



Instituto Nacional  
de Tecnologías  
de la Comunicación

# GUÍA PRÁCTICA DE GESTIÓN DE REQUISITOS

**LNCS**

## AVISO LEGAL

- CMMI® es una marca registrada en la Oficina de Marcas y Patentes de EEUU por la Universidad Carnegie Mellon.
- Las distintas normas ISO mencionadas han sido desarrolladas por la International Organization for Standardization.
- SWEBOOK® (Software Engineering Body of Knowledge) es una marca registrada por el IEEE (Institute of Electrical and Electronics Engineers).

Todas las demás marcas registradas que se mencionan, usan o citan en la presente guía son propiedad de los respectivos titulares.

INTECO cita estas marcas porque se consideran referentes en los temas que se tratan, buscando únicamente fines puramente divulgativos. En ningún momento INTECO busca con su mención el uso interesado de estas marcas ni manifestar cualquier participación y/o autoría de las mismas.

Nada de lo contenido en este documento debe ser entendido como concesión, por implicación o de otra forma, y cualquier licencia o derecho para las Marcas Registradas deben tener una autorización escrita de los terceros propietarios de la marca.

Por otro lado, INTECO renuncia expresamente a asumir cualquier responsabilidad relacionada con la publicación de las Marcas Registradas en este documento en cuanto al uso de ninguna en particular y se eximen de la responsabilidad de la utilización de dichas Marcas por terceros.

El carácter de todas las guías editadas por INTECO es únicamente formativo, buscando en todo momento facilitar a los lectores la comprensión, adaptación y divulgación de las disciplinas, metodologías, estándares y normas presentes en el ámbito de la calidad del software.

## ÍNDICE

<b>1.</b>	<b>INTRODUCCIÓN</b>	<b>6</b>
<b>2.</b>	<b>DESARROLLO DE REQUISITOS</b>	<b>9</b>
2.1.	Obtención de requisitos	9
2.2.	Definición de requisitos	10
2.3.	Verificación de requisitos	11
2.4.	Revisión de especificación	12
<b>3.</b>	<b>GESTIÓN DE LOS REQUISITOS</b>	<b>13</b>
3.1.	Gestión de cambios	13
3.1.1.	Evaluar el impacto	14
3.1.2.	Aceptación del cambio	14
3.1.3.	Implementación del cambio	14
<b>4.</b>	<b>MEJORES PRÁCTICAS</b>	<b>18</b>
4.1.	Mejores prácticas en el desarrollo de requisitos	18
4.2.	Mejores prácticas en la gestión de requisitos	18
<b>5.</b>	<b>ENFOQUE DE ALGUNOS MODELOS</b>	<b>20</b>
<b>6.</b>	<b>REFERENCIAS</b>	<b>21</b>

## ÍNDICE DE TABLAS

Tabla 1	Tipos de requisitos.....	7
Tabla 2	Roles y responsabilidades en el desarrollo y gestión de requisitos.....	8
Tabla 3	Técnicas de recogida de requisitos .....	10
Tabla 4	Buenas prácticas en la documentación de requisitos.....	11
Tabla 5	Matriz hacia atrás / hacia delante .....	16
Tabla 6	Matriz de dependencias.....	17



Instituto Nacional  
de Tecnologías  
de la Comunicación

## ÍNDICE DE FIGURAS

Figura 1	Desarrollo de requisitos .....	9
Figura 2	Proceso de gestión de cambios.....	14

## 1. INTRODUCCIÓN

---

Los mejores productos, desde el punto de vista del usuario, son aquellos creados por desarrolladores que tienen muy claro lo que se pretende conseguir con el producto y cómo obtenerlo. Para llegar a este punto, se debe entender el trabajo del usuario, cómo afectará el producto a su trabajo y cómo se adecuará a los objetivos de la organización.

Lo que hace el producto y las condiciones que debe satisfacer en este contexto son los **requisitos del producto**.

### ¿Por qué necesitamos requisitos?

El **coste** de una buena recogida de requisitos y análisis del sistema a desarrollar es **menor** comparado con el coste resultante de tener requisitos pobres, es decir, el coste de reparar productos deficientes o de poca calidad, el coste de los proyectos cancelados y el coste de haber perdido la oportunidad de tener el producto correcto en el momento correcto.

El fundamento básico de cualquier software recae sobre **su proceso de ingeniería de requisitos**. El éxito o fallo del software depende casi siempre de cómo de bien se hayan capturado, entendido y usado los requisitos como base para el desarrollo. La ingeniería de requisitos es la fase de la ingeniería del software donde se definen las propiedades y la estructura del software. La ingeniería de requisitos comprende el desarrollo y gestión de requisitos.

- El **desarrollo de requisitos** implica entender los requisitos de negocio, identificar los requisitos de usuario y trasladar los requisitos de usuario y de negocio a requisitos de sistema/software.
- La **gestión de requisitos** implica gestionar los cambios de requisitos y mantener la consistencia entre los requisitos y otros productos de trabajo del proyecto.

### ¿Qué es un requisito?

Un **requisito** es algo que el producto debe hacer o una característica que debe tener. Un requisito existe por el tipo de demanda que tiene el producto o porque el cliente quiere que el requisito sea parte del producto entregado. La tarea de todo analista de requisitos es hablar con la gente, entenderla, escuchar lo que dicen y también lo que no dicen, para entender lo que necesitan.

Tipos de requisitos	Descripción
<b>Requisitos de negocio</b>	Dan una <b>descripción a alto nivel</b> de lo que el sistema debe hacer. Representan: los objetivos, la base del negocio, estrategias, visión, alcance y el valor esperado del desarrollo del software
<b>Requisitos de usuario</b>	Son una descripción de las tareas que el sistema ha de ejecutar cuando el usuario opera con él.  Describen la funcionalidad necesaria para <b>satisfacer</b> tareas específicas, necesidades operacionales y grupos de usuarios.
<b>Requisitos del sistema/software</b>	Definen las funcionalidades y características que debe tener el sistema para satisfacer tanto los requisitos de negocio como los de usuario.  Van a servir como <b>base</b> para llevar a cabo la arquitectura, diseño y planes de pruebas del sistema
<b>Restricciones</b>	Son condiciones que <b>limitan</b> las elecciones disponibles al diseñador o programador. Pueden ser restricciones del propio proyecto o del diseño del producto.

*Tabla 1 Tipos de requisitos*

### Roles y responsabilidades

Los roles y responsabilidades que toman parte en el proceso de desarrollo y gestión de los requisitos vienen reflejados en la siguiente matriz RACI.

#### Roles:

RM Gerente

PM Jefe de proyecto

SQA Responsable de Aseguramiento de la Calidad del Software

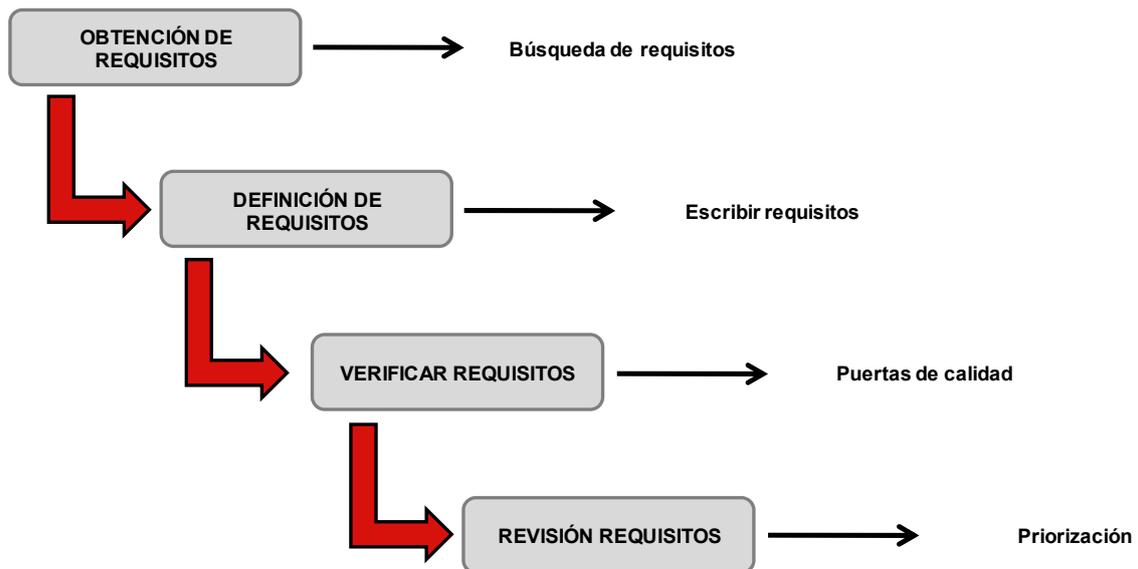
CCB Equipo de Control de Configuración

Actividad	Responsabilidad				Salida
	Preparación	Revisión	Aprobación	Responsabilidad	
Identificar los proveedores de requisitos y las autoridades firmantes	PM			PM	
Documentar los requisitos de usuario y de negocio	Equipo de proyecto	Cliente /RM	Cliente /RM	PM	Requisitos de usuario y de negocio
Documentar los requisitos de software/ sistema	Equipo de proyecto	Cliente /RM	Cliente /RM	PM	Especificación de requisitos software y especificación de casos de uso
Preparar y actualizar la matriz de trazabilidad de requisitos	Equipo de proyecto	SQA	PM/SQA	PM	Matriz de trazabilidad de requisitos
Analizar los requisitos				Equipo de proyecto/ PM	Matriz de trazabilidad de requisitos
Verificar y validar los requisitos. Obtener acuerdo		Cliente	Cliente	PM	Requisitos de usuario y de negocio, especificación de requisitos software, especificación de casos de uso
Línea base de los requisitos		SQA		PM	Línea base de los requisitos de usuario y de negocio, de la especificación de requisitos software y de la especificación de casos de uso
Gestionar cambios a los requisitos	PM	SQA	CCB	PM	Registro de peticiones de cambio, matriz de trazabilidad requisitos.

**Tabla 2 Roles y responsabilidades en el desarrollo y gestión de requisitos**

## 2. DESARROLLO DE REQUISITOS

Las principales actividades realizadas durante el proceso de desarrollo de requisitos son las que se muestran en el gráfico siguiente:



*Figura 1 Desarrollo de requisitos*

### 2.1. OBTENCIÓN DE REQUISITOS

La **obtención de requisitos** se define como el proceso de identificar las necesidades del negocio, solucionando las posibles disparidades entre las personas involucradas en el mismo, con el propósito de definir y destilar los requisitos para cumplir las restricciones impuestas por las distintas partes.

Un buen proceso de obtención de requisitos soporta el desarrollo de la especificación de los requisitos, de tal forma que tengan los siguientes **atributos**:

- Deben ser **completos, consistentes** y han de estar **dentro del alcance** del proyecto
- Deben tener un único identificador
- **Cumplen** con los **objetivos** de los clientes
- Son **viabes** y apropiados para el desarrollo
- Los requisitos han de ser “testeables” (deben tener **capacidad de prueba**).

#### Técnicas de recogida de requisitos

Existen múltiples técnicas que pueden ayudar a la hora de recoger los requisitos de un producto.

Algunas son bastante conocidas como por ejemplo: realizar entrevistas, reuniones, cuestionarios.... y otras como la utilización de casos de uso, escenarios o prototipos, que quizás no son tan comunes y son las que vamos a ver brevemente.

Técnicas	Descripción
Casos de Uso	<p>Describen las <b>interacciones entre el usuario y el sistema</b>, centrándose en qué hace el sistema para el usuario, es decir, la totalidad del <b>comportamiento del sistema</b>.</p> <p>Los pasos a seguir:</p> <ol style="list-style-type: none"> <li>1. <b>Identificar actores</b></li> <li>2. <b>Identificar escenarios</b></li> <li>3. <b>Identificar casos de uso</b></li> </ol>
Prototipos	<p>Un prototipo es un <b>borrador de un producto</b> potencial o de una parte del mismo. Es una simulación de los requisitos.</p> <p>Los prototipos se pueden utilizar cuando los analistas de requisitos no pueden continuar su trabajo porque les <b>faltan datos</b> debido a:</p> <ul style="list-style-type: none"> <li>- el usuario no ha dado suficientes detalles</li> <li>- el producto es tan innovador que nadie conoce realmente sus requisitos</li> </ul>
Escenarios	<p>Son una <b>descripción</b> paso a paso de la <b>funcionalidad</b> de un <i>caso de uso</i> del producto o del negocio, sin demasiado detalle, con el objetivo de hacer entender cómo funciona este caso de uso.</p> <p>Los pasos están escritos en el lenguaje que utilizan las personas involucradas en el negocio</p>

*Tabla 3 Técnicas de recogida de requisitos*

## 2.2. DEFINICIÓN DE REQUISITOS

Conseguir que los requisitos estén claramente definidos puede ser difícil. Para ello es importante:

- **Definir** los requisitos teniendo en cuenta la **perspectiva del usuario**
- **Reutilizar** requisitos, revisando proyectos ya finalizados para ver si contienen material potencialmente reutilizable. La ventaja de esta reusabilidad es que, una vez que un requisito ha sido especificado satisfactoriamente para un producto y que el producto ha tenido éxito, el requisito no tendrá que volverse a inventar, podrá ser utilizado las veces que se desee
- **Documentar** los requisitos de la forma correcta. Aunque escribir los requisitos puede parecer una tarea tediosa, es la única manera de asegurar que la esencia de los requisitos ha sido capturada correctamente, y que esto pueda ser probado.

### Buenas prácticas

Los requisitos son escritos de una forma **tecnológicamente neutra**, es decir, especifican lo que el producto hace y no qué tecnología se usará para crearlo.

Su especificación **no** debe ser **ambigua**.

**Eliminar** todos los **pronombres** de la especificación de requisitos, sustituyéndolos por los sujetos.

Tener cuidado con adjetivos y adverbios ya que pueden llevar a confusiones.

Se debe **evitar usar palabras** que lleven a **confusión** al escribir los requisitos, como 'debería', ya que da a entender que el requisito es opcional.

Al escribir los requisitos una buena técnica es **leerlos en alto**. Y si es posible pedir a alguien que los lea.

**Confirmar** que los involucrados en el negocio tienen el **mismo entendimiento** acerca de los requisitos que la persona que los escribe.

Utilizar una **convención de nombres** y **definiciones común** dentro de la organización. De esta forma se aseguran que en todos los proyectos se está usando el mismo vocabulario.

*Tabla 4 Buenas prácticas en la documentación de requisitos*

Un mal uso del lenguaje puede llevar a un mal entendimiento, horas de trabajo perdido, una mala comunicación entre miembros del equipo, y en definitiva una especificación de requisitos de poca calidad.

## 2.3. VERIFICACIÓN DE REQUISITOS

En esta fase, el usuario final añade **criterios de aceptación** para cada requisito. Además, apoya el hecho de que los requisitos han de ser correctos antes de que sean entregados a los diseñadores y desarrolladores.

La **puerta de calidad** es un punto por el que pasan cada uno de los requisitos antes de formar parte de la especificación.

Una de las tareas de las puertas de calidad es asegurarse de que cada requisito cumple con el criterio que tiene asignado. Este criterio es una medida del requisito que le hace entendible y con capacidad para ser probado.

Otra razón por la que el proyecto tiene puertas de calidad es para prevenir posibles **fugas de requisitos**. Los requisitos, algunas veces, aparecen en las especificaciones sin que nadie realmente sepa de donde vienen o qué valor añaden al producto. Asegurándose de que la única forma de que los requisitos entren a formar parte de las especificaciones sea a través de las puertas de calidad, el equipo del proyecto tiene un **total control de los requisitos**.

## 2.4. REVISIÓN DE ESPECIFICACIÓN

Cuando un requisito pasa por una puerta de calidad, se puede tener confianza acerca de la corrección y viabilidad de los requisitos. *¿Pero qué ocurre con la especificación en conjunto? En este punto se sabe que los requisitos son correctos, pero ¿se puede asegurar que todos estos requisitos, en conjunto, describen la 'historia' completa?*

El término **especificación de requisitos** hace referencia a la colección de requisitos especificados y definidos previamente. La especificación no tiene que estar en un determinado formato, puede ser una especificación sobre papel, o un blog, o algo similar.

Una vez que la especificación de los requisitos está completa se tendrá un conocimiento preciso del alcance y funcionalidad del producto. Este es el momento de llevar a cabo la revisión de la especificación. En esta revisión final **se valida que no falta ningún requisito**.

Otro punto muy importante a tener en cuenta es asegurarse de que los requisitos tengan **consistencia**, y en caso contrario, que cualquier conflicto entre los requisitos ha sido resuelto. Dos requisitos **están en conflicto** si no pueden implementarse juntos, es decir, si la solución a un requisito impide la implementación de otro.

Cuando las expectativas del cliente son altas, la fecha de entrega y los recursos son limitados, es importante asegurarse de que las funciones más importantes del producto sean entregadas y además tan pronto como sea posible. Otro problema que se puede plantear es que haya demasiados requisitos.

La solución a los problemas anteriores es **la priorización** de los requisitos.

### 3. GESTIÓN DE LOS REQUISITOS

---

La **gestión de requisitos** es el conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento. Es decir, la gestión de requisitos consiste en gestionar los cambios de los requisitos, las relaciones entre ellos, las dependencias entre la especificación de requisitos y otros documentos producidos por el proceso de desarrollo de software. De esta forma se asegura la consistencia entre los requisitos y el sistema construido.

Por lo tanto, los **objetivos** principales del proceso de gestión de requisitos serán:

- **Gestionar la recogida de requisitos**
- **Obtener la aprobación de los participantes del proyecto**
- **Gestionar los cambios (trazabilidad)**

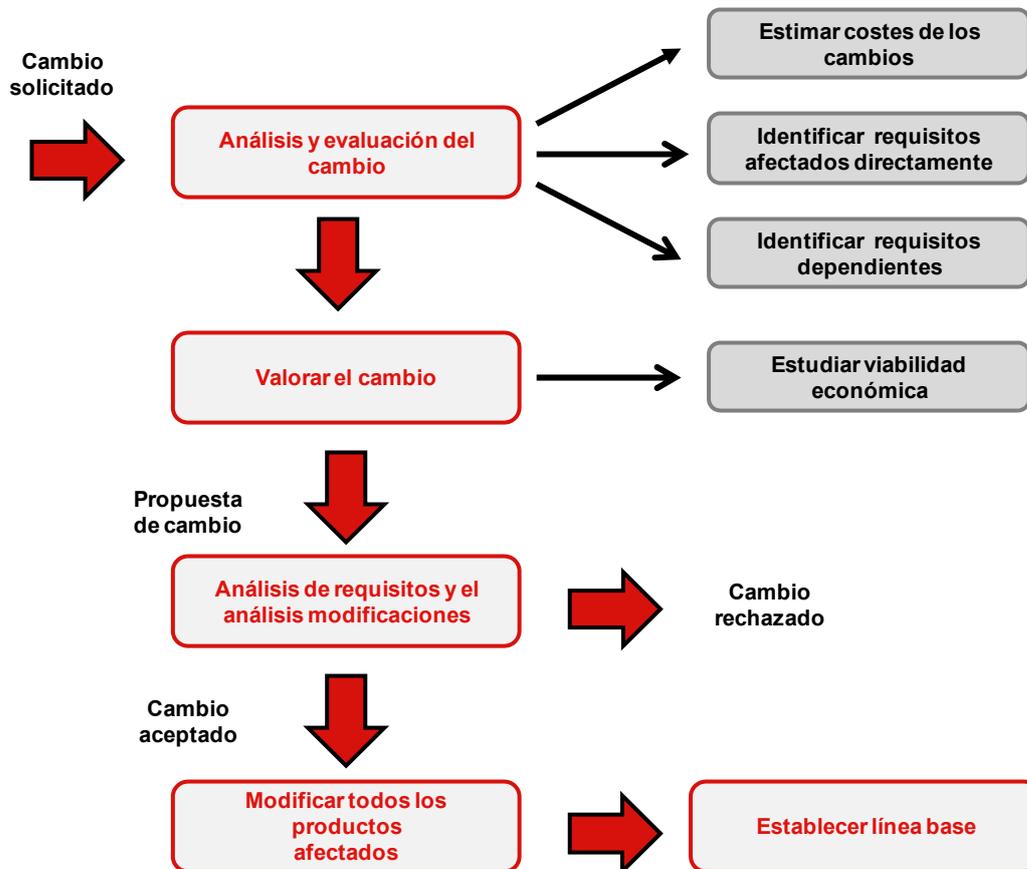
La gestión de requisitos es un proceso que se desarrolla a lo largo de toda la vida del producto.

#### 3.1. GESTIÓN DE CAMBIOS

Los requisitos cambian durante todo el ciclo de vida de desarrollo del producto como se vio en apartados anteriores. Los cambios deben controlarse y documentarse, es decir, hay que convivir con ellos y por ello la gestión de cambios es esencial para tratar dichos cambios.

Cuando, durante el proyecto y una vez aceptada la línea base de requisitos, se **solicita un cambio sobre esta línea base**, estos cambios no se pueden aceptar sin más ya que podrían afectar al desarrollo de todo el sistema, o alguna parte esencial del mismo.

El siguiente gráfico muestra el proceso de gestión de cambios con las actividades a llevar a cabo durante el desarrollo del mismo:



*Figura 2 Proceso de gestión de cambios*

En definitiva, podríamos destacar tres importantes actividades dentro del proceso de gestión de cambios:

### 3.1.1. Evaluar el impacto

La primera tarea a realizar tras recibir una petición de cambio es valorar el impacto del mismo. Para ello se deberá ir recorriendo todo el árbol de requisitos viendo como les afecta el cambio, y aquí es donde entra la trazabilidad de los requisitos.

### 3.1.2. Aceptación del cambio

Una vez analizado el impacto del cambio, se debe tomar una decisión. Si se acepta el cambio, tras negociarlo con el cliente, se continuará con la actividad de implementar el cambio. En caso contrario, se deberá negociar con el cliente el siguiente paso a realizar.

### 3.1.3. Implementación del cambio

Si se ha aceptado el cambio, hay que reflejar ese cambio en todos los productos que resulten afectados por dicho cambio (si el cambio es mínimo algunos productos como el

plan del proyecto, puede que no sea necesario modificar). Además se deberá generar un nuevo punto de partida (**línea base**) de requisitos.

### **Trazabilidad**

Un concepto clave en el proceso de gestión de cambios es la **Trazabilidad**. **Los requisitos deben ser trazables**, es decir, “rastreables”. Se podría decir que un requisito es trazable si se pueden identificar todas las partes del producto existente relacionadas con ese requisito. Todos los requisitos deberían ser trazables para mantener consistencia entre los distintos documentos de un proyecto.

Es importante conocer aspectos de los requisitos tales como:

- Su **origen** (Quién los propuso)
- **Necesidad** (Por qué existe)
- Relación con otros requisitos (**Dependencias**)
- Relación con otros elementos (**Dependencias**)

El uso de matrices de trazabilidad es una buena técnica para llevar a cabo esta actividad de forma eficiente. A continuación se propone una posible matriz de trazabilidad:

Matriz de trazabilidad de Requisitos		Requisitos				Diseño alto nivel	Diseño detallado	Código	ID Caso prueba unitario	ID Caso prueba integración	ID Caso prueba sistema	Peticion de cambio	
		Req. negocio	Req. usuario	Req. Sistema / SW	Caso de uso								

Tabla 5 Matriz hacia atrás / hacia delante

En la matriz se irán registrando los requisitos de negocio. Por cada requisito de negocio se identificarán los requisitos de usuario correspondientes. De cada requisito de usuario se identificarán cuales son los requisitos de sistema asociados a cada uno de ellos.... Y así sucesivamente se irá rellenando toda la matriz de requisitos.

La siguiente matriz se utiliza para relacionar requisitos. Es una matriz de dependencias:

		Requisitos (A)							
		Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8
Requisitos (B)	Req 1		X			X			X
	Req 2							X	
	Req 3					X	X		
	Req 4								X
	Req 5	X							
	Req 6		X						
	Req 7					X			
	Req 8		X						

*Tabla 6 Matriz de dependencias*

En este caso los Requisitos (A) representan los requisitos que originan las dependencias y los Requisitos (B) serían los requisitos que dependen de otros requisitos, de los Requisitos (A). Vamos a poner un ejemplo para verlo más claro.

Por ejemplo, se puede ver como los requisitos 1,3 y 7 dependen del requisito 5, o que el requisito 3, además de depender del requisito 5 también depende del requisito 6. De esta forma se puede ver de qué manera se relacionan los requisitos, para analizar mejor el impacto de los cambios.

## 4. MEJORES PRÁCTICAS

---

En el desarrollo de software, una **mejor práctica** es un método bien definido que contribuye a una implementación exitosa del proyecto software. Las organizaciones crean mejores prácticas para que les ayuden a resolver problemas pudiendo disminuir la tasa de fallo de sus proyectos.

### 4.1. MEJORES PRÁCTICAS EN EL DESARROLLO DE REQUISITOS

A continuación se describen una serie de mejores prácticas orientadas al desarrollo de requisitos:

- **Documentar el alcance y visión del proyecto:** permitirá tener un mejor entendimiento de los requisitos y asegurará que todas las personas involucradas en el proyecto trabajen hacia la misma meta.
- **Mantener un glosario del proyecto:** facilitará una comunicación efectiva asegurando un entendimiento unánime.
- **Uso de técnicas de obtención de requisitos de usuario:** para facilitar esta tarea.
- **Involucrar a toda la gente implicada:** asegura una validación temprana del entendimiento de los requisitos.
- **Desarrollo incremental de requisitos:** puede minimizar la cantidad de re-trabajo del proyecto.
- **Captura de requisitos usando casos de uso:** será más fácil gestionar los requisitos y hacer un seguimiento de los mismos.
- **Validar requisitos:** para mejorar el éxito de los proyectos es crítico que se validen los requisitos de forma adecuada.
- **Verificar requisitos:** para asegurar que los requisitos proporcionan una base adecuada para llevar a cabo el diseño, la construcción y las pruebas.

### 4.2. MEJORES PRÁCTICAS EN LA GESTIÓN DE REQUISITOS

A continuación se muestran una serie de mejores prácticas relacionadas con la gestión de requisitos:

- **Priorizar requisitos:** para determinar aquellos que se deberían cumplir en la primera versión o producto y aquellos que pueden llevarse a cabo en sucesivas versiones.
- **Establecer líneas base de los requisitos:** para asegurar que cualquier modificación en los requisitos que cambie la línea base se trata como cambios de alcance.
- **Comunicación abierta:** para asegurar que la información relacionada con los requisitos se comunica de forma consistente. Una comunicación abierta también implica comunicar a la gente correcta y al conjunto mínimo de personas.



Instituto Nacional  
de Tecnologías  
de la Comunicación

- **Gestión de cambios de los requisitos:** es esencial gestionar estos cambios de forma efectiva y eficiente.
- **Uso de herramientas para la gestión de requisitos:** para facilitar la gestión de requisitos.
- **Mantener trazabilidad de requisitos:** para llevar un seguimiento de la vida de un requisito.
- **Establecer un plan de mejora de procesos para la ingeniería de requisitos:** para cumplir con las necesidades actuales y futuras de forma más eficiente y con mayor calidad.
- **Formar a los analistas de requisitos:** para asegurar que los analistas de requisitos tienen el conocimiento, entre otros aspectos, de cómo escribir buenos requisitos, etc.

## 5. ENFOQUE DE ALGUNOS MODELOS

---

Tanto la gestión como el desarrollo de requisitos son procesos contemplados por los principales modelos de mejora de procesos orientados al desarrollo de software.

**CMMI® y SPICE** son modelos de mejora de procesos que describen los procesos que una organización debe ejecutar para la adquisición, desarrollo y mantenimiento de productos y servicios software. Ambos modelos contemplan, entre sus áreas de proceso, la **gestión y desarrollo de requisitos**. Para implementar correctamente esta área, ambos modelos proponen una serie de prácticas a seguir.

Según CMMI®, el propósito de la gestión de requisitos es gestionar los requisitos de los productos del proyecto y de sus componentes e identificar las inconsistencias entre requisitos, planes del proyecto y productos de trabajo. Las prácticas que propone para llevarlo a cabo son:

- Obtener un entendimiento de requisitos
- Obtener un compromiso con los requisitos
- Gestionar los cambios de los requisitos
- Mantener una trazabilidad bidireccional de los requisitos
- Identificar inconsistencias entre los requisitos y los productos de trabajo del proyecto

CMMI® también contempla el desarrollo de requisitos. Su propósito en aquí es producir y analizar requisitos de usuario, de producto, y de los componentes de producto. Las metas que habría que cumplir para satisfacer esta área de proceso son:

- Desarrollar los requisitos de cliente
- Desarrollar los requisitos de producto
- Analizar y validar los requisitos

Por otra parte, los procesos del modelo SPICE que contemplan las actividades de desarrollo y gestión de requisitos son:

- ENG.1 Recogida de requisitos: recoger, procesar y registrar las necesidades y requisitos cambiantes del cliente a lo largo de la vida del producto y/o servicio para establecer una línea base de los requisitos que sirva de base para definir los productos de trabajo necesarios.
- ENG.2 Análisis de requisitos del sistema: para transformar los requisitos definidos por el cliente en un conjunto de requisitos técnicos deseados que guiarán el diseño del sistema.
- ENG.4 Análisis de requisitos del software: para establecer los requisitos de los elementos software del sistema.

## 6. REFERENCIAS

---

A. Abran, J.W. Moore, P. Bourque, R. Dupuis, *Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, 2004.

Gilb, Tom. *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*. Butterworth-Heinemann, 2005

K.E. Emam, J.N. Drouin, W. Melo, *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE Computer Society Press, 1998. Leffingwell, Dean, and Don Widrig. *Managing Software Requirements: A Use Case Approach*. Second edition. Addison-Wesley, 2003

M.B. Chrissis, M. Konrad, S. Shrum, *CMMI® Second Edition. Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 2007.

R.S. Pressman, *Software Engineering: A Practitioner's Approach*, Sixth ed., McGraw-Hill, 2004.

Sommerville, Ian and Pete Sawyer. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, 1998

Suzanne Robertson y James Robertson, *"Mastering the Requirements Process"*, Segunda edición (2006)