



UNIDAD 11

VALIDACION DE REQUISITOS

11. VALIDACIÓN DE REQUISITOS	1
11.1. REVISIÓN DE REQUISITOS	3
11.2. PROTOTIPOS	6
11.3. GENERACIÓN DE CASOS DE PRUEBA.....	9

El proceso de validación de requisitos comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de requisitos, tal y como se indica en la Figura 1.

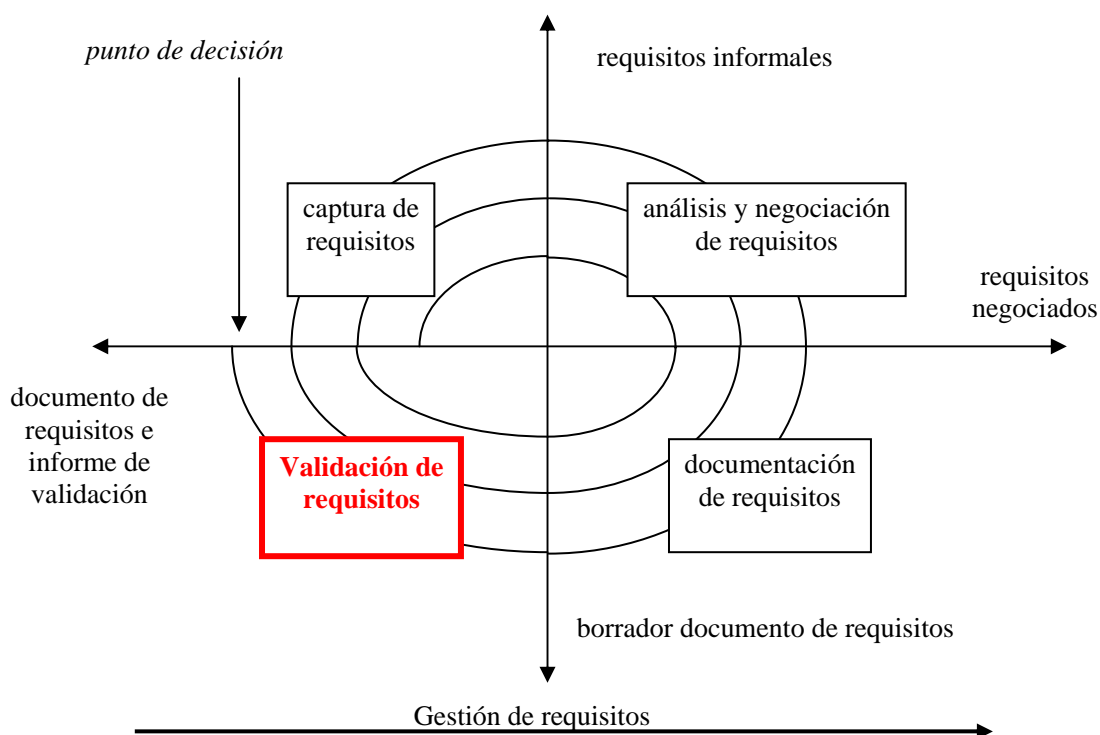


Figura 1. La validación en el proceso de requisitos.

Este proceso tiene por finalidad comprobar que los requisitos del software poseen todos los atributos de calidad enunciados en la UNIDAD 10; en breve: son consistentes, completos, precisos,



realistas, verificables y definen lo que el usuario desea del producto final. La realización de estas actividades en este momento pretende evitar los altos costos que significaría el tener que corregir una vez avanzado el desarrollo.

La actividad de validación tiene como entrada el documento de requisitos, los estándares relacionados y el conocimiento de la organización, y como salida se obtiene una lista de problemas y una lista de acciones recomendadas, tal y como muestra la Figura 2.

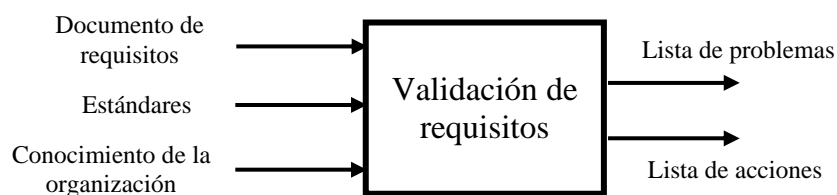


Figura 2. La validación en el proceso de requisitos.

La validación se realiza a través de diversos métodos. Los tres métodos más habituales son:

- **Revisión de requisitos.** Las revisiones de requisitos consisten en reuniones donde un equipo de analistas intenta localizar errores en el documento de especificación.
- **Prototipado.** El método del prototipado consiste en construir una maqueta del futuro sistema software a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud.
- **Generación de casos de prueba (test de requisitos).** Este método tiene como objetivo comprobar la verificabilidad de los requisitos. Consiste en la definición de casos de prueba que permitan verificar el cumplimiento de los requisitos funcionales.

Además de los tres métodos anteriores, existe un cierto número de otras técnicas alternativas, que no se estudiarán de forma específica, pero es interesante mencionar. Éstos son:

- **Creación de manuales de usuario¹.** Este método consiste en verificar si la especificación de requisitos contiene el suficiente detalle² como para preparar el manual de usuario del sistema. De no ser así, podría ocurrir que la especificación fuera incompleta.
- **Animación y validación de modelos o especificaciones formales.** Este método sólo puede emplearse cuando los requisitos se han utilizado modelos durante el análisis o métodos de especificación formal durante la documentación. Consiste en la animación tanto de modelos como de especificaciones formales, con el objetivo de verificar que el funcionamiento del sistema, tal y como está especificado, es el deseado por cliente y usuarios. Asimismo,

¹ La creación de manuales de usuario es un método que puede usarse conjuntamente con el prototipado (ver sección 11.2). Adicionalmente, es una técnica muy similar a la generación de casos de prueba (ver sección 11.3).

² El detalle con el que puede confeccionarse el manual de usuario es variable, ya que depende del grado de detalle del documento de especificación. Si el documento de especificación es muy detallado, también lo será el manual, y viceversa.



algunos modelos y todos los tipos de especificaciones formales admiten la posibilidad de validación de los requisitos mediante demostraciones formales.

Las secciones siguientes describen brevemente los métodos de la revisión de requisitos, prototipado y generación de casos de prueba, indicados anteriormente.

11.1. Revisión de requisitos

La revisión de requisitos es uno de los mejores métodos de validación de requisitos. Generalmente hablando, las revisiones de requisitos permiten:

- Descubrir una gran cantidad de defectos en los requisitos.
- Reducir los costos de desarrollo entre un 25% y un 30%.
- Reducir el tiempo de pruebas entre un 50% y un 90%.

Las revisiones de requisitos consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados. Éstas reuniones son llevadas a cabo por el analista encargado del proyecto y un conjunto de colegas³ que, preferiblemente, no están relacionados con el proyecto⁴ y, además, son competentes en la actividad de requisitos. El resultado final de las reuniones de revisión es un documento que contiene la lista de defectos localizados y una lista de acciones recomendadas⁵.

Las reuniones de revisión se realizan habitualmente siguiendo un procedimiento compuesto por seis pasos:

- **Preparar el plan de la revisión.** Este plan debería incluir al menos lo siguiente: Las tareas a realizar (esto es, las que se indican a continuación), la planificación temporal y las personas que participarán en la revisión.
- **Distribuir los documentos a revisar.** Habitualmente, el único documento a revisar será el documento de especificación. Junto con este documento, también es necesario remitir a los participantes en la revisión todos aquellos documentos que ayuden a comprender adecuadamente el documento de especificación. Típicamente, tales documentos son los documentos referenciados y los anexos a la especificación.
- **Preparar la reunión.** Esta tarea es muy distinta, dependiendo de quien la realice:
 - Para el analista promotor de la reunión, esta tarea es fundamentalmente logística. Básicamente consiste en reservar una sala donde realizar la revisión, solicitar los materiales que sean necesarios (desde papel y lápiz a cañones de proyección),

³ Los cuales pueden ser voluntarios, seleccionados por el analista o asignados a la revisión por cualesquiera otros medios.

⁴ Debido al ya conocido principio de que una persona no debe revisar su propio trabajo.

⁵ Debido al ya conocido principio de que una persona no debe corregir el trabajo de otro.



preparar justificantes del tiempo empleado por el personal participante⁶ (en caso de que sea preciso), etc.

- Para los analistas participantes, la preparación de la reunión es, probablemente, la más importante de las tareas de revisión. Durante esta tarea se deben leer cuidadosamente los documentos recibidos y anotar todas aquellos defectos (o cosas que parezcan defectos) con la finalidad de ponerlos de manifiesto durante la reunión posterior.
- **Realizar la reunión de revisión.** Básicamente, El formato de esta reunión puede ser muy diverso: Puede oscilar desde una total falta de control⁷ hasta grupos muy formalizados y sujetos a protocolos de actuación.
- **Identificar los defectos y acciones a realizar.** La lista de defectos y acciones recomendadas es el documento final obtenido en las revisiones de requisitos. El formato de este documento puede ser similar al indicado en la UNIDA 6, tal y como se muestra en la Figura 3.
- **Realizar las correcciones que sean precisas a los documentos revisados.** El analista promotor de la reunión (esto es, el analista encargado del documento de especificación) debe evaluar y, si lo estima conveniente, llevar a cabo, las acciones recomendadas que han surgido de la reunión de revisión.
- **Informar de las modificaciones realizadas a los participantes en la reunión.** Una vez que los defectos en la especificación han sido subsanados, debería enviarse un breve informe de las tareas realizadas, y una copia corregida de los documentos de especificación, a los participantes en la reunión para su visto bueno⁸.

Nº de requisito	Defectos detectados	Acciones recomendadas
1	Error de estilo, que lleva a ambigüedad	Modificar el texto del requisito de tal forma que diga algo como “El sistema deberá permitir el registro de los fondos bibliográficos”
2	Idem	Idem
11	Ambigüedad	Precisar la duración de las reservas
12	Concisión	No se han identificado diferencias entre profesores y alumnos a todo lo largo de la lista de requisitos. Este requisito se debería eliminar, ya que proporciona ningún tipo de información relevante.
15	Realizabilidad	El sistema no puede realizar automáticamente los préstamos. Quizás se refiere el requisito a que debe proporcionar el máximo de automatización? Precisar, en este caso, o eliminar por irreal.
17	Concisión	Separar lo referido a libros prestados de lo referido a libros reservados.

Figura 3. Un posible formato para la lista de errores y acciones recomendadas

⁶ Independientemente de cómo los analistas hayan sido asignados, típicamente dichos analistas trabajan en otros proyectos distintos del que van a revisar (ya que, en caso contrario, estarían revisando su propio trabajo). Por ello, puede ser necesario proporcionarles documentos que justifiquen cuánto de su tiempo ha sido ocupado por la revisión.

⁷ Por ejemplo: no se registran ni evalúan las reuniones, no se controla el rendimiento individual de cada participante, etc.

⁸ Esta última tarea tiene dos propósitos: Por una parte, mostrar a los analistas participantes el resultado de sus sugerencias y, por otra, obligar al analista a realizar las acciones recomendadas (o, al menos, a justificar por qué no los ha llevado a cabo).



Para llevar a cabo tanto la preparación de la reunión de revisión, como la reunión en sí misma, se utilizan checklists de validación. Dos checklist de validación se adjuntan, a modo de ejemplo, en la sección de Materiales Adicionales. Dichos checklist, al igual que lo que ocurriría durante el análisis, consisten en una serie de preguntas a realizar, o una serie de aspectos a considerar, los cuales pretenden apoyar al analista durante la identificación de defectos en la especificación.

Por ejemplo, el checklist mostrado en la Figura 4 (uno de los adjuntados en la sección de Materiales Adicionales) está compuesto por dos partes importantes: (1) la columna de la derecha, en donde se propone una pregunta que el analista debe realizarse acerca de los requisitos y (2) la columna de la izquierda, que es el atributo de calidad que dicha pregunta pretende examinar. Si la respuesta a alguna pregunta no es afirmativa, entonces se habrá localizado un defecto en la calidad de la especificación.

Example Requirements Review Checklist

Intended use of this checklist	When examining a set of requirements (which may be in a Statement of Work), the reviewer might use the following questions to see that the document is adequate, perhaps while doing some type of review.	
ID	Defect Type	Items to Examine
Organization and Structure of the Documentation		
1	Standards	Have appropriate requirements documentation standards been followed?
2	Standards	Are all figures, tables, and diagrams labeled and referenced?
3	Standards	Are all terms and units of measure defined?
4	Standards	Are all requirements written at a consistent and appropriate level of detail?
5	Standards	Are individual requirements rated (or ranked), with descriptions of priority provided?
6	Unverifiable	Do the requirements provide an adequate basis for design and system test?
Completeness and Correctness		
7	Correctness	Are all internal cross-references to other requirements correct? [For

Figura 4. Checklist de validación

Es aconsejable que cada organización desarrolle y mantenga checklists propios y adaptados a los sistemas que desarrolla. La experiencia muestra que los checklists adaptados son los proporcionan siempre los mejores resultados. No obstante, en caso de no disponer de un checklist propio, puede utilizarse un checklist más o menos general, como los suministrados, o incluso es posible realizar la revisión sin disponer de checklist alguno.

Para finalizar, debe indicarse que la experiencia ha mostrado que antes de las revisiones, es interesante realizar lo que se denomina pre-revisiones. Una pre-revisión es simplemente una revisión rápida e informal de los documentos de especificación. Las pre-revisiones permiten habitualmente identificar errores sencillos (por ejemplo, desviaciones de los estándares, omisiones, etc.) que pueden detectarse fácilmente sin necesitar de un proceso tan largo como son las revisiones. En consecuencia, las pre-revisiones son un buen medio de

- **Permitir que los participantes se concentren en los defectos difíciles de identificar.** Ello es una consecuencia directa de eliminar los defectos más sencillos y evidentes.
- **Disminuir el número de errores y acciones a realizar** ya que, previsiblemente, el número de defectos identificados será mucho menor.



Las pre-revisiones son realizadas por un colega del analista encargado de los requisitos. La única restricción en la selección de esta persona es que, como en el caso general, la persona que realice la pre-revisión no haya participado en la confección de la especificación. Por último, no es necesario utilizar checklists durante las pre-revisiones, ya que su objetivo es identificar errores sencillos y, además, se desea que la pre-revisión finalice rápidamente. Por ello, la experiencia del analista es habitualmente suficiente.

11.2. Prototipos

Los prototipos son un método de validación ampliamente utilizado en muchas disciplinas, y en todos los casos, los principios subyacentes son los mismos: el prototipado consiste en la creación de una maqueta o versión del producto final. Los objetivos de los prototipos varían en función de la disciplina. En el caso de la actividad de requisitos, los prototipos se utilizan, fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos.

Existen varios tipos de prototipos, cada uno de los cuales permite la realización de un tipo determinado de pruebas y con un determinado nivel de realismo. En ingeniería de requisitos, los prototipos más comunes son los siguientes:

- **Mock-ups.** Se trata de pantallas, típicamente dibujadas a mano en papel, que representan un aspecto concreto del sistema. El soporte que proporcionan a la validación es muy limitado, con la excepción, quizás, de aclarar el interfaz gráfico deseado en casos complejos.
- **Storyboards.** Son una evolución de los mock-ups, ya que además del interfaz, se muestra la secuencia de acciones, o escenarios⁹, que se deben realizar con el programa. Por ejemplo: es habitual que, antes de cerrar un programa, se pregunte si se desea cerrar el fichero de trabajo. En un storyboard, la tarea “cerrar programa” podría representarse como se muestra en la Figura 5.

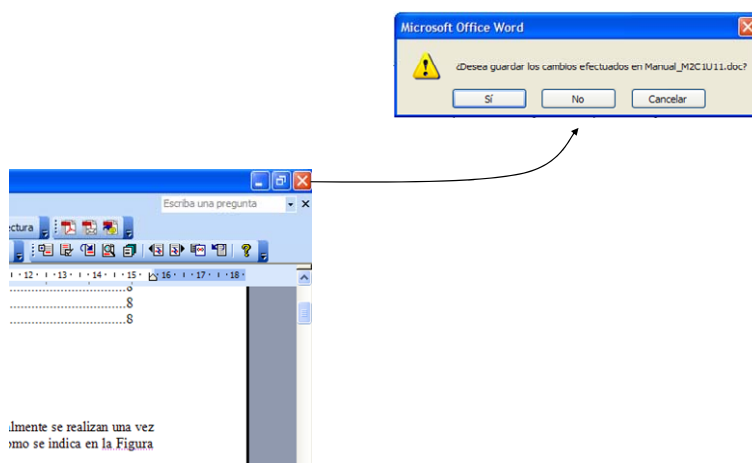


Figura 5. Storyboard

⁹ La utilización de storyboards, o la creación de escenarios (que es aproximadamente lo mismo) es muy similar a la creación de casos de uso esenciales (o incluso reales) expandidos, y por lo tanto extremadamente útil para el diseño orientado a objetos.



Existen varias alternativas para crear storyboards.

- La opción más sencilla es confeccionar pantallas individuales. Posteriormente, a medida que se valida el prototipo, éstas pueden pegarse sobre una superficie donde se pueda pintar (una pizarra, por ejemplo) y relacionarse con flechas. De este modo se pueden reutilizar muchas pantallas distintas en distintos escenarios.
- Una opción más compleja es dibujar, sobre una hoja de papel, el conjunto completo de pasos para llevar a cabo una tarea.
- Se pueden utilizar diversos programas (desde Powerpoint o Visio a programas especializados de creación multimedia como Macromedia Director) para confeccionar storyboards. La posibilidad de enlazar páginas que ofrecen estos programas dotan de gran dinamismo al storyboard y lo hacen muy similar a las maquetas, comentadas más abajo.

Por último, nótese que se pueden crear storyboards sin necesidad de ningún artefacto gráfico. Por ejemplo, un flowchart, un DFD o un DTE¹⁰ pueden hacer las veces de storyboard.

- **Maquetas.** Una maqueta es una versión simplificada del sistema software deseado¹¹. Típicamente, una maqueta representa únicamente el interfaz del sistema y, opcionalmente, las conexiones entre pantallas mediante la utilización de elementos activos como los botones. Si fuera necesaria mayor fidelidad, podrían codificarse partes del sistema, de tal modo que además, del interfaz, el software pudiera ofrecer algunos resultados reales. Ello es lo que se conoce como “prototipo funcional”.

Las maquetas acostumbran a realizarse mediante programas gráficos de diseño, tales como Visual Basic o PowerBuilder, por citar únicamente dos ejemplos.

Una pregunta que surge con frecuencia es: ¿Qué tipo de prototipo se debe construir? La respuesta depende de dos factores:

- **Los recursos y tiempo disponibles.** Cuanto más limitados sean estos recursos, más conveniente es construir prototipos sencillos como los mock-ups o storyboards.
- **La fidelidad deseada.** Este es un problema de índole psicológico. Si el usuario experimenta rechazo por el prototipo, al no ser capaz de entender que el prototipo es sólo una versión simplificada del producto final, se debe tender a construir prototipos más complejos. Tampoco se debe olvidar que el analista deberá suplir al usuario la información que el prototipo no pueda suministrar (mensajes de error, resultados de los cálculos, etc.). Ello es también un argumento para preferir prototipos más complejos.

¹⁰ La animación de modelos tiene cierta relación con esto. Básicamente, animar un modelo consiste en simular que el modelo realiza una determinada tarea. Esta simulación podría consistir, por ejemplo, en marcar los procesos del DFD, o estados del DTE, que están “activos” en cada momento. Ello, obviamente, es muy similar a un storyboard.

¹¹ Cuando se habla de prototipos, habitualmente se quiere decir “maquetas”.



No obstante, construir el prototipo es únicamente una parte, y quizás la más sencilla, de la tarea de validación. Para validar los requisitos, es necesario realizar adicionalmente las tareas indicadas en la Figura 6.

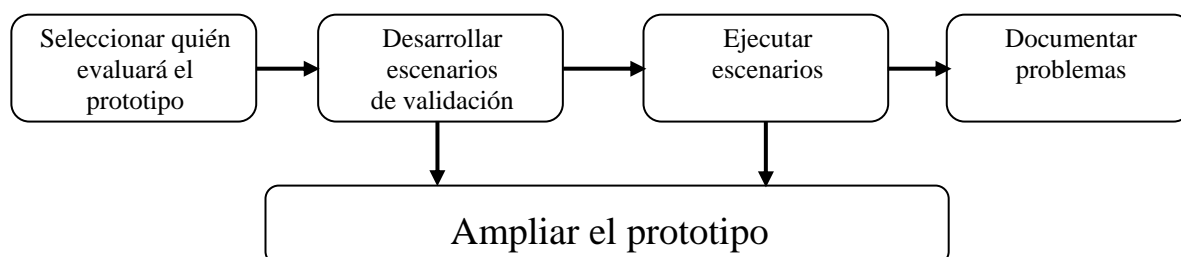


Figura 6. Prototipo en el proceso de validación de requisitos.

Las tareas indicadas en la Figura 6 son las siguientes¹²:

- **Seleccionar quién evaluará el prototipo.** Para que la evaluación del prototipo sea lo más efectiva posible, deben seleccionarse adecuadamente los usuarios que participarán en la evaluación. Siempre que sea posible, debe seleccionarse un conjunto de usuarios representativos de los distintos perfiles, de tal modo que sea posible validar el software en todos sus modos de utilización. Por ejemplo: Si un software X es utilizado por administrativos, ingenieros y contables, al menos un usuario de cada uno de estos tipos de usuarios debería evaluar el prototipo de X. En la mayoría de los casos, los usuarios pueden identificarse directamente a partir del detalle de la especificación.
- **Desarrollar escenarios de validación.** En muchos textos, se dice que el prototipo se presenta a los usuarios para que “jueguen con él”. Ello, no obstante, dista de ser cierto. Para evaluar correctamente el prototipo, deben identificarse una serie de escenarios o tareas, los cuales deben ser llevados a cabo por los usuarios utilizando el prototipo. Ejemplos de estos escenarios podrían ser, por ejemplo: crear un pedido para el que hay existencias, crear un pedido con algún producto sin existencias suficientes, crear un pedido para un producto inexistente, etc.
En la mayoría de los casos, dichos escenarios pueden identificarse directamente a partir del detalle de la especificación. Otra alternativa para generar los escenarios es intentar confeccionar un manual de usuario para el sistema. Si este manual está correctamente confeccionado, deberá contener las alternativas de utilización del sistema, lo cual es exactamente lo mismo que un escenario. Por último, nótese que confeccionar un storyboard es lo mismo que diseñar gráficamente un escenario de utilización.
- **Ejecutar los escenarios.** Esta tarea consiste en que el usuario realice, o ejecute, cada uno de los escenarios previstos. Dependiendo del tipo de prototipo utilizado (mock-up, storyboard, maqueta, etc.), el analista deberá orientar al usuario acerca de la diferencia entre el producto final y el prototipo (de tal modo que el usuario no se muestre descontento). Asimismo, el analista deberá paliar las carencias del prototipo, proporcionando al usuario la

¹² Nótese que no existe ningún camino de “marcha atrás” en la Figura 6. Ello se debe a que el prototipo, al ser usado para VALIDAR, y no para EDUCIR, no se perfecciona incrementalmente y, en teoría, se debería desechar. Adicionalmente, la actividad de validación tiene un punto de terminación bien definido.



información que el prototipo no ofrezca y que el usuario necesite para ejecutar los escenarios; Esto es: el analista debe suplir la funcionalidad no presente en el prototipo.

- **Documentar los problemas.** Esta tarea consiste, simplemente, en confeccionar la lista de problemas encontrados. Dado que el analista que confecciona los requisitos es, habitualmente, el mismo que valida el prototipo con los usuarios, no es usual confeccionar una lista de problemas recomendados.

A medida que se localizan problemas, pueden corregirse y retroalimentar al prototipo para que refleje los cambios, o no realizar ninguna acción. Hacer uno o lo otro dependerá de los defectos identificados. Si estos son tantos, o tan severos, que inutilizan el prototipo como herramienta de validación, sin duda los problemas debe corregirse inmediatamente. En caso contrario, es mucho más eficiente terminar completamente la validación y, posteriormente, corregir todos los defectos en la especificación.

11.3. Generación de casos de prueba

Uno de los atributos de calidad de los requisitos es que sean verificables. Existen dos razones para ello. La primera, que es normalmente la que enuncia, es que los requisitos deben ser verificables para poder determinar su cumplimiento o ausencia durante las pruebas de sistema y aceptación.

Existe, sin embargo, una segunda razón: Además de lo indicado anteriormente, la verificabilidad exige que los requisitos estén muy bien enunciados, sean consistentes y estén razonablemente completos. Por ello, comprobar que los requisitos sean verificables implica indirectamente una cierta confianza en la calidad de los requisitos.

El único método existente para comprobar la verificabilidad de un requisito es que sea posible definir uno o varios casos de prueba para dicho requisito.

Los casos de prueba son artefactos bien definidos en el contexto de la prueba del software. En dicho contexto, un caso de prueba es la descripción de una acción bien definida que se debe realizar con el software. Por bien acción bien definida, debe entenderse que están perfectamente descritos tanto los datos de entrada como las tareas a realizar y los resultados esperados.

Durante la validación de requisitos, los casos de prueba se utilizan del mismo modo que durante la prueba del software: es necesario poder describir, como se ha indicado anteriormente, tanto los datos de entrada como las tareas a realizar y los resultados esperados, para lo cual es necesario que estén perfectamente descritos los requisitos.

Por ejemplo; dado el requisito:

X. El sistema deberá generar facturas



Es posible describir la tarea a realizar (invocar la función *generar factura*), pero no los datos de entrada ni los resultados a obtener. Ello ocurre porque el requisito está incompleto y, lo que es más importante, es fácil indicar qué aspectos del requisito no están bien definidos. Éstos son:

- No se indica cómo se deben generar las facturas (¿una única factura, una factura por cliente, una factura por pedido, etc.?)
- No se indica qué se factura (¿un pedido específico, los pedidos entre dos fechas, etc.?)

Los defectos anteriores no están presentes en la siguiente definición del mismo requisito:

X. El sistema deberá generar facturas

X.1. Se generará una factura por cliente. Esta factura incluirá todos los pedidos no facturados del cliente en cuestión.

X.2. El formato de la factura será el especificado en el anexo 1

Para el que el caso de prueba correspondiente sería¹³:

1. *Se introducirán en el sistema los pedidos A, B, C, ...*
Estos pedidos A, B y C se tratarían de pedidos bien definidos, indicando todos los datos necesarios (cantidades, productos, etc.)
2. *Se invocará la función de generar factura*
3. *Se deberán obtener las facturas X, Y, ...*
Estas facturas X e Y se tratarían, de nuevo, de facturas bien definidas

El método de la generación de casos de prueba produce dos resultados:

- De igual modo, que los demás métodos de validación, una lista de errores (que el requisito X no es verificable) y acciones recomendadas (los aspectos del requisito X que es necesario completar: datos de entrada, datos de salida, etc.).
- Un conjunto de casos de prueba que pueden ser utilizados en la posterior prueba de sistema y aceptación.

¹³ Un problema de la definición de casos de prueba es conjugar la indefinición del documento de especificación con la precisión requerida por un caso de prueba. Dicho de otro modo: en ciertas ocasiones, la especificación de requisitos es deliberadamente ambigua porque el cliente no exige un producto altamente específico. En estos casos, y a medida que la indefinición aumenta, el método de la generación de casos de prueba se vuelve cada vez más inefectivo.