

## **LAS METODOLOGÍAS AGILES**

### **1. Metodología Ágil Scrum:**

#### **Descubriendo la Historia de Scrum**

El concepto de Scrum tiene su origen en un estudio de 1986 sobre los nuevos procesos de desarrollo utilizados en productos exitosos en Japón y los Estados Unidos (cámaras de fotos de Canon, fotocopadoras de Xerox, automóviles de Honda, ordenadores de HP y otros). Los equipos que desarrollaron estos productos partían de requisitos muy generales, así como novedosos, y debían salir al mercado en mucho menos del tiempo del que se tardó en lanzar productos anteriores. Estos equipos seguían patrones de ejecución de proyecto muy similares. En este estudio se comparaba la forma de trabajo de estos equipos altamente productivos y multidisciplinarios con la colaboración entre los jugadores de Rugby y su formación de Scrum.

En 1993 se realizó el primer Scrum para desarrollo de software y en 1995 el proceso fue formalizado. En 2001 un grupo de personas muy relevantes en lo que empezaba a ser el desarrollo ágil escribieron los valores fundamentales de los procesos ágiles.

Desde 1995 miles de proyectos en todo el mundo han utilizado Scrum para el desarrollo de productos, tanto en empresas pequeñas, con tan sólo 3 personas desarrollando un producto, como en multinacionales.

**Organización de Scrum:**

**1. Artefactos**

Tenemos los siguientes artefactos:

## Artefactos

- Product backlog
- Sprint backlog
- Burndown charts

**1.1. Product Backlog:** El product backlog es un documento de alto nivel para todo el proyecto.

Contiene descripciones genéricas de todos los requerimientos y funcionalidades deseables

Se realiza la lista de Requerimientos a trabajar en el Scrum.

Es abierto y cualquiera puede modificarlo.

## Ejemplo de Product Backlog

Backlog item	Estimación
Permitir que un invitado a hacer una reserva.	3
Como invitado, quiero cancelar una reserva.	5
Como invitado, quiero cambiar las fechas de una reserva.	3
Como un empleado de hotel, puedo ejecutar informes de los ingresos por habitación disponible	8
Mejorar el manejo de excepciones	8
...	30
...	50

jorge.abad@gmail.com

**1.2. Sprint Backlog:** es el documento donde se establecen y se compromete el equipo a cumplir los Requerimientos Funcionales durante el desarrollo del Sprint.

El Sprint dura de 2 a 4 semanas.

## Ejemplo de Sprint Backlog

Tareas	L	M	M	J	V
Codificar UI	8	4	8		
Codificar negocio	16	12	10	4	
Testear negocio	8	16	16	11	8
Escribir ayuda online	12				
Escribir la clase foo	8	8	8	8	8
Agregar error logging			8	4	



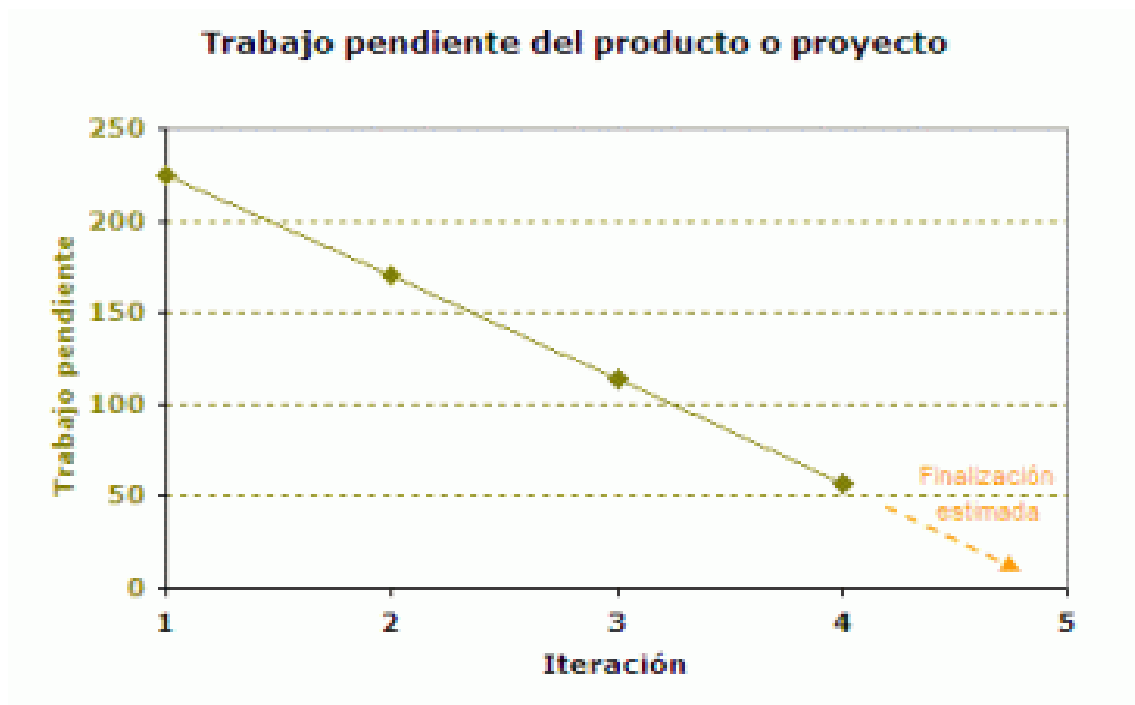
En el ejemplo se puede observar las tareas que se comprometen a realizar en el Sprint y la duración de cada tarea.

### 1.3. Burndowns Charts

Eje Horizontal: muestra los sprint que se desarrollan.

Eje Vertical: los requisitos que faltan completar al final de cada sprint.

Veamos un ejemplo de Burndowns Charts:



En el Burn Down Chart se realiza el seguimiento diario de cómo se van desarrollando los requerimientos funcionales para el desarrollo del proyecto.

## **2. Roles:**

### **2.1. Product Owner:**

- Define Funcionalidades
- Diseño del Producto
- Representa al cliente
- Prioriza funcionalidad de acuerdo al valor del negocio.
- Acepta o rechaza resultados del trabajo en equipo.

### **2.2. Product Owner:**

- Define Funcionalidades del producto.
- Es el dueño del producto.
- Representa al Cliente
- Vuelca lo que hay que hacer en el trabajo.
- Define Funcionalidades del producto.
- Es el dueño del producto.
- Representa al Cliente
- Vuelca lo que hay que hacer en el trabajo.

### **2.3. El Team o Equipo:**

- 5 a 9 integrantes
- Autoorganizado
- Realiza tareas multifuncionales
- Solo puede haber cambios cuando termina el Sprint.
- Miembros Full Time.

#### 2.4. Scrum Master:

Durante la iteración el [Facilitador \(Scrum Master\)](#) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

#### *No es un Lider Project.*

- Protege al equipo.
  - No es un Lider Project.
  - Representa la Gestión del Proyecto
  - Resuelve impedimentos.
  - Asegura un equipo funcional y productivo.
  - Responsable de promover los valores y prácticas de Scrum.
- Se asegura de que el equipo es completamente funcional y productivo.
  - Permite la estrecha cooperación en todos los roles y funciones.
  - Escudo del equipo de interferencias.

3. Reuniones:

## Reuniones

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting



### 3.1. Sprint Planning Meeting:

- Es Reunión de Planificación del Sprint (Sprint Planning Meeting).
- Al inicio del ciclo Sprint (cada 15 o 30 días), una “Reunión de Planificación del Sprint” se lleva a cabo.
- Selecciona el trabajo a realizar.
- Interviene el Product Owner con el Equipo que cumplirá los requerimientos y queda detallado en el Sprint Backlog
- Los requerimientos se extraen del **PRODUCT BACKLOG**.
- Prepara con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.
- Identifica y comunica cuánto del trabajo es probable que se realice durante el actual Sprint.
- **8 hs como límite.**

### 3.2. El Sprint

- ✓ El Sprint es el período en el cual se lleva a cabo el trabajo en sí.
- ✓ Es recomendado que la duración de los sprints sea constante y definida por el equipo en base a su propia experiencia.
- ✓ Se puede comenzar con una duración de sprint en particular (2 o 4 semanas) e ir ajustándolo en base al ritmo del equipo.
- ✓ El final de cada sprint, el equipo deberá presentar los avances logrados, y deberían entregar un producto con características de utilizable por el cliente.
- ✓ Asimismo **se recomienda no cambiar los objetivos del sprint backlog a menos que la falta de estos cambios amenacen al éxito del proyecto.**

✓ La constancia hace a la concentración y la mejor productividad del equipo de trabajo.

### **3.3. Daily Scrum:**

- Duración 15 minutos.
- Todos los integrantes se mantienen de pie para evitar pérdida de tiempo y desconcentración.

Preguntas:

¿Qué has hecho ayer?

¿Qué es lo que estas planeando hacer hoy?

¿Has tenido algún problema que te haya impedido alcanzar el objetivo?

#### **3.3.1. Roles en el Daily Scrum:**

En la reunión nos encontramos con los roles de cerdo y gallinas. El cerdo se compromete con el proyecto, la gallina pone el huevo y se va.

Se suele tener un castigo para el que llega tarde, por ejemplo se le cuelga una gallina en el pecho.

**Todos son bienvenidos, pero sólo los "cerdos" pueden hablar.**

**La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.**

**Todos los asistentes deben mantenerse de pie (esto ayuda a mantener la reunión corta)**

La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.

**Durante la reunión, cada miembro del equipo contesta a tres preguntas:**

¿Qué has hecho desde ayer?

¿Qué es lo que estás planeando hacer hoy?

¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del Scrum Master recordar estos impedimentos).

**Asumen roles de Cerdo:**

- Scrum Master(Facilitador)
- Product Owner: representa al cliente. Dueño del Producto.
- Equipo: Realiza las tareas

**Asumen roles de Gallina:**

- Usuario.
- Gerente.

**3.4. Scrum de Scrum:**

- ✓ Se realiza cada día normalmente después del “Daily Scrum”
- ✓ Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.
- ✓ Asiste una persona asignada por cada equipo.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

La agenda será la misma que la del Daily Scrum, añadiendo además las siguientes cuatro preguntas:

Preguntas:

- ¿Qué ha hecho tu equipo desde nuestra ultima reunión?
- ¿Qué hará tu equipo antes que nos volvamos a reunir?
- ¿Existe algo de demora o estorbo en tu equipo?
- ¿Estás a punto de poner algo en el camino de otro equipo?

#### **4. Descubriendo la potencialidad de un equipo:**

Tenemos un Sprint que dura 2(dos) semanas días hábiles de 8 horas de lunes a viernes.

8 horas x 10 días= 80 horas

Intervienen 3(Tres) Personas: Juan, María y Jose.

Si Jose falta un días tiene 8 horas menos:

Actor	Total	Presente
Juan	80	80
María	80	80
José	80	72
Total Potencialidad	240	<b>232</b>

Se puede observar que el equipo tuvo una potencialidad de 232 horas debido que José faltó un día.

## 2. Metodología Ágil X.P(Extreme Programing)

Utiliza desarrollo iterativo y participación del cliente a niveles "Extremos"

### Características

- ✓ En la programación extrema todos los requerimientos se expresan como escenarios llamados historias de usuario los cuales son implementados directamente como una serie de tareas.
- ✓ Se propone que los desarrolladores trabajen en pareja y desarrollan pruebas para cada tarea antes de escribir el código.
- ✓ Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema.



# Ingeniería de Software I

## Clase Número 5

### Metodología Ágiles

Dr. Sergio D. Conde  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Principio o práctica	Descripción
Planificación incremental	Los requerimientos se registran en tarjetas de historias y las historias a incluir en una entrega se determinan según el tiempo disponible y su prioridad relativa. Los desarrolladores dividen estas Historias en «Tareas» de desarrollo. Véanse las Figuras 17.6 y 17.7.
Entregas pequeñas	El mínimo conjunto útil de funcionalidad que proporcione valor de negocio se desarrolla primero. Las entregas del sistema son frecuentes e incrementalmente añaden funcionalidad a la primera entrega.
Diseño sencillo	Sólo se lleva a cabo el diseño necesario para cumplir los requerimientos actuales.
Desarrollo previamente probado	Se utiliza un sistema de pruebas de unidad automatizado para escribir pruebas para nuevas funcionalidades antes de que éstas se implementen.
Refactorización	Se espera que todos los desarrolladores refactoricen el código continuamente tan pronto como encuentren posibles mejoras en el código. Esto conserva el código sencillo y mantenible.
Programación en parejas	Los desarrolladores trabajan en parejas, verificando cada uno el trabajo del otro y proporcionando la ayuda necesaria para hacer siempre un buen trabajo.
Programación en parejas	Los desarrolladores trabajan en parejas, verificando cada uno el trabajo del otro y proporcionando la ayuda necesaria para hacer siempre un buen trabajo.
Propiedad colectiva	Las parejas de desarrolladores trabajan en todas las áreas del sistema, de modo que no desarrollen islas de conocimientos y todos los desarrolladores posean todo el código. Cualquiera puede cambiar cualquier cosa.
Integración continua	En cuanto acaba el trabajo en una tarea, se integra en el sistema entero. Después de la integración, se deben pasar al sistema todas las pruebas de unidad.
Ritmo sostenible	No se consideran aceptables grandes cantidades de horas extras, ya que a menudo el efecto que tienen es que se reduce la calidad del código y la productividad a medio plazo.
Cliente presente	Debe estar disponible al equipo de la XP un representante de los usuarios finales del sistema (el cliente) a tiempo completo. En un proceso de la programación extrema, el cliente es miembro del equipo de desarrollo y es responsable de formular al equipo los requerimientos del sistema para su implementación.

### Roles de X.P.

Los roles de acuerdo con la propuesta original de Beck son:

- ✓ **Programador:** El programador escribe las pruebas unitarias y produce el código del sistema.
  - ✓ **Cliente:** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
  - ✓ **Encargado de pruebas (Tester).** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
  - ✓ **Encargado de seguimiento (Tracker).** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
- Entrenador (Coach).** Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- ✓ **Consultor.** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
  - ✓ **Gestor (Big boss).** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

**Diferencia entre Metodologías:**

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1. Diferencias entre metodologías ágiles y no ágiles



### 3. Metodología Ágil Kanban:

Kanban (en kanji, donde kan significa "visual", y ban significa "tarjeta" o "tablero") es un concepto de producción justo-a-tiempo (JIT).

El kanban es una tarjeta física que se utiliza en el Sistema de Producción para soportar un control productivo descentralizado por demanda.

Kanban es una herramienta proveniente de la filosofía Lean, de tipo “pull”, lo que significa que los recursos deciden cuándo y cuánto trabajo se comprometen a realizar.

Kanban refleja a través de un tablero la situación de cada tarea del proyecto, con una visualización comprensible para todo el mundo, y permitiendo a la gente pensar por sí mismos y sacar las expectativas razonables sin necesidad de una interpretación excesiva por escrito del estado.

Los siguientes elementos son esenciales:

**Stream:** visualizar el flujo real de trabajo a través de los equipos, y la comprensión de lo que están haciendo en cada paso. El tablero de Kanban consiste en una serie de columnas que representan las diferentes etapas de trabajo.

**Contenido:** representar los tipos diferentes de trabajo, y asegurarse de que todo el trabajo se visualiza de alguna manera que sea comprensible para todos los involucrados.

Normalmente, esto se hace con las tarjetas que se colocan en las columnas definidas anteriormente.

**Límites:** definir claramente los límites de la cantidad de trabajo que el equipo puede soportar dentro de cada etapa del stream. Estos límites denotan el punto a partir del cual un equipo no logra progresos adicionales al tratar de tomar un trabajo adicional. Estos límites se definen como el número de tarjetas permitidas en cada columna.

**Políticas:** estas políticas toman la forma de declaraciones comprobables que declaran en forma explícita y específica, qué significa avanzar una tarea (tarjeta) de una columna a la siguiente. Las políticas ayudan a crear confianza en la visualización de todos los interesados.

### **3.1. Trabajo en Kanban**

#### **1. Visualiza el Workflow:**

- ✓ Divide el trabajo en piezas, y escribe cada una de ellas en tarjetas que se colocan en el tablero (“kanban board”)
- ✓ Utiliza columnas con nombres para ilustrar dónde se encuentra dentro del proceso o workflow cada ítem o tarea.

**2. Limita el “WIP” (work in progress):** asigna límites específicos de cuantos ítems pueden estar siendo procesados a la vez dentro de cada columna del workflow.

**3. Mide el “Lead Time” (también llamado “ciclo de tiempo”):** es el tiempo promedio para completar un ítem, o sea, que el mismo haya pasado por todas las columnas del workflow hasta llegar al final. El Kanban tiene por objetivo optimizar el proceso para hacer el lead time lo más pequeño y predecible que se pueda.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Lo único que Kanban prescribe es que el workflow pueda visualizarse y ser comprendido por todos.

El propósito es crear un flujo continuo a través del sistema y minimizar el lead time o cycle time.

Cada columna representa un estado dentro del workflow, o un buffer entre dos estados del workflow. La cantidad de columnas la determina cada uno de acuerdo a su experiencia y necesidad. Cuando se ha llegado al límite Kanban de una columna y no se tiene otra cosa que hacer, comience a buscar los cuellos de botella y solucionarlos. Si no existen es un indicativo que el límite podría ser demasiado bajo. Si se nota que los ítems permanecen en una columna por largo tiempo, es un indicativo de que el límite es demasiado alto. La regla es la siguiente:

- ✓ Límite kanban demasiado bajo = recursos desperdiciados = baja productividad
- ✓ Límite kanban demasiado alto = tareas en cola = erróneo lead time.

### **Beneficios de Kanban**

- ✓ La visibilidad de los cuellos de botella (“bottlenecks”) en tiempo real. Esto permite al equipo que colabore para optimizar la cadena de valor en lugar de ocuparse cada uno de su parte. El síntoma de un cuello de botella es cuando vemos que la columna X está repleta de ítems mientras que la columna que sigue se encuentra vacía.
- ✓ Permite una evolución más gradual para pasar de waterfall a un desarrollo ágil de software, para aquellas empresas que aún no se animan a intentar dar ese paso.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

✓ Con Kanban no se usan iteraciones. La realidad es que las iteraciones son opcionales.

✓ Con Kanban no hay que estimar. La realidad es que las estimaciones son opcionales también.

Para tener una idea clara sobre estimación y velocidad en Kanban repasemos muy brevemente lo relacionado con Scrum.

Uno de los aspectos únicos de un proyecto Agile es que la carga de trabajo para cada iteración se determina al comienzo de cada iteración. Sólo hay necesidad de planificar para la iteración actual y no una planificación anticipada pesada y completa como el método tradicional.

Esto permite que el proyecto Agile sea mucho más flexible

Al comienzo de cada iteración se realiza una reunión entre “product owner” y el equipo del proyecto para determinar la carga de trabajo para la nueva iteración. Durante la reunión, se evalúa la cartera de los requisitos y se selecciona la siguiente serie de casos de uso (o historias de usuario) que son de más alta prioridad. El nivel de esfuerzo para cada caso de uso debería haber sido asignado cuando se añadió al backlog. Allí la estimación no debe ser horas de esfuerzo, sino más bien una serie de "story points."

Los story points son números arbitrarios utilizados para estimar el tamaño relativo de un caso de uso. La escala exacta es diferente de un equipo a otro. Por ejemplo, si un caso de uso tiene 10 puntos, otro caso de uso que es dos veces más complejo debería tener 20 puntos. La estimación total de los puntos incluye el tiempo de análisis, diseño, construcción, prueba e implementación. Durante la reunión de planificación, el equipo del proyecto toma tantos story points como los que pueda completar dentro de

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

la iteración, dentro de los más priorizados. Por supuesto, las prioridades pueden cambiar durante el proyecto.

Cuando esto sucede en un proyecto Agile, el cambio de prioridades es simplemente una cuestión de reasignar el orden de los casos de uso en el backlog.

Es importante que el equipo del proyecto pueda determinar rápidamente la cantidad de trabajo que puede completar en cada iteración. Esto permitirá que la carga de trabajo se mantenga relativamente uniforme de iteración a iteración. Si el equipo del proyecto considera que no fue capaz de completar una serie de casos de uso en la iteración anterior, se ponen de acuerdo para tomar menos trabajo en la siguiente iteración. Del mismo modo, si el equipo se da cuenta de que podían haber hecho más trabajo en una iteración, deberían tomar más trabajo en la siguiente iteración. Veamos un ejemplo:

Digamos que el equipo inicia el proyecto y estima que pueden completar 40 puntos en una iteración. Sin embargo, a medida que avanza la iteración, se encuentran con que tienen que trabajar muchas horas extras y que no pueden completar todo el trabajo durante la iteración. En la siguiente iteración, se decide entonces escoger sólo 32 story points. El equipo tiene éxito, pero durante la revisión final de la iteración se dan cuenta que podría haber hecho más. En la siguiente iteración deciden tomar 36 puntos de historia. Después de la iteración el equipo decide que este es un ritmo exigente pero totalmente factible para ellos. Por lo tanto, a partir de entonces el equipo toma 36 puntos de historia de casos de uso por iteración.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Este ritmo al que el equipo puede completar los puntos de historia del backlog se conoce como la velocidad del equipo. Los casos de uso que son seleccionados para una iteración necesita ser completados en esa iteración. En un proyecto ágil es importante mantenerse en un ciclo de repetición constante. Si el caso de uso no está listo cuando la iteración está lista para pasarla a producción, el código de dicho caso de uso debe ser retirado para que el resto de la iteración pueda ser liberada a tiempo. No hay retrasos en la fecha de finalización de la iteración. El equipo se centra en completar todos los story points en la fecha comprometida una y otra vez. Este ritmo constante en cada iteración también se conoce como el ritmo del equipo o cadencia.

En Kanban, la estimación no está prescrita, por este motivo algunos equipos eligen hacer estimaciones y medir la velocidad al utilizando un tamaño de los requerimientos funcionales a cumplir: S, M, L, XL.

## **Metodología Ágil SCUMBAN**

### Metodología Ágil Scrumban

Scrum aporta el enfoque de trabajo en equipos pequeños, obsesionados por estar muy cerca y dar valor continuamente a los usuarios y a los clientes finales de forma efectiva. Kanban, por su parte, promueve el enfoque de eficiencia de procesos que nos permite estar constantemente en un círculo virtuoso de mejora continua. Es eficaz y efectivo en el mismo párrafo demostrado mejorar el compromiso de las personas.

Scrumban es, por tanto, ese equilibrio perfecto entre eficacia y eficiencia. Cuando se aplica, se garantiza la posibilidad de navegar por la incertidumbre y de, a la vez, buscar siempre la mejora continua mediante la eficiencia.

La metodología Scrumban nace de la combinación de principios de los métodos ágiles de gestión de proyectos más importantes en la actualidad: Scrum y Kanban. Aunque en principio pueden parecer iguales, las dos estrategias de gestión presentan diferencias en la manera de ejecutar el proyecto. Es por eso que el novedoso plan Scrumban se encarga de combinar aquellos elementos que resultan complementarios. Por ejemplo, una de las combinaciones más usadas en el plano empresarial, es la de gestionar las tareas previstas con el método Scrum y planificar los errores con el método Kanban. Sin embargo, la mezcla de una y otra implica una

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

nueva manera de gestión. ¿Qué cambia en la metodología Scrumban?  
¿Cómo aplicarla a un plan de empresa?

Al margen de lo que puede ganar o perder una empresa al aplicar este método, la premisa inicial de Scrumban es emplear lo mejor de Scrum y Kanban. Es decir, aquello que al combinarlo sirva para mejorar la productividad de un plan de empresa. En el terreno de la ejecución, el objetivo de Scrumban es llevar a cabo una secuencia de mejoras evolutivas a través de prácticas eficientes. El flujo de trabajo sigue siendo el mismo de Kanban (etapas relacionadas entre sí), aunque con la inclusión de algunos elementos de Scrum como las reuniones diarias de 15 minutos entre el grupo de trabajo y el gestor o los análisis retrospectivos para incorporar mejoras al proceso. Por ello, existen ciertos proyectos que se adecúan mejor al método mixto. En general, se trata de aquellos con un mayor nivel de complejidad. Algunos ejemplos son:

Proyectos de mantenimiento: aquellos en los que resulta indispensable la presentación de resultados de forma parcial para seguir avanzando.

Proyectos en los que los requisitos varíen con frecuencia: aquellos en los que el cliente no tiene fijadas las condiciones y expectativas del proyecto y éstas se van introduciendo a lo largo de las distintas etapas.



**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Proyectos en los que surjan errores de ejecución: aquellos en los que se deba replantear el método usado y analizar retrospectivamente la evolución de las tareas.

En cuanto al plan de etapas, el mapa o interfaz de la aplicación (en caso de que sea informática) ya no se limita a nombrar las tareas con los rótulos «sin comenzar», «en progreso» o «finalizadas» de Kanban, sino que, tras la revisión de Scrum, añade otras categorías, como por ejemplo «probadas» o «entregadas». Incluso, en algunos casos se recomienda la anexión de casillas con la descripción del contexto en el que se han realizado dichas tareas, algo muy propio del método Scrum.

#### Beneficios de la metodología Scrumban

Permite conocer en estado real el proceso de ejecución del proyecto.

Introduce soluciones oportunas ante eventuales errores.

Permite un mayor análisis de tareas realizadas.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Características de Scrumban:

Metodología Ágil Scrumban

Scrum aporta el enfoque de trabajo en equipos pequeños, obsesionados por estar muy cerca y dar valor continuamente a los usuarios y a los clientes finales de forma efectiva. Kanban, por su parte, promueve el enfoque de eficiencia de procesos que nos permite estar constantemente en un círculo virtuoso de mejora continua. Es eficaz y efectivo en el mismo párrafo demostrado mejorar el compromiso de las personas.

Scrumban es, por tanto, ese equilibrio perfecto entre eficacia y eficiencia. Cuando se aplica, se garantiza la posibilidad de navegar por la incertidumbre y de, a la vez, buscar siempre la mejora continua mediante la eficiencia.

La metodología Scrumban nace de la combinación de principios de los métodos ágiles de gestión de proyectos más importantes en la actualidad: Scrum y Kanban. Aunque en principio pueden parecer iguales, las dos estrategias de gestión presentan diferencias en la manera de ejecutar el proyecto. Es por eso que el novedoso plan Scrumban se encarga de combinar aquellos elementos que resultan complementarios. Por ejemplo, una de las combinaciones más usadas en el plano empresarial, es la de gestionar las tareas previstas con el método Scrum y planificar los errores con el método Kanban. Sin embargo, la mezcla de una y otra implica una

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

nueva manera de gestión. ¿Qué cambia en la metodología Scrumban?  
¿Cómo aplicarla a un plan de empresa?

Al margen de lo que puede ganar o perder una empresa al aplicar este método, la premisa inicial de Scrumban es emplear lo mejor de Scrum y Kanban. Es decir, aquello que al combinarlo sirva para mejorar la productividad de un plan de empresa. En el terreno de la ejecución, el objetivo de Scrumban es llevar a cabo una secuencia de mejoras evolutivas a través de prácticas eficientes. El flujo de trabajo sigue siendo el mismo de Kanban (etapas relacionadas entre sí), aunque con la inclusión de algunos elementos de Scrum como las reuniones diarias de 15 minutos entre el grupo de trabajo y el gestor o los análisis retrospectivos para incorporar mejoras al proceso. Por ello, existen ciertos proyectos que se adecúan mejor al método mixto. En general, se trata de aquellos con un mayor nivel de complejidad. Algunos ejemplos son:

Proyectos de mantenimiento: aquellos en los que resulta indispensable la presentación de resultados de forma parcial para seguir avanzando.

Proyectos en los que los requisitos varíen con frecuencia: aquellos en los que el cliente no tiene fijadas las condiciones y expectativas del proyecto y éstas se van introduciendo a lo largo de las distintas etapas.

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Proyectos en los que surjan errores de ejecución: aquellos en los que se deba replantear el método usado y analizar retrospectivamente la evolución de las tareas.

En cuanto al plan de etapas, el mapa o interfaz de la aplicación (en caso de que sea informática) ya no se limita a nombrar las tareas con los rótulos «sin comenzar», «en progreso» o «finalizadas» de Kanban, sino que, tras la revisión de Scrum, añade otras categorías, como por ejemplo «probadas» o «entregadas». Incluso, en algunos casos se recomienda la anexión de casillas con la descripción del contexto en el que se han realizado dichas tareas, algo muy propio del método Scrum.

### Beneficios de la metodología Scrumban

Permite conocer en estado real el proceso de ejecución del proyecto.

Introduce soluciones oportunas ante eventuales errores.

Permite un mayor análisis de tareas realizadas.

### **Metodología Ágil Scrumban**

Scrum aporta el enfoque de trabajo en equipos pequeños, obsesionados por estar muy cerca y dar valor continuamente a los usuarios y a los clientes finales de forma efectiva. Kanban, por su parte, promueve el enfoque de eficiencia de procesos que nos permite estar constantemente en un círculo virtuoso de mejora continua. Es eficaz y efectivo en el mismo párrafo demostrado mejorar el compromiso de las personas.

Scrumban es, por tanto, ese equilibrio perfecto entre eficacia y eficiencia. Cuando se aplica, se garantiza la posibilidad de navegar por la incertidumbre y de, a la vez, buscar siempre la mejora continua mediante la eficiencia.

La metodología Scrumban nace de la combinación de principios de los métodos ágiles de gestión de proyectos más importantes en la actualidad: Scrum y Kanban. Aunque en principio pueden parecer iguales, las dos estrategias de gestión presentan diferencias en la manera de ejecutar el proyecto. Es por eso que el novedoso plan Scrumban se encarga de combinar aquellos elementos que resultan complementarios. Por ejemplo, una de las combinaciones más usadas en el plano empresarial, es la de gestionar las tareas previstas con el método Scrum y planificar los errores con el método Kanban. Sin embargo, la mezcla de una y otra implica una nueva manera de gestión. ¿Qué cambia en la metodología Scrumban? ¿Cómo aplicarla a un plan de empresa?

Al margen de lo que puede ganar o perder una empresa al aplicar este método, la premisa inicial de Scrumban es emplear lo mejor de Scrum y

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

Kanban. Es decir, aquello que al combinarlo sirva para mejorar la productividad de un plan de empresa. En el terreno de la ejecución, el objetivo de Scrumban es llevar a cabo una secuencia de mejoras evolutivas a través de prácticas eficientes. El flujo de trabajo sigue siendo el mismo de Kanban (etapas relacionadas entre sí), aunque con la inclusión de algunos elementos de Scrum como las reuniones diarias de 15 minutos entre el grupo de trabajo y el gestor o los análisis retrospectivos para incorporar mejoras al proceso. Por ello, existen ciertos proyectos que se adecúan mejor al método mixto. En general, se trata de aquellos con un mayor nivel de complejidad. Algunos ejemplos son:

**Proyectos de mantenimiento:** aquellos en los que resulta indispensable la presentación de resultados de forma parcial para seguir avanzando.

**Proyectos en los que los requisitos varíen con frecuencia:** aquellos en los que el cliente no tiene fijadas las condiciones y expectativas del proyecto y éstas se van introduciendo a lo largo de las distintas etapas.

**Proyectos en los que surjan errores de ejecución:** aquellos en los que se deba replantear el método usado y analizar retrospectivamente la evolución de las tareas.

En cuanto al plan de etapas, el mapa o interfaz de la aplicación (en caso de que sea informática) ya no se limita a nombrar las tareas con los rótulos «sin comenzar», en progreso o finalizadas de Kanban, sino que, tras la revisión de Scrum, añade otras categorías, como por ejemplo «probadas» o

**Ingeniería de Software I**  
**Clase Número 5**  
**Metodología Ágiles**  
**Dr. Sergio D. Conde**  
[sconde@unaj.edu.ar](mailto:sconde@unaj.edu.ar)  
[drcondesergio@gmail.com](mailto:drcondesergio@gmail.com)

«entregadas». Incluso, en algunos casos se recomienda la anexión de casillas con la descripción del contexto en el que se han realizado dichas tareas, algo muy propio del método Scrum.

### **Beneficios de la metodología Scrumban**

Permite conocer en estado real el proceso de ejecución del proyecto.

Introduce soluciones oportunas ante eventuales errores.

Permite un mayor análisis de tareas realizadas.

### **Características de Scrumban:**

	Scrum	Kanban	Scrumban
Procesos	Iterativo e incremental	Continuo	Iterativo e Incremental de forma continua
Personas	Las personas son el centro	Las personas son el pilar	Equipo motivado con personas como pilar y en el centro.
Producto	Foco en la efectividad	Foco en la eficiencia	Balance inteligente entre Efectividad y Eficiencia
Organización	Mejora continua del Producto	Mejora continua del proceso	Mejora continua del producto y del proceso