

Técnicas para verificar y validar requisitos

Tabla de contenido

Introducción	2
Verificación y validación de requerimientos.....	2
Pruebas y revisiones	4
Tipos de revisiones.....	4
Revisiones de pares.....	5
Revisiones (Walkthroughs).....	5
Definición	5
Etapas	5
Inspecciones	6
Definición	6
Etapas	6
Reglas	7
Beneficios	8
Prototipos.....	8
En la validación de requerimientos	8
Conclusiones.....	9
Resumen.....	9

Introducción

Verificación y validación de requerimientos

Habíamos visto en capítulos anteriores cuáles son las actividades de la ingeniería de requerimientos y hoy nos ocupan 2: La verificación y la validación. La verificación apunta a determinar si hemos descrito correctamente los requerimientos, mientras que la validación apunta a evaluar si entendimos bien los requerimientos. Parece un juego de palabras, pero vamos a intentar profundizar un poquito más en detalle qué significa cada una de estas cosas.

Antes que nada, es importante hacer un par de comentarios, o dar un par de definiciones con respecto a algunos temas:

- Por un lado, ¿qué es **calidad**? Cada uno de los autores que han estado trabajando en el tema ha dado su propia definición, nosotros vamos a tomar una que es la de la ISO (Organización Internacional de Estándares) que dice que **la calidad es la totalidad de prestaciones y características de un producto o servicio que genera su capacidad de satisfacer necesidades explícitas o implícitas**.
- También es importante que aclaremos un poquito esa diferencia entre verificación y validación:
 - **Verificación** apunta a determinar si construimos bien, es decir, si **aplicamos correctamente las técnicas para transformar correctamente el input en output**. Es el típico caso de la prueba de software, por ejemplo. La prueba que apunta a ver si el programa funciona bien.
 - La **validación**, por otra parte, tiene como objetivo determinar si **hemos construido el producto correcto**. Y es el ejemplo clásico de la prueba de aceptación. Nosotros podemos hacer todas nuestras pruebas, estamos seguros de que nuestro sistema funciona bien, pero hasta que no validamos con el usuario que lo que construimos es lo que quería, no vamos a poder dar por terminado nuestro proyecto, no vamos a poder dar por terminado nuestro desarrollo. La validación, en general, tiene que ver con poder utilizar el/la producto/solución desarrollado/a en el entorno operativo deseado o en un entorno muy similar al entorno donde finalmente va a ser operado ese producto. Esta es la diferencia fundamental.

La verificación...

- ☒ **es aplicar las técnicas para transformar correctamente el input en output**
- ☐ es por ejemplo una prueba de aceptación
- ☒ **apunta a determinar si construimos bien**
- ☒ **es por ejemplo una prueba de software**
- ☐ tiene como objetivo determinar si se ha construido el producto correcto
- ☐ tiene que ver con poder utilizar la solución desarrollada en el entorno operativo deseado

La validación...

- ☐ es por ejemplo una prueba de software
- ☐ apunta a determinar si construimos bien
- ☒ **tiene que ver con poder utilizar la solución desarrollada en el entorno operativo deseado**
- ☐ es aplicar las técnicas para transformar correctamente el input en output
- ☒ **tiene como objetivo determinar si se ha construido el producto correcto**

También es importante que aclaremos 2 términos que a veces se utilizan en forma intercambiable: Control de calidad y aseguramiento de la calidad.

- ◆ El **aseguramiento de la calidad** es algo que no tiene nada que ver ni con el testing ni con las pruebas. El aseguramiento de la calidad apunta a evaluar si los productos y mis procesos, es decir, los procesos que estoy utilizando para producir mi producto, se adhieren a los estándares aplicables. Eso es **Quality Assurance**.
- ◆ **Quality Control** apunta más a la evaluación operativa, si se quiere, para determinar si lo que hemos producido cumple con los requerimientos de calidad. **Claramente la verificación y la validación caen dentro de esta categoría de control de calidad.**

Pero recuerden, siempre que hablen de Quality Assurance, por las dudas, pregunten de qué es lo que están hablando, porque a veces hay gente que utiliza de forma incorrecta aseguramiento de la calidad como un sinónimo de control de calidad o de prueba. Pero lo cierto es que son 3 cosas diferentes. Podríamos hablar de una *trilogía de elementos en calidad*:

- **Control de calidad** que es lo más operativo, por ejemplo, las pruebas, las revisiones, las inspecciones.
- **Aseguramiento de la calidad** que nos permitan determinar si estamos utilizando/aplicando correctamente los procesos y los estándares organizacionales
- **La mejora de la calidad** que apunta a tratar de, continuamente, mejorar la forma en la cual estamos produciendo nuestros productos y proveyendo nuestros servicios.

[V / F] El aseguramiento de la calidad apunta a evaluar si los productos desarrollados y los procesos utilizados para ese desarrollo se adhieren a los estándares aplicables.

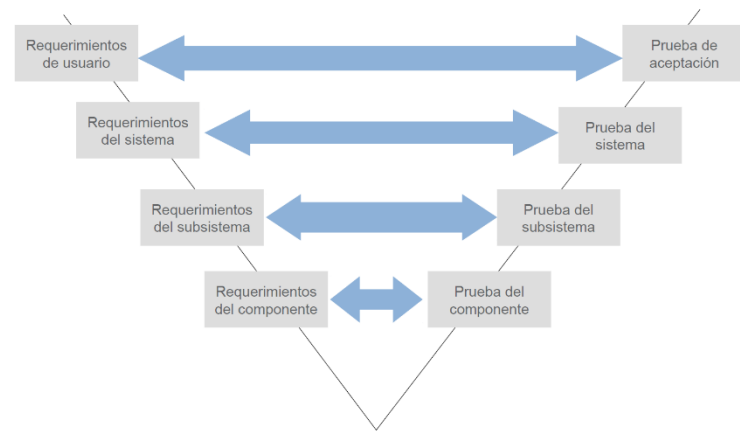
¿Qué tienen que ver la verificación y la validación con respecto al mundo de los requerimientos? Bueno, como decíamos al comienzo, la verificación de requerimientos apunta a confirmar que los requerimientos hayan sido correctamente formulados. Es decir, ¿escribí bien la especificación de requerimientos en función del input que tuve para especificar esos requerimientos? O, ¿el modelo que construí está bien construido de acuerdo a las reglas de construcción del modelo y aparte está bien desarrollado a partir del input que tuve? La validación consiste en definir si hemos entendido correctamente los requerimientos a partir de lo que nos propuso nuestro usuario o los interesados. Dicho en otras palabras, por ejemplo, me reuní con el usuario, discutimos una serie de funcionalidades, prepare una minuta de reunión a manera de resumen de lo charlado, chequeo que ese resumen de la reunión sea consistente con las notas que tomé, y después le devuelvo esa minuta de reunión para que el usuario me valide si lo que entendí está bien. O, por ejemplo, yo produzco un modelo de casos de uso o una especificación de requerimientos detallando lo que fui aprendiendo de lo que el sistema tiene que hacer, la verifique, es decir, chequeé la calidad de esos artefactos, desde el punto de vista mío, y ahora viene la etapa en donde le planteó, le devuelvo el documento o la especificación al usuario para que me diga si entendí bien lo que me pidió.

[V / F] La verificación y la validación son parte del control de calidad.

El otro tema sumamente importante que hay que aclarar es que, contrario a lo que mucha gente cree, **el control de calidad o la verificación-validación no son cosas que se hacen al final de un proyecto**, sino que existen múltiples oportunidades para que esa confirmación, esas actividades de verificación y validación sean continuas a lo largo del ciclo de vida. No es algo que se hace al final, sino que hay que tratar de hacer en puntos intermedios del proyecto.

¿Cuál es el papel que tienen los requerimientos en las actividades en general de control de calidad a lo largo del proyecto? Bueno, importantísimas. Este es el **modelo V** ampliamente difundido en la industria. Fíjense que lo que nos plantea es que los requerimientos del usuario nos permiten hacer lo que llamamos la prueba de aceptación. La prueba de aceptación es la que apunta a determinar si el producto que construimos es el correcto. ¿Contra qué vamos a contrastar la prueba de aceptación? Con los requerimientos del usuario. Los requerimientos del sistema, es

decir, la especificación de lo que el software tiene que hacer, la vamos a contrastar contra la prueba del sistema. La prueba del sistema es la prueba que hacemos internamente nosotros para chequear que el sistema funcione bien y que cumpla con los requerimientos que nosotros entendimos, con nuestras especificaciones. Y si seguimos bajando en la descomposición del sistema, en subsistemas y componentes, nos vamos a encontrar que en el último nivel hay requerimientos que son aplicables a una pieza determinada y lo que vamos a hacer es utilizar, para generar los casos de prueba o por lo menos para como parámetro para esa actividad de verificación, la especificación de esa componente para hacer una prueba unitaria o prueba del componente. Así que claramente, los requerimientos forman parte integral del proceso de gestión de la calidad del proyecto, de gestión de la calidad del producto, y son un input sumamente importante. Es por eso que venimos diciendo ya desde los primeros capítulos que no hay que esperar a terminar de desarrollar el software para empezar a pensar cómo vamos a hacer las pruebas y qué cosas vamos a probar. Una muy buena práctica es cuando uno define el alcance del software que va a construir es, como hemos dicho ya, también una oportunidad de definir cuáles son las condiciones de aceptación, los criterios de aceptación y entre ellos, los tipos de pruebas y los casos de prueba en el mejor de los casos, que se van a ejecutar. ¿Y de dónde sale eso? Claramente de los requerimientos que nosotros hayamos acordado.

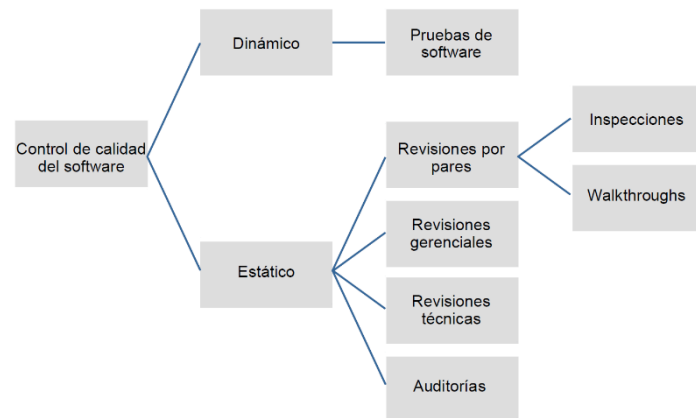


[V / F] La verificación y la validación se realizan generalmente hacia el final de los proyectos.

Pruebas y revisiones

¿Qué tipo de actividades de control de calidad tenemos en el mundo del software? Bueno, básicamente tenemos 2 grandes tipos de actividades de control de calidad:

- Las que tienen un enfoque un poco más **dinámico**, que son las pruebas. Es un control de calidad dinámico porque se ejecuta el software para ver cómo funcionan.
- Y hay otro grupo de enfoques que son **estáticos**. Es un control de calidad estático porque no se ejecuta el software, se hace algún tipo de examinación o de evaluación. Entre ellas nos vamos a encontrar con lo que se llaman: **Revisiones de pares**, o por pares del autor, que son una familia que incluye a las inspecciones y a los walkthroughs o revisiones propiamente dichas.



Tipos de revisiones

Esta es la clasificación que hay de los tipos de revisiones que se puede llegar a encontrar uno en el mundo del software. Hay algunas que tienen que ver con la revisión gerencial, otras tienen que ver con revisiones técnicas, algunas imponen limitaciones a la cantidad de material a revisar, otras no. Y ahí están estas 2 técnicas que mencionábamos recién: las inspecciones y los walkthroughs. Vamos a ir viendo cada una de estas un poquito más en detalle. Dicho sea de paso, ambas sirven para revisar distintos tipos de artefactos del ciclo de vida. Podemos utilizar esas revisiones para evaluar código, casos de prueba. En el caso nuestro vamos a utilizarlas para evaluar requerimientos, pero tengan presente que se pueden usar para cualquier tipo de artefacto.



Revisiones de pares

Las revisiones de pares son una familia de métodos. La verdad que se utilizan métodos similares en otras áreas, digamos, por ejemplo, cuando uno escribe un paper y lo publica un congreso, hay pares del autor que revisan ese documento, ese paper, y hacen sugerencias. La idea de las revisiones de pares aplicadas a nuestra disciplina justamente tiene ese mismo enfoque, hay pares del autor de lo que queremos revisar, que participan de la revisión y que tratan de encontrar defectos en ese artefacto que estemos revisando.

Las revisiones por pares...

- ☐ están relacionadas a las revisiones gerenciales y técnicas
- ☒ **pertenecen al enfoque estático de control de calidad**
- ☒ **poseen límites en cuanto a la cantidad de participantes y el material a revisar**
- ☐ se utilizan solamente para evaluar los requerimientos dentro del ciclo de vida del software
- ☒ **incluye a las inspecciones y a las revisiones**

No hay específicamente una técnica de peer review o de revisión de pares o por pares, hay distintas alternativas. Nosotros vamos a hablar de los Walkthroughs o revisiones y de las inspecciones.

Revisiones (Walkthroughs)

Definición

Este tipo de revisiones son relativamente informales. Las lidera el autor del artefacto a revisar. No siempre es necesaria una preparación previa para poder participar de la revisión. Y en general, no hay límites a la cantidad de material que se revisa, ni tampoco a la duración, ahí depende de cada uno como la quiera organizar.

Etapas

En general, lo que se sugiere es que haya una hora de duración y alternativamente, una hora de preparación. La hora de preparación simplemente es para que los participantes de la revisión puedan familiarizarse con el material que se va a revisar. Después, tiene lugar la revisión, que es básicamente una reunión en donde participa el autor y quienes haya invitado él a participar de la revisión. Normalmente es el autor quien auspicia de presentador o coordinador de la reunión. Pueden haber algunos roles diferenciados, algún secretario que tome nota, algún revisor que tenga un rol específico. Y en definitiva, de ahí en más el formato es relativamente libre. Se puede utilizar para la verificación. Se puede utilizar para la validación. En general, la idea es que dependiendo la audiencia se oriente más hacia el lado de la verificación. La regla fundamental en este tipo de actividades es que el management no participa.

Las revisiones (walkthroughs)...

- ☒ **son informales**
- ☐ tienen etapas bien definidas
- ☒ **se permite la discusión**
- ☐ pueden participar otros roles como secretarios y management
- ☒ **las lidera generalmente el autor del artefacto**
- ☐ poseen moderador

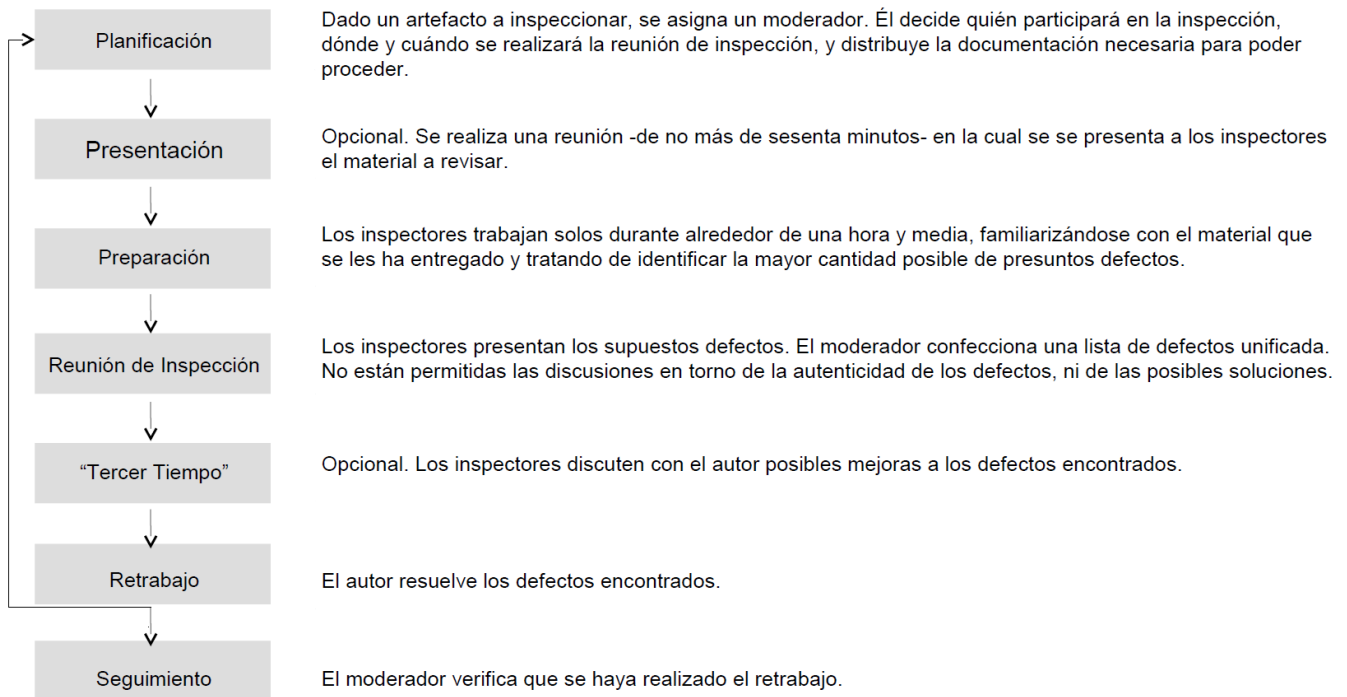
Inspecciones

Definición

Las inspecciones son mucho más formales. Hay roles, etapas, artefactos detallados. Tienen su origen en un trabajo que se hizo en IBM en los 70. Se implementaron exitosamente para inspeccionar código fuente. Y al igual que el anterior, no hay ningún participante que tenga relación con el management o con el liderazgo del proyecto.

Etapas

Básicamente las etapas son las que estamos viendo aquí:

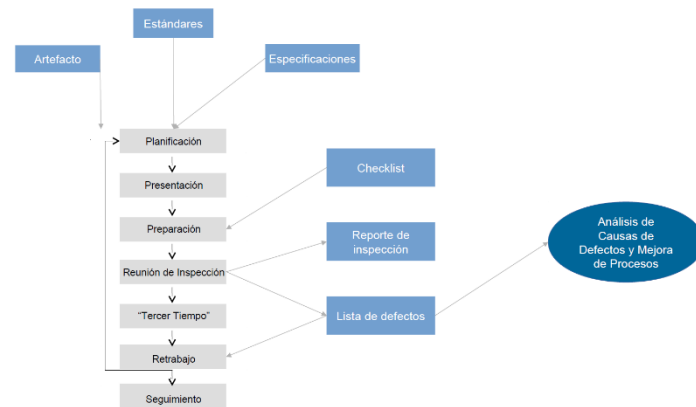


Hay una etapa de **planificación** en donde, identificado un artefacto a inspeccionar (una pieza de código, un documento de requerimientos, un plan, lo que ustedes definan que se puede utilizar como elemento a inspeccionar), se asigna un moderador. El moderador tiene que estar capacitado en la técnica. Y él se encarga de seleccionar a los participantes de inspección, definir dónde y cuándo se va a realizar la reunión de inspección; Y finalmente, distribuye la documentación que hace falta para hacer la revisión propiamente dicha. Cada uno de los participantes de la inspección, trabajan en forma individual, tratando de buscar defectos. Esa es la etapa de **preparación**. Aproximadamente trabajan en solitario una hora y media y el propósito de ese trabajo es detectar la mayor cantidad posible de potenciales defectos, potenciales cosas que estén mal en lo que se está revisando. Puede pasar que haga falta una actividad previa, en donde los inspectores se familiaricen con el proyecto y/o el artefacto que se va a revisar, pero eso es una actividad opcional. Una vez que los inspectores tienen sus propias listas de defectos o de potenciales defectos, tiene lugar la reunión de inspección. La **reunión de inspección** también está acotada, es una hora u hora y media, la coordina el moderador, y la idea ahí es que cada uno de los inspectores presente los supuestos defectos que encontró. Se consensua una lista de defectos única. Entonces, ¿hay varios inspectores que encontraron el mismo defecto? Bueno, probablemente eso sea un defecto que debemos informar. Entonces se trata de armar una lista unificada de los defectos encontrados por los distintos inspectores. No se discute en la reunión de inspección cómo resolver los defectos. No se pierde tiempo en eso. Se utiliza sí, una posible etapa de **"tercer tiempo"** en donde los inspectores se reúnen con el autor para discutir cómo se pueden llegar a resolver esos defectos que se encontraron. Lo que sí es importante es que no participan de las inspecciones los autores. En general, no participan. Sin embargo, hay una variante, hay 2 enfoques para llevar adelante las inspecciones: Fagan y Gil son los 2 autores. Fagan no objeta que estén el/los autor/es del producto que se está revisando/inspeccionando. Gil dice mejor que no estén, porque si no uno toma las cosas a lo mejor en forma personal, los inspectores por ahí se sienten incómodos, entonces es una situación que puede potencialmente a no llegar a ser productiva, entonces más vale que no participe. Si lo que puede pasar entonces es que los inspectores se reúnan luego con el autor para explicarle qué es lo que se les ocurrió a ellos con respecto a cómo resolver esos

defectos encontrados en el artefacto. Lo que sí es importante es que luego de terminar la reunión de inspección y eventualmente la reunión de tercer tiempo tiene que tener lugar el **retrabajo**. Con lo cual el moderador no termina su trabajo al finalizar la reunión de inspección, sino que tiene que chequear que el autor haga el retrabajo correspondiente si es que se encontraron defectos.

[V / F] Las inspecciones tienen como objetivo encontrar defectos y luego se puede discutir el origen de los mismos.

Es importante, para que esto funcione bien, que uno tenga estándares. Si vamos a revisar código, lo ideal es tener estándares de programación, tener estándares que nos permitan clasificar a los defectos, por ejemplo, a ver qué es un defecto grave, es un defecto que se resuelve fácilmente, es un defecto invalidante. Bueno, hay que definir ahí algún esquema de clasificación. También lo otro que es sumamente importante es contar con checklist, con listas de verificación que nos permitan hacer más fácil el trabajo previo de preparación de los inspectores. A ver, por ejemplo, la lista de chequeo podría ser: Está tal sección del documento. Tal sección ¿es clara?, ¿es consistente con el resto? Y, obviamente, esa lista de defectos que armemos nos tiene que servir no sólo para hacer el retrabajo, sino para hacer análisis de causas de defectos y hacer análisis de posibles mejoras. Si vemos que toda la gente se equivoca en tal aspecto del artefacto, en tal parte del documento, evidentemente hay algo que no funciona bien y deberíamos plantearnos si realmente no tenemos que hacer algo con eso. Por ahí capacitar, cambiar el estándar o lo que sea que corresponda.



Reglas

¿Cuáles son las reglas? Bueno, las reglas de las inspecciones es que el objetivo es encontrar defectos. Como decíamos antes, el objetivo no es repararlos, ni discutir cómo hacerlo, porque sino la discusión es eterna. La reunión de inspección tiene que durar una hora u hora y media y si nos disponemos a discutir acerca de cuál es la mejor manera de acceder de tal manera a una tabla o de mostrar tal caso de uso, vamos muertos. Aparte, recuerden que el origen de esto era para inspeccionar código, con lo cual ahí podemos estar discutiendo años acerca de cómo hacer el código más performante, cómo hacer esto, cómo hacer aquello. También hay que tener presente que no hay que tomarse personalmente las actividades de control de calidad y, fundamentalmente, las inspecciones. No es que nos están evaluando a nosotros, están evaluando lo que nosotros generamos. Hay que separar a la gente de lo que se está evaluando. El equipo tiene que estar formado por entre 3 y 6 personas, que deben ser pares del autor. Las reuniones no tienen que durar más de 2 horas, porque la cantidad de material tiene que ser acotado, no puede ser una reunión de inspección para revisar un libro entero. Tiene que ser algo que podamos evaluar en 2 horas. Y hay que tener en cuenta que hay roles claramente diferenciados: Está el moderador que coordina la reunión; el presentador, que participa en la reunión previa, contándonos de qué se trata el producto que estamos construyendo; el Secretario que toma notas y que es el que está preparando la lista de defectos, pasando en limpio la discusión o la conversación que tiene lugar durante la reunión.

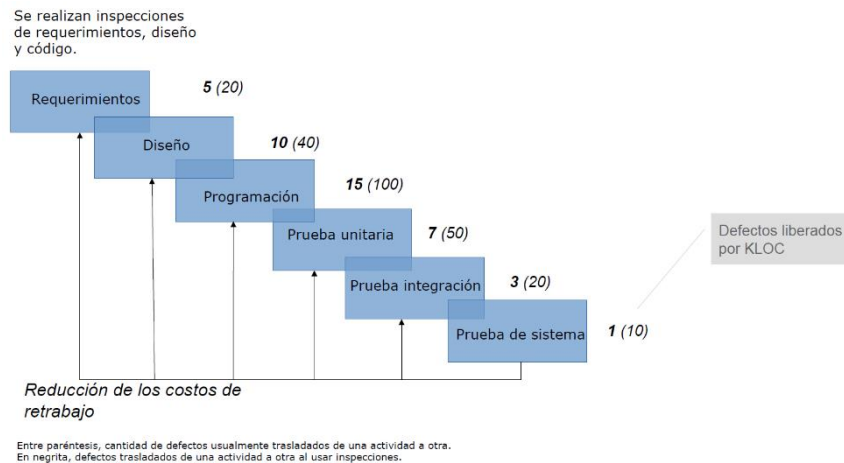
Las etapas opcionales del proceso de inspecciones son:

- ☐ Planificación y tercer tiempo
- ☒ **Tercer tiempo y presentación**
- ☐ Presentación y retrabajo

Beneficios

La verdad es que la técnica de inspecciones es muy buena. Pueden identificar tempranamente hasta un 80% de los defectos de un producto de software. Junto con el testing, pueden reducir los defectos en un orden de 1 a 10, permiten un incremento del 25% al 30% de la productividad. Requieren una inversión de alrededor del 15% del costo total del desarrollo y en general se utilizan, en el contexto de ingeniería de requerimientos, en la verificación al igual que la revisión de pares, y se utilizan fundamentalmente en la verificación. La revisión estructurada o walkthrough, adicionalmente se podría llegar a utilizar para la validación, pero son técnicas para emplear para la verificación.

Aquí tenemos un gráfico explicando los beneficios. Suponiendo que se realizan inspecciones de requerimientos, diseño y código, fíjense que los errores o defectos que se van trasladando de una actividad a otra está representado por estos números que están ahí a la derecha, y fíjense que normalmente de una actividad a otra se trasladan 20 defectos, 40 defectos, 100 defectos, etc¹. Con la aplicación de esta técnica, lo que los estudios demuestran es que se reducen notablemente. Fíjense que la cantidad de defectos que terminan en producción sin aplicar inspecciones son 10 por cada 1000 líneas de código y utilizando inspecciones junto con testing es 1 defecto por cada 1000 líneas de código.



Este es un claro ejemplo de una técnica de ingeniería de software que le llevó a la ingeniería de software 30 años terminar de adoptar. Es un dato interesante porque muchas veces salen métodos, técnicas que se ponen de moda y la verdad que algunas se quedan y otras pasan, pero es un buen estudio para entender que muchas de las cosas que se plantean en ingeniería de software llevan décadas hasta ser aceptadas en forma general y ser adoptadas como una buena práctica. De hecho, las inspecciones se utilizan muchísimo hoy en muchos segmentos de la industria. Sin ir más lejos, la revisión que tiene lugar durante la programación de a pares, que propone XP², por ejemplo, básicamente es una inspección que se está haciendo online. Pero pensemos en la comunidad open source en donde claramente hay revisiones de pares. Realmente son sumamente potentes como instrumento para evaluar la calidad.

[V / F] Utilizando la técnica de inspecciones se reducen notablemente la cantidad de defectos que se trasladan entre las actividades dentro del ciclo de desarrollo de software.

Prototipos

En la validación de requerimientos

Una técnica alternativa para la verificación y la validación son los prototipos. Se pueden emplear en la validación con los usuarios, con la salvedad de que tienen que entender que el prototipo no es necesariamente el diseño final de la interfaz del usuario, sino que hay que concentrarse en los detalles funcionales, más que en los detalles de diseño.

¹ Hace referencia al número entre paréntesis que está después del número en negrita en cada caso.

² Extreme Programming

Conclusiones

Resumen

- ★ Hemos visto entonces que la verificación se ocupa de determinar si el producto está bien construido, es decir, si satisface las especificaciones.
- ★ La validación está orientada entonces a determinar si el producto construido es el producto correcto, es decir, si satisface el propósito para el cual fue creado.
- ★ En el contexto que nos ocupa, es decir, el de la ingeniería de requerimientos, la verificación está orientada a evaluar la calidad de los requerimientos, mientras que la validación se ocupa de garantizar que estos requerimientos resuelven el problema.
- ★ La verificación y la validación de los requerimientos es continua a lo largo de todo el proceso y que básicamente podemos utilizar revisiones, inspecciones y prototipos.
- ★ Las revisiones son más informales que las inspecciones y las inspecciones requieren una serie de pasos ordenados, roles claramente establecidos y probablemente algún soporte de checklist y de listas estándar. También vimos que las inspecciones son un muy buen mecanismo para mejorar, en general, la calidad del producto que entregamos.

[V / F] Los prototipos se utilizan generalmente para validar.