

# Módulo 4. Metodologías ágiles

## Unidad 4.1 Introducción a las metodologías ágiles

El concepto de *metodologías ágiles* se aplica cada día más en empresas y emprendimientos, puesto que son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo así mayor flexibilidad y rapidez en la respuesta para amoldar el proyecto y su desarrollo a las necesidades del cliente y las circunstancias específicas del entorno. Se podría decir, entonces, que los emprendimientos que apuestan por estas metodologías consiguen gestionar sus proyectos de forma eficaz reduciendo los costes e incrementando su productividad.

Esta metodología surge en el ámbito del desarrollo de *software*, pero es aplicable a cualquier industria y rubro.

Veamos rápidamente algunos de los principios básicos de las metodologías ágiles que se presentan en el artículo “Las 8 grandes ventajas de las metodologías ágiles” (Martínez, 2014).

1. Las metodologías ágiles mejoran la satisfacción del cliente dado que este se involucrará y comprometerá a lo largo del proyecto. En cada etapa del desarrollo se informará al cliente sobre los progresos del mismo. De ese modo, el cliente puede sumar su experiencia, comentarios o pedidos de ajustes para optimizar las características del producto final. Se pueden evitar así numerosos malentendidos dado que el cliente poseerá en todo momento una completa visión del estado del producto.
2. Asimismo, mejoran la motivación e implicación del equipo de trabajo. Pero esta mejora no es casual: las metodologías ágiles permiten a todos los miembros del equipo conocer el estado del proyecto en cualquier momento. Los compromisos son negociados y aceptados por todos los miembros del equipo y las ideas de cualquiera de sus integrantes son tenidas en cuenta.
3. Los procesos ágiles permiten ahorrar tanto tiempo como costos. El desarrollo ágil trabaja de un modo más eficiente y rápido que otras metodologías. Además, estos procesos ponen el foco en cumplir

estrictamente el presupuesto y los plazos pactados a la hora de definir y planificar el proyecto.

4. Se trabaja con mayor velocidad y eficiencia, realizando entregas parciales pero funcionales del producto. De ese modo, es posible entregar en el menor intervalo de tiempo posible una versión funcional del mismo.

5. Gracias a las entregas parciales (centradas en entregar en primer lugar aquellas funcionalidades que en verdad aportan valor) y a la implicación del cliente será posible eliminar aquellas características innecesarias del producto.

6. Las metodologías ágiles permiten mejorar la calidad del producto. La continua interacción entre los miembros del equipo emprendedor y los clientes tienen como objetivo asegurar que el producto final sea exactamente lo que el cliente quiere y necesita. Además, este enfoque permite abrazar la excelencia, lo que permite obtener un producto superior.

7. Por otro lado, estas metodologías permiten alertar rápidamente tanto de errores como de problemas. En la etapa de planificación, el equipo ha presentado una hoja de ruta anticipando y dando respuesta a los principales problemas técnicos y a la velocidad en la que se puede trabajar. Con metodologías más tradicionales, los errores no identificados en las primeras fases del proyecto suelen acarrear costes muy altos.

8. Y, finalmente, las metodologías ágiles permiten rentabilizar las inversiones más rápidamente. Gracias a la realización de entregas tempranas el cliente tendrá rápido acceso a aquellas funcionalidades que en verdad aportan valor acelerando el retorno de la inversión. (Martínez, 2014, <https://goo.gl/a4kqpG>).

### 4.1.1 Estilos de gestión

A continuación, en el siguiente continuo se presentan dos historias que muestran los dos estilos de gestión que se pueden encontrar en empresas y emprendimientos:

Figura 1: Historias emprendedoras

# Mark - El hombre de negocios



## Mark

- 30 años Norteamericano
- Software Services Company Owner
- MBA - Ing de Software
- Miembro de Grupo Empresarios



Cansado de Facturar horas y de Tener problemas

## Idea: Producto de Software



## Enero de 2002

Comienza a trabajar fuertemente



## Inicio de Fase de planificación

Resultado: Plan de Proyecto (gantt)  
Duración de la fase: 4 meses  
Fecha Fin: Enero 2003



## Plan de Negocio

Contratando a la mejor empresa en consultoría de los EEUU.  
TimeLine: 6 Meses  
Fecha Fin: Septiembre de 2002



## Socio Inversor

- John C.
- U\$S 1,000,000
- 50% o equity



## Fase de Analisis

Resultado: ERS  
Duración de la fase: 6 meses  
Cantidad de Requerimientos: 70  
Fecha Fin: Junio 2003



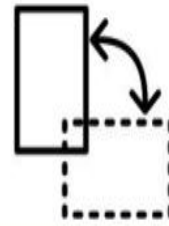
## Equipo de Desarrollo

Arquitecto  
6 Devs  
2 QA  
1 BA  
1 PM  
Se elige un framework State of the Art



6 Meses dps Tienen la primer release interna.

Mark decide Cambiar 3 desarrolladores



### Primer Cliente

Un reconocido banco se interesa por el producto. No lo quieren para gestionar proyectos, lo quieren para Reclutamiento.

Comienza la modificación del sistema.

### ROI

John C. Quien no estuvo involucrado comienza a presionar para saber cuando podrá obtener dividendos del producto

Hasta ahora: 0 Clientes finales han probado el producto

### Rotación

Comienza la rotación del equipo. Su Arquitecto Renuncia. Es difícil conseguir desarrolladores que conozcan el framework

Otros 6 meses se han ido



### Cambio

6 meses después de muchas quejas, visitas al cliente, el Banco decide contratar una plataforma SASS 30 U\$\$ por usuario por mes.

### Cansancio

Mark esta muy cansado atribuye la mayoría de sus problemas a que su equipo le falta compromiso

Mark decide Archivar el producto esperando que vengan mejores tiempos

**8 años han pasado y el código sigue intacto**

# The "Dropouts"



## Scott & Mike

- 21 años
- Ingenieros en Sistemas

## Su idea preferida: Plataforma de Gestión de proyectos

Su hobby: intentar ideas chicas y probar si funcionan

## Credit Card Loan

Con U\$S 10,000 dólares fundaron la empresa. No dieron Equity a Nadie



## Modelo de Comercialización

- Invertir 0 U\$S en Vendedores.
- Se debía vender solo.
- Vender Barato
- El Cliente tiene que comprar, no nosotros vender.

## First Release

Lanzan la primera version funcional: Issue Tracker: 6 Semanas después.

Comienzan a Utilizarla como su propia herramienta.

"We built the first version of our product while working in the garage before we even had our first office."

## Requerimientos

**Alto Nivel: crear un sistema de Gestión de Tickets o Issues.**

## Dropouts



**Junio 2006 -  
4,340 clientes y  
50 staff  
empleados**

## 2010 - Inversión

\$60 million investment from Accel Partners.

**\$100M+  
Business With  
No Sales  
People -2010**

Fuente: Elaboración propia.

Nota: los dropouts se definen como estudiantes que dejan la universidad.

Repasemos, entonces, las diferencias de las dos perspectivas de desarrollo:

**Tabla 1: Diferencias en estilos de gestión**

Mark - El hombre de negocios	Dropouts
<ul style="list-style-type: none"><li>● Necesitamos un plan de negocios detallado.</li><li>● No tocar una línea de código hasta que nos pongamos de acuerdo con lo que hay que hacer.</li><li>● Hagamos un muy buen proceso para optimizar el trabajo en equipo.</li><li>● Se necesitan los mejores vendedores.</li><li>● ¡Que no nos roben el producto o la idea!</li><li>● Esto es culpa de la falta de compromiso.</li></ul>	<ul style="list-style-type: none"><li>● Priorizan la construcción de prototipos y MVP's.</li><li>● Buscan modelar un negocio utilizando herramientas livianas y no hacer un gran plan de negocios antes de empezar.</li><li>● Se focalizan en validar las ideas y no en hacer grandes estudios de mercado.</li><li>● Hacer lo necesario al mínimo costo.<ul style="list-style-type: none"><li>● Visibilidad.</li><li>● Equipos autodirigidos.</li><li>● Elegir bien a las personas.</li></ul></li></ul>

Fuente: Elaboración propia.

Estas dos historias indican que hay dos estilos diferentes de gestión que se pueden encontrar:

### 1. Tradicional

- El estilo de planificación es más largo, se planifica a meses, los proyectos tiene una larga duración.
- Los planes son más rígidos.
- El jefe tiene el rol fundamental de controlar al equipo y los procesos.
- Las especificaciones del producto son detalladas y largas.
- El equipo sigue órdenes.
- El cliente no es parte del equipo.
- El cambio genera negociaciones y fricciones.

### 2. Ágil

- Hay planificaciones a largo plazo, pero con entregables permanentes.
- El equipo es autodirigido, donde el jefe es facilitador y no controlador.
- Las especificaciones se detallan, pero son más importantes los prototipos.
- El equipo participa de las decisiones, propone alternativas.
- El cliente es parte del equipo.
- El cambio es parte de la forma de trabajo.

## 4.1.2 Manifiesto ágil

Todo esto surge de un manifiesto ágil, redactado a inicios del 2000, por un grupo de ingenieros que analizaron las características de los proyectos que salen bien, identificando cuatro características principales:

### **1. Individuos e interacciones sobre los procesos y herramientas:**

Podemos tener las mejores herramientas y procesos pero si el equipo no es el indicado o se lleva mal, ese proyecto no va a avanzar, es más importante trabajar en las personas y cómo se relacionan que en los procesos.

Esto no quiere decir que no se tengan que definir procesos y herramientas, pero lo principal es trabajar el equipo.

### **2. PRODUCTO / SERVICIO funcionando sobre documentación extensiva:**

Tienen más valor los prototipos que la documentación extensiva. Acá surge un concepto conocido como la parálisis por análisis, donde sólo se analiza y por ello no llega la ejecución.

### **3. Colaboración con el cliente sobre negociación contractual:**

Muchos proyectos tradicionales, cuando tiene planes largos generan fricciones con el cliente o proveedor, interno o externo.

En un proyecto Ágil se prioriza la colaboración, muchas veces cediendo a acuerdo contractuales.

### **4. Respuesta al cambio sobre seguir un plan:**

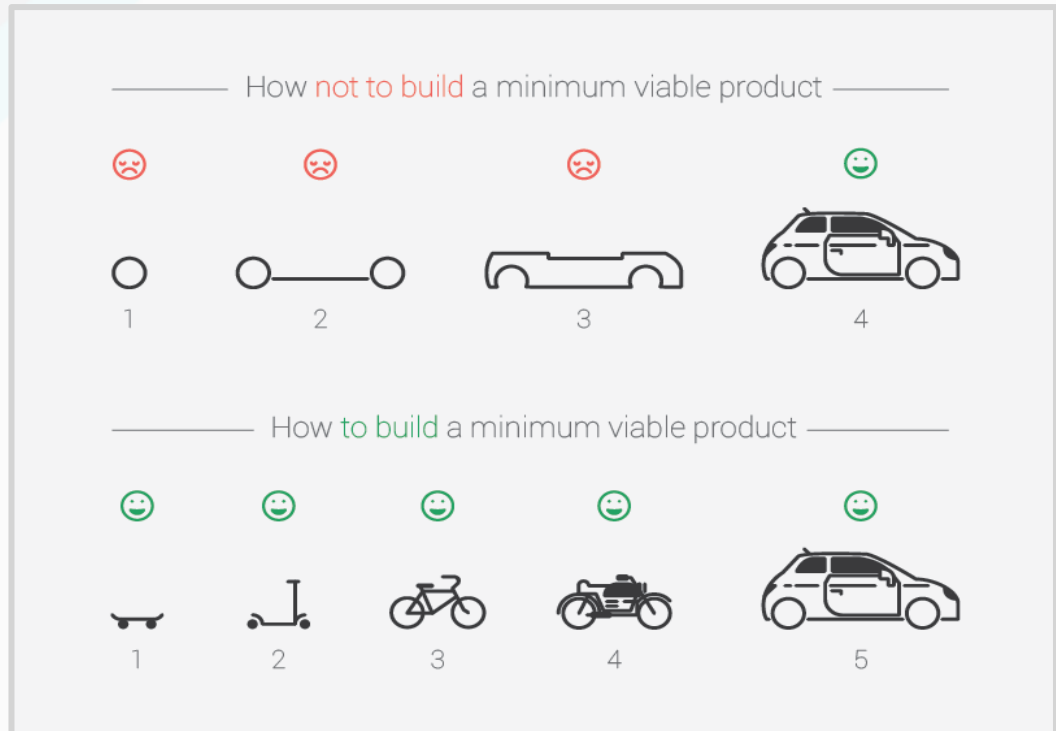
Este punto tiene que ver con no ajustarse a cuestiones definidas en el plan y tener la flexibilidad de que si surgen cuestiones que perdieron validez tiene que ser cambiadas. (Beck et al., 2000, <https://goo.gl/JY4H6D>).

Veamos a continuación un ejemplo:

### **Spotify**

Parece una aplicación sencilla, pero es una obra de ingeniería gigante con muchos módulos y versiones. Uno de sus mentores es quien firmó el manifiesto ágil. Todo lo que hacen en Spotify sobre construcción de producto es guiado por metodologías ágiles. Esta persona realizó el gráfico que se observa a continuación.

**Figura 2: Enfoque ágil de Spotify**



Fuente: Help Scout(S.f.)Recuperado de: <https://goo.gl/uTk4pc>

En la parte de arriba, se presenta cómo **no** se construyen los productos en Spotify. Por ejemplo, viene un cliente y le dice que quiere un auto. Ellos toman el pedido y le dicen que regrese en un mes.

Cuando el cliente regresa, le cuentan que comenzaron a hacer la rueda. “¿Te gusta? Bueno, el mes que viene te mostramos otro avance”. Cuando regresa el cliente, ya le muestran las dos ruedas, y así sucesivamente. Al mes regresa a ver las ruedas con el chasis y así, al cuarto mes, podrá ver el auto terminado.

Muchas empresas trabajan con este enfoque de metodología tradicional.

En el caso de Spotify, ante el pedido de un cliente de un auto, en lo primero que indaga es para qué lo quiere. Ante la respuesta de que lo precisa como medio de transporte, lo citan a las dos semanas para ver avances. A las dos semanas el cliente vuelve y le ofrecen una patineta, con el objetivo de entregar algo rápidamente que le permita satisfacer su necesidad de movilidad. Y de esa manera se va mejorando el vehículo en cada iteración, y al final se entrega el mismo producto.

La diferencia radical de esto es que con cada iteración el cliente recibe algo de valor, un incremental de valor. Esto es clave en metodologías ágiles: brindarle al cliente en cada etapa algo concreto, que se pueda tocar, que se pueda probar. Este es el concepto de la entrega de *prototipos que validar*.

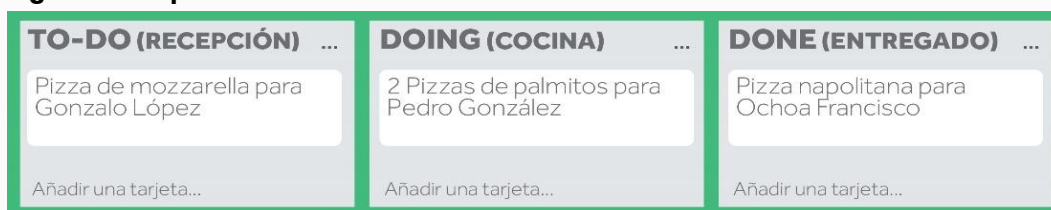
A continuación, desarrollaremos dos de las metodologías ágiles más importantes: Kanban y Scrum.

### 4.1.3 La metodología Kanban

Originalmente desarrollada por Toyota, la metodología Kanban revolucionó la forma en que las empresas maximizan la eficiencia en la producción. Esta consiste en una herramienta logística muy liviana para producir cosas de manera eficiente, con base en lo que se vaya necesitando.

Una de sus grandes ventajas reside en la sencillez del método. Es recomendada especialmente para áreas o proyectos donde hay muchas entradas y salidas y poca planificación. Vamos a un ejemplo concreto vinculado con la producción de pizzas:

**Figura 3: Esquema inicial**

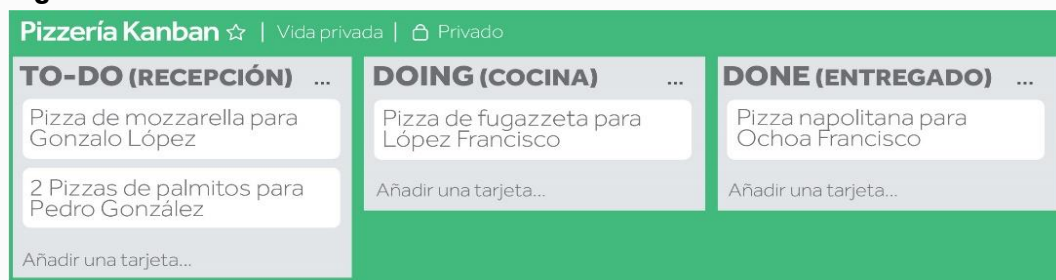


Fuente: Elaboración propia.

Se utilizan tarjetas para planificar y seguir la producción desde la recepción, inventario y entrega. Luego, se debe armar un tablero donde están todas las fases del proceso:

- **TODO: lista de cosas por hacer.**
- **DOING: cosas que se están haciendo o procesando.**
- **DONE: cosas que están hechas.**

**Figura 4: Modelo de tablero**

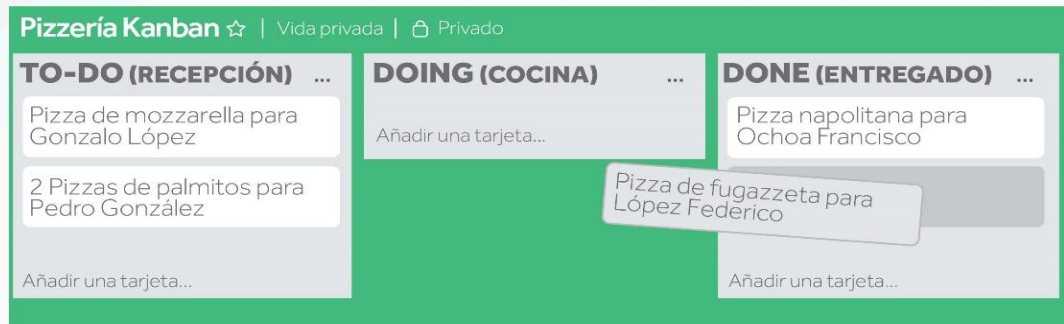


Fuente: Elaboración propia.

Aparece acá un concepto importante que es el *push* frente al *pull*:

- **PUSH = empujar.**
- **PULL = tirar.**

**Figura 5: Última fase tirando (pulling). Tarjeta desde doing a done**



Fuente: Elaboración propia.

- **A medida que se van necesitando, se van pidiendo los ingredientes (solo los necesarios).**
- **Hay una clara visibilidad de lo que se está haciendo en todo momento.**

Cuando las siguientes áreas van tomando las tarjetas que dejan las áreas anteriores, están haciendo pull. Por el contrario, en una metodología tradicional, el jefe o cliente le daría una tarjeta.

La diferencia principal, entonces, reside en que cuando se hace pull se independiza a la persona de su trabajo y se logra que no haya recursos en espera. A comienzos del día, se presenta la lista de tareas y el equipo sabe cómo ejecutarla.

Con push, al comenzar el día, se dan las indicaciones al equipo.

#### 4.1.4 La metodología Scrum

Scrum es un enfoque ágil para la gestión de un proyecto, el cual utiliza procesos iterativos e incrementales, está orientado a resultados y compromisos y también cuenta con una implementación muy simple.

En cuanto a la metodología, se elige un período de tiempo de planificación; por ejemplo, 2 semanas. Esto sería un **sprint de planificación de 2 semanas**. Puede que un proyecto tenga 20 sprints, pero el horizonte de planificación es de 2 semanas.

Allí comienza una serie de actividades de la gestión del proyecto:

1. Lo primero es realizar una reunión donde se desglosan las tareas que se requieren para cumplir con el objetivo, lo que se denomina *backlog*.
2. El primer día de la semana, se hace una planificación para esas 2 semanas. Allí se definen actividades por tiempos. A esta tarea se la denomina *grooming*.
3. Definidas las actividades y prioridades, el equipo se pone a trabajar.
4. Cada día el equipo realiza reuniones de 15 minutos, a primera hora del día (o alguna hora del día donde todos los integrantes puedan, es importante que siempre sea en el mismo horario para generar el hábito), lo que se

denomina *stand up meeting* o *daily*, donde los integrantes responden tres preguntas:

1. ¿Qué hiciste ayer?
2. ¿Qué vas a hacer hoy?
3. ¿Tienes algún problema?

A medida que transcurren los días y las dos semanas, se va actualizando el tablero de seguimiento.

El hecho de que esas reuniones se realicen con todos sus participantes de pie (por eso *stand up meeting*) es crucial para que duren menos de 15 minutos. Así nadie se pone cómodo y estira la reunión.

Es muy importante ser concisos en estas reuniones y responder de manera escueta y específica a cada una de las tres preguntas. Cualquier problema que surja se trata después o por fuera de esa reunión para respetar los 15 minutos. A su vez, ese problema se trata solo con las personas involucradas, así no se retiene o demora a todo el resto del equipo.

Los participantes de las *daily* o *stand up meetings* son los responsables y miembros del equipo que esté llevando adelante el proyecto.

5. Al finalizar las dos semanas, se realizan la *demo* o *sprint planning* y la *retrospectiva*:
  - Para la **demo** el equipo se reúne con el cliente, se muestra todo lo trabajado y se recibe el *feedback* identificando ajustes que realizar. Esta reunión o evento puede durar 1 hora. Aquí es recomendable que participe el cliente, ya que es a quien se muestran los avances y se pide retroalimentación.
  - La **retrospectiva** se realiza después de la *demo*, y es una reunión donde se trabaja con un tablero donde se va definiendo lo siguiente:

**Tabla 2: Tablero de retrospectiva**

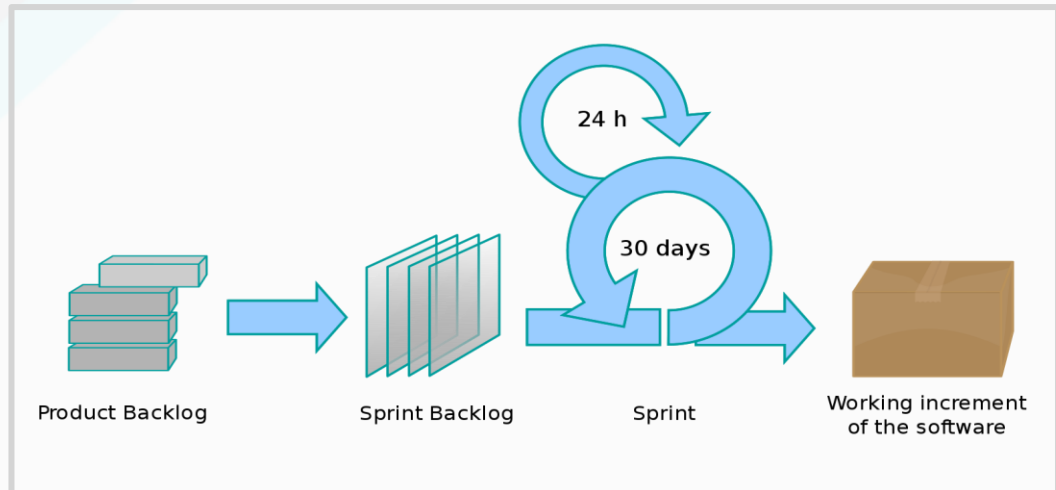
COSAS QUE SALIERON BIEN	COSAS QUE SALIERON MAL	COSAS PARA MEJORAR

Fuente: Elaboración propia.

En esta instancia, puede o no participar el cliente. Esta actividad puede llegar a durar una hora también, pero es muy importante para generar la instancia de aprendizaje y mejora continua.

A modo de revisión, veamos el ciclo de vida Scrum:

**Figura 6: Ciclo de vida Scrum**



Fuente: Lakeworks, 2009, <https://goo.gl/MijwBN>

## **Roles y responsabilidades en Scrum**

### **1. Dueño del producto:**

- Define las funcionalidades.
- Decide la fecha de entrega y su contenido.
- Prioriza el backlog de producto.
- Puede cambiar las funcionalidades y prioridades de cada sprint.
- Acepta o rechaza los resultados del sprint.
- Participa en las reuniones de planificación de sprint y revisión.
- Puede ser el mismo cliente (en muchos casos) o el miembro del equipo que más conozca del cliente.

### **2. Scrum master:**

- Asegura que el equipo sea completamente funcional y productivo.
- Promueve la cooperación entre todos los roles y remueve barreras.
- Protege al equipo de interferencias externas.
- Asegura que el proceso sea cumplimentado.
- Participa en las reuniones diarias, de planificación de sprint y revisión de él.

### **3. Equipo:**

- Típicamente, de 5 a 9 personas.
- Selecciona el backlog de sprint.
- Tiene el derecho de hacer lo que sea dentro de los límites de las guías del proyecto para alcanzar el objetivo del sprint.
- Se organiza a sí mismo y a su trabajo.
- Demuestra los resultados del trabajo al dueño del producto.

**Veamos algunas claves para la implementación:**

- Agregar valor en cada etapa.
- Separar el proyecto en fases cortas y priorizar requerimientos.
- Aprender de cada etapa.
- Mejorar continuamente.

Para finalizar, repasemos los **beneficios de esta metodología**:

- No se desperdicia tiempo en fases “no productivas”.
- El desarrollo comienza de manera temprana.
- Entregas frecuentes.
- *Feedback* temprano.
- Los cambios son normales, no son un problema.
- Se hace foco en el valor de negocio, calidad y *time to market*.
- ¡Menos papel! Se documenta solamente lo que es útil.
- El cliente participa activamente en todas las etapas (*product owner*).

## Unidad 4.2 Herramientas colaborativas

Las herramientas colaborativas nos ofrecen muchos beneficios; por ejemplo, permiten a distintas personas estar comunicadas y trabajar en conjunto estando o no en el mismo lugar físico. Las modificaciones son alertadas de manera *online*. Hay una medida de la productividad e instancias de rendición de cuentas más acotadas en el tiempo.

Gracias a la tecnología y conectividad a través de internet, equipos que estén en diferentes locaciones pueden trabajar juntos perfectamente. El seguimiento de cumplimiento de objetivos se hace más sencillo, las comunicaciones fluyen más rápido y la documentación de resultados se simplifica. Hay que tener especial atención al exceso de virtualización de las relaciones, ya que a veces cuestiones que fácilmente pueden resolverse con un llamado telefónico se extienden y complejizan a través de la utilización de *e-mails* kilométricos. Cuando la relación está construida, hay confianza y se llega a un punto específico de detalle, es mejor acudir al llamado por teléfono o cualquier otro medio en vez de los *e-mails* o mensajes.

Veamos algunos ejemplos de ellas:

### 4.2.1 Google Calendar

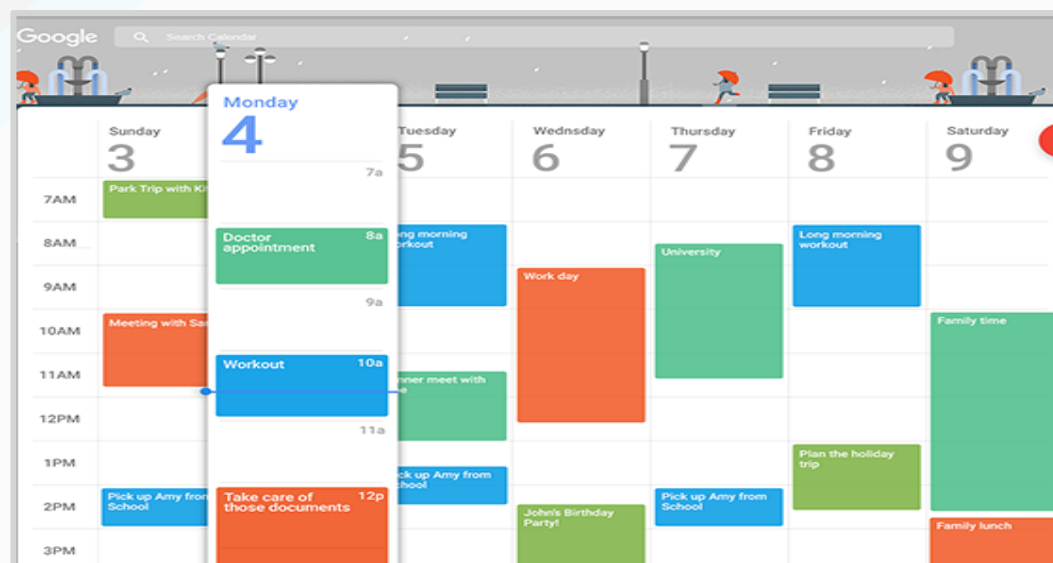
El calendario de Google, que se sincroniza con tu teléfono y con el *e-mail*, es muy útil para agendar reuniones e, inclusive, para tener el enlace para conectarse virtualmente (a través de Hangout, herramienta de video llamadas de Google) y comenzar la reunión.

En cada evento que se agenda en el calendario, se pueden invitar otras personas/participantes, y se pueden incluir archivos o comentarios, entre otras cosas. También pueden agendarse reuniones con cierta frecuencia y periodicidad, y utilizarse como autorecordatorio para realizar ciertas actividades.

Para reuniones que son fuera de la oficina, por ejemplo, se puede agregar la dirección del lugar de la reunión con acceso directo a un mapa para llegar a dicho lugar a través de cualquier medio de transporte. También se puede configurar cada evento para que Google de manera automática envíe una notificación cierta cantidad de minutos antes para preparar la reunión o para trasladarse al lugar donde esta se va a llevar a cabo.

Las reuniones de estas metodologías (*daily*, *planning*, *demo*, *retro*) se pueden agendar en el calendario y contar con un enlace para que rápidamente los participantes cuenten con acceso a toda la información necesaria para ella.

**Figura 7: Ejemplo de Google Calendar**



Fuente: Ilan, s. f., <https://goo.gl/UprmQA>

## 4.2.2 Trello

Esta herramienta es particularmente poderosa para trabajar con Kanban o Scrum. Es gratuita y puede utilizarse con equipos en tiempo real, a través de diferentes usuarios.

Cada tablero puede ser un proyecto, e inclusive se puede tener uno particular del emprendimiento y luego otros por proyecto o áreas (por ejemplo, área de desarrollo o área técnica).

La herramienta permite armar tableros con diferentes listas y tarjetas dentro de cada lista donde se describen las actividades, se agregan los responsables, se pueden adjuntar archivos, colocar fecha de vencimiento, etiquetas, comentarios o *checklists*, entre otros.

Las listas pueden definirse como la lista del backlog donde esté todo el cúmulo de tareas por realizar, la cual se alimenta durante la grooming. Se puede tener otra lista con el *sprint backlog*, donde cada tarjeta (tarea) esté priorizada de acuerdo con lo que se defina en el planning.

Puede haber otra lista que se denomine *En progreso*, que contenga todas las tareas que estén en ejecución en ese día en particular, y finalmente una lista que contenga las tarjetas terminadas.

Adicionalmente, se puede sumar una lista llamada *Información importante*, que contenga archivos, accesos, claves, artículos y cualquier información que resulte clave, de recurrente necesidad y de fácil acceso para el equipo.

Trello cuenta con muchos *plugins* o *power-ups* para conectar esta herramienta a otras, como Google Drive o Google Calendar, entre otros. Esto permite eficientizar tiempos y conectar archivos.

Esta herramienta se encuentra online, y cada miembro del equipo tiene un usuario. A cada tarjeta (tarea) se le asignan miembros y responsables. Además, en los comentarios internos que permite colocar cada tarjeta se pueden etiquetar miembros del equipo para compartirles información por ese canal o dejarles comentarios o aclaraciones.

**Figura 8: Ejemplo de Trello**



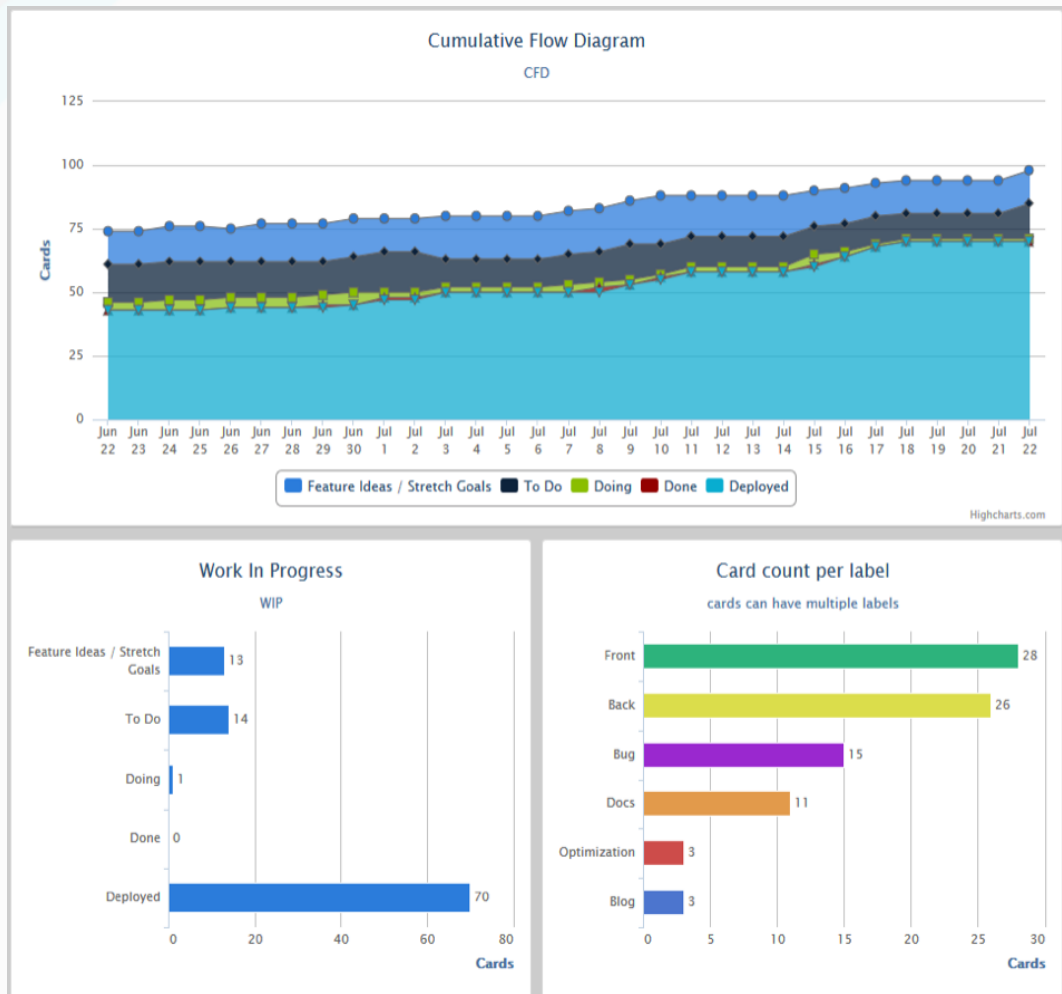
Fuente: ChicaGeek, 2013, <https://goo.gl/6K4dfX>

### 4.2.3 Ollertap

Es una herramienta que se conecta con Trello para medir la productividad. Ollertap permite medir la cantidad de tarjetas (tareas) realizadas, las que quedan pendientes, la velocidad de trabajo en función de tareas realizadas y objetivos cumplidos y, por ende, la capacidad de trabajo del equipo.

Recomendamos utilizar esta herramienta una vez que el equipo haya podido generar el hábito de las reuniones dailys y ya esté usando el tablero activamente.

**Figura 9: Ejemplo Ollertap**



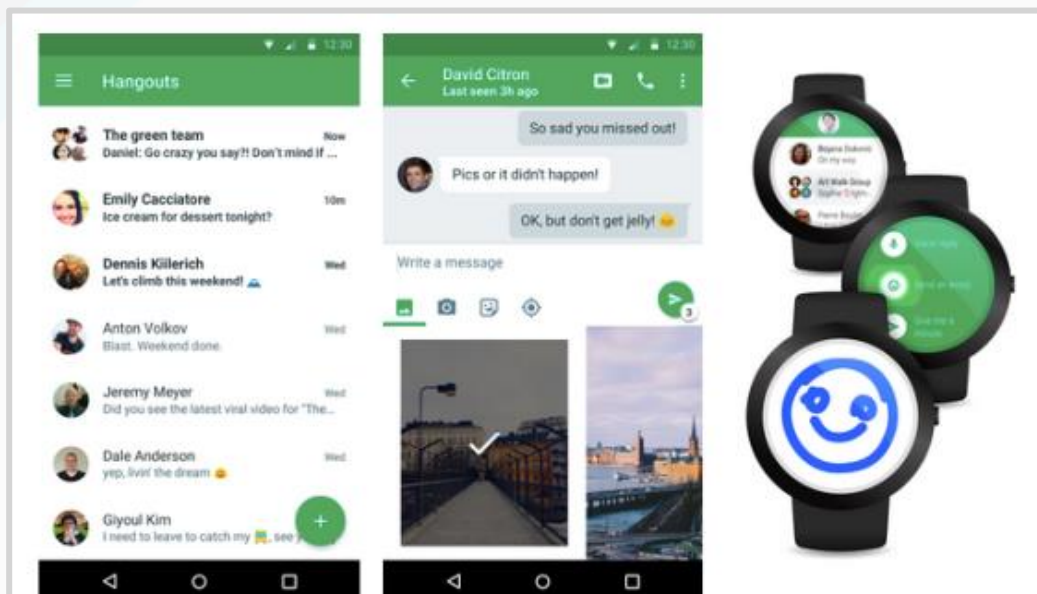
Fuente: <https://ollertapp.com/>

## 4.2.4 Google Hangout

Esta herramienta permite hacer videollamadas gratuitas, en grupo o uno a uno. Puede ser con video o solo voz. Pueden participar hasta 25 personas en simultáneo.

Todos deben contar con un usuario de Gmail para poder ingresar. Se pueden generar reuniones privadas o de acceso público. Se pueden compartir archivos a través de un chat con el que cuenta la herramienta. También puede compartirse la pantalla para que los otros miembros del equipo puedan ver una presentación o archivo en vivo y en directo, y de esta manera puede generarse trabajo colaborativo.

**Figura 10: Ejemplo de Google Hangout**



Fuente: Causey, 2015, <https://goo.gl/1s8kjU>

### 4.2.5 Google Drive

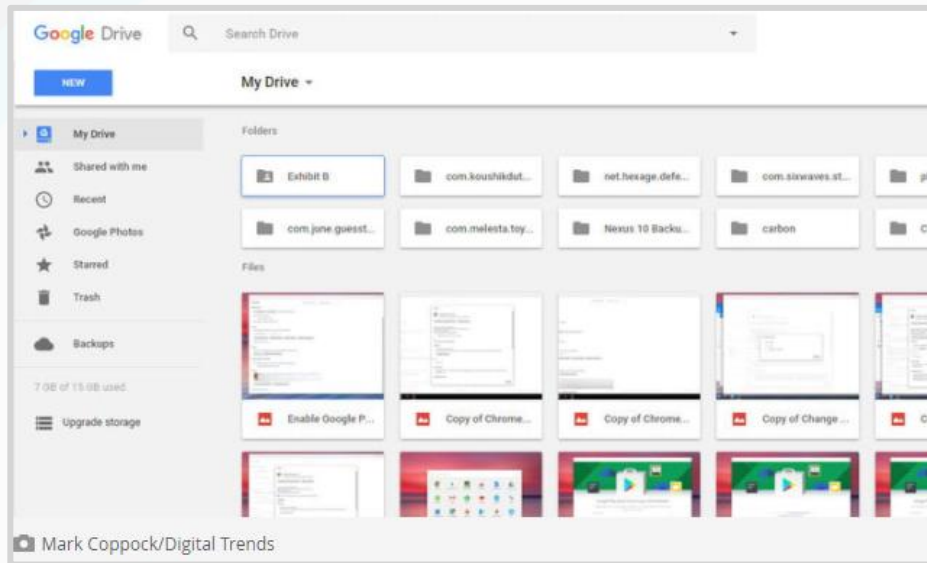
Otra de las herramientas del paquete gratuito de Google es Drive, un servicio de alojamiento de archivos. Es accesible a través del sitio web desde computadoras y dispone de aplicaciones para Android e iOS que permiten editar documentos y hojas de cálculo.

Esta herramienta permite, entre otras cosas:

- Contar con archivos versionados, permitiendo así acceder a versiones anteriores de un archivo luego de haberlo modificado.
- Crear y editar archivos y tenerlos disponibles en la nube.
- Proporciona respaldo automático.

Google Drive también cuenta con aplicaciones integradas como Google Docs, Slides, programa gratuito para crear documentos colaborativos en línea, editor de formularios destinados a encuestas, creador de dibujos, entre otros.

**Figura 11: Ejemplo de Google Drive**



Fuente: Martindale, 2017, <https://goo.gl/fVeeH8>

*Estas herramientas son muy poderosas y simples, y permiten potenciar el trabajo colaborativo y el retorno entre diferentes miembros del equipo, e incluso entre clientes, proveedores, socios y aliados estratégicos.*

# Referencias

**Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M..., & Thomas, D.** (2000). Manifiesto for Agile Software Development. Recuperado de <http://agilemanifesto.org/>

**Causey, J.** (2015). [Imagen sin título sobre Google Hangouts]. Recuperado de <http://www.talkandroid.com/260250-google-hangouts-4-0-launches-today/>

**ChicaGeek.** (2013). [Imagen sin título sobre Trello]. Recuperado de <http://www.chicageek.com/organiza-facilmente-cualquier-proyecto-trello-2294368/>

**Ciotti, G.** (2016). Why a Visual Really is Worth 1,000 Words. Recuperado de <https://www.helpscout.net/blog/visual-communication/>

**Ilan, R.** (s. f.). Google Calendar Redesign – WIP [Imagen]. Recuperado de <https://www.uplabs.com/posts/google-calendar-web-redesign-work-in-progress>

**Lakeworks.** (2009). Scrum process [Imagen]. Recuperado de [https://commons.wikimedia.org/wiki/File:Scrum\\_process.svg](https://commons.wikimedia.org/wiki/File:Scrum_process.svg)

**Martindale, J.** (2017). [Imagen sin título sobre Google Drive]. Recuperado de <https://www.digitaltrends.com/computing/google-drive-vs-dropbox/>

**Martínez, E.** (2014). Las 8 grandes ventajas de las metodologías ágiles. Recuperado de <https://www.iebschool.com/blog/que-es-agile-agile-scrum/>