



# REQUIREMENTS REUSE

Material producido por la Universidad Peruana de Ciencias Aplicadas  
UPC, 2021.



## CONTENIDO

---

Objetivos de aprendizaje.....	3
Introducción .....	3
Contexto .....	6
Modos de reutilización.....	8
Reuse approach.....	10
Best practices for requirements reuse.....	15
Conclusiones.....	18
Referencias .....	18

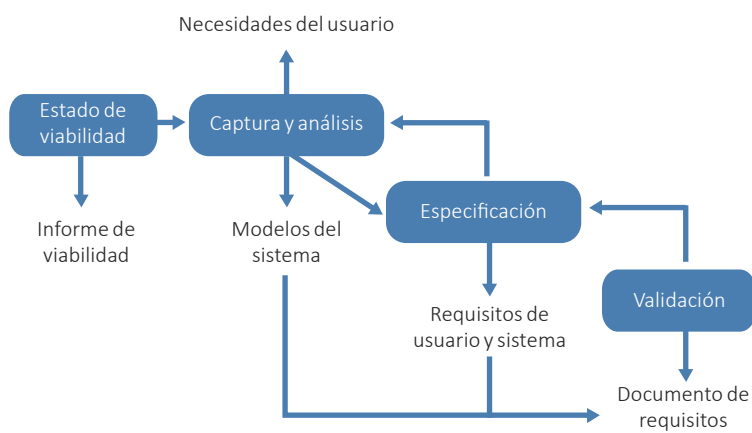


## 1. OBJETIVO DE APRENDIZAJE

Al finalizar la unidad, el estudiante valida los requisitos a partir de las metodologías y frameworks actuales de desarrollo de software, de acuerdo con las necesidades de los involucrados, comunicándose de manera clara y fluida.



## 2. INTRODUCCIÓN

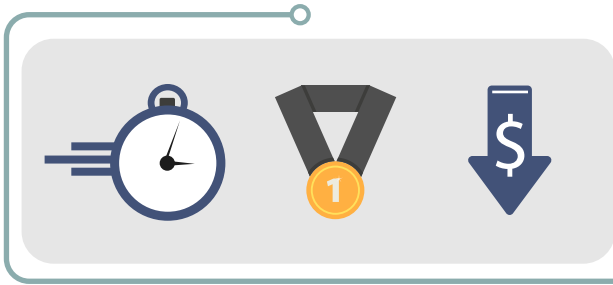


La reutilización de requisitos depende primero de la capacidad de identificar los requisitos reutilizables y, en segundo lugar, de tener la reutilización de requisitos como parte de su proceso de requisitos.

“La reutilización de software es el proceso de crear sistemas de software a partir de software existente en lugar de crearlos desde cero”.

Cuando la reutilización está presente en un proceso de desarrollo software, este debe integrar todas las actividades necesarias para producir y reutilizar componentes de software. Así, se distinguen dos actividades principales dentro de la reutilización: El desarrollo para la reutilización y el desarrollo con reutilización.



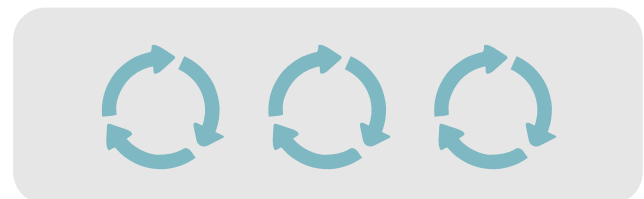


La reutilización de software es una solución reconocida para los tres impulsores principales del negocio de software: **Más rápido, mejor y más barato.**

## 2.1. REUTILIZACIÓN DEL SOFTWARE: ESTRUCTURA

- ✓ Reutilización del software
- ✓ Beneficios de la reutilización
- ✓ Dificultades para la reutilización
- ✓ Assets

- ✓ Niveles de reutilización
- ✓ Modelo de procesos con reutilización
- ✓ Desarrollo basado en componentes



## 2.2. BENEFICIOS

Mejora de la productividad

Disminución del tiempo en el desarrollo:

- Mejor adaptación, requisitos cambiantes
- ¡Los requisitos no son estables!

Disminución de costos

Mejora de la calidad del software

Mayor fiabilidad

Mayor eficiencia (aunque al principio puede parecer que no)



## 2.3. DIFICULTADES

- En muchas empresas no existe plan de reutilización (no se considera prioritario)
- Escasa formación
- Resistencia del personal
- Pobre soporte metodológico:
  - Uso de métodos que no promueven la reutilización (estructurados)
  - Métodos necesarios para:
    - Desarrollo para reutilización
    - Desarrollo con reutilización



<https://www.youtube.com/watch?v=nsVUaNFqfe8>

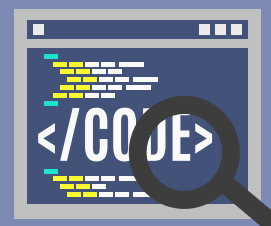
## 2.4. SOFTWARE REUSE & REUSABILITY (R&R)

La reutilización de software es el uso de software existente, o conocimiento de software para construir software nuevo.



La reutilización es una propiedad de un activo de software que indica su probabilidad de reutilización.

Los activos reutilizables pueden ser software reutilizable o conocimiento de software.





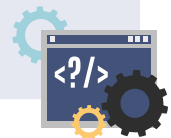
## 3. CONTEXTO

### 3.1. DEVELOPMENT FOR REUSE (DESARROLLO PARA LA REUTILIZACIÓN)

Consiste en la realización de un software que cumple con un conjunto de restricciones sobre su reutilización y su calidad. A la hora de desarrollar componentes reutilizables, es fundamental centrarse en criterios de calidad y en la disminución de los costos de producción de los componentes.



Cuando se va a crear un componente reutilizable, debe analizarse la variabilidad de los requisitos que satisface dicho componente, de forma que se construya como un componente genérico que pueda ser especializado en el momento de su reutilización para ajustarse a unos requisitos específicos.



### 3.2. DEVELOPMENT WITH REUSE (DESARROLLO CON REUTILIZACIÓN)

- Las técnicas utilizadas en el proceso de desarrollo con reutilización, dependen en gran medida de los componentes que se hayan preparado en el proceso de desarrollo para la reutilización.
- El desarrollo con reutilización consiste en la generación de nuevos productos de software, integrando elementos existentes de forma directa o pasando por un proceso de adaptación.



### 3.3. CONTEXTOS DE REUTILIZACIÓN

Los componentes arquitectónicos y el código de software pueden estar sujetos a reutilización en dos contextos separados:

**A**

Para desarrollar aplicaciones similares.

- Se conoce como Software Product Lines (SPL).

**B**

Para implementar versiones de la misma aplicación.

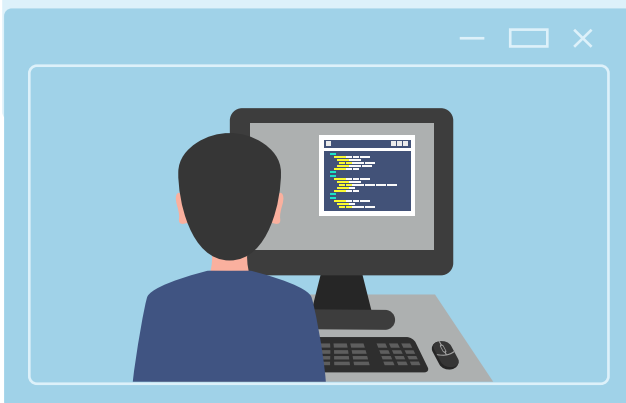
- Se conoce como mantenimiento de software.



La reutilización de requisitos para el mantenimiento del software tiene más oportunidades en comparación con el caso de SPL, por dos razones:

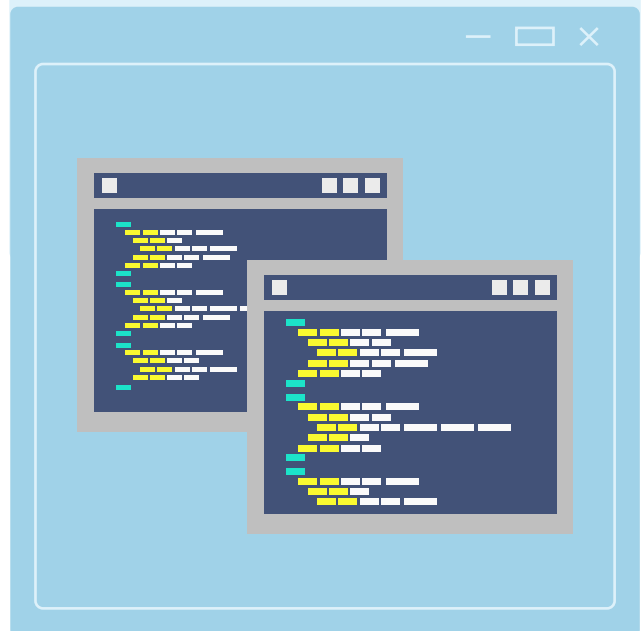
1

El mantenimiento de software es un caso muy común en la práctica, ya que, la mayoría de las aplicaciones en un departamento de TI típico, por ejemplo, en las industrias financieras u hoteleras, son sistemas de producción existentes que pasan por ciclos regulares de lanzamientos de producción; mientras que, las nuevas aplicaciones representan solo una pequeña parte de la cartera de TI. Deja pocas oportunidades para que los equipos de proyectos de TI apliquen y se beneficien de la reutilización de requisitos en el contexto de SPL.



2

La mayoría de las veces, los cambios asignados a futuras versiones se superponen con la funcionalidad existente de las aplicaciones. Por lo tanto, esto brinda oportunidades para reutilizar los requisitos de productos existentes para futuras versiones.



### 3.4. PATRONES DE DISEÑO



Son soluciones basadas en la experiencia y que se ha demostrado que funcionan. Es evidente que a lo largo de multitud de diseños de aplicaciones, hay problemas que se repiten o que son análogos, es decir, que responden a un cierto patrón. Sería deseable tener una colección de dichos patrones con las soluciones óptimas para cada caso.



Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables. Han revolucionado el diseño orientado a objetos y todo buen arquitecto de software debería conocerlos.



## 4. MODOS DE REUTILIZACIÓN



### Reuse by copying (copiar y pegar)

Hay varias formas de reutilizar los requisitos, y copiar y pegar es la técnica más utilizada. Copia independiente de la especificación. No hay conexión con la especificación original. Aunque es muy básico, puede ser un buen punto de partida. Existen múltiples variantes de este método, como copiar y pegar en modo de solo lectura.



### Reuse with linking (copiar y enlace)

Esta es una opción mucho más poderosa, ya que nos permitirá crear una rama (una versión divergente con el mismo identificador único) a partir del requisito original, pero aún así recibir actualizaciones si se modifica el requisito original. Esto nos permitirá seguir trabajando en nuestra propia sucursal, o fusionarnos con la original en cualquier momento.



### Reuse with inheritance (enlazar)

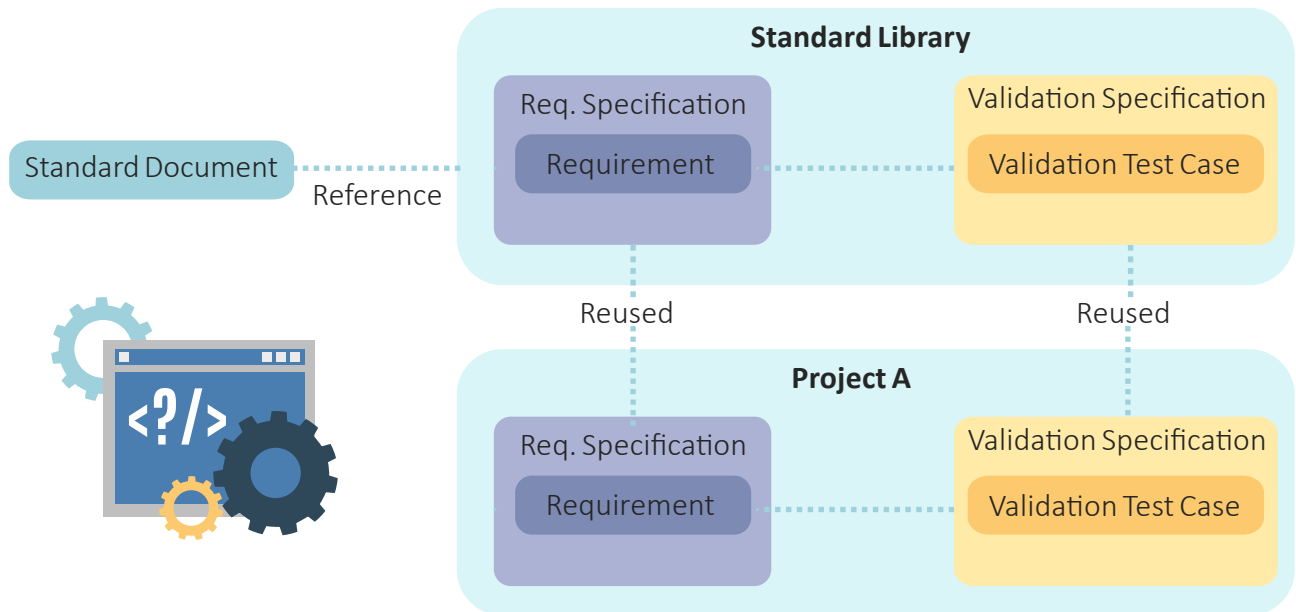
Este modo permite mostrar una representación de un conjunto de requisitos de otro proyecto en modo de solo lectura. Cualquier cambio realizado en los requisitos originales se completa automáticamente con el resto de los proyectos reutilizándolos. Este modo es especialmente útil cuando se trata de estándares y normas, en los que los proyectos no necesitan modificarlos, pero deben asegurarse de que se mantengan actualizados.

### 4.1. CENTRAL REQUIREMENT DATABASE FOR STANDARDS

En lugar de copiar los requisitos estándar entre proyectos individuales, se debe establecer una base de datos centralizada común con propietarios de datos con nombre para atender todos los proyectos.

Los requisitos en la base de datos deben formar un conjunto completo de alta calidad, que represente toda la documentación estándar relevante para sus proyectos, y que haya sido revisada con las partes relevantes.

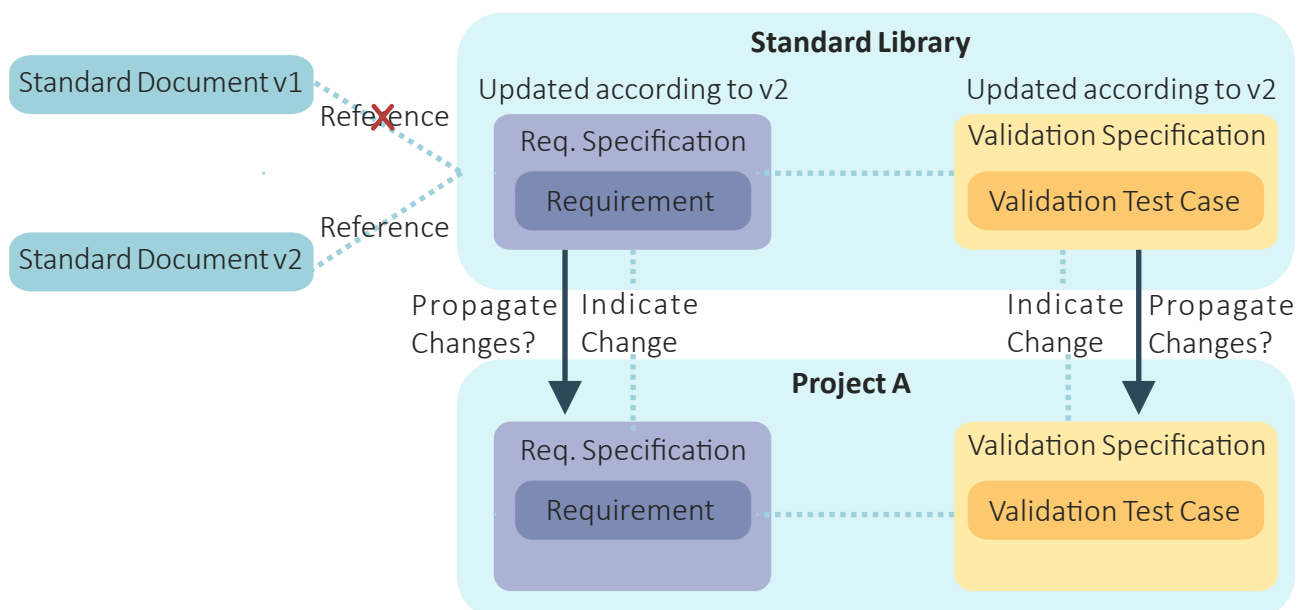




¿Qué sucede cuando cambian los estándares?

Cuando la **reutilización** se realiza con una copia en vivo vinculada con herencia, los cambios se pueden propagar fácilmente a las copias en los proyectos. Cuando se actualizan las especificaciones en la base de datos central, se indica automáticamente a las copias reutilizadas.

Los proyectos pueden decidir si toman la nueva versión en uso y propagan el cambio en sus copias de las especificaciones. De nuevo, depende de la herramienta seleccionada si los cambios se pueden propagar realmente a las copias.





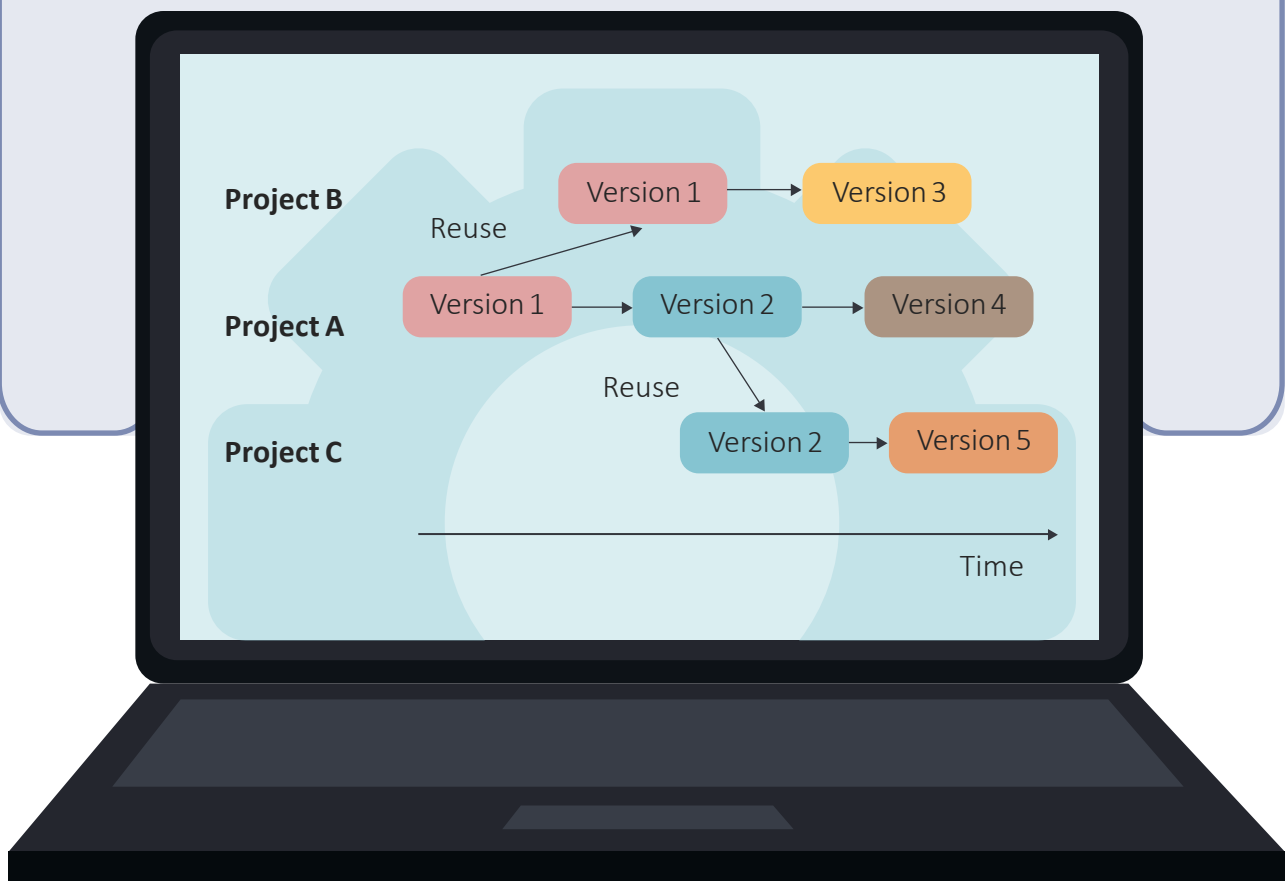
## ¿Cómo empezar?

Los requisitos estándar son un objetivo fácil para la reutilización porque son exactamente los mismos para cualquier producto, pero nada le impide utilizar los métodos de reutilización también para otros tipos de requisitos, si se aplican a varios proyectos o productos.

## 5. REUSE APPROACH

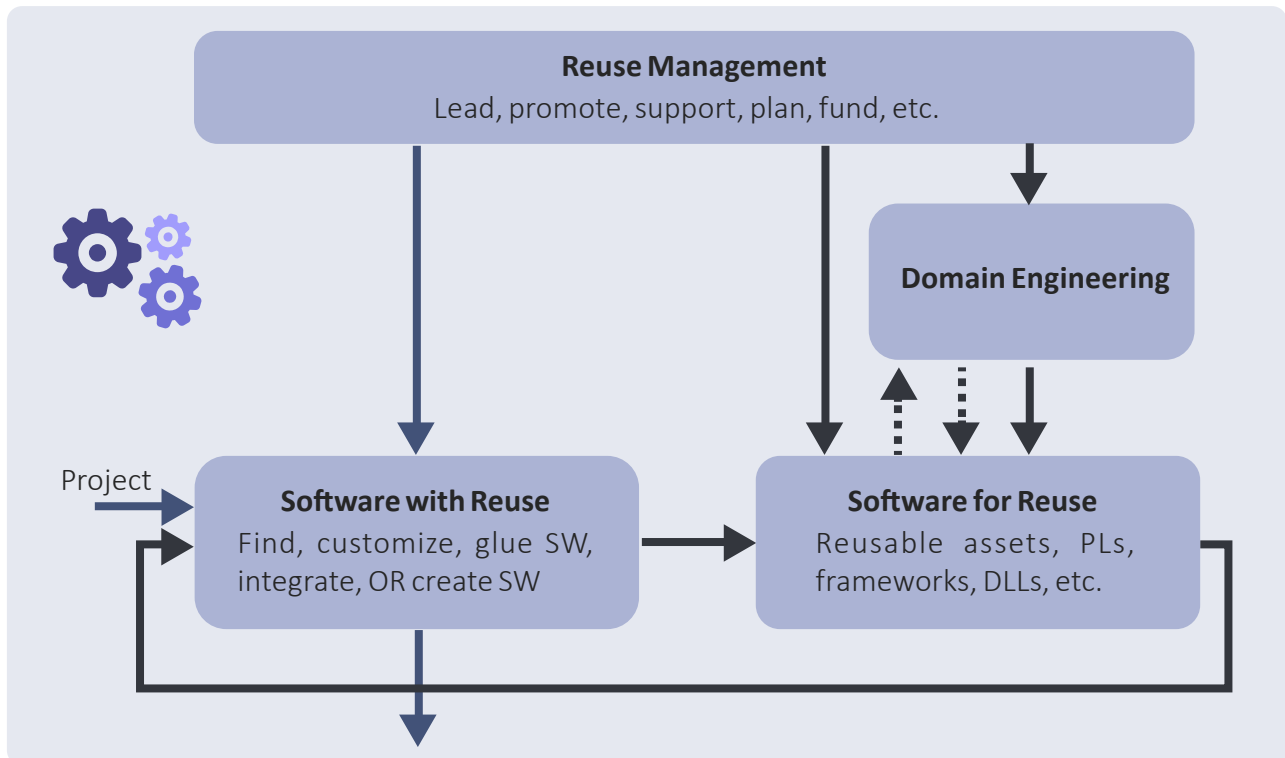
### 5.1. THE REQUIREMENTS REUSE

La reutilización de requisitos puede potenciar el ciclo de vida del software. Es un enfoque para utilizar sistemáticamente los documentos de requisitos existentes para reducir el esfuerzo general dentro del ciclo de vida del software. La reutilización de los primeros productos de software y los procesos pueden afectar el ciclo de vida desde dos puntos básicos de vista: mejorar el proceso de ingeniería de requisitos y apoyando el proceso de desarrollo con reutilización.





## 5.2. REQUIREMENTS REPRESENTATION FOR REUSE



Se han propuesto algunas alternativas diferentes, basadas en la representación del conocimiento y el razonamiento analógico para reutilizar los requisitos de una base de conocimiento. Estas técnicas ponen énfasis en la semántica de los requisitos de documentos y, exige aplicaciones de inteligencia artificial para adquirir y gestionar el conocimiento codificado en la documentación de requisitos.

## 5.3. ALCANCE DE LA REUTILIZACIÓN

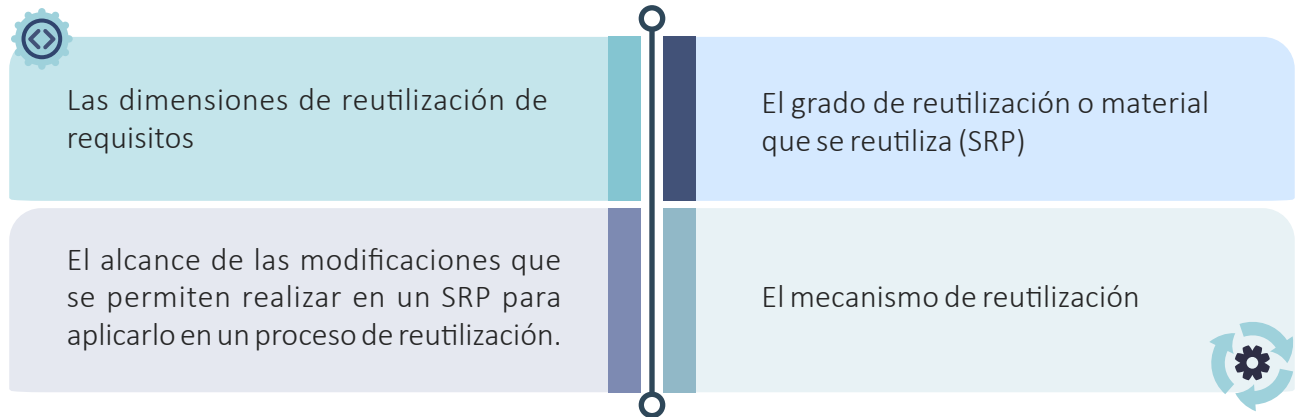
Los Software Requirement Patterns o SRP (patrones de requisitos de software) es un conjunto de requisitos que persiguen el mismo objetivo en un sistema a desarrollar, y donde todos los elementos específicos de un determinado proyecto se han eliminado y convertido en plantillas.

El alcance de la reutilización puede ser de SRP individual, o de hecho de partes de un SRP, a conjuntos de SRP que abordan el mismo dominio de software, funcionalidad, regulación estándar, etc.

Los conjuntos de requisitos que pueden reutilizarse conjuntamente son establecidos por la clasificación de SRP en un catálogo. Para aceptar un SRP en el catálogo, debe analizarse la calidad de las especificaciones de requisitos que se obtendrán al aplicarlo.

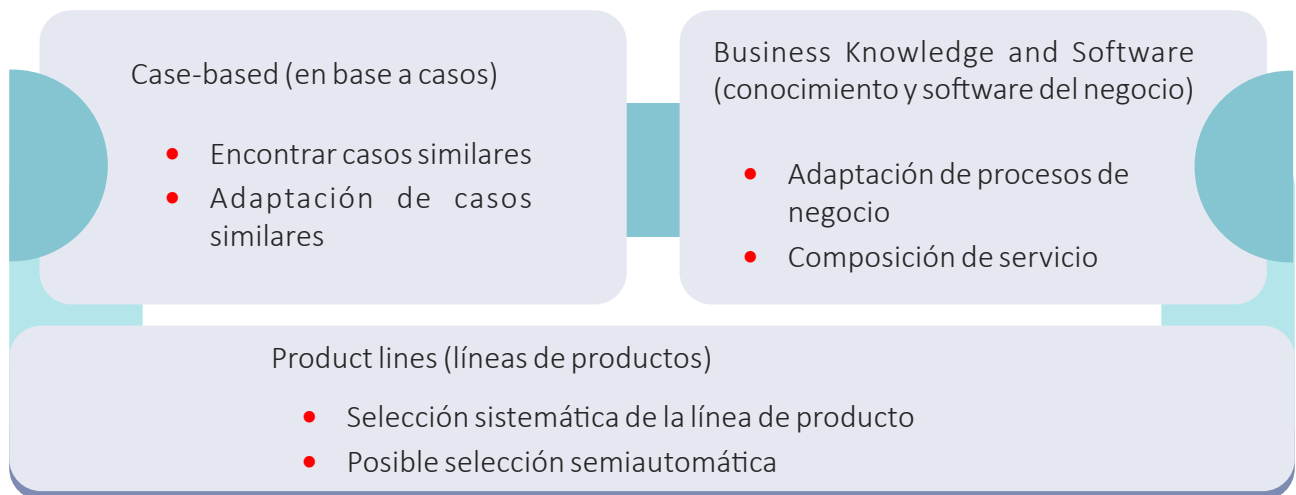


## 5.4. CONSIDERACIONES PARA REQUIREMENTS REUSE



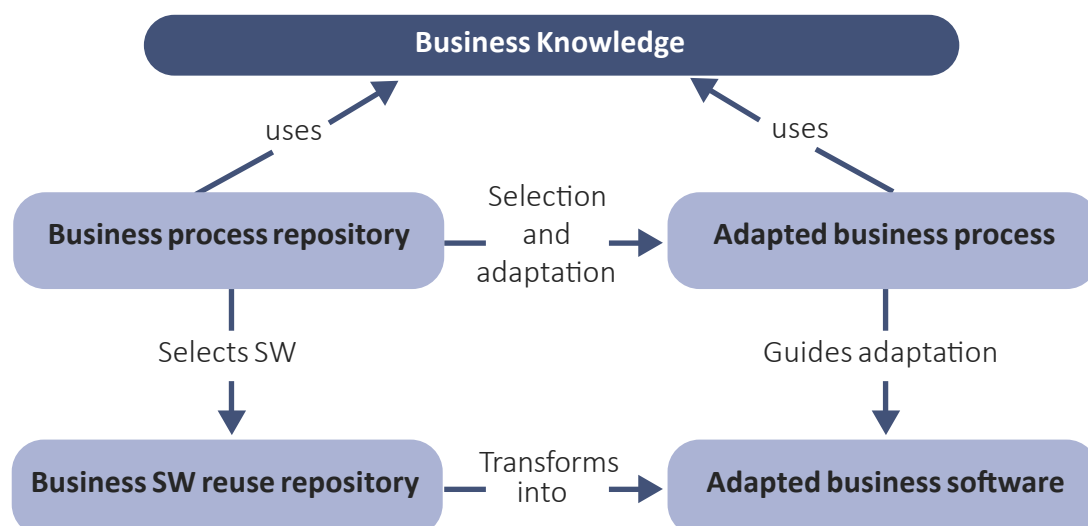
## 5.5. REUSE APPROACHES (ENFOQUES)

Hay tres approaches principales para la reutilización de requisitos:



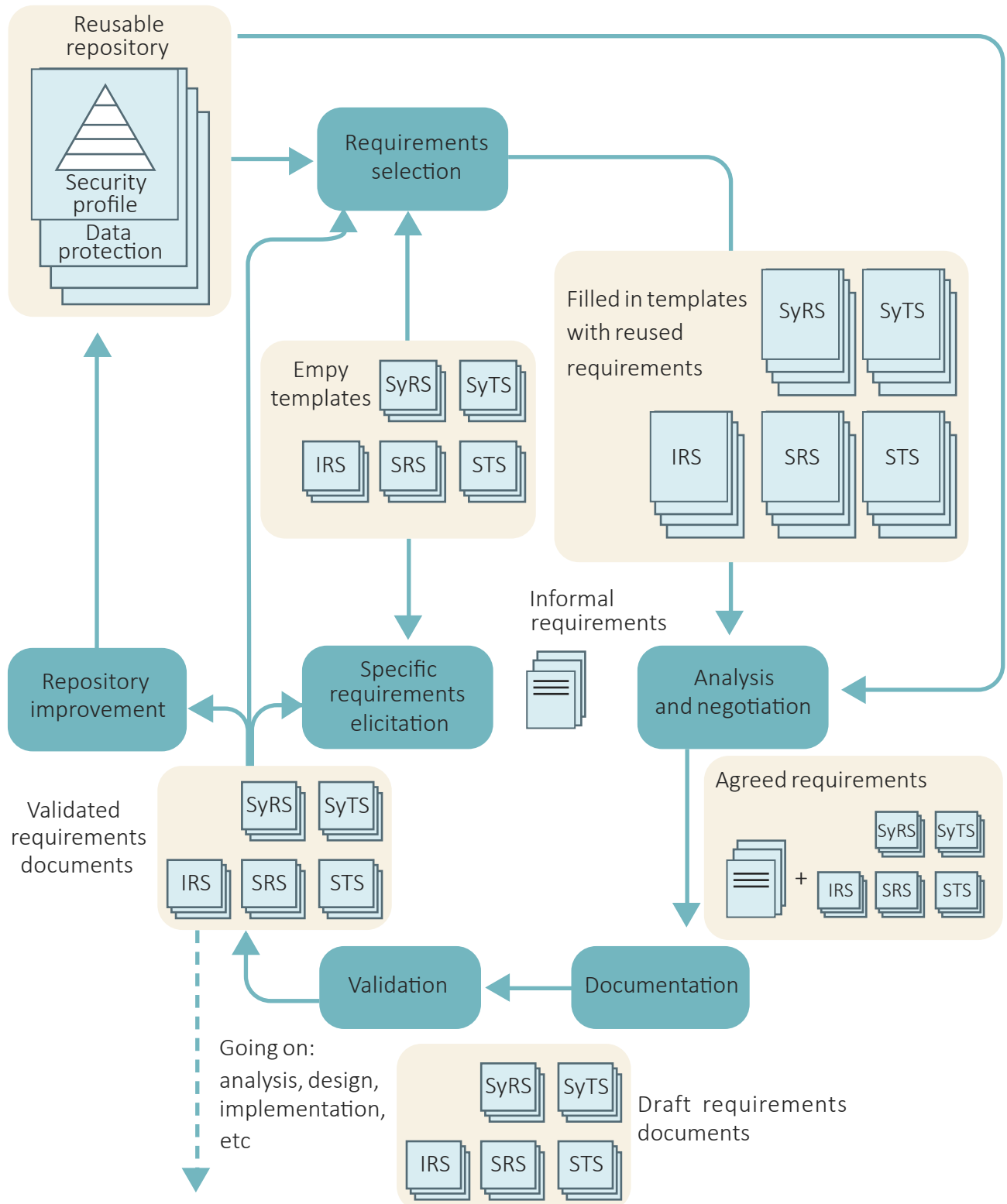
## 5.6. BUSINESS SOFTWARE REUSE APPROACH

Para este approach, el conocimiento del negocio se va almacenando en repositorios, cubriendo procesos de negocio y software del negocio. Sobre la base de ellos, se realizan procesos de selección y adaptación como una alternativa para cubrir nuevas necesidades.





## 5.7. MÉTODO DE REUTILIZACIÓN DE REQUISITOS SIREN



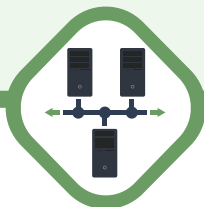
La propuesta SIREN se basa en la utilización de un modelo de proceso en espiral, unas plantillas de documentos de requisitos y un repositorio de requisitos reutilizables, que se encuentra organizado por catálogos.



### 5.8. COMPUTER-AIDED REQUIREMENTS ENGINEERING

#### EXAMPLE: ENTERPRISE ARCHITECT

Cuando configura un repositorio, tiene la opción de estructurarlo para un solo proyecto o para múltiples proyectos, que a su vez podrían organizarse mediante varios programas de trabajo.



Enterprise Architect le da al modelador un control completo sobre cómo está estructurado el repositorio, lo que permite que los paquetes se configuren por encima del nivel de los proyectos donde se pueden agregar algunos requisitos, como los requisitos comerciales, reglamentarios y arquitectónicos.



#### CREAR UN MODELO BASE

Cuando crea un nuevo repositorio en productos de software que cubren la ingeniería de requisitos, como por ejemplo Enterprise Architect, tiene la opción de crear un modelo en blanco usando el asistente de modelo para ayudar a configurar una estructura de repositorio, o puede usar un modelo base como plantilla para el nuevo modelo.

El modelo base es un buen lugar para almacenar activos reutilizables, como requisitos comerciales, reglamentarios y arquitectónicos, y políticas y normas comerciales.

Se puede crear un modelo base usando la opción 'Guardar proyecto como ...' bajo el ícono 'Administración de archivos' (lista de proyectos).

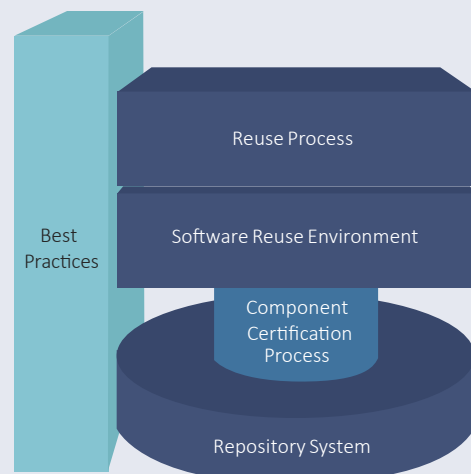


## 5.9. REUSABLE ASSETS (ACTIVOS REUSABLES)



## 6. BEST PRACTICES FOR REQUIREMENTS REUSE

Para acelerar el tiempo de comercialización y reducir los costos de desarrollo, muchos productos de los equipos toman los requisitos escritos para proyectos similares anteriores y los reutilizan para un nuevo proyecto. La reutilización de requisitos permite a los equipos lograr sus objetivos de ahorro de tiempo y costos, sin sacrificar la calidad del producto.



01



### Documentar los requisitos

No puedes reutilizar un requisito a menos que lo tengas documentado en algún lado. Crear un repositorio de búsqueda independiente para requisitos reutilizables, y comenzar a construir en proyectos futuros. Con una gran base de datos de requisitos, sería imposible leer todos los requisitos para ver que son buenos candidatos para su reutilización en el nuevo proyecto.

02

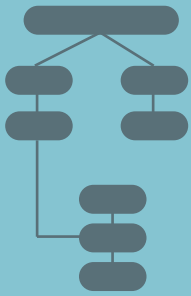


### Tune-up de requisitos existentes

Es poco probable que puedas reutilizar los requisitos de proyectos pasados sin hacer algunos cambios, por lo que deberás examinar tu inventario de los requisitos existentes para ver cómo tendrán que ser ajustado para volverse reutilizable. Si un requisito es específico para un proyecto, se visualiza si puedes hacerlo aplicable a una gama de proyectos generalizándolo.



03



## Iniciar con el propósito en mente

Cuando tienes que escribir un nuevo requisito, se escribe con la reutilización en mente. Generalizar el requisito suficiente que se aplicará a otros proyectos y sistemas, sin dejar de ser útil para el proyecto para el que lo estás escribiendo.

Se necesita más esfuerzo para escribir requisitos reutilizables. La recompensa es que ahorrarás esa cantidad de esfuerzo en el próximo proyecto, y el siguiente, y el uno después de eso, y así sucesivamente.

04



## Evitar la excesiva granularidad

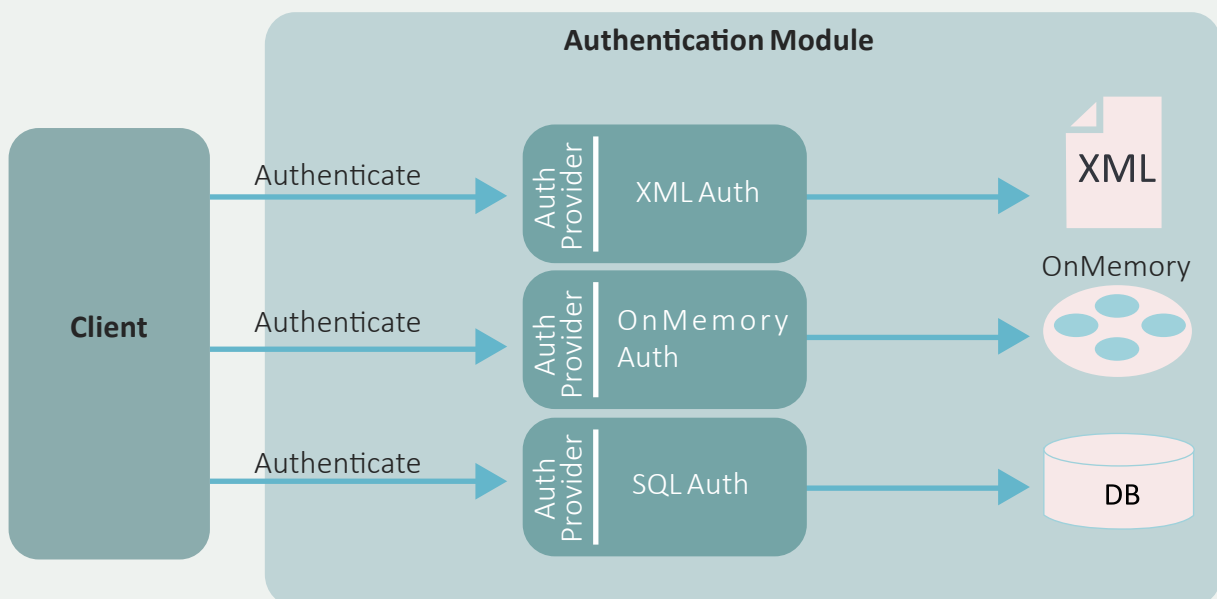
Se puede pensar en la respuesta para mantener los requisitos genéricos y documentar los detalles de cada requisito por separado. Desafortunadamente, esto crea más trabajo a largo plazo, no menos.

05

## Desarrollar un patrón

Un patrón de requisitos puede ayudarte a escribir altamente requisitos reutilizables. Los patrones de requisitos de software tienen el objetivo de permitir escribir requisitos de mayor calidad, más rápido y con menos esfuerzo.

### Execution Flow



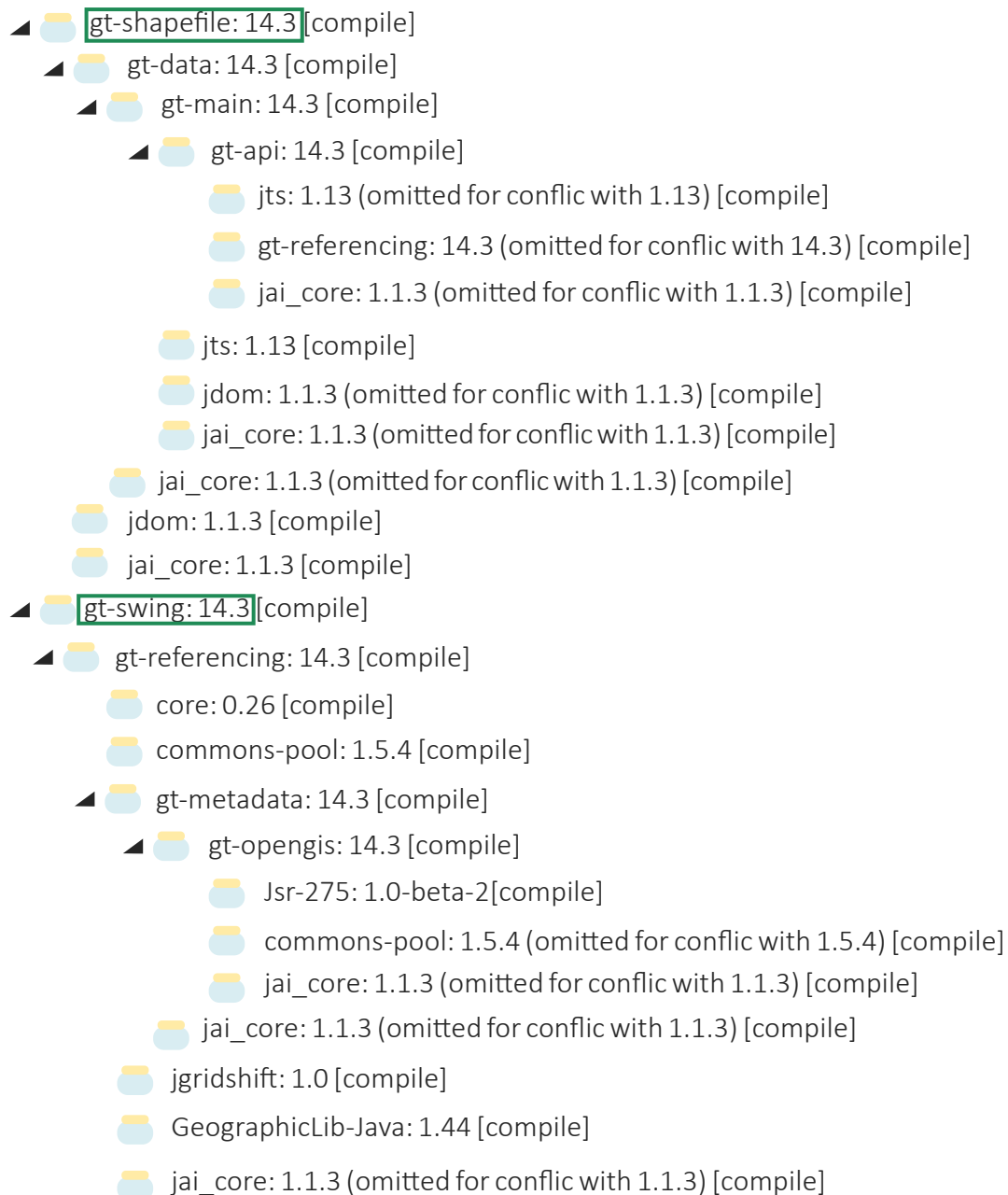




## Vincular las dependencias

Algunos requisitos deben reutilizarse juntos o dependen de otro requisito para trabajar, o tienen un requisito (o varios) que depende de que funcione. Para asegurar estos dependientes, los requisitos no se pierden durante la reutilización, establecen enlaces y hacen que los dependientes sean rastreables el uno al otro.

### Dependency Hierarchy





### 7. CONCLUSIONES

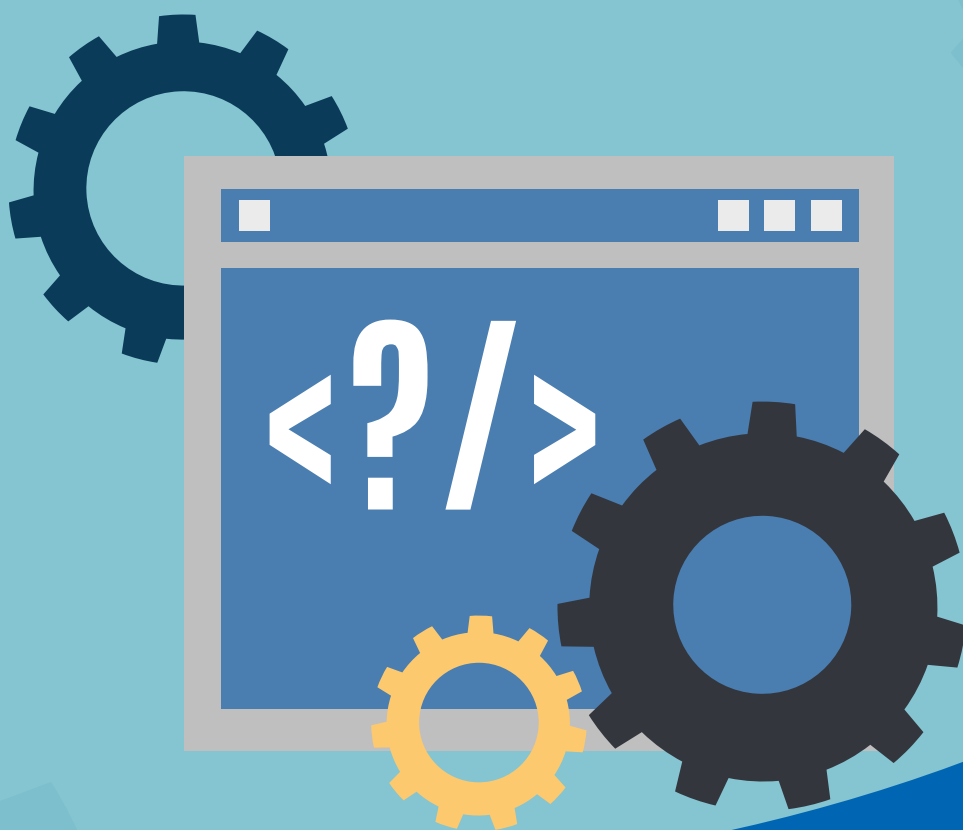
- La reutilización de requisitos parte de identificar que software se pueden reutilizar para la creación de nuevos sistemas a raíz de los ya existentes, integrando elementos de forma directa o pasando por algún proceso de adaptación, lo que permite ahorrar tiempo, ya que no se empieza desde cero.
- La reutilización se basa en 3 principales ventajas: es rápido, factible y económico.
- El desarrollo con reutilización, en el caso que sean requisitos estándar, es más sencillo replicar a varios proyectos porque son exactamente lo mismo para cualquier producto. Se debe tener en cuenta que cuando se configura un repositorio se tiene la opción de estructurarlo para un solo proyecto o múltiples proyectos.
- Los requisitos escritos para proyectos anteriores con similitud ayudan a acelerar el tiempo de comercialización y reducir costos de desarrollo de nuevos productos. Ello conlleva a cumplir los objetivos de ahorro de tiempo y costos sin dejar de lado la calidad del producto.



### REFERENCIAS

Para profundizar:

- <http://www.methodsandtools.com/archive/archive.php?id=68>
- <https://es.slideshare.net/pto0404/seminario-3-reutilizacin-del-software>
- [https://www.ecured.cu/Reutilizaci%C3%B3n\\_de\\_requisitos](https://www.ecured.cu/Reutilizaci%C3%B3n_de_requisitos)
- <https://sites.google.com/site/desarrollodesoftwareuba/reutilizacion>



© UPC. Todos los derechos reservados.

Autor: Ángel Augusto Vasquez Nuñez