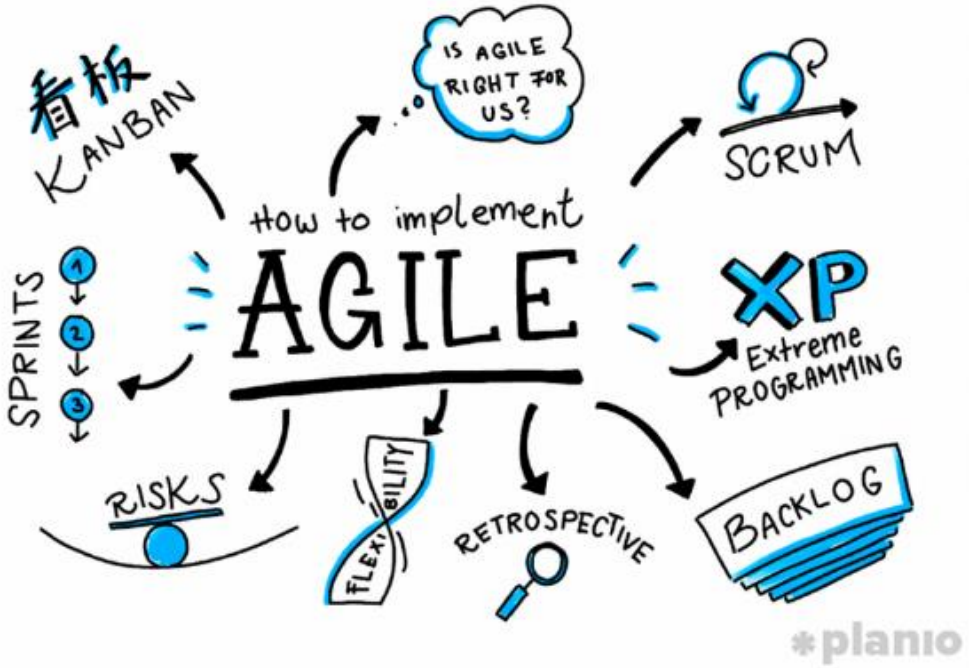
IMPLANTACIÓN DE SISTEMAS ERP-CRM

METODOLOGÍAS ÁGILES



INDICE

Ciclo de Vida del Software

El desarrollo de software va unido a un ciclo de vida compuesto por una serie de etapas que comprenden todas las actividades, desde el momento en que surge la idea de crear un nuevo producto software, hasta aquel en que el producto deja definitivamente de ser utilizado por el último de sus usuarios.

Es importante delimitar en cierta manera estas las etapas y ver la forma en la que se afrontan (existen diversos modelos de ciclo de vida)

Etapas de un Ciclo de Vida clásico

Definición de necesidades: Toma de *requerimientos y especificaciones*. Se crea un documento con las necesidades del cliente tras una serie de entrevistas.

Análisis: Determinar qué elementos intervienen en el sistema a desarrollar, así como su estructura, relaciones, evolución en el tiempo, detalle de sus funcionalidades, ... que van a dar una descripción clara de qué sistema vamos a construir, qué funcionalidades va a aportar y qué comportamiento va a tener.

Diseño: En esta etapa se determina ¿cómo debe ser construido el sistema?; aquí se definirán en detalle entidades y relaciones de las bases de datos, casos de uso (user case), se seleccionará el lenguaje más adecuado, el Sistema Gestor de Bases de Datos a utilizar en su caso, librerías, configuraciones hardware, redes, etc.).

Implementación: Codificación de algoritmos y estructuras de datos definidos en las etapas anteriores, utilizando el correspondiente lenguaje de programación y/o para un determinado sistema gestor de bases de datos.

Pruebas: El objetivo de estas pruebas es garantizar que el sistema ha sido desarrollado correctamente, sin errores de diseño y/o programación. Es conveniente que sean planteadas al menos tanto a nivel de cada módulo (aislado del resto), como de integración del sistema (según sea la naturaleza del proyecto en cuestión se podrán tener en cuenta pruebas adicionales, p.ej. de rendimiento).

Validación: Esta etapa tiene como objetivo la verificación de que el sistema desarrollado cumple con los requisitos expresados inicialmente por el cliente y que han dado lugar al presente proyecto (para esta fase también es interesante contar con los casos de uso (use cases), generados a través de las correspondientes fases previas, que servirán de guía para la verificación de que el sistema cumple con lo descrito por estos.

Mantenimiento y evolución:

Finalmente la aplicación resultante se encuentra ya en fase de producción (en funcionamiento para el cliente, cumpliendo ya los objetivos para los que ha sido creada). A partir de este momento se entra en la etapa de mantenimiento, que supondrá ya pequeñas operaciones tanto de corrección como de mejora de la aplicación (p.ej. mejora del rendimiento), así como otras de mayor importancia, fruto de la propia evolución (p.ej. nuevas opciones para el usuario debidas a nuevas operaciones contempladas para el producto).

La mayoría de las veces en que se desarrolla una nueva aplicación, se piensa solamente en un ciclo de vida para su creación, olvidando la posibilidad de que esta deba sufrir modificaciones futuras (que tendrán que producirse con casi completa seguridad para la mayor parte de los casos).

METODOLOGÍA DE IMPLANTACIÓN

Cuando nos enfrentamos a un nuevo proyecto de implantación debemos realizar un proceso estructurado y metodológico para lleva a buen término el desarrollo, este proceso se denomina metodología de implantación.

Tras muchos estudios, metodologías, informes de implantación y demás planes realizados, la conclusión a la que se llega siempre es que una implantación desorganizada llevará al fracaso en el proyecto.

No hay una metodología o mecanismo de implantación genérico que funcione en todos los proyectos de implantación, ya que cada uno es particular y debemos adaptarnos a sus características especiales.

EVOLUCIÓN

Desde la aparición de los primeros programas computacionales hasta la actualidad, el desarrollo de software ha pasado por un proceso evolutivo tan marcado y propio de cualquier área tecnológica.

Primero se gestaron **metodologías las tradicionales**, que nacieron debido a los excesivos costos, entregas fuera de tiempo y unas expectativas poco claras por parte de los clientes, se formalizan como compendios de criterios, procedimientos, técnicas, herramientas y documentos para guiar a los desarrolladores de software en sus esfuerzos por la implementación de sistemas de información.

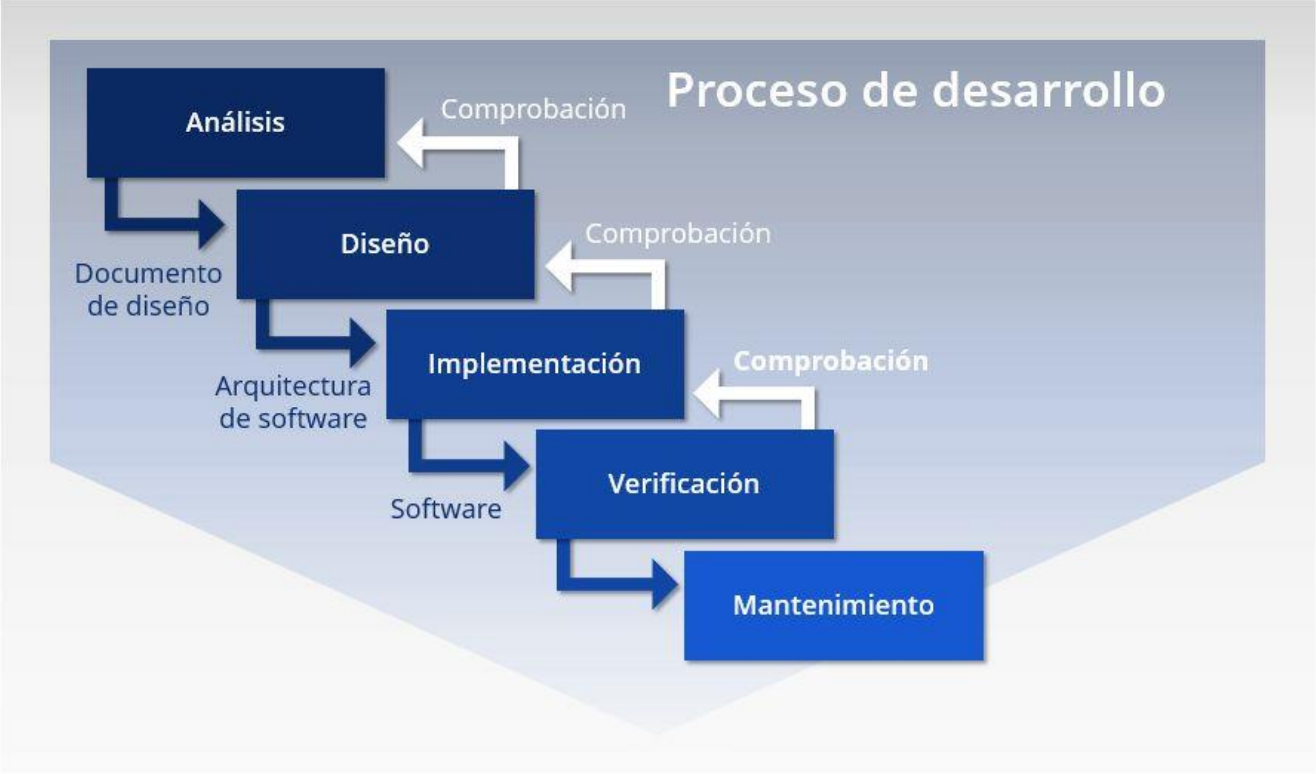
Años más tarde, debido al exceso de controles, énfasis en la arquitectura, formalidad de la documentación, la llegada de internet y la necesidad de contextos cambiantes y rápida adaptación, llegaron las **metodologías ágiles, que se conciben como una práctica orientada a las personas y no a los procesos.**

Las **metodologías tradicionales** se centran en un férreo control de los procesos, definiendo de forma rigurosa las actividades a realizar, además de marcar las herramientas y protocolos que se deben utilizar.

Se caracterizan por ser **rígidas y guiadas por la documentación** que se genera en cada una de las fases de los proyectos. Sin embargo, este enfoque no resulta el más adecuado para muchos proyectos actuales, en los que el entorno del sistema Les muy cambiante y se exige reducir drásticamente los tiempos de desarrollo sin perder la calidad.

METODO EN CASCADA (metodología waterfall)

La cascada se basa en que los equipos sigan una secuencia de pasos y nunca avancen hasta que se haya completado la fase anterior. Esta estructura es apta para proyectos más pequeños con entregables que son fáciles de definir desde el inicio.



METODO EN CASCADA (metodología waterfall)

Las **ventajas** más destacadas de este método en cascada son las siguientes:

- Es simple y sencillo de usar.
- Es fácil de administrar porque cada fase consta de entregables concretos.
- El proceso es predecible, ya que todos tienen una idea anterior acerca de cómo se evolucionará. Los clientes conocen el cronograma y su resultado final.
- Las fases se ejecutan y completan una a la vez.
- Las metodologías de desarrollo de software son perfectas para los proyectos que cuentan con requisitos específicos.
- Si hay mucha rotación de personal, no incidirá demasiado en el proyecto.

METODO EN CASCADA (metodología waterfall)

Desventajas

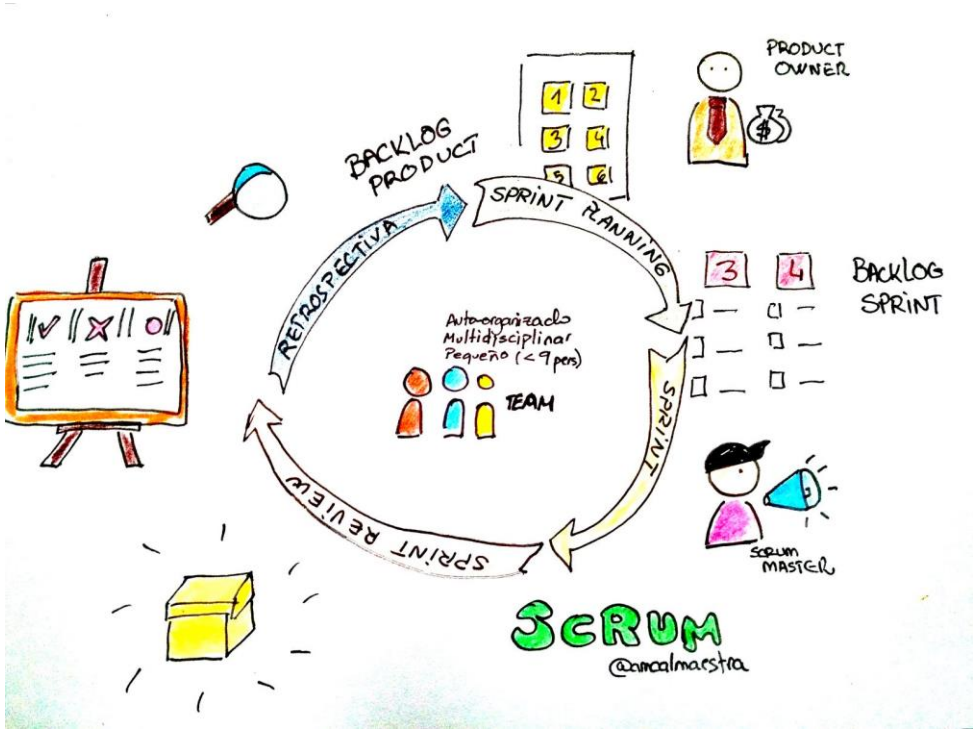
- En caso de encontrar errores o cambios, cada proyecto se deberá empezar con un nuevo código.
- En el momento en que el producto esté en etapa de prueba, no será fácil retroceder y modificar aspectos que no estén formulados debidamente.
- No es un método adecuado para proyectos que sean largos con requisitos que sean más complejos.
- Tampoco es el ideal para proyectos que son volátiles, donde los cambios están a la orden del día.

Las **metodologías ágiles** surgen como alternativa a las metodologías tradicionales ya que son demasiado burocráticas y rígidas para las actuales características del mercado.

Ventajas de adoptar metodologías ágiles:

- Son una de las claves de éxito de las empresas que lanzan una apuesta decidida por la transformación digital. Ya que se gana en **flexibilidad e inmediatez** dado que adaptan la forma de trabajo a las cambiantes condiciones de cada proyecto.
- Permite reducir costes e incrementar la productividad.
- Permite gestionar los proyectos en un mundo VUCA dominado por los siguientes principios:
  - V de Volatilidad.
  - U de Incertidumbre *(uncertainty en inglés)*
  - C de Complejidad.
  - A de Ambigüedad

SCRUM


PRODUCT OWNER
BACKLOG PRODUCT
SPRINT PLANNING
RETROSPECTIVA
Auto-organizado
Multidisciplinar
Pequeño (<9 pers)
TEAM
BACKLOG SPRINT
SPRINT
SPRINT REVIEW
SCRUM MASTER
SCRUM
@amcalmarstra

Scrum es un proceso en el que se aplican de manera regular un conjunto de procesos para trabajar en equipo, y obtener el mejor resultado posible de un proyecto.

Dentro del ciclo de vida del desarrollo de sowftware (SDLC) ágil, el trabajo se divide en sprints, ciclos breves para el desarrollo, con el objetivo de producir un producto funcional al final de cada sprint.

Se caracteriza por:

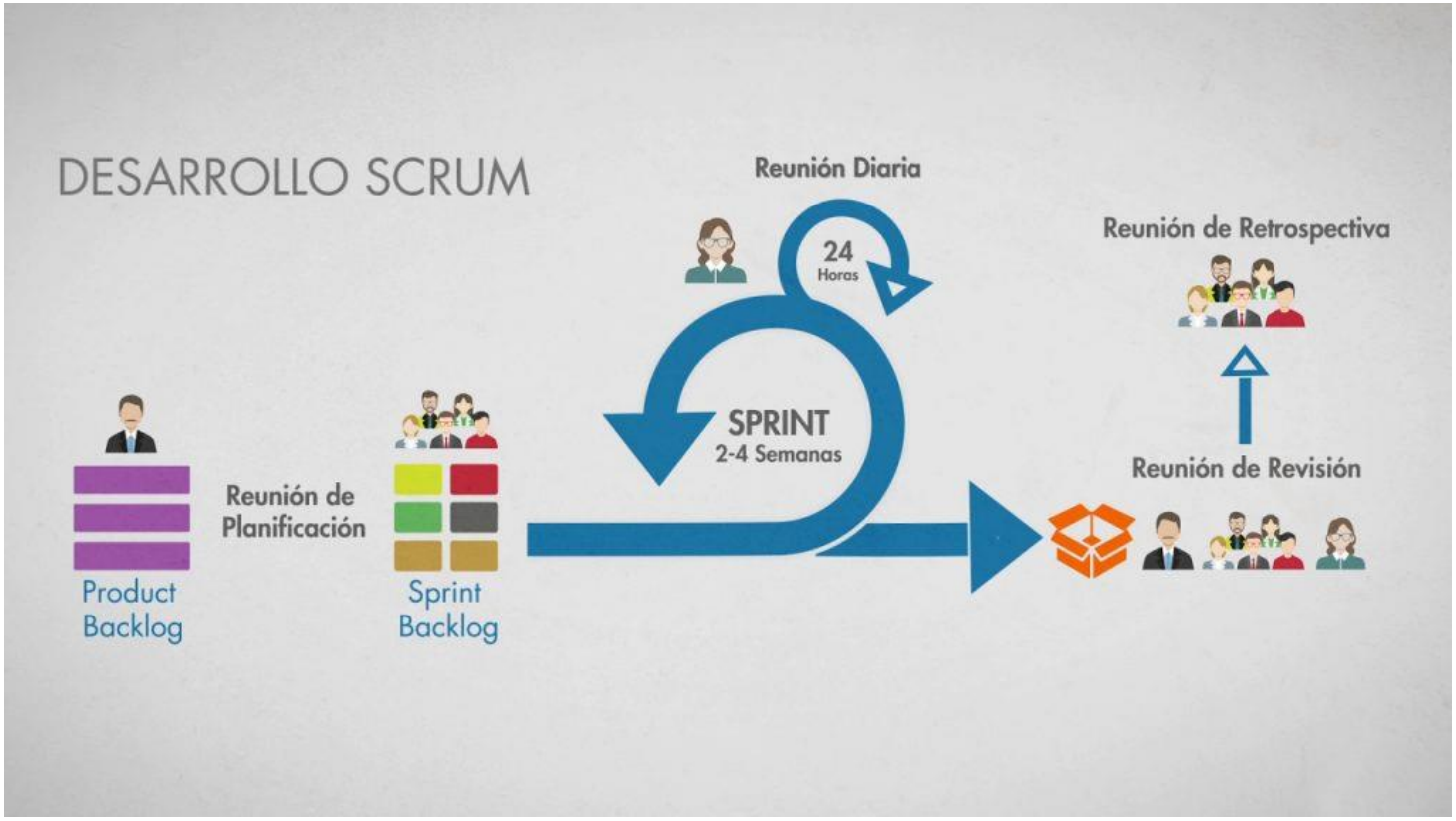
- Adoptar una idea total de la realización del producto, en lugar de la planificación y ejecución completa del producto.

- Enfocarse mas en las zonas de solapamiento, en lugar de realizar una tras otra en un ciclo de cascada.

Elementos que intervienen en el proceso:

1. El punto de partida es una **lista de objetivos** que deben priorizarse. Esta fase es esencial ya que dependiendo de lo que se priorice serán necesarios unos u otros requisitos. Será esta priorización la que determine las posteriores interacciones así como las entregas. El equipo que la realiza es el que determinará, también, qué tareas son necesarias y cómo se asignan los trabajos entre los miembros.
2. Los **sprints.** La metodología SCRUM se caracteriza porque se ejecuta en bloques de tiempo periódicos y cortos. Se denominan sprints y son la base del método. Cada sprint dura entre 2 y 4 semanas y da un resultado completo. Este será un entregable. Todos los sprints tienen un formato idéntico. Para que funcionen debe existir una comunicación continuada. Además, se representan los avances de manera visual.
3. **Feedback y reflexión**. Al terminar cada ciclo se presenta el resultado para se aprobado o rechazado. Es el momento de que el equipo involucrado reflexione sobre cómo se ha trabajado en el sprint, lo que ha ido bien y mal y, sobre todo, cómo mejorarlo.

Como ves, la metodología SCRUM se centra en la repetición de estos ciclos, los sprints, que tienen la siguiente estructura:



HERRAMIENTAS SCRUM:

1. REUNIONES
2. ROLES
3. DOCUMENTOS

REUNIONES

1. ***Planificación del Backlog: (Sprint***
   - Se definirá un documento en el que se reflejan los requisitos del sistema por prioridades.
   - También se define la planificación del **Sprint 0**, en la que se decidirá cuales van a ser los objetivos y el trabajo que hay que realizar para esa iteración.
   - Se obtiene un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint.

2. ***Daily Sprint (Reuniones diarias)***
   - La reunión comienza puntualmente a su hora.
   - 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:
     1.¿Qué has hecho desde ayer?
     2.¿Qué es lo que estás planeando hacer hoy?
     3.¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

3. ***Reunión de Planificación del Sprint***
   - Al inicio del ciclo Sprint (cada 15 o 30 días), una "Reunión de Planificación del Sprint" se lleva a cabo.
   - Seleccionar qué trabajo se hará.
   - Ocho horas como límite.
   - Al final del ciclo Sprint, dos reuniones se llevarán a cabo: la "Reunión de Revisión del Sprint" y la "Retrospectiva del Sprint".

4. ***Revisión del Sprint:***
   - Revisar el trabajo que fue completado y no completado.
   - Presentar el trabajo a los interesados.
   - El trabajo incompleto no puede ser demostrado.
   - Cuatro horas como límite.

4. *Retrospectiva del Sprint:*

Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado.

El propósito de la retrospectiva es realizar una mejora continua del proceso.

Esta reunión tiene un tiempo fijo de cuatro horas.

En ella, el equipo trabaja los siguientes aspectos:

- Una revisión de cómo funcionó el sprint en lo referente a las personas implicadas las herramientas utilizadas.
- Se identifican los errores y se definen las potenciales mejoras que los subsanen

ROLES

Para que este método funcione se han establecido una serie de roles que son los que garantizan que el método se cumple.

Hay 3 roles principales y uno auxiliar.

A.- Product owner
B.- Scrum Master
C.- Equipo de desarrollo
D.- Stakeholders

Product owner

Se trata de la persona que está en contacto directo con el cliente. Por tanto, tiene una tarea importante como interlocutor con todos los stakeholders del proyecto. Es quien conoce las peticiones y requerimientos de los clientes. Pero su labor más importante es maximizar el valor del trabajo del equipo de desarrollo.

Sólo debe haber un product owner para que los sprints funcionen. Tiene la responsabilidad de mantener el Product Backlog bien estructurado, detallado y priorizado.

Scrum Master

Es el responsable de que la metodología Scrum sea comprendida y aplicada en la empresa. Es el manager de Scrum. Por eso, su principal labor es ayudar en la adopción de esta metodología en todos los equipos. Para ello, no sólo formará al equipo sino que, también, debe ser el facilitador en todas las reuniones.
Sus tareas básicas son:

- Gestionar el proceso Scrum para que aporte valor a la organización que lo adopta.
- Eliminar impedimentos.

Equipo de desarrollo

Se trata de las personas encargadas de realizar las tareas. Debe ser un equipo multifuncional y autoorganizado y tienen la responsabilidad compartida de haber realizado el trabajo o no haberlo conseguido. De ahí que no se debe intervenir en sus dinámicas de funcionamiento. Es el equipo el que decide gestionarse internamente de una manera concreta.

Y, por eso, tendrá que rendir cuentas como uno solo.

Stakeholders

Existen unos roles auxiliares que son aquellos que no tienen un rol formal y no se involucran en el proceso. Pero que, sin embargo, su opinión debe ser tomada en cuenta. Pueden ser desde expertos en negocio que pueden asesorar hasta clientes o proveedores. Algunos de ellos participan durante la revisión del sprint.

DOCUMENTOS

Product backlog

- El product backlog es un documento de alto nivel para todo el proyecto.
- Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. del producto a desarrollar.
- Se estructura en TEMAS, EPICS, HISTORIAS DE USUARIO, TAREAS, ETC..,
- Se prioriza por historias de usuario, según su valor para el negocio (business value).
- Una historia de usuario se definiría

> **Como** [rol del usuario], **quiero** [objetivo], **para poder** [beneficio].

- Es abierto y cualquiera puede modificarlo.

DOCUMENTOS

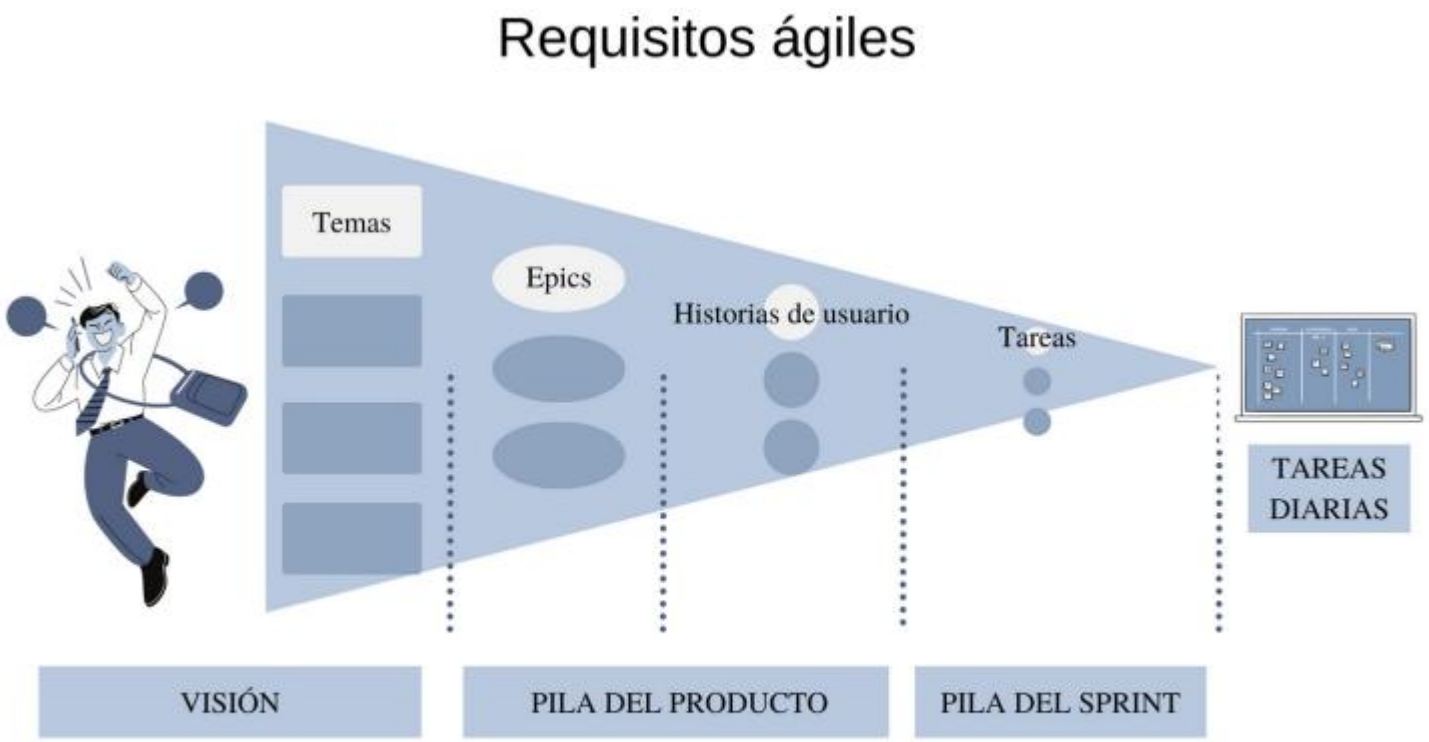
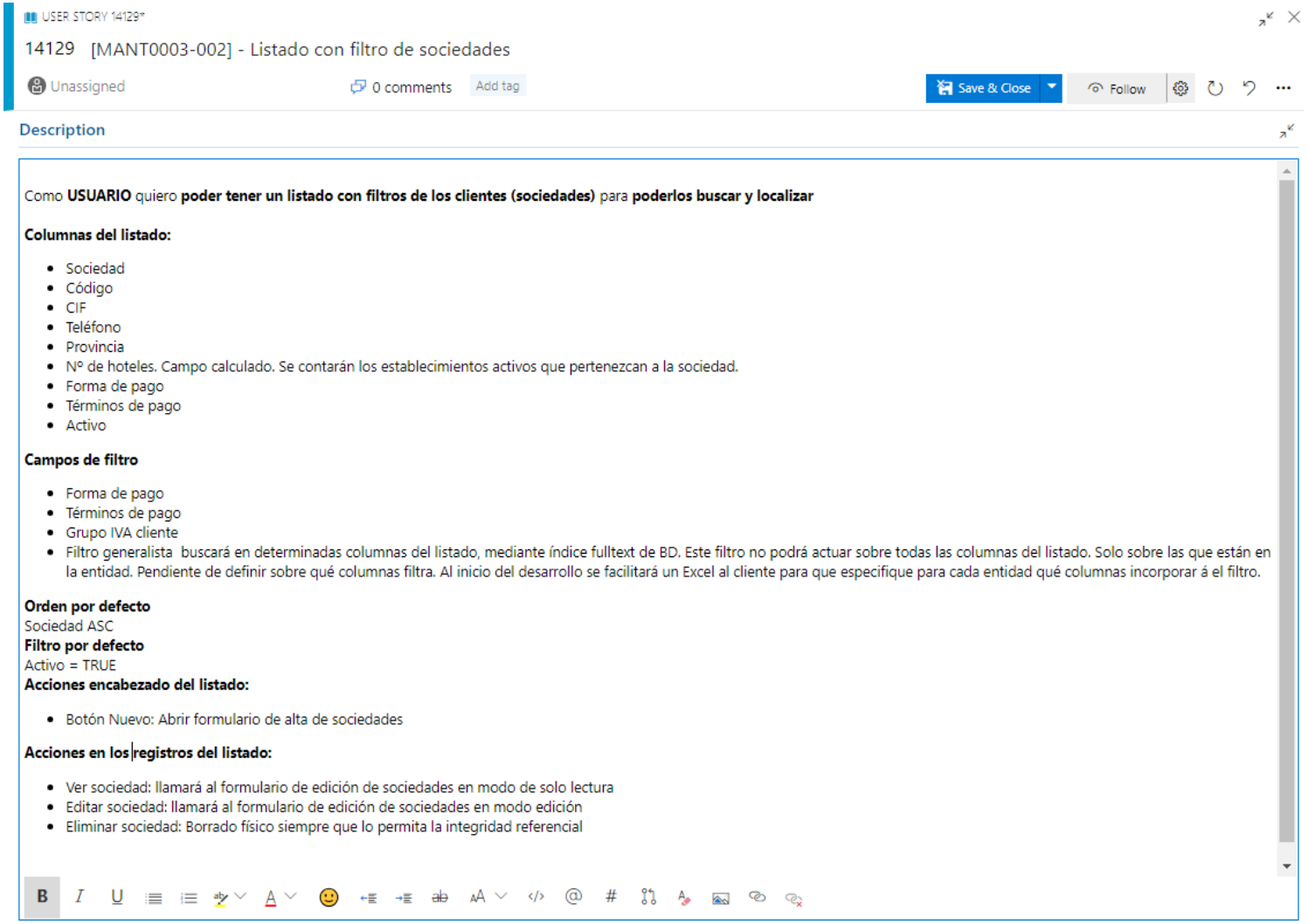


Ilustración 3: gráfico con los cuatro niveles de tamaño con que trata los requisitos la gestión ágil.

Ejemplo de Blacklog

Epic	**OKESCENA: Mantenimientos entidades principales	New
Feature	**[MANT0001] - Usuarios	New
Feature	[MANT0002] – Cadenas	New
User Story	[MANT0002-002] - Listado de cadenas	QA Pending
User Story	[MANT0002-003] – CRUD de cadenas: Datos generales	QA Pending
User Story	[MANT0002-004] – CRUD de Cadenas: Documentos adjuntos	New
Feature	[MANT0003] – Sociedades (Clientes)	New
User Story	[MANT0003-002] - Listado con filtro de sociedades	QA Pending
Task	Implementar modelo	Closed
Task	Implementar listado	Closed
Task	Implementar botones de acción	Closed
Task	Implementar filtrado	Closed
Task	Implementar campo calculado 'Nº de hoteles'	Closed
Task	Implementar campos de filtro e indice fulltext	Closed
User Story	[MANT0003-003-001] - CRUD de sociedades: Datos generales	QA Pending
Task	Implementar CRUD en backend y frontend	Closed
Task	Editar formularios alta y edición	Closed
Task	Implementar botones de acción	Closed
Task	Implementar lógica mensaje de confirmación al cerrar formulario	Closed
Task	Implementar campos con búsqueda en otras entidad	Closed
User Story	[MANT0003-003-002] - CRUD de sociedades: Listado de establecimientos (hoteles)	New
User Story	[MANT0003-005] – Sociedades: Documentos adjuntos	New

Ejemplo de historia de usuario



USER STORY 14129*

14129 [MANT0003-002] - Listado con filtro de sociedades

Unassigned | 0 comments | Add tag | Save & Close | Follow

Description

Como **USUARIO** quiero **poder tener un listado con filtros de los clientes (sociedades)** para **poderlos buscar y localizar**

Columnas del listado:

- Sociedad
- Código
- CIF
- Teléfono
- Provincia
- Nº de hoteles. Campo calculado. Se contarán los establecimientos activos que pertenezcan a la sociedad.
- Forma de pago
- Términos de pago
- Activo

Campos de filtro

- Forma de pago
- Términos de pago
- Grupo IVA cliente
- Filtro generalista buscará en determinadas columnas del listado, mediante índice fulltext de BD. Este filtro no podrá actuar sobre todas las columnas del listado. Solo sobre las que están en la entidad. Pendiente de definir sobre qué columnas filtra. Al inicio del desarrollo se facilitará un Excel al cliente para que especifique para cada entidad qué columnas incorporar á el filtro.

Orden por defecto
Sociedad ASC
Filtro por defecto
Activo = TRUE
Acciones encabezado del listado:

- Botón Nuevo: Abrir formulario de alta de sociedades

Acciones en los registros del listado:

- Ver sociedad: llamará al formulario de edición de sociedades en modo de solo lectura
- Editar sociedad: llamará al formulario de edición de sociedades en modo edición
- Eliminar sociedad: Borrado físico siempre que lo permita la integridad referencial

DOCUMENTOS

Sprint backlog

- El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint.
- Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser rota en mayor detalle.
- Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

Daily Scrum Se trata del seguimiento diario que se hace a las tareas asignadas.

Burn down

El documento de control en el que se detalla la evolución de las tareas y los requerimientos hasta llegar a la entrega final

¿Cómo funciona la metodología Scrum?

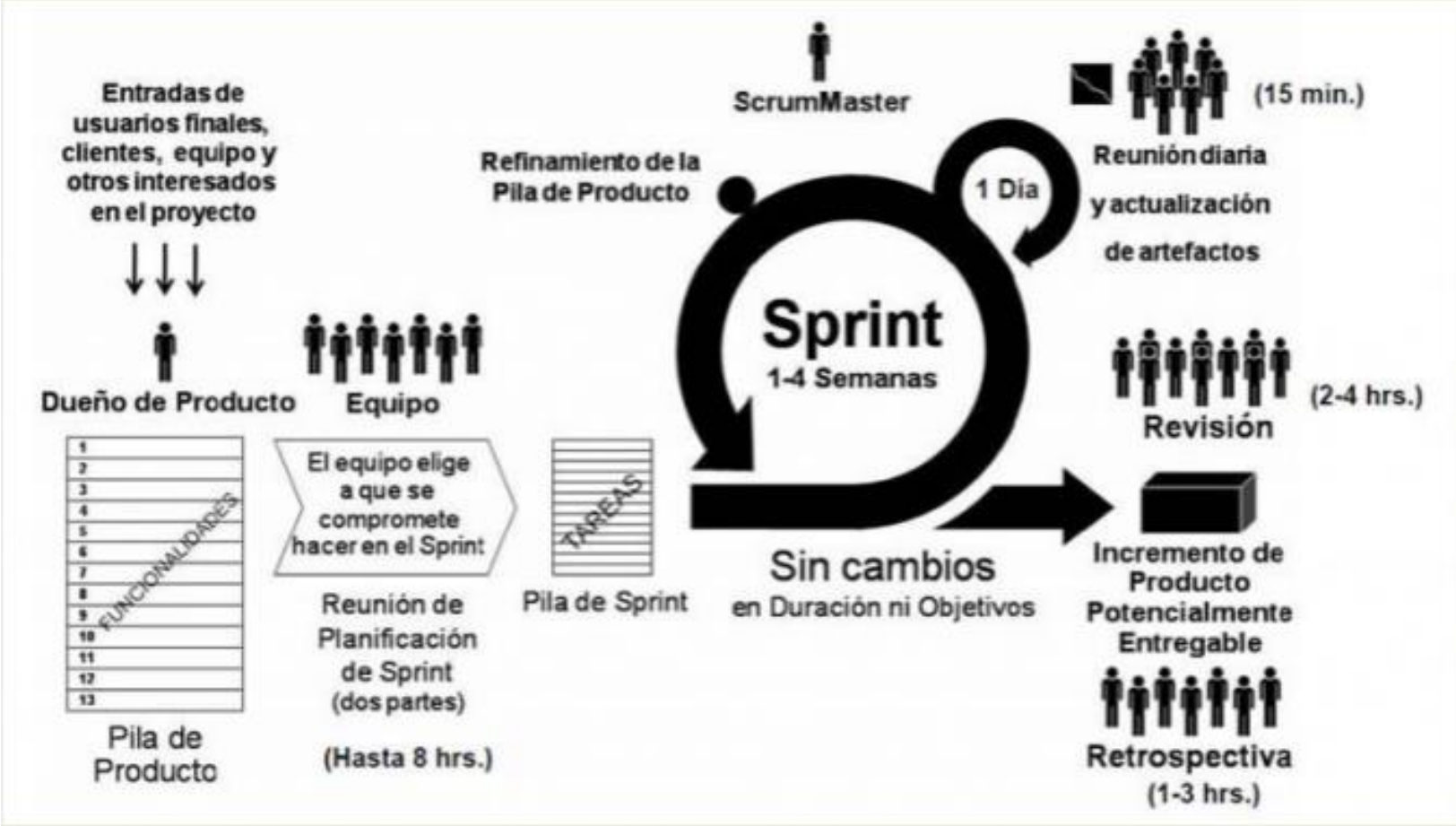
•El encargado de producto elabora una lista de deseos con diferentes prioridades (***Product backlog***)

•El equipo scrum toma una pequeña parte de la lista de deseos (***Sprint backlog***) y planea su implementación.

•El equipo completa sus tareas del sprint backlog en un ***Sprint*** (que suelen ser período de 2 a 4 semanas). Se evalúa el progreso en una reunión diaria (***Daily scrum***)

•Cuando acaba el Sprint, se envía o revisa el trabajo completado y tras su revisión se cierra este sprint, para comenzar con el siguiente.

Herramientas básicas para la metodología Scrum



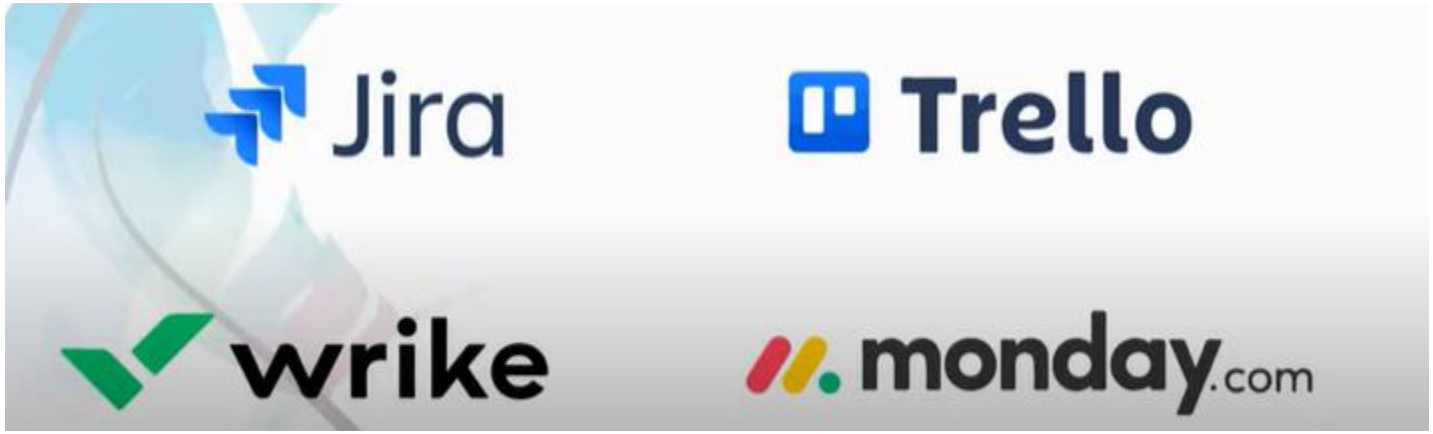
Ventajas de la metodología Scrum

- Permite crear productos listos para salir al mercado en un tiempo menor.
- Ofrece el marco necesario para cambiar de rumbo o de enfoque cuando un proyecto así lo necesita.
- Ayuda a comunicarse de manera interna al mantener reuniones diarias que ayudan a retroalimentar el proceso y encontrar los errores.
- Es un medio para evaluar el rendimiento individual y del equipo, así como para propiciar su mejora.
- Fomenta la revisión por parte del cliente y sus necesidades pueden ser gestionadas de manera rápida.

Desventajas de la metodología Scrum

- Si los objetivos no están definidos correctamente, habrá ambigüedad en todo el proceso.
- Si el cliente tiene demandas nuevas en partes avanzadas del proceso, puede estancarse todo el proyecto, por lo que es necesario establecer fechas límite.
- Si no hay confianza en el equipo, los controles excesivos disminuirán el rendimiento y desvirtuarán el propósito de este marco de trabajo.
- Debido a que es una forma de trabajo exigente, es necesario contar con miembros de gran experiencia o capacidad; de lo contrario, el proyecto puede fallar.

SOFTWARE PARA ADMINISTRAR FLUJOS DE TRABAJO SCRUM




Gracias