

RESUMEN TEORÍA SISTEMAS INFORMACIÓN

INDICE

1. Requisitos del software
 - a. Scrum
 - b. Requisitos del software
2. Calidad del software
 - a. Pruebas del software
 - b. Calidad del software

REQUISITOS DEL SOFTWARE

METODOLOGÍA SCRUM

El proceso SCRUM es un proceso basado en incrementos. Un incremento es una versión funcional del proyecto donde se priorizan los requisitos.

El objetivo principal de este es proporcionar una metodología ágil e incremental basada en sprints, en base a la elaboración de requisitos a través del Backlog, donde se recogen las necesidades del Product Owner.

Los requisitos cambian de prioridad con frecuencia según el proyecto va avanzando y, las características del sistema se comienzan a elaborar en el Product Backlog. En cada sprint se recogen las características con más prioridad para el Sprint Backlog.

No más de 4 semanas

El proceso SCRUM se basa en una evaluación continua del Product Backlog, y se realiza de la siguiente manera:

- Al comienzo del sprint, se realiza un Sprint Planning Meeting, el cual está compuesto por dos reuniones.
 - Reunión 1: En esta reunión el equipo define los que se va a hacer, incluyendo lo que desea el dueño del producto. El dueño del producto y el equipo discuten, revisan y refinan la pila del producto (contiene los elementos de mayor prioridad que no han sido finalizados) incluyendo historias y criterios de aceptación.
 - El dueño del producto decide las prioridades.
 - El equipo decide las tareas a realizar desde el principio de la pila.
 - El SCRUM Máster presta atención a todos.
 - Reunión 2: EL equipo define la planificación detallada de tareas (como se va a hacer), mediante la obtención de información adicional, diseño de pantallas (prototipos), cambio en el modelo de datos, programación de las partes del entregable, pruebas, etc. En esta reunión además se realiza la selección definitiva de los elementos para el Sprint Backlog en función de la estimación final de cada una.
- Durante la ejecución del Backlog, se realiza un Stand-Up Meeting diario. En esta reunión se reúne el equipo para que cada miembro informa del estado de las historias asignadas. No suele acudir el jefe del equipo.

- Finalmente se finaliza el sprint. Para ello se realiza un Review and Restrospective (demostración y revisión) en dos reuniones más una actualización.
 - o Reunión 1: En esta reunión se revisa el sprint. Se demuestra el producto y se realiza una conversación en profundidad entre el equipo y el dueño del producto.
 - o Reunión 2: En esta reunión se realiza la retrospectiva. El equipo inspecciona y adapta el proceso (como ha ido y posibles mejoras).
 - o Actualización: Actualizar la pila del producto y gráfica de trabajo restante de todo el producto (Backlog Burndown Chart).

Para facilitar la planificación de sprints posteriores se pueden realizar refinamientos, los cuales se realizan durante el sprint, en referencia a elementos futuros no incluidos en él:

- Analizar los requisitos.
- Separar elementos grandes en pequeños.
- Estimar nuevos elementos.
- Reestimación de los elementos existentes.

Este suele abarcar entre un 5-10 % del tiempo total del sprint y se suele realizar al final de este en forma de taller.

Durante el proceso SCRUM se realizan varias funciones, en las cuales intervienen diferentes personas como:

- Equipo: Este esta formado por el Product Owner + SCRUM Máster + equipo SCRUM.
- Product Owner: Este representa el papel del cliente, el cual se encarga de priorizar, describir y solucionar preguntas. También puede representar al negocio o al conjunto de clientes (Stakeholders). Este tiene total autoridad en la toma de decisiones. Interactúa activa y frecuentemente con el equipo.
- SCRUM Máster: Ayuda a aplicar la metodología SCRUM. Elimina impedimentos y puede realizar trabajos como el equipo. Es importante remarcar que este no es el jefe del equipo (pues no hay jefe).
- Equipo SCRUM: Este NO se encuentra dentro de los participantes de las reuniones periódicas, pero si pertenecen a los roles de SCRUM. Realizan labores como autogestión, estimación de tiempos a emplear, decide cómo se realiza el trabajo, se encarga del proceso de desarrollo. Suele estar formado por 7 personas con experiencia y estabilidad.

Son buenas prácticas ágiles:

- Adaptarse a los cambios.
- Realizar microincrementos con versiones funcionales cada pocas semanas.
- Interactuar frecuentemente con el cliente.
- Realizar pruebas continuas.

Las historias de usuario son descripciones claras y concisas de la funcionalidad en términos del valor que aporta al usuario final del producto. La estructura típica es en forma de tarjeta:

Como [ROL] quiero [CARACTERÍSTICA] para [VALOR DE NEGOCIO]

En la descripción de las HU son necesarios detalles adicionales y los criterios de aceptación para mostrar cuando está completa (requisitos). Estas presentan una serie de características que son:

- Independientes.
- Criterios suficientes para implementar.
- Pequeñas, de forma que se puedan estimar y probar.

Existen otros tipos de historias como:

- Epic: Solo en el Product Backlog. Estas son funcionalidades generales, sin dividir.
- Técnicas: Estas no aportan funcionalidad, pero son necesarias.

REQUISITOS DE SOFTWARE

Los requisitos de software se obtienen mediante la elaboración de las historias de usuario y un supuesto práctico.

“Se ha formalizado un contrato con un cliente para construir un sistema software. Partimos de una definición de la oferta realizada por el cliente, y que está incluida en los pliegos”.

- Descripción general del sistema y los objetivos.
- Lista (de algunos) requisitos de usuario.

Como equipo de desarrollo, se nos da la tarea de desarrollar dicho sistema, definiendo el producto que será diseñado, programado y probado mediante la elaboración de historias de usuario. Para ello se siguen los siguientes pasos:

1. Se define el contexto, así como el objetivo a llevar a cabo. Que es lo que el cliente desea que se haga.
2. Se identifican los principales procesos del proyecto.
3. Se identifican aquellas partes interesadas del proyecto.
4. Se establecen las necesidades y objetivos del proyecto.
5. A continuación, se definen los requisitos de usuario, comenzando por los requisitos funcionales y luego los no funcionales.
6. Se realiza la valoración de las alternativas.
7. Se describe la solución propuesta. También se organiza el proyecto.
8. Se decide el planteamiento del desarrollo.
9. Se elabora el caso de estudio. El objetivo es definir los detalles del sistema que se va a implementar, las tareas a realizar de forma incremental, resultando en la formación del Product Backlog. Esto se suele realizar mediante las historias de usuario.

Durante la planificación del sprint (reunión 1), se define que se va a hacer. Si las historias de usuario definidas anteriormente son demasiado grandes (Epic), se refinan en historias de usuario más pequeñas, empezando siempre por la más prioritaria. Esto se lleva a cabo en la primera reunión del Sprint Planning Meeting. Se completa y detalla el Product Backlog. Una parte clave son las conversaciones con el Product Owner.

Los criterios de aceptación especifican los detalles adicionales y las condiciones que ha de satisfacer una historia de usuario. Expresan las necesidades del usuario y son obtenidas mediante conversaciones. NO son pruebas de aceptación.

Durante la planificación del sprint (reunión 2), se da compromiso de que se va a realizar en el sprint dadas las historias de usuario del Product Backlog priorizadas, y seleccionar aquellas que pueden realizarse en un sprint o iteración. Se define como profundizar en los criterios de aceptación de cada historia de usuario, completando dichos criterios. Los criterios de aceptación pueden tener varias formas:

- Reglas de negocio.
- Escenarios.

Los prototipos describen más ampliamente los criterios de aceptación en relación con las funcionalidades del sistema. Se debe discutir con el Product Owner para saber si satisface sus necesidades.

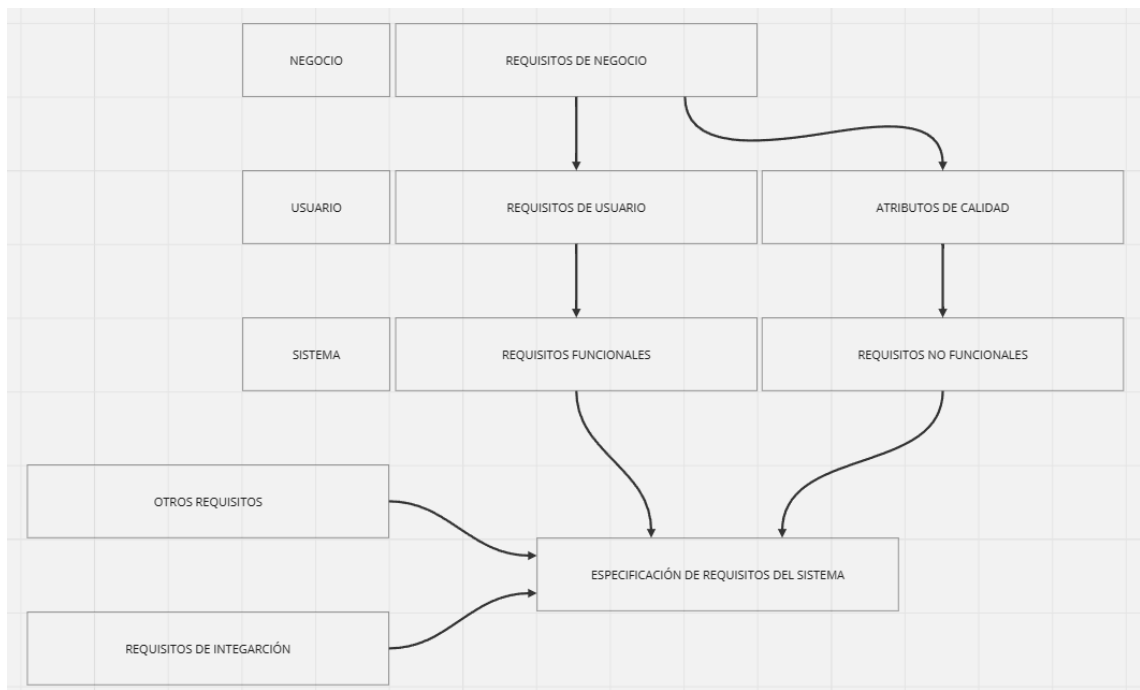
Los modelos de datos deben realizarse de forma incremental, partiendo de un modelo básico, para cada sprint ir añadiendo las tablas o columnas necesarias. Puede ser de clases de dominio o entidad relación.

Los requisitos del software describen el qué de un producto software, es decir:

- Qué debe hacer el software para servir a las partes interesadas.
- Qué debe ser el software para servir a las partes interesadas.
- Qué limitaciones tendrá el producto software.

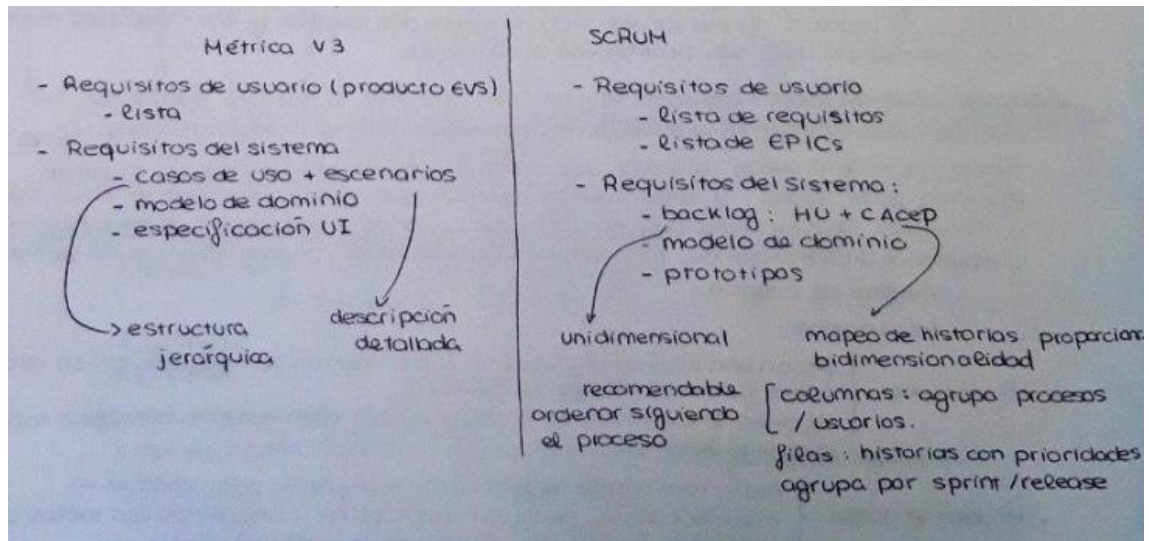
Hay de varios tipos:

- Requisitos de usuario: Tratan de definir los requisitos para un sistema que pueda proporcionar los servicios requeridos por los clientes y las partes interesadas.
- Requisitos del sistema: Transforman los requisitos de las partes interesadas de los servicios deseados en una vista técnicas del producto que pueda ofrecer dichos servicios.
- Requisitos de negocio: Se representan en términos de los objetivos de la organización.



Los requisitos se clasifican frecuentemente en base a los atributos de calidad. Estos son características que definen la calidad de un producto software como la funcionalidad, la fiabilidad, la eficiencia, la usabilidad, la mantenibilidad o la portabilidad.

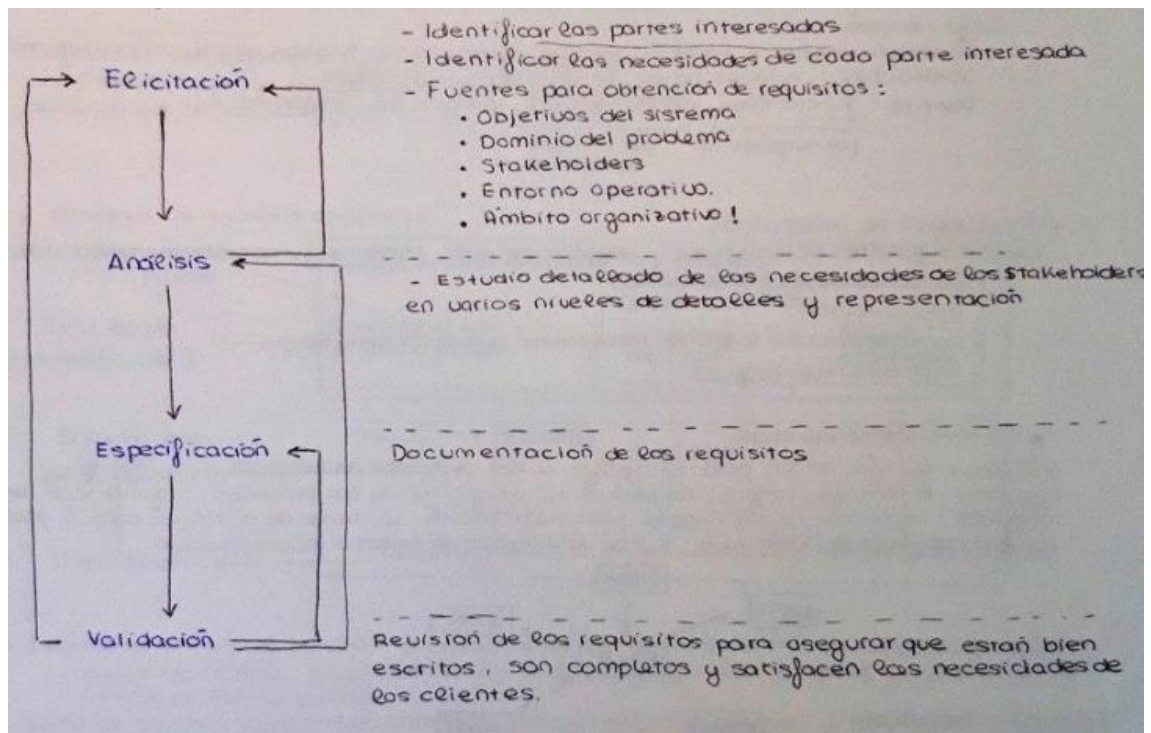
Una clasificación habitual es en funcionales o no funcionales, integración con otros sistemas, implantación y migración, normativos.



Estos presentan una serie de características principales, que son:

- Identificables: Identificador único.
- Singulares: Especifican una idea o propiedad del sistema.
- Inequívocos: No expresan ambigüedades.
- Trazables: Se debe poder identificar qué productos se obtienen a partir de un determinado requisito.
- Verificables: Debe poder comprobarse que el software satisface el requisito.

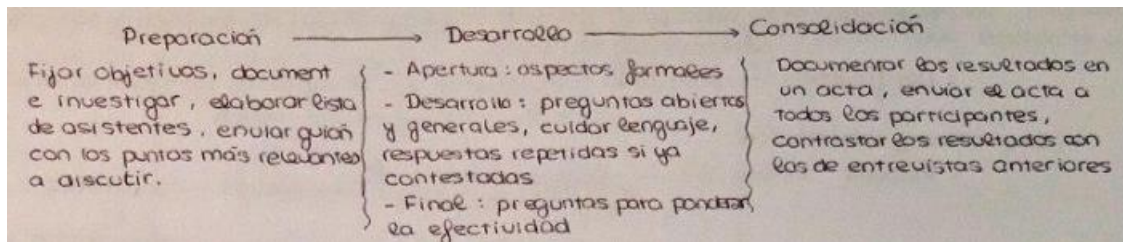
Para realizar estos requisitos, se suele seguir una serie de pasos:



Estos requisitos se pueden obtener de diversas fuentes como:

- Escenarios: Ejemplos de casos de uso.
- Prototipos.
- Reuniones.
- Observación in situ.
- Análisis de los sistemas existentes y de la información manejada.

Las entrevistas son una técnica efectiva para obtener información sobre el funcionamiento del sistema actual, la organización de las unidades, las responsabilidades y funciones de los usuarios y los requisitos del software.

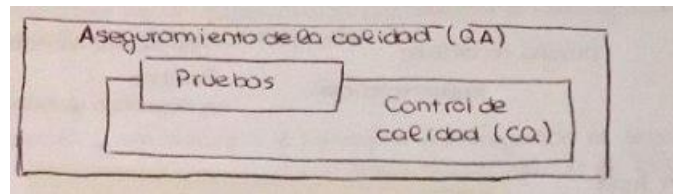


La especificación de los requisitos consiste en describir los requisitos software del producto que se va a desarrollar, el cual incluye una descripción general, requisitos de usuario y requisitos del sistema con trazabilidad. Se organizan jerárquicamente e identificados de forma única. Los niveles altos son los requisitos más abstractos y los más bajos los más detallados.

La evaluación de alternativas consiste en valorar diferentes alternativas y seleccionar una.

Los sistemas empaquetados son sistemas de información para dar apoyo a los procesos de negocio internos de la empresa de manera integral. Hace énfasis en la integración de los procesos, organización en módulos, implantación en base a parametrización. La base de datos es común para facilitar el flujo de información.

CALIDAD DEL SOFTWARE



PRUEBAS DEL SOFTWARE

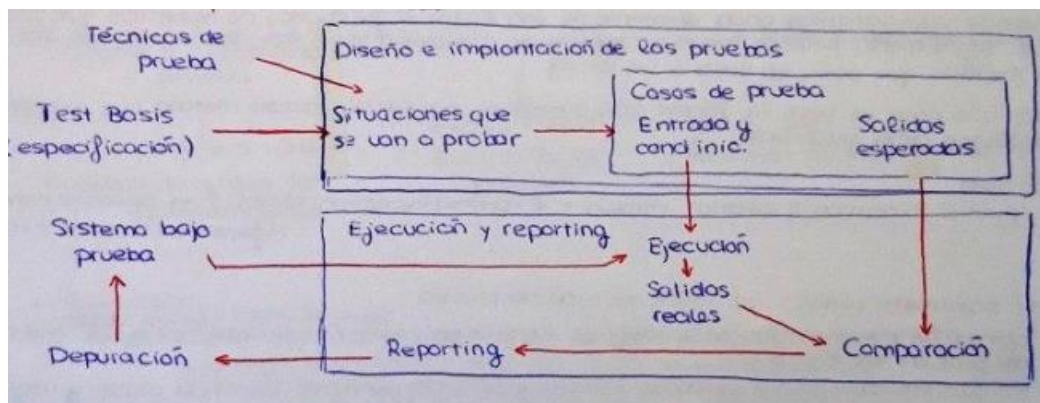
Realizar pruebas del software es un proceso que consiste en ejecutar un programa con la intención de encontrar fallos. Un buen caso de prueba es el que tiene una alta posibilidades detectar un nuevo fallo. Un caso de prueba con éxito es el que detecta un fallo nuevo.

Un error es aquel que ocurre debido a la equivocación de un ser humano.

Un fallo es la diferencia entre el comportamiento esperado en el software y el observado.

Un defecto es aquel que ocurre debido a un desperfecto en un componente / sistema.

El proceso de pruebas dinámicas



Las técnicas de prueba: Es un procedimiento que permite derivar un conjunto de buenos casos de prueba, maximizando la probabilidad de encontrar fallos y cumpliendo los límites de tiempo y coste.

Los casos de prueba: Son un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo en particular. A un conjunto de casos de prueba se le denomina test suite.

Las clases de equivalencia: Representan un conjunto de datos para los que se supone que el programa tiene un comportamiento similar. Los dominios se dividen en clases de equivalencia disjuntas.

Para identificar las clases de equivalencia se siguen los siguientes pasos:

1. Se examina cada condición de entrada.
2. Cada condición de entrada se divide en CE conjuntos.
3. Si hay razones para creer que los elementos de una clase no se tratarán de la misma forma, dividirla la clase en otra más pequeña. Se puede aplicar lo mismo para las salidas.
4. Se derivarán casos de prueba para cubrir clases de equivalencia.

CALIDAD DEL SOFTWARE

La calidad del software se puede definir como el grado en el cual un componente, sistema o proceso cumple con los requisitos especificados y las necesidades del cliente o usuario.

El aseguramiento de la calidad es un patrón sistemático y planificado de todas las acciones necesarias para proveer de la confianza adecuada a un producto u objeto de acuerdo con los requisitos técnicos. EL conjunto de actividades diseñadas para evaluar un proceso por el cual están desarrollando dichos productos.

El control de la calidad es el conjunto de actividades diseñadas para evaluar la calidad de un producto desarrollado o manufacturado. El proceso de verificar tu propio trabajo o el de un trabajador.

Para el aseguramiento de la calidad se suelen realizar una serie de actividades como:

- Verificación y validación: Si estamos construyendo el producto correcto y correctamente.
- Pruebas (dinámicas): Ejecutar el software con el objetivo de descubrir propiedades respecto a la calidad y proporcionar confianza en el producto.
- Revisiones (estáticas): Lectura de un artefacto con la intención de detectar defectos en la calidad.
- Gestión de la configuración: Asegurar la integridad de los productos y servicios desarrollados durante su primer desarrollo y evaluaciones posteriores.
- Evaluación y mejora del proceso software: Trabajar de acuerdo con una serie de acciones sistemáticas y cada vez mejor.

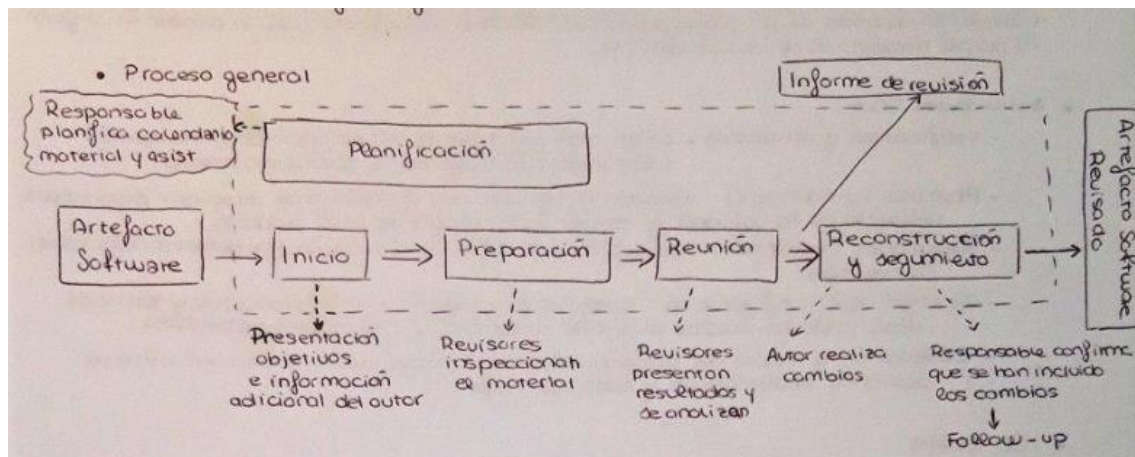
Las revisiones son una forma temprana de contribuir a la detección temprana de defectos, de forma que estos defectos puedan eliminarse lo antes posible para eliminar las posibles modificaciones posteriores que se lleven a cabo. En la práctica se llevan a cabo para múltiples fines. Existen varios tipos de revisiones como:

- Muy formales: Se crea un equipo de inspectores que representan las diferentes perspectivas. El autor forma parte del equipo, además de otras partes interesadas. Para una inspección de requisitos son necesarios:
 - El analista (autor) que ha elaborado los requisitos.
 - El usuario / cliente, que es la fuente de los requisitos.
 - El personal de diseño que usara la especificación.
 - ...

- Más informales: Existen varios tipos:
 - o Recorrido: El autor expone el trabajo a una audiencia, solicita preguntas y comentarios.
 - o Revisión por pares: Distribución del trabajo a otras partes buscando discrepancias y no conformidad con especificación y/o estándares.
 - o Revisión técnica: Similar a revisión por pares, pero con más formalidad.

En las revisiones suelen intervenir los siguientes roles:

- Director: Este es el responsable/jefe. Planifica y asigna los recursos.
- Moderador: Se encarga de dirigir la reunión, hacer de mediador y concluye los resultados.
- Autor: Expone el trabajo y material a revisar, llevara a cabo los cambios recomendados.
- Revisor: Se encargan de detectar los defectos.
- Secretario: Registra y documenta todos los alumnos.



La gestión de la configuración del software:

- SCM: consiste en un conjunto de actividades diseñadas para gestionar los cambios durante todo el ciclo de vida de un producto o servicio con el objetivo de mantener la integridad del producto o servicio, controlar y evaluar los cambios y hacer visible el producto a todo el equipo.
- Elemento de configuración del software (ECS): Cualquier producto de trabajo cuyo cambio pueda resultar crítico en el proyecto y que es expuesto explícitamente para su control.
- Elemento de configuración hardware.
- Línea base: Conjunto de ECS con estados estables. Antes de la línea base hay cambios rápidos e informales y después de esta debe seguirse un proceso más formal.
- Release (Entrega): Estado de configuración que se utiliza fuera del contexto de desarrollo.

La separación de entornos tiene como objetivo tener versiones operativas en entornos diferentes.

- Desarrollo: Consiste en el desarrollo de componentes, pruebas unitarias, algunas de integración.
- Integración: Integrar componentes, pruebas de integración.
- Preproducción: Pruebas de aceptación, formación de usuarios.
- Producción: El utilizado por los usuarios reales.