



plan  
avanza2,»» ) inteco (

Instituto Nacional  
de Tecnologías  
de la Comunicación

# CURSO DE CALIDAD DE UN PRODUCTO SOFTWARE

**Laboratorio Nacional de Calidad del  
Software**

## NOTA DE EDICIÓN

Este curso ha sido desarrollado por el Laboratorio Nacional de Calidad del Software de INTECO. Esta primera versión ha sido editada en Junio del 2009.

Copyright © 2009 Instituto Nacional de Tecnologías de la comunicación (INTECO)



El presente documento está bajo la licencia Creative Commons Reconocimiento-No comercial-Compartir Igual versión 2.5 España.

Usted es libre de:

- copiar, distribuir y comunicar públicamente la obra
- hacer obras derivadas

Bajo las condiciones siguientes:

- **Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
- **No comercial.** No puede utilizar esta obra para fines comerciales.
- **Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Esto es un resumen legible por humanos del texto legal (la licencia completa) disponible en <http://creativecommons.org/licenses/by-nc-sa/2.5/es/>

El presente documento cumple con las condiciones de accesibilidad del formato PDF (Portable Document Format).

Se trata de un documento estructurado y etiquetado, provisto de alternativas a todo elemento no textual, marcado de idioma y orden de lectura adecuado.

Para ampliar información sobre la construcción de documentos PDF accesibles puede consultar la guía disponible en la sección [Accesibilidad > Formación > Manuales y Guías](#) de la página <http://www.inteco.es>.

## ÍNDICE

---

<b>1.</b>	<b>ESCENARIO DE APERTURA</b>	<b>4</b>
<b>2.</b>	<b>CONTEXTO DE CALIDAD</b>	<b>5</b>
2.1.1.	El papel de la calidad en un proyecto exitoso	6
2.1.2.	¿Qué quieren los clientes?	7
<b>2.2.</b>	<b>Realidad de los proyectos</b>	<b>7</b>
<b>2.3.</b>	<b>Definición de calidad</b>	<b>8</b>
2.3.1.	Distintas perspectivas de calidad	9
	<i>Perspectiva de calidad del cliente</i>	<i>9</i>
	<i>Perspectiva de calidad del desarrollador</i>	<i>9</i>
<b>2.4.</b>	<b>Niveles de calidad</b>	<b>9</b>
<b>3.</b>	<b>COSTE DE LA CALIDAD</b>	<b>11</b>
<b>4.</b>	<b>GESTIÓN EFECTIVA DE LA CALIDAD DEL PRODUCTO</b>	<b>14</b>
4.1.	Gestión de requisitos	15
4.2.	Diseño de pruebas	17
4.3.	Gestión de defectos	19
4.4.	Gestión de la puesta en marcha	21
<b>5.</b>	<b>CARACTERÍSTICAS DE CALIDAD SEGÚN ISO 9126</b>	<b>23</b>
<b>6.</b>	<b>ESCENARIO DE CLAUSURA</b>	<b>25</b>
<b>7.</b>	<b>GLOSARIO</b>	<b>27</b>

## Escenario de apertura

La empresa COMPASS S.A. dedicada al desarrollo de productos SW está teniendo grandes pérdidas económicas en los últimos meses.

El director de la empresa tras recibir informes con estos resultados negativos decide reunir a los jefes de proyecto de su organización para tomar medidas y solucionar esta situación tan desfavorable que está provocando la pérdida de clientes y la mala fama de la empresa.



Figura 1. Escenario de apertura I

## Contexto de calidad

---

En el mercado tan competitivo de hoy en día, la calidad ha llegado a ser un factor crucial y cada vez más empresas insisten en este punto. Las organizaciones también se están dando cuenta de que sólo conseguirán atraer a nuevos clientes y mantener a los que ya tienen proporcionándoles productos y servicios de alta calidad.

Aunque los responsables de los proyectos software se esfuerzan por cumplir los requisitos de los clientes y entregar productos de calidad, en muchas ocasiones fallan en este intento. Esto es debido a que su **percepción de calidad** es muy distinta a la de los clientes.

El **éxito de un proyecto** normalmente se mide sobre 3 ejes:

- Agenda: ¿El proyecto fue entregado a tiempo?
- Coste: ¿El proyecto se mantuvo dentro del presupuesto?
- Calidad: ¿El producto tuvo una buena calidad?

Un proyecto exitoso es aquel que se completa a tiempo, permanece dentro del presupuesto y tiene desarrolladas las funciones requeridas. Cumplir los requisitos de calidad es uno de los aspectos a considerar en un proyecto exitoso.



Figura 2. Identificando las causas de la situación de la empresa

## El papel de la calidad en un proyecto exitoso

La mayoría de las veces, los responsables de proyectos software tienden a centrarse principalmente en hacer cumplir los hitos del proyecto y en estar dentro del presupuesto y simplemente esperan que la calidad del producto sea aceptable. Esto es debido a que tanto el presupuesto como los hitos (el cronograma) son fácilmente medibles y por lo tanto cualquier desviación que ocurra será fácilmente identificable por la dirección.

Los jefes de proyecto y demás responsables en el proyecto están demasiado ocupados estableciendo hitos de entrega y haciendo cálculos para ajustarse en el presupuesto que la calidad toma un último lugar entre sus tareas.

También debido a que la calidad del software no es visible ni medible, los problemas no llegan a ser obvios hasta que son extremadamente difíciles de corregir. Además, algunos métodos de mantener el cronograma previsto y de controlar costes tienen como resultado o incluso se basan en disminuir la calidad en el producto.

## ¿Qué quieren los clientes?

Hasta ahora hemos visto cómo el éxito de un proyecto depende del cronograma, del coste, y de la calidad. Pero, ¿cuál de estos factores es más importante para el cliente?

Algunos responsables del proyecto piensan que los clientes están más interesados en una entrega del proyecto a tiempo o de un desarrollo a bajo coste. Sin embargo, están confundidos. Los clientes pueden perdonar que un proyecto esté ligeramente fuera de fechas. Pueden incluso perdonar un proyecto con un coste mayor de lo planificado. Pero **ningún cliente perdonaría un proyecto de baja calidad**. Por lo tanto hay que trabajar para conseguir una alta calidad.

## Realidad de los proyectos

Hoy en día las compañías son cada vez más conscientes de la importancia de una buena gestión de la calidad de sus productos. Existen multitud de estadísticas que demuestran con datos esta realidad. A continuación se expone un ejemplo. Son los resultados del **informe CHAOS del Standish Group** que es una de las estadísticas más reconocidas a nivel mundial.

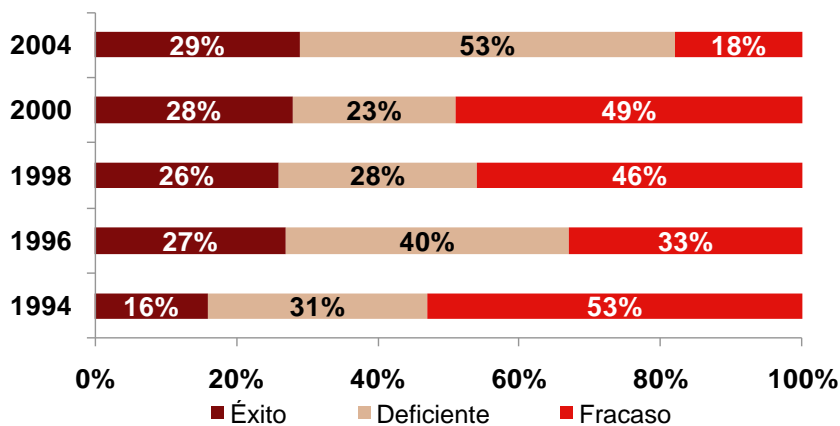


Figura 3. La realidad de los proyectos. Fuente: Chaos Report 2004

Algunas de las conclusiones del informe son:

- **Menos de un tercio** de todos los proyectos TI a nivel mundial **cumplen sus objetivos**

- Cerca del **20% fallan** provocando enormes costes financieros y de recursos.
- Uno de los **factores que más afecta** a los proyectos de software es la forma de implementar y ejecutar la **gestión de la calidad**.
- A pesar de los avances en procesos TI y en la tecnología, los proyectos casi siempre finalizan **fuera del tiempo** establecido y no disminuyen sus costes.
- En la actualidad los proyectos son cada vez más complejos y la **necesidad de una mejora en su gestión y una adecuada planificación** es cada vez mayor, sobre todo cuando hay fuertes restricciones de presupuesto y tiempo.
- En la mayoría de los casos **cuando el tiempo escasea** la gente tiende a **sacrificar la calidad** del producto y esto es un claro error.

## Definición de calidad

*“Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas.”*

**ISO 8402**

*“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”* **R.S.Pressman (1992)**

*“Calidad es la idoneidad de uso. Es decir, las características del producto que satisfacen las necesidades del cliente y, por tanto, producen satisfacción de producto. La calidad es la inexistencia de deficiencias”* **Juran**

*“La calidad se define, desde el punto de vista del cliente, como cualquier cosa que aumenta su satisfacción”* **Deming**

*“Nivel al que una serie de características inherentes satisfacen los requisitos”*  
**ISO 9000: 2000**

*“La capacidad de un conjunto de características inherentes de un producto, o componente del producto, o proceso, de satisfacer por completo los requisitos del cliente”* **CMMI® (S.E.I.)**



## Distintas perspectivas de calidad

---

Para cualquier organización que entregue productos software, la calidad juega un papel importante. Por lo tanto es importante entender lo que compromete la buena calidad. Pero la calidad se puede ver desde diferentes perspectivas.

### Perspectiva de calidad del cliente

---

Un producto está construido para cumplir los requisitos del cliente. Estos requisitos son de dos tipos:

- Explícitos: Lo que el cliente plantea explícitamente
- Implícitos: Lo que el cliente no especifica pero espera

### Perspectiva de calidad del desarrollador

---

Para los desarrolladores un producto es de calidad cuando todas las especificaciones dadas por el cliente se han cumplido. Sin embargo, los clientes pueden fallar al especificar algunos requisitos o especificarlos de forma no clara.

A menudo, el fracaso al construir un producto de calidad es el resultado de una mala especificación de requisitos o de la especificación de requisitos ambiguos. En el contexto del software, el entendimiento entre los desarrolladores es distinto cuando los requisitos implícitos no son recogidos durante la fase de análisis. Por lo tanto estos requisitos no son incluidos en las especificaciones que los desarrolladores intentan cumplir.

Además, el desarrollador puede interpretar especificaciones ambiguas de forma diferente a lo que el cliente intentó. Como resultado, el producto creado puede no ser útil para el cliente.

## Niveles de calidad

La calidad del software puede gestionarse a distintos niveles:

- A **nivel de producto**: cuando nos centramos en el proceso de desarrollo de software y hacemos una serie de pruebas en paralelo con cada etapa, para detectar y corregir los posibles defectos que puedan surgir.

- A nivel de proyecto: cuando nos centramos en controlar todas las fases y áreas de gestión de proyecto, implantando metodologías y mejores prácticas que aseguren la correcta gestión de las mismas.
- A nivel de proceso: cuando nos centramos en gestionar todas las áreas de proceso de una organización, mediante la implantación de una metodología. De esta forma se consigue tener mayor información de los procesos de modo que puedan controlarse y mejorarse, y produzcan así, un aumento de la calidad de los productos y servicios relacionados con ellos.

Mejorar en algunos de los niveles implica la mejora en el resto, dado que en la realidad los niveles de producto, proyecto y proceso están interrelacionados, no siendo posible la mejora independiente de uno de ellos.



Figura 4. Gestión de la calidad

## Coste de la calidad



*Figura 5. Coste de la calidad*

Todo proyecto tiene su coste. En el siguiente gráfico hemos clasificado este coste en dos grandes grupos:

- **Coste de ejecución** que son los costes que supone la planificación y el propio desarrollo del proyecto
- **Coste de calidad** que son los costes derivados de temas relacionados con la calidad. Estos a su vez se dividen en dos grandes grupos:
  - Costes de conformidad
  - Costes de no conformidad

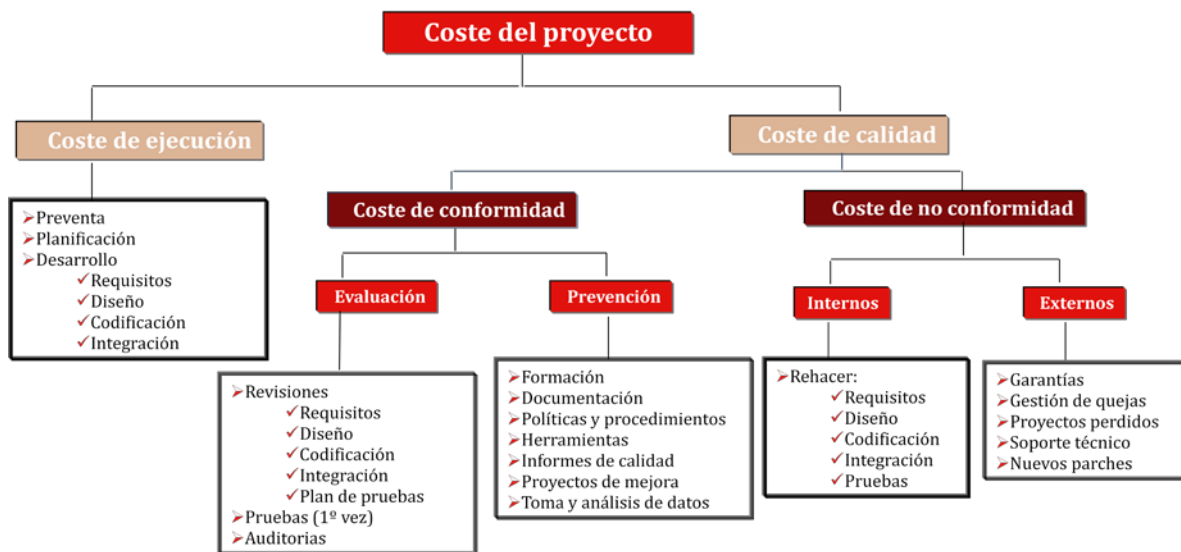
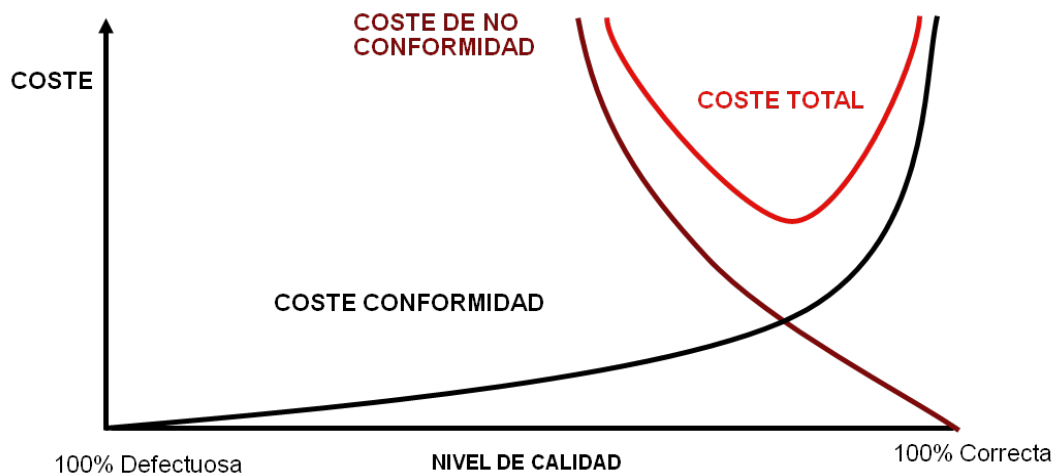


Figura 6. Coste de un proyecto

Sería obvio decir que la calidad no es gratuita, siendo los costes de conformidad los costes de las actividades de calidad. Sin embargo también existen los llamados costes de no conformidad que surgen ante la ausencia de calidad.

El **coste de calidad** se define como la suma del coste de conformidad más el coste de no conformidad.

Los costes de alcanzar calidad (**costes de conformidad**) y los costes debidos a la falta de calidad (**costes de no conformidad**) tienen una relación inversa el uno respecto del otro, cuando la inversión para conseguir calidad aumenta, los costes debidos a la falta de calidad disminuyen.



*Figura 7. Relación del coste con la calidad*

No alcanzar el nivel de calidad deseado es consecuencia de los fallos, es decir de la no calidad, no conformidad o mala calidad. Los costes incurridos por los fallos cometidos, en general, disminuyen a medida que aumenta el tiempo transcurrido desde que la empresa ha implantado un programa de calidad, y también en función del importe invertido, así como de la efectividad y eficiencia de la inversión destinada a tareas de prevención y evaluación.

## Gestión Efectiva de la Calidad del Producto

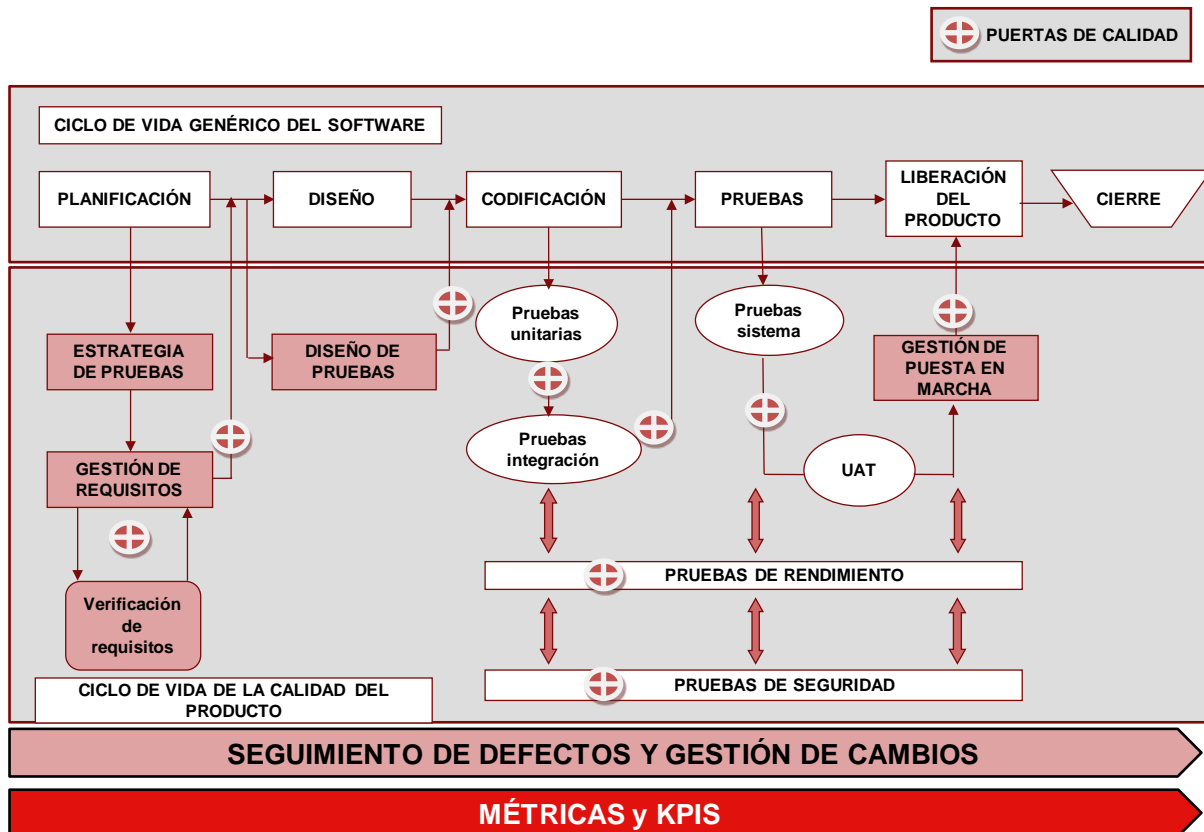


Figura 8. Ciclo de vida de la calidad del producto

Como se ve en la figura anterior, la gestión de calidad del producto es un proceso que va en paralelo con el ciclo de vida del producto:

- Durante la etapa de **planificación**:
  - Los requisitos tienden a evolucionar con el tiempo por lo que es importante llevar a cabo una gestión de los mismos. También hay que especificarlos de forma clara y precisa. (**Gestión de requisitos**)
  - También es importante establecer una buena **estrategia de pruebas**.
- La siguiente fase de desarrollo es el **diseño del producto** que trae consigo el **diseño de casos de pruebas**.

- Durante las siguientes fases de **codificación y pruebas** del producto se **ejecutan las pruebas unitarias** de sistemas, de integración, etc.
- Una vez que el software ha superado las pruebas oportunas, se **libera el producto**, **gestionando antes su puesta en marcha** para verificar que su calidad es la adecuada.
- La **gestión de defectos** es un proceso que se ejecuta durante todas las fases del ciclo de vida del producto. La detección y corrección temprana de los defectos suele suponer un importante ahorro de dinero, lo cual es uno de los objetivos principales de los proyectos.
- Se **recogerán métricas y KPIs** durante todas las fases, para tener un control acerca de la evolución del proyecto y obtener datos significativos que puedan utilizarse en otros proyectos o fases del mismo.

## Gestión de requisitos

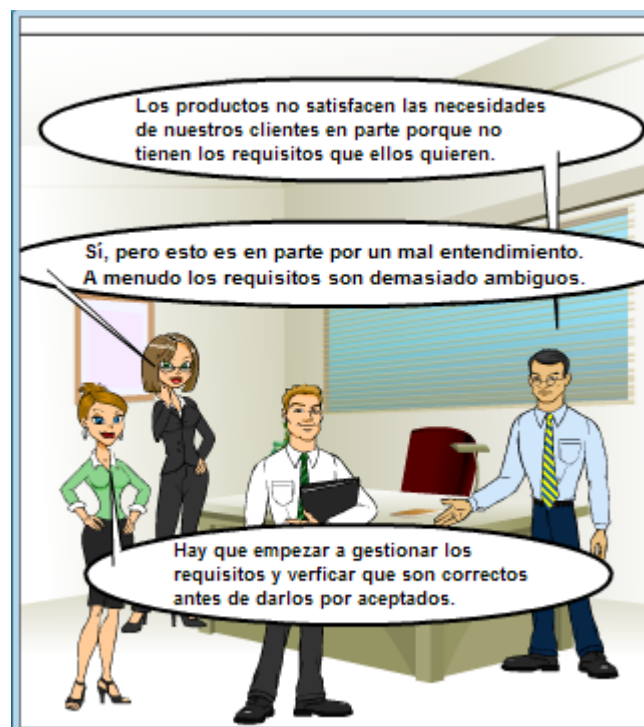


Figura 9. Importancia de la gestión de requisitos

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**.

Excepto en unos cuantos casos fortuitos, ningún producto tendrá éxito sin un claro **entendimiento previo de sus requisitos**. Si no se entiende cómo se quiere que el producto sea y funcione, no se podrá construir correctamente

Los requisitos del producto deben ser entendidos por todas las partes (cliente y desarrollador) antes de que comience su construcción o el proyecto fracasará. Sólo cuando se conocen los requisitos correctos se podrá diseñar y construir un producto que permita a los usuarios hacer su trabajo de forma que **satisfaga las necesidades del negocio**.

Por desgracia, los requisitos no son siempre entendidos correctamente. Existen muchos estudios que lo demuestran, por ejemplo, de acuerdo con el Instituto Nacional de Estándares y Tecnología de Estados Unidos, los requisitos incompletos, imprecisos y conflictivos normalmente causan un 70% de los defectos de una aplicación.

Aunque los desarrolladores tienen la oportunidad de subsanar la mayoría de los errores en la definición de requisitos, muchas veces se precipitan o hacen suposiciones que, como consecuencia, dan lugar a un **producto erróneo** por falta de tiempo o de presupuesto entre otros motivos.

La **verificación de los requisitos** (mostrada también en el gráfico de gestión de calidad de un producto software) es un punto clave dentro del proceso de gestión 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.

## Diseño de pruebas

A continuación se describen los pasos a llevar a cabo durante esta fase del ciclo de vida de calidad de un producto software:

1. Antes de ejecutar las pruebas, es necesario saber: qué se va a probar, cuáles van a ser las entradas y los resultados producidos y cómo se van a preparar y ejecutar las pruebas.

El **análisis de pruebas** es el proceso de examinar todo aquello que pueda proporcionar información para las pruebas. Estas serán las bases para las pruebas, que pueden ser un requisito del sistema, una especificación técnica, el código en sí o un proceso de negocio.

Algunas veces las pruebas se pueden basar en el conocimiento del sistema de un usuario experto y podrán no ser documentadas. Las bases de pruebas incluyen cualquier cosa en la que se basen las pruebas.

2. El siguiente paso sería definir **condiciones de pruebas**. Una condición de prueba es un elemento o evento de un componente o sistema que puede ser verificado mediante uno o más casos de prueba, es decir, algo que podría probarse. Para definir las, habrá que acudir a las bases de pruebas. Las condiciones de pruebas deberían poderse enlazar/trazar con su origen en las bases de pruebas.
3. Crear **casos de pruebas**: Las **condiciones de pruebas** pueden ser imprecisas y cubrir un gran rango de posibilidades. Sin embargo al crear **casos de pruebas** es necesaria una especificación detallada de los mismos. No es recomendable usar descripciones generales.

El siguiente gráfico muestra cómo llevar a cabo un caso de prueba:

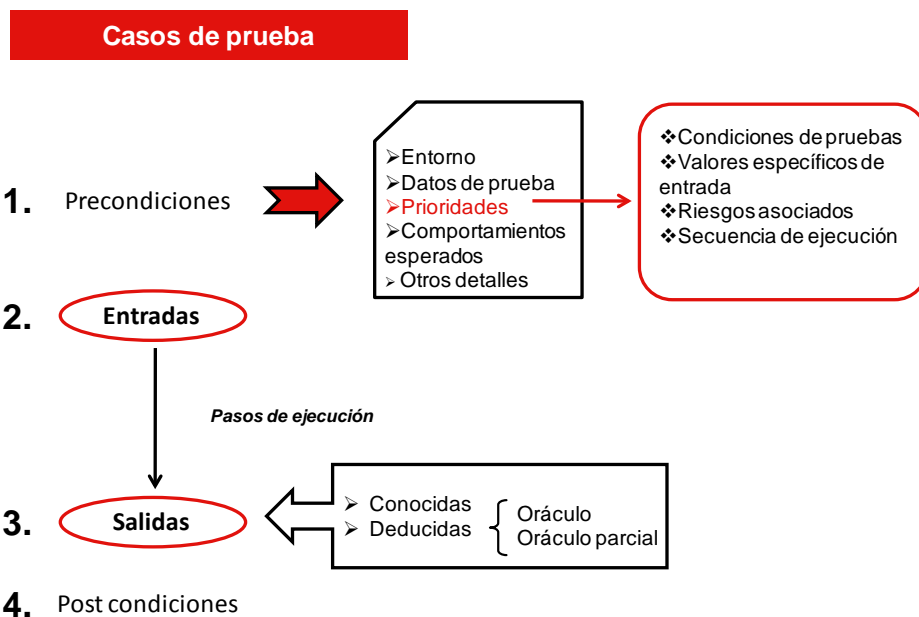


Figura 10. Desarrollo de un caso de pruebas

4. Una vez que tenemos los casos de pruebas, el siguiente paso es **agrupar los casos de pruebas** en el orden de su ejecución y especificar los pasos secuenciales que se necesitan llevar a cabo para ejecutar la prueba. Por ejemplo, un conjunto de pruebas simples que cubra el alcance de un sistema puede formar un grupo de pruebas de regresión.

## Gestión de defectos



Figura 11. Detección de defectos

Es común realizar informes de defectos encontrados en el código o en el propio sistema. Sin embargo también se ven casos donde los defectos se refieren a los requisitos o a la especificación del diseño, a la documentación e incluso a las pruebas. Es más, **la mayoría de los errores son introducidos en las primeras fases** del ciclo de vida del proyecto como muestra la siguiente figura:

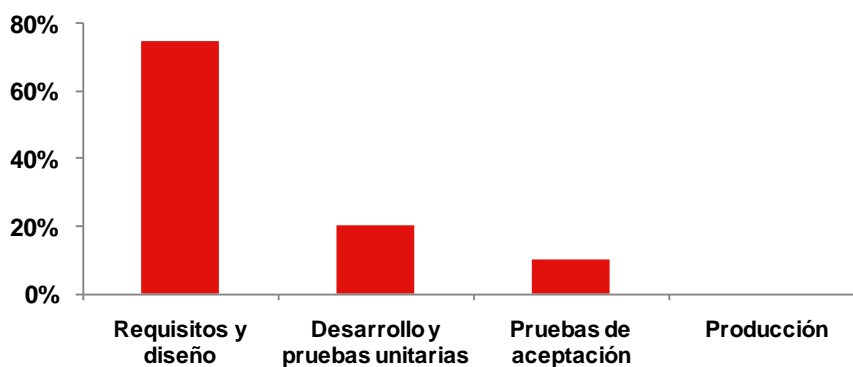


Figura 12. Introducción de defectos en las etapas de desarrollo

Encontrar defectos desde las primeras etapas tempranas del proceso de desarrollo permite su pronta corrección y que el resto de fases no se desvíen de la previsión inicial, de modo que se satisfaga finalmente tanto a las personas de negocio como a los clientes.



Figura 13. Coste detección tardía de defectos

Sin embargo, la mayoría de ellos se detectan durante las pruebas de aceptación y producción como refleja la siguiente figura:

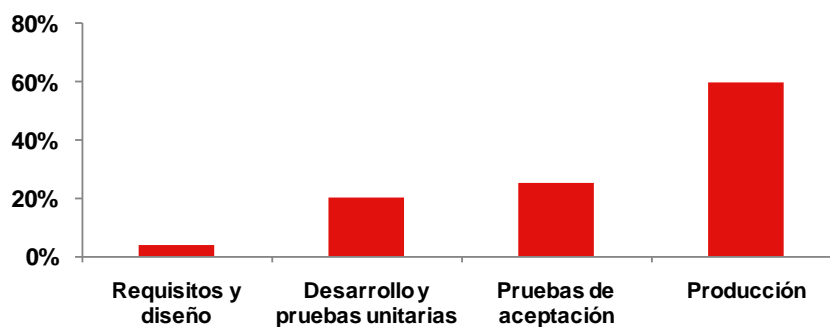


Figura 14. Detección de defectos en las distintas fases de desarrollo

Esto es debido a que no se dedica el esfuerzo necesario en gestión de calidad en las etapas iniciales del desarrollo de software. Lo más habitual es dedicar el tiempo a buscar posibles

defectos en la etapa de ejecución pruebas, cuando esta etapa es ya bastante tardía en el desarrollo, y los posibles defectos introducidos en fases anteriores ya se han propagado.

Esta tendencia debe cambiarse y empezar la gestión de defectos desde la etapa de definición de requisitos.

En fabricación, por ejemplo, un ejemplo de acciones a tener en cuenta en lo relativo a la gestión de defectos sería:

- Cualquier persona en la planta de fábrica puede parar el proceso entero de fabricación si se dan cuenta de que algo no va bien.
- Cualquier persona en la línea de montaje es validador de calidad.
- Tener claro que cuando un problema es detectado pronto, es mucho más barato arreglarlo en ese momento en vez de desperdiciar un grupo de coches defectuosos, por ejemplo, y más tarde tener que rediseñarlos o incluso retirarlos del mercado.

## Gestión de la puesta en marcha

La gestión de la puesta en producción tiene como **objetivos**: entregar, distribuir y hacer un seguimiento de los cambios que se presenten en la puesta en producción. Es conveniente que este proceso esté integrado con los procesos de:

- gestión de la configuración
- gestión de cambios

Antes de la puesta en producción hay que tener en cuenta una serie de **acciones y criterios** a seguir.

- Hay que establecer políticas de congelación de código
- Todos los requisitos han de estar cerrados.
- Se deben realizar evaluaciones de métricas de riesgos.

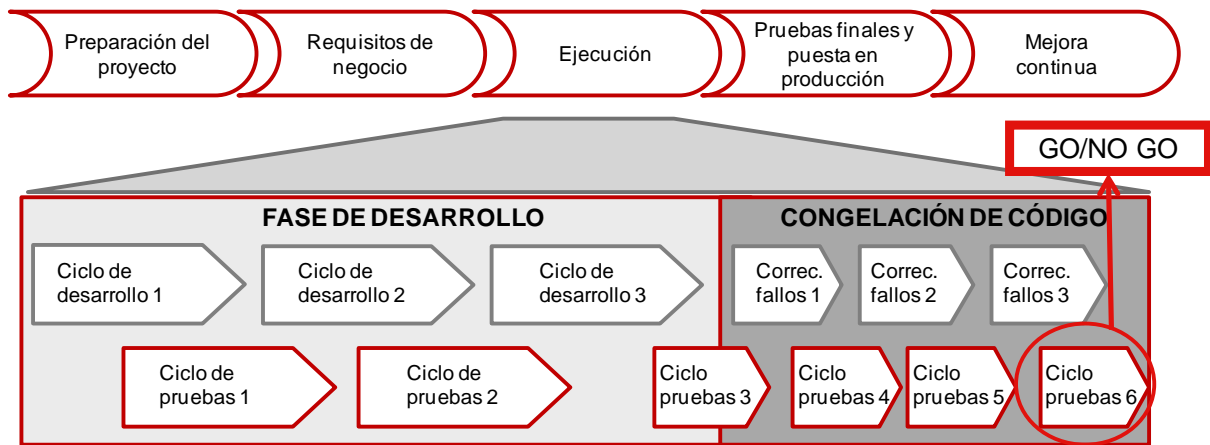


Figura 15. Criterio de salida a producción

Además, para llevar a cabo la puesta en producción hay que cumplir con una serie de **criterios**.

El nivel de calidad (basado en el último ciclo de pruebas) debe cumplir con los **criterios acordados**. Hay que asegurar la calidad de las aplicaciones software antes de pasarlas a producción, ya que ese es el **momento** en el que es **más costoso encontrar un defecto**.

Deben existir **estrategias para mitigar los riesgos post-puesta en producción**. Para cumplir con este criterio se debe contar con un plan que ayude a abordar los riesgos que puedan surgir en producción.

Se deben identificar los distintos tipos de riesgos que pueden surgir: mejoras, solicitudes de cambio, incidentes... Se hace necesaria una integración entre la gestión del cambio, los requisitos que genera y el aseguramiento de la calidad.

## Características de calidad según ISO 9126

La ISO 9126 define un modelo de calidad para calidad tanto interna como externa. Categoriza los atributos de calidad del software en 6 características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Estas características están subdivididas en sub características que pueden ser medidas por métricas internas o externas.

La **calidad interna** es la totalidad de características del producto software desde una visión interna (componentes/productos intermedios). La calidad interna es medida y evaluada contra los requisitos de calidad internos. Los detalles de calidad del producto software pueden ser mejorados durante la implementación, revisión y pruebas del código, pero la naturaleza fundamental de la calidad del producto software representado por calidad interna permanece sin cambios a menos que se lleven a cabo actividades de re diseño.

La **calidad externa** es la totalidad de características del producto desde una visión externa (producto final). Es la calidad cuando el software es ejecutado, que es típicamente medido y evaluado durante las pruebas en un entorno simulado con datos simulados utilizando métricas externas. Durante las pruebas, la mayoría de los fallos deberían ser descubiertos y eliminados. Sin embargo, algunos fallos pueden permanecer después de las pruebas. Como es difícil corregir la arquitectura del software u otros aspectos fundamentales del diseño del software, el diseño fundamental normalmente permanece invariable durante las pruebas.

Funcionalidad	Fiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Adecuación	Madurez	Capacidad de entender	Tiempo de respuesta	Capacidad analizar	Adaptabilidad
Exactitud	Tolerancia a fallos	Capacidad de aprender	Utilización de recursos	Capacidad de cambio	Instalabilidad
Seguridad de acceso	Capacidad de recuperación	Capacidad para operar	Cumplimiento de eficiencia	Estabilidad	Coexistencia
Cumplimiento funcional	Cumplimiento de fiabilidad	Capacidad de atracción		Capacidad para pruebas	Capacidad de reemplazo
		Cumplimiento usabilidad		Cumplimiento mantenibilidad	Cumplimiento portabilidad

Figura 16. Características de calidad ISO 9126

### Funcionalidad

Capacidad del producto software de proporcionar funciones que cumplan las necesidades cuando el software es utilizado bajo condiciones específicas.

### **Fiabilidad**

Capacidad del producto software para mantenerse a un nivel específico de rendimiento cuando es utilizado bajo condiciones específicas.

### **Usabilidad**

Capacidad del producto de ser entendido, utilizado y ser atractivo al usuario cuando es utilizado bajo condiciones específicas.

### **Eficiencia**

Capacidad del producto para proporcionar un apropiado rendimiento, relativo al conjunto de recursos utilizados bajo condiciones establecidas.

### **Mantenibilidad**

Capacidad del producto de ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en requisitos o en especificaciones funcionales.

### **Portabilidad**

Capacidad del producto software de ser transferido desde un entorno a otro.



## Escenario de clausura

El director de COMPASS S.A se vuelve a reunir con los jefes de proyecto para comunicarles los resultados obtenidos tras volver a analizar la situación de la empresa 9 meses después.



Figura 17. Escenario de clausura I



*Figura 18. Escenario de clausura II*

## Glosario

---

- **Metodología:** estrategias de gestión, desarrollo y mantenimiento del software orientadas a la calidad
- **Modelo:** conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión, desarrollo y mantenimiento de productos software
- **Norma:** documento de aplicación voluntaria que contiene especificaciones técnicas basadas en los resultados de la experiencia y del desarrollo tecnológico. Debe aprobarse por un organismo de normalización reconocido.
- **Gestión de la calidad:** actividades coordinadas para dirigir y controlar una organización en lo relativo a la calidad.
- **Certificación:** acción llevada a cabo por una entidad reconocida como confiable e independiente de las partes interesadas, mediante la que se manifiesta la conformidad de una empresa, producto, servicio o persona con los requisitos definidos en normas o especificaciones técnicas