

Este documento ha sido generado para facilitar la impresión de los contenidos.
Los enlaces a otras páginas no serán funcionales.

La empresa **SI Andalucía**, desde su nacimiento, ha tenido que competir en el mercado con otras empresas más grandes, más antiguas, y más conocidas que ellos. Desde un principio, **José, María y Jesús** se plantearon que tenían que diferenciarse de las demás empresas, ofreciendo algo más que no ofrecieran los demás. Rápidamente se dieron cuenta de que su diferenciación tenía que ser ofrecer **un servicio de mayor calidad** que el resto de sus competidores.



Al mismo tiempo, se dieron cuenta de que algunas empresas de su zona, dedicadas a otros sectores de actividad que no eran la informática, habían decidido antes que ellos que la **calidad** era lo que marcaba las diferencias. Esas empresas eran potenciales clientes suyos, y casi todos estaban certificados por algún organismo externo que acreditaba que se trabajaba con calidad.

Así, por ejemplo, son varios los pequeños comercios que se han certificado en calidad por **AENOR** en la norma **ISO 9001:2000**. Tienen un logotipo de calidad en la entrada de sus comercios, que los acredita, y usan el logotipo de la certificación en calidad en todos sus documentos, porque lo consideran un elemento importante, y una garantía para sus clientes, que lo valoran positivamente.

José ha pensado que si estos pequeños comercios se han gastado un dinero y han perdido un tiempo en la implantación de sus sistemas de **Gestión de calidad**, seguramente valorarán muy positivamente el hecho de que una empresa de informática que vaya a trabajar con ellos esté certificada también en calidad.



María tenía muchas dudas, porque no veía claramente qué se podía establecer como parámetro de calidad de una aplicación software, que es algo con unas marcadas diferencias frente a otros servicios, que tienen un proceso de producción más complejo. ¿Qué se puede medir como calidad en el software? **José** le comenta que existen técnicas de gestión y certificación de la calidad pensadas exclusivamente para conseguir y para medir la calidad del software.



A partir de ese momento, se informaron a conciencia, y tras un corto periodo implantaron un sistema de gestión de calidad del software en su empresa, obtuvieron la

*certificación de la entidad externa, **AENOR**, que les ha ayudado enormemente a elaborar mejores programas, más fáciles de mantener, más robustos y mejor documentados. Y el efecto no se ha hecho esperar: desde que han empezado a trabajar con una política de calidad del software, son muchos los nuevos clientes que les han elegido a ellos en vez de a las empresas de la competencia, y el grado de **fidelidad** de estos clientes es, además, bastante alto. Su trabajo se ha hecho más fácil, producen software con más calidad, y han conseguido mejorar su imagen como empresa y aumentar la satisfacción de sus clientes, obteniendo una clara ventaja frente a sus competidores.*

Calidad del software

¿Tienes una marca favorita de ropa? ¿Qué coche te gustaría comprar? ¿Te has decepcionado alguna vez con algún producto que no ha cumplido tus expectativas?

Cuando compramos algo o contratamos un servicio esperamos que cumpla con nuestros gustos y necesidades, y si no lo hace, la próxima vez cambiaremos de empresa de servicios o de producto, buscando otra que responda a lo que necesitamos. De igual forma, las empresas cuando ofrecen un servicio o venden un producto saben que deben cumplir las **expectativas** del cliente, porque si no, lo perderán para próximas adquisiciones.



En un mercado tan competitivo como el actual con tantas empresas ofreciendo productos y servicios iguales o parecidos, no basta con producir un producto y distribuirlo, además hay que hacerlo con garantías de que va a satisfacer las necesidades y expectativas de los consumidores. Esto implica realizar varios **procesos**, estudios de mercado, elaboración de especificaciones, diseño y producción del producto, publicidad, distribución, etc. Pero todo esto no es suficiente:

Para que tengamos éxito, hay que realizar todos estos procesos y hay que hacerlos con **calidad**.

Pero, ¿qué es la **calidad**?

Cuando hablamos de **calidad** podemos fijarnos en diferentes aspectos:

- la excelencia (jamón de pata negra),
- la calidad de los materiales (oro, diamantes),
- el origen de la fabricación (champagne francés),
- el método de fabricación (alfombra hecha a mano),
- el prestigio de la marca (coche Mercedes),
- la categoría del punto de venta (restaurante de lujo), etc.



Pero quizás la comida ofrecida en el restaurante de lujo sea peor que otra ofrecida en un restaurante familiar, sin embargo podemos pensar que el restaurante de lujo nos dio más "calidad". Puede haber vinos espumosos de otras regiones de igual o más calidad que el francés. La relación calidad/precio puede ser mucho peor en un vehículo de marca de prestigio que en otro de marca generalista. En este sentido podríamos definir **la calidad como las propiedades de un producto o servicio que permiten apreciarlo como mejor, igual o peor que el resto de productos o servicios del mismo tipo.**

Nuestro concepto de **calidad** sobre un mismo producto puede evolucionar a lo largo del tiempo, las prendas de abrigo antes eran apreciadas por su peso porque indicaba más protección frente al frío, y ahora se prefieren prendas ligeras. Además, las **personas** pueden apreciar diferentes calidades, los niños piensan que los mejores caramelos son los más dulces, y los mayores podemos pensar que los mejores son los caramelos sin azúcar. También puede ser distinto en los diferentes **mercados**, hay comidas que se tienen por manjares en ciertos países y en otros se desprecian. Por tanto la calidad no es un concepto absoluto sino multidimensional que puede depender de su funcionalidad ofrecida, su oportunidad en el mercado y su coste económico. Por otro lado, la calidad cuando está en el producto a veces no se nota ni se aprecia, pero cuando está ausente se hace notar demasiado, por ejemplo cuando el coche se rompe.



Debemos por tanto proporcionar una definición de calidad que recoja todas estas peculiaridades. Para ello recogemos la propuesta por la Asociación Española de Normalización y Certificación ([AENOR](http://www.aenor.es)):

La calidad es el "**grado en el que un conjunto de características inherentes cumple con los requisitos**".

En esta definición podemos ver que las características se refieren a cualquier tipo de rasgo diferenciador, luego es válido para cualquier producto o servicio. También, podemos observar que hablamos de requisitos para referirnos a las necesidades o expectativas establecidas por la empresa suministradora, los clientes o el propio mercado. Por tanto, la **definición** cumple con todo lo que hemos indicado antes, ya que está abierta a cualquier característica del producto y a cualquier expectativa del cliente y centra la calidad en la satisfacción de dichas expectativas y necesidades. No debemos olvidar que los requisitos pueden evolucionar en el tiempo, por lo que habrá que tener prevista su revisión.

Para saber más

Hemos hablado de la Asociación Española de Normalización y Certificación (AENOR) que es la agencia española de referencia sobre la gestión de la calidad, si quieres conocerla visita su página web:

AENOR

<http://www.aenor.es> [versión en cache]

Tipos de calidad

Hemos introducido el concepto de **calidad** pero la calidad será distinta dependiendo del prisma desde el que se mire. Una cosa es la calidad que necesita o quiere el cliente y otra la que la empresa estima que debe aplicar. Es más, puede ser diferente la calidad que pretende obtener la empresa de la que luego es capaz de producir. Por tanto, vemos que hay distintos orígenes de la calidad, ¿podríamos explicitarlos? Para ello nos vamos a fijar en el diagrama siguiente:



- **La calidad realizada** es la que se es capaz de proporcionar por las personas implicadas en el proceso. Se puede potenciar mejorando la formación personal y técnica de los desarrolladores e ingenieros.
- **La calidad programada** es la que pretendíamos obtener y que se ha descrito en el documento de especificaciones. Se potencia mejorando la formación y el desempeño en relación con la metodología de desarrollo.
- **La calidad necesaria** es la que el cliente espera y exige. Se potencia realizando una buena [especificación](#) que recoja las necesidades del cliente.

En nuestro proceso de desarrollo debemos intentar que los tres círculos coincidan entre sí, porque la **intersección** de los mismos será la calidad adecuada. Todo lo que quede fuera será motivo de derroche, gasto superfluo o insatisfacción para el cliente. Para poder conseguirlo, es fundamental realizar una buena especificación, porque sería inútil desarrollar unas pruebas y verificaciones muy completas sobre un software diseñado con unas especificaciones que no recogen bien las necesidades del cliente. Por ejemplo, un programa puede estar perfectamente realizado técnicamente, pero si no hace lo que quiere y necesita el cliente, poca utilidad tiene, y en consecuencia poca calidad.

Autoevaluación

¿Cuál es el nivel de calidad que se propone obtener el fabricante en un desarrollo?

- a) Calidad percibida.
- b) Calidad realizada.
- c) Calidad programada.
- d) Calidad necesaria.

Comprobar

Calidad del software

Evolución de la calidad en el mundo empresarial

Hemos hablado del concepto general de calidad, pero

¿qué implica este concepto en la gestión y organización de las empresas?

Para responder a esta pregunta debemos conocer la evolución de la gestión de la calidad a lo largo del tiempo. Vamos a distinguir **cuatro** grandes etapas en las que se van introduciendo nuevas características que complementan la etapa anterior:

- **Sin mecanismos explícitos de control de calidad:** no se establecen mecanismos de control de la calidad ni de mejora del producto, se confía en el diseño, la producción y el mercado.
- **Control de la calidad:** en esta primera etapa se busca crear productos que satisfagan al cliente, y para ello se establecen inspecciones a los productos terminados, con la intención de detectar defectos para retirar productos defectuosos e introducir mejoras en próximas producciones. La búsqueda de la calidad está centrada en cumplir con las características técnicas del producto y evitar la salida de bienes defectuosos. Podemos definir el control de calidad como el **conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio.**
- **Aseguramiento de la calidad:** en este caso nos centramos en introducir mecanismos y estrategias que garanticen un determinado nivel de calidad. Para ello, las acciones se dirigen a la prevención de los defectos, y la satisfacción del cliente. Esto requiere modificar sistemas y procedimientos en la organización para garantizar la prevención de los fallos y ser competitivos, por ello surgen los primeros modelos de calidad. Podemos definir el aseguramiento de la calidad como el **conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad.**



- **Calidad total:** La Calidad Total es el estadio más evolucionado dentro de las sucesivas etapas. Se trata de un sistema de gestión empresarial íntimamente relacionado con el concepto de **Mejora Continua** y que incluye las dos etapas anteriores. Requiere la puesta en marcha de un **sistema de gestión de la calidad que determina y aplica la política de la calidad, los objetivos y las responsabilidades de la planificación, el control, la garantía y la mejora de la calidad.**

La calidad total se introduce con estas **finalidades**:

- Consecución de la plena **satisfacción** de las necesidades y

- expectativas del cliente, tanto interno como externo.
- Ser altamente **competitivo**, con una toma de decisiones de gestión basada en información objetiva, evitando la gestión basada en la intuición y optimizando los costes.
 - Desarrollo de un proceso de **mejora continua** en todas las actividades, procesos y personas implicadas con la empresa (incluidos clientes, proveedores, etc.) con un total compromiso y liderazgo de todo el equipo directivo que fomente el trabajo en equipo.
 - Identificación y gestión de los **procesos** clave de la organización, superando las barreras departamentales y estructurales de la empresa, para propiciar los cambios necesarios en la planificación, fijación de objetivos, coordinación, formación, etc.

La siguiente presentación te muestra cómo evoluciona el concepto de gestión de la calidad, hacia la calidad total.

Evolución histórica

Esta evolución nos ayuda a comprender la **necesidad de ofrecer una mayor calidad del producto o servicio** que se proporciona al cliente, y cómo poco a poco se ha ido involucrando a toda la organización en la consecución de este fin. La calidad no se ha convertido únicamente en un requisito esencial del producto sino que **es un factor estratégico** clave del que dependen las organizaciones, **no sólo para mantener su posición en el mercado sino incluso para asegurar su supervivencia**. El concepto de calidad total proporciona una respuesta a la necesidad de hacer coincidir los distintos tipos de calidad que vimos antes: programada, realizada y necesaria, ya que se centra en la satisfacción del cliente e implica a todo el desarrollo.

Autoevaluación

¿Cuál es un rasgo inherente a la calidad total?

- a) Ser competitivo
- b) Satisfacer al cliente
- c) Implicar la mejora continua de la productividad
- d) Todas las respuestas anteriores son correctas

Comprobar

Calidad del software

¿Y qué es el software de calidad?

Hemos introducido el concepto general de la calidad, pero lo que nos interesa a nosotros es **la calidad del software**.

¿Qué entendemos por calidad referida al software?

Si nos fijamos en las definiciones anteriores de la calidad



vemos que lo importante es establecer las características y requerimientos del producto para poder satisfacer las necesidades y expectativas del cliente. Esto enlaza con los objetivos del **desarrollo del software**, por lo que la definición es válida.

Sin embargo, **el software es diferente** a otros productos, ya que:

- el resultado obtenido es **lógico, no físico**,
- por lo que, **no se degrada con el uso** y
- **puede reproducirse sin pérdida de calidad**.
- Es **flexible**, puede modificarse fácilmente aunque con efectos **difíciles de controlar**. Aunque se está incrementando la reutilización de aplicaciones anteriores, aún hay muchos desarrollos hechos a medida.
- En muchos casos **se entrega el producto con errores**, todas las grandes aplicaciones tienen un sistema de actualizaciones con "parches" que corrigen y mejoran los programas, esto no sería admisible en otros sectores. ¿Te imaginas que fuese necesario llevar tu coche al taller trimestralmente para que le corrigiesen fallos de seguridad?
- Por otro lado, en otras ingenierías el proceso de definición de la solución o el producto se realiza previamente a la construcción y está claramente separada, una vez diseñado el producto entra en las plantas de producción y se repite continuamente. Sin embargo, **en el software, la definición, que incluye el análisis y el diseño, se realiza de manera ligada e inseparable a la construcción**.



En el desarrollo del software se producen importantes diferencias que hacen que el **seguimiento** de un proceso sistematizado de desarrollo sea fundamental, y en concreto que la especificación de requisitos sea ajustada a las necesidades y expectativas del cliente pues todo el proceso depende de ella. Esto implica que debemos tener en cuenta estas **particularidades del software** cuando hablamos de calidad.

La obtención de software de calidad implica la utilización de **metodologías y procedimientos estándares** para el desarrollo del software que serán proporcionados por la **Ingeniería del Software**, que nos ofrece un conjunto de metodologías, actividades, métodos, prácticas y transformaciones utilizados para desarrollar software y los productos asociados:

- **planes del proyecto**,
- **documentos de análisis/diseño**,
- **codificación**,
- **casos de prueba**,
- **manuales de usuario**.

La utilización de **las metodologías de desarrollo** que



hemos ido mostrando a lo largo de este módulo nos ayudará a la consecución de software de calidad. Pero además, es necesaria la implantación de un modelo de calidad que nos asegure:

- que esas metodologías se están llevando a cabo de forma correcta,
- que se están obteniendo los resultados esperados, y
- que se establecen mecanismos para la mejora continua.

Con todo esto, ya podemos definir el **concepto de calidad del software**:

Es el grado con el que un sistema, componente o proceso cumple los requisitos especificados y las necesidades o expectativas del cliente.

Esta definición implica que el software debe producirse cumpliendo con:

- los **requisitos funcionales** (por ejemplo, el sistema debe emitir las nóminas) y no funcionales (por ejemplo, el sistema debe responder en un tiempo razonable) fijados explícitamente durante la fase de análisis,
- los **requisitos implícitos** que espera el cliente y que no quedaron explicitados en el análisis (por ejemplo, la facilidad del mantenimiento).
- los **estándares y normas** de desarrollo proporcionados por la Ingeniería del software que aseguren la calidad técnica del producto (por ejemplo, para garantizar la ausencia de defectos).

Autoevaluación

¿Cómo calificarías el hecho de que un cliente necesita que su programa gestione el stock de un almacén?

- a) Es un requisito funcional que debe aparecer reflejado en el documento de análisis de requisitos.
- b) Es un requisito no funcional que debe aparecer reflejado en el documento de análisis de requisitos.
- c) Es un requisito que puede quedar implícito y no es necesario que aparezca en el análisis de requisitos.
- d) Es una exigencia para la aplicación de estándares y normas en la metodología de desarrollo.

Comprobar

Para saber más

Puedes conocer todas las novedades sobre la calidad del software en esta página web dedicada de manera exclusiva a esta temática:

Calidad del software

<http://www.calidaddelsoftware.com/index.php> [versión en cache]

Aplicación de la calidad en ingeniería del software

Es importante que comprendas que los **modelos generales** que hemos estado explicando se aplican a cualquier desarrollo empresarial y no sólo a la producción de software. La propia empresa de desarrollo de software puede aplicar la calidad total a su gestión y administración, y no sólo al desarrollo del software, ésta es la idea fundamental de la calidad total. Es por todo ello, que la aplicación de la calidad en Ingeniería del Software puede realizarse en distintos niveles tal y como vemos a continuación:



- **Nivel de la realización del personal:** Se trata de aplicar los conceptos de calidad al trabajo de cada uno de los **trabajadores** de la empresa implicados en el desarrollo del producto. Está relacionado con la calidad realizada, es decir, la calidad máxima que podemos obtener con los recursos humanos y materiales existentes. Para mejorar este nivel hay que aplicar una estrategia de mejora de las capacidades técnicas y profesionales individuales de los desarrolladores e ingenieros.
- **Nivel del proyecto y su organización:** el enfoque del nivel de realización anterior no es suficiente, puesto que la calidad total exige la implicación de todo el personal y el trabajo en equipo. Por ello, se introduce este nivel de calidad que implica a todo el desarrollo del proyecto. En este nivel, las actividades que se llevan a cabo son las siguientes:



- **Aseguramiento de la calidad (Software Quality Assurance):** Pretende dar confianza en que el producto tendrá calidad. Se puede definir como el conjunto de acciones planificadas y sistemáticas, implementadas, que son necesarias para proporcionar la confianza adecuada de que un producto satisfará los requisitos dados sobre la calidad. El aseguramiento de calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla e incluye tres actividades fundamentales:

- La **gestión de la configuración** del software, ligado a la aplicación de una metodología de desarrollo.
 - **Verificación y validación** del software a lo largo del ciclo de vida.
 - Aplicación de **métricas** de software para el control del proyecto.

- **Control de calidad (Software Quality Control):** Se trata de aplicar los procesos de evaluación a los productos obtenidos en el desarrollo para conocer su nivel de calidad. Está muy ligada al anterior, pero mientras que las acciones del aseguramiento buscan crear un marco de trabajo de procedimientos y estándares de organización que conducen a la calidad del software, el control se asegura de que dicho marco se lleva a cabo y se desarrolla adecuadamente. Las actividades concretas serían:

- **Verificación y validación (Software Verification and Validation):** este proceso se explicó en la unidad anterior e incluye las pruebas y los procesos de revisión e inspección realizados para comprobar que se cumplen los requisitos y necesidades del cliente fijados en el análisis.



- **Medición:** se trata de evaluar que el software desarrollado cumple con ciertas características de calidad, para ello se han desarrollado distintos modelos de calidad y técnicas de medición.

Podemos completar el control de calidad interno con auditorías externas, revisiones independientes y peritajes.

- **Nivel de empresa:** en este nivel se organizan las actividades de calidad para toda la empresa con el objetivo de alcanzar la calidad total. Esto hace necesario la implantación de un sistema de **Gestión de la calidad software (Software Quality Management)** que determina y aplica la política de calidad marcando los objetivos y directrices generales de la calidad de la empresa. Las **actividades** relacionadas con la calidad se extienden a toda la organización para lo que se debe aplicar un modelo de calidad general aplicable a cualquier entorno productivo (como las normas ISO 9001), o bien un modelo específico para empresas de desarrollo de software (como CMM o SPICE). Para no comprometer la calidad, es importante, que en función del tamaño de la empresa, exista un equipo de administración de la calidad, que debe ser independiente de los equipos de desarrollo del producto. Para cada proyecto deben adaptarse los planes generales de calidad a las necesidades y requisitos concretos del proyecto, buscando la mejora de los procesos de desarrollo y mantenimiento del software.

Quality is
Our Business.

Para ayudarnos a **implantar modelos de calidad** en nuestra empresa podemos contratar empresas **consultoras** especializadas en gestión de la calidad que nos pueden **asesorar** y proporcionar expertos que faciliten la puesta en marcha de los mecanismos y las estrategias necesarias. En cualquier caso, para que tenga éxito un proyecto de calidad total como el descrito, es necesario crear una **cultura de la calidad** en la empresa donde cada persona responsable del desarrollo del producto es motivada para que logre un alto nivel de calidad del producto.

Para saber más

Hemos comentado la existencia de la norma ISO 9001, esta norma ha sido establecida por la organización internacional de estándares ISO. Si quieres saber más sobre ella puedes visitar su página web oficial:

ISO

<http://www.iso.org> [versión en cache]

Calidad del software

Caso.

Cuando en **SI Andalucía** se decidieron a implantar un sistema de gestión de calidad, se encontraron con que tenían distintas posibilidades, cada una de las cuales presentaba sus propias características.



Se encontraron con que hay sistemas de gestión de calidad orientados a cualquier tipo de empresa, como la norma ISO 9001:2000, y otros orientados específicamente para calidad software, como SPICE o CMMI.



*En principio, **Jesús** y **María** defendían que puesto que su empresa se dedicaba en gran medida al desarrollo software, debían elegir uno de esos sistemas de calidad específicos para el software.*

***José**, sin embargo, argumentaba que un sistema general como la norma ISO 9001:2000, era perfectamente válido también para el caso de empresas dedicadas al desarrollo del software. Además, entre otras cosas por ser válido para cualquier empresa o proceso productivo, era quizás el sistema con mayor reconocimiento a nivel internacional, y de hecho, el que tenían implantado la mayoría de las empresas que eran sus potenciales clientes, con lo que no debían desaprovechar el componente publicitario extra que les podía ofrecer este sistema frente a otros, más centrados en el desarrollo software, pero menos conocidos por sus clientes.*

*Tras deliberar un tiempo, optaron por adoptar la norma ISO 9001:2000. El sentido práctico de **José** los convenció a todos.*

¿Pero todo esto sirve para algo?

Hemos descrito la calidad y sus implicaciones en la empresa en general y en el desarrollo del software en particular. A esta altura de la unidad seguro que ya te has imaginado que poner todos estos modelos en marcha exige un **esfuerzo económico y de tiempo**, por lo que surge la pregunta obvia, ¿merece la pena?



Cuando en una empresa se propone implantar un modelo de calidad, e incluso cuando ya está en marcha es posible escuchar comentarios del tipo: "No merece la pena, una pérdida de tiempo", "demasiado papeleo pesado, muchas reuniones para establecer un proceso", "sólo sirve para conseguir un sello que nos proporcione publicidad", etc.

Para que la **implantación de un modelo de calidad** tenga éxito es condición indispensable **implicar a todo el personal y motivarlo** para que comprenda su utilidad y necesidad. Aunque la resistencia al cambio es congénita al ser humano, cuando se oyen comentarios como los anteriores, puede ser debido a que no se hizo ese trabajo previo de motivación y justificación, o debido a que el proceso de implantación y desarrollo del modelo de calidad no se hizo adecuadamente. Si todo el mundo está motivado, hay un derroche de ilusión y trabajo pero luego no se implanta el modelo correctamente, los desarrolladores e ingenieros terminarán desilusionados y abandonando el modelo. Por ello es fundamental tener el asesoramiento adecuado de expertos.



A continuación comentamos las **motivaciones, necesidades y objetivos** que deben buscarse a la hora de implantar un modelo de calidad en cualquier empresa

y que justifican su impulso:

- El modelo debe ser **eficaz y comprendido** por todos, desde las bases hasta la dirección.
- Debe dar **confianza** en que se van a satisfacer las necesidades de los clientes.
- Hay que hacer comprender a todos que la filosofía subyacente es poner más énfasis en la **prevención** que en la detección y corrección.
- Asegurarse de que todos asumen que el modelo nos da la oportunidad de conseguir un **producto mejor** que el de la competencia.
- La calidad nos permitirá **repetir el éxito** con plazos y costos conocidos.
- El modelo nos ayudará a **sistematizar** el desarrollo con procesos definidos para los requisitos, análisis, diseño, implementación, gestión, pruebas, documentación, etc.
- Una vez puesto en marcha, el modelo no se detiene nunca, siempre se buscará la **mejora continua** para lo que es necesario un seguimiento continuo de todo el proceso.
- El personal debe recibir la **formación** necesaria para conocer los modelos de gestión de la calidad y su implantación.
- Es importante entender que para implantar un modelo de calidad debemos ayudarnos de **herramientas** de soporte que automaticen los procesos burocráticos y nos ayuden en el tratamiento de la información y la toma de decisiones.
- La empresa debe proporcionar los **recursos** necesarios para llevar a cabo la implantación del sistema de calidad.



Si el **personal de la empresa y la dirección** no asumen estos objetivos y motivaciones difícilmente tendrá éxito la implantación de un modelo de calidad. Pero si se reúnen las condiciones propicias y pone en marcha el modelo, está claro que proporcionará grandes ventajas, como se ha demostrado para las empresas que lo han llevado a cabo. A lo largo de esta unidad podrás comprender estas ventajas y la necesidad de su implantación en cualquier empresa que quiera ser **competitiva** y subsistir en el mercado.

Autoevaluación

¿Qué condiciones deben reunirse para que tenga éxito la implantación de un modelo de calidad en la empresa?

- a) Todo el personal de la empresa debe estar motivado.
- b) Todo el mundo debe conocer y comprender la necesidad y ventajas de implantar un modelo de calidad.
- c) El sistema debe implantarse adecuadamente, con asesoramiento profesional y utilizando las herramientas de ayuda necesarias.
- d) Deben reunirse todas las condiciones anteriores.

Comprobar

Calidad del software

Los sistemas de gestión de la calidad en la empresa

Ya sabemos qué es la calidad y estamos motivados para impulsarla en nuestra empresa, ¿y ahora qué?

Si queremos crear un marco de trabajo basado en la calidad debemos poner en marcha un **sistema de gestión de la calidad** que lo vamos a definir como el conjunto de elementos interrelacionados de una empresa u organización por los cuales se administra de forma planificada la calidad de la misma, en la búsqueda de la satisfacción de sus clientes. Entre dichos elementos, los principales son:



- la **estructura** de la organización,
- los **procesos**,
- los **documentos** y
- los **recursos**.

El sistema de gestión de la calidad debe organizar también, la **evaluación de la calidad**, que incluye las actividades de control y supervisión de todo lo relativo al grado de desarrollo y aplicación de la calidad en la empresa. Para realizar esta evaluación se pueden usar técnicas de [benchmarking](#) que miden la calidad. Normalmente existe un equipo responsable de calidad que velará por el cumplimiento de los planes programados sobre calidad, pero la **dirección** de la empresa debe ejercer el **liderazgo** y ser la responsable de fijar la política de calidad y las decisiones relativas a iniciar, desarrollar, implantar y actualizar el sistema de calidad.

Para llevar a cabo todos estos compromisos se utilizan **modelos de calidad**, que permiten estandarizar el proceso de implantación, desarrollo, control y evaluación de resultados. Para desarrollar estos modelos se han establecido una serie de **normas de calidad**. Vamos a definir una **norma de calidad** como un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad. Ejemplos de organizaciones internacionales emisoras de normas de calidad son: [ISO](#) (**Organización Internacional de Estándares**) y [IEC](#) (**Comisión Electrotécnica Internacional**).

Hay diferentes modelos y **normas de calidad**, comentamos algunos:

- El propuesto por la **Fundación Europea para la Gestión de la Calidad (E.F.Q.M. European Foundation for Quality Management)**, con el fin de reforzar la posición de las empresas europeas en el mercado mundial impulsando en ellas la calidad como factor estratégico clave para lograr una ventaja competitiva global. La EFQM otorga todos los años el Premio Europeo a la



Calidad a las empresas que se han distinguido por su incremento de la eficacia y la eficiencia, y por la aplicación de la calidad total. La implementación de programas de calidad total y mejora continua permite alcanzar beneficios significativos, tales como un incremento en la eficiencia y eficacia, reducción de costos, mejora de los resultados de negocio y mayor satisfacción de los clientes.

- El **Marco Común de Evaluación (CAF, Common Assessment Framework)** es un modelo organizativo de la Unión Europea para que las administraciones públicas apliquen un sistema de gestión de la calidad. Permite realizar diagnósticos, y a partir de ellos, planes de acción con el objetivo de lograr administraciones públicas comprometidas con la búsqueda de la excelencia en los servicios ofrecidos.
- Una de las normas más conocidas y utilizadas a nivel internacional para gestionar la calidad, es la norma **ISO 9001:2000** que comentaremos posteriormente.

Para saber más

Hemos comentado el modelo europeo EFQM, es posible que quieras conocer más de este modelo para lo que puedes ir a la Wikipedia:

Wikipedia: Modelo de Excelencia EFQM

http://es.wikipedia.org/wiki/EFQM#Modelo_de_Excelencia_EFQM
[versión en cache]

Puedes completar la información en la web oficial de EFQM, aunque está en inglés.

Web oficial EFQM

<http://www.efqm.org/> [versión en cache]

Calidad del software

La necesaria normalización: documentación

Hemos dicho que es necesario seguir un modelo de calidad para implementar un sistema de gestión de la calidad, y estos modelos están normalizados, pero ¿qué nos ofrece esta **normalización**?

Para comprender qué significa la normalización, es importante que antes describamos los **elementos** fundamentales que componen un **sistema de calidad**:

- **Elementos físicos:** se refiere a todo lo relativo a las instalaciones, herramientas, ordenadores, etc. de que disponemos.
- **Aspectos humanos:** entre los que se incluye la formación del personal tanto en temas de calidad como otras temáticas necesarias, la creación y coordinación de equipos de trabajo, las normativas de personal, etc.
- **Documentación:** son una serie de documentos en los que se describen los aspectos fundamentales del sistema de calidad. Puede haber manuales a nivel de empresa, departamento, proyecto, etc. Fundamentalmente se



componen de:

- El **Manual de calidad**: Es el documento principal aplicable a toda la empresa para establecer e implantar un sistema de calidad. El Manual de Calidad describe la estructura del Sistema de Gestión de la Calidad y especifica la **misión y visión** de la empresa con respecto a la calidad así como la política de la calidad y los objetivos que apuntan al cumplimiento de dicha política. Puede ser un documento público.
- El **Manual de procedimientos**: es el documento utilizado por los departamentos y equipos de desarrollo, y contiene una **descripción precisa de cómo deben desarrollarse las actividades de cada empresa**. Ha de ser un documento interno que incluirá la especificación y documentación acerca de cómo realizar los procesos y productos a lo largo del ciclo de vida del desarrollo del software: especificación de requisitos, planes de pruebas, diseño del código, etc.
- Los manuales anteriores pueden complementarse con las **instrucciones técnicas** de trabajo, los registros (resultados obtenidos en las actividades) y la normativa externa que completan los procedimientos o detallan procesos específicos. Estos documentos son utilizados por los desarrolladores. Si nuestra empresa participa en un proyecto de sostenibilidad y ecología es posible que debamos recoger normativa al respecto.



Ya conocemos los elementos, pero ¿cómo los organizamos?, ¿cómo podemos hacer comparaciones entre organizaciones si cada una usa su propio modelo?. Si a una empresa le funciona bien su modelo ¿no podemos traspasarlo a la nuestra?

La respuesta a todas estas preguntas sólo puede ser una: la **estandarización de los modelos y metodologías**.

Podemos definir la **normalización** como el proceso de elaboración, aprobación, aplicación y mejora de normas que se aplican a distintas actividades científicas, industriales o económicas con estos fines:



- Reducir y **simplificar** los modelos quedándose únicamente con los más necesarios.
- **Unificar** sistemas, documentos, estrategias, etc. para permitir la compatibilidad y los intercambios de sistemas a nivel internacional.
- Evitar errores de identificación e interpretación en los sistemas mediante una **especificación** que usa un lenguaje claro y preciso
- Garantizar la **calidad** de los productos y su seguridad de funcionamiento.

Existen diferentes **organismos** internacionales encargados de realizar estas normalizaciones y que seguro que conoces:

- ¿Has usado el puerto FireWire para conectar una videocámara digital a tu ordenador? Pues la norma que incluye la especificación de todas las características de este puerto es la IEEE 1394.
- ¿Tienes un reproductor de vídeos compatible con MPEG-4? Pues su funcionamiento viene marcado por la norma ISO/IEC 14496.
- Otro ejemplo, la norma ISO 8859 especifica las codificaciones de caracteres que incluye al código ASCII.

Estos son algunos de estos organismos normalizadores:

- **IEEE** - Institute of Electrical and Electronical Engineers.
- **ISO**- Organización Internacional para la Estandarización.
- **AENOR** - Asociación Española de Normalización y Certificación.
- **ANSI** - American National Standards Institute.



Con respecto a los sistemas de calidad, la norma ISO 9001:2000 exige un Manual de Calidad y un Manual de Procedimientos muy detallados y les dan una importancia capital, sin embargo, otros modelos de calidad no dan tanta importancia al Manual de Calidad. En los próximos apartados aclararemos estas cuestiones explicando las principales características de los modelos más extendidos:

- **ISO 9001:2000,**
- **SPICE y**
- **CMMI.**

Autoevaluación

¿Qué documentos se incluyen en el sistema de calidad a nivel de empresa?

- a) Manual de Calidad
- b) Manual de Calidad, Manual de Procedimientos
- c) Manual de Calidad, Manual de Procedimientos, Instrucciones y normativa externa
- d) Manual de Calidad, Manual de Procedimientos, instrucciones y normativa externa, Planes de calidad adaptados a los proyectos

Comprobar

Calidad del software

Norma ISO 9001:2000

Hemos nombrado varias veces en esta unidad la norma **ISO 9001:2000**, ha llegado el momento de describirla puesto que es la más extendida en España.

Con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores, se publicaron en 1994 las normas ISO 9000, que son un conjunto de normas editadas y revisadas periódicamente por la **Organización Internacional**



de Normalización (ISO) sobre la garantía de calidad de los procesos. La última versión es la ISO 9001:2000, que puede ser aplicada a cualquier tipo de organización (empresa de producción, empresa de servicios, administración pública, etc.), proporcionando una guía para la gestión, diseño e implantación de sistemas de calidad.

Esta norma propone una **visión integral y dinámica de la calidad** en el sentido de la calidad total que antes enunciábamos. El modelo se implanta para conseguir la **calidad total en el diseño, desarrollo, producción, instalación y servicio post-venta**. Crea un marco de trabajo para la mejora continua orientado a la satisfacción del cliente basándose en el ciclo [PDCA](#). El sistema de gestión de la calidad propuesto tiene que ser algo dinámico que se va enriqueciendo continuamente alimentado por la satisfacción/insatisfacción de los clientes y por sus diferentes demandas a lo largo del tiempo. Podemos ayudarnos de la **reingeniería de procesos**, que es una técnica para analizar en profundidad el funcionamiento de uno o varios procesos dentro de una empresa con el fin de rediseñarlos y mejorar.

El modelo del sistema de calidad tiene una **estructura** basada en procesos y se compone de **cuatro subsistemas** de gestión de calidad que se integran en la organización:



- **Responsabilidad de la Dirección:** se trata de establecer una política integral de calidad cumpliendo con los requisitos y necesidades de los clientes. Se deben establecer revisiones formales de la gestión de calidad realizada.
- **Gestión de los Recursos:** incluye la gestión de todos los recursos como la información, las infraestructuras y sobre todo, los recursos humanos. Debe crearse una política adecuada de asignación de tareas, con las acciones de formación y dualización necesarias.
- **Realización del Producto o Servicio:** se realiza una gestión de los procesos relacionados con el cliente, el diseño y desarrollo de los productos y servicios con un sistema de prevención de las "[no conformidades](#)" con respecto a los requisitos establecidos.
- **Medición, Análisis y Mejora:** deben realizarse evaluaciones y auditorías internas para comprobar el éxito del sistema, el grado de satisfacción del cliente, el desarrollo del sistema de calidad, etc. A partir de la información obtenida se emprenden acciones de mejora para corregir errores, prevenir problemas futuros y establecer procesos de mejora continua.



La **implantación** de este modelo supone un esfuerzo para las empresas pero a cambio obtienen grandes **ventajas**:

- Proporciona **confianza** a los clientes, reduciendo los rechazos e incidencias en la producción o prestación del servicio.
- Proporciona una **garantía** de la calidad.
- Aumento de la **productividad** y la competitividad
- Mayor **compromiso** con los requisitos del cliente.
- **Mejora** continua en todos los procesos.
- La estandarización del modelo que se ha adaptado a más de 90 países e implantado en todo tipo de organizaciones industriales y de servicios, tanto en el sector privado como en el público.
- Hace que los desarrollos sean más predecibles en cuanto a tiempo y costes.

Es posible certificar a una empresa que haya implantado el modelo ISO 9001:2000, para ello, existen entidades certificadoras que entregan un sello de **certificación** que demuestran públicamente que la empresa dispone de un sistema de calidad ISO 9001:2000. Un sistema de certificación de calidad permite una valoración independiente que debe demostrar que la organización es capaz de desarrollar productos y servicios de calidad. En España, AENOR se encarga del registro de las certificaciones.

Para la implantación del modelo, es muy conveniente que apoye a la organización una **empresa de consultoría**, y el firme compromiso de la Dirección de que quiere implantar el Sistema.

Para saber más

Si quieres saber más sobre la certificación de las empresas que implantan un modelo ISO 9001:2000 puedes consultar la web de la Entidad Nacional de Acreditación:

ENAC

<http://www.enac.es> [versión en cache]

SPICE

Hemos comentado la norma ISO 9001:2000 como un modelo para implantar un sistema de calidad en una empresa, pero como hemos indicado en su descripción, este modelo es válido para cualquier tipo de empresa. Podríamos preguntarnos si hay **modelos específicos para empresas de informática** que desarrollen productos software, y la respuesta es que sí. En concreto vamos a describir dos, SPICE en este apartado y CMMI en el siguiente.



SPICE (Software Process Improvement and Capability dEtermination) sigue la norma **ISO/IEC 15504** y aunque no es un modelo en sí, define un marco para la evaluación y mejora continua de los procesos de desarrollo del software. Incluye un conjunto de **actividades** para:

- La **evaluación** de procesos
- La **mejora** de procesos
- La determinación de la **capacidad** de cualquier proceso en función de grado de consecución de sus atributos, determina el nivel de calidad que se ha alcanzado para ese proceso.

El **modelo de referencia de SPICE** describe los **procesos** que una organización puede realizar para comprar, suministrar, desarrollar, operar, mantener y soportar el software, así como los atributos que caracterizan la capacidad de estos procesos. Estos procesos deben ser evaluados a partir de su capacidad para realizar propuestas de mejora.

Quality

Por **ejemplo**, pensemos en el proceso de realizar **pruebas de caja negra sobre los módulos**. En un momento dado se determina que la capacidad de dicho proceso se encuentra sobre los 5 niveles que tiene el modelo, en el nivel 2. Esto significa que el proceso está en fase de definición y debe ser evaluado y mejorado. El proceso estaría en revisión y evaluación, como consecuencia se comprueba que se detectan pocos errores con relación a lo previsto. Se analizan las causas y se descubre que algunas actividades no están sistematizadas como consecuencia de que el personal que aplica dicho proceso no está suficientemente cualificado. Se propone la mejora de impartir formación a este personal. En la siguiente revisión del proceso se observa una mejoría y se han introducido mecanismos para seguir mejorando. La capacidad del proceso se eleva a nivel 3, que significa que el proceso queda definido.

Los procesos están agrupados en **categorías** en función del tipo de actividad al cual se aplican:

- Procesos relacionados con el **Cliente** y los **Proveedores**.
- Procesos de **Ingeniería** que corresponden con los del ciclo de vida del software.
- Procesos de **Soporte** formados por procesos que dan soporte al resto de procesos en distintos puntos del ciclo de vida del software.
- Procesos de **Gestión** del proyecto.
- Procesos de **Organización** que establecen los objetivos de negocio de la empresa.

Autoevaluación

¿Cuál de estas características corresponde a SPICE?

- a) Es un modelo de calidad aplicable a cualquier empresa y sector productivo.
- b) Exige el desarrollo de un Manual de Calidad y un Manual de Procedimientos.
- c) Se basa en la norma ISO 9001:2000.
- d) Define un marco para la evaluación y mejora de los procesos de desarrollo del software.

Comprobar

Calidad del software

CMMI

Caso.

Con el tiempo, en **SI Andalucía** han ido aumentando su cartera de clientes, y el departamento de desarrollo software es quizás el que más ha crecido. Tras un tiempo con el **sistema de certificación en calidad ISO 9001:2000**, piensan que se han cubierto la mayoría de los objetivos iniciales de mejora de su trabajo, de los productos software que desarrollan, y de la satisfacción de los clientes. No obstante, **María** plantea que quizás es el momento de dar otro salto cualitativo, y usar alguna metodología de calidad específica para el desarrollo del software, que estuviera ampliamente extendida, y que se integrara bien con RUP y UML, que son dos tecnologías que usan en su empresa.



José le pregunta que si ha pensado en algo en concreto, aunque conociendo a **María**, sabe que seguro que así será. Efectivamente, ella le habla del modelo CMMI, que cumple todos esos condicionantes a la perfección, y que les permitirá volver a marcar diferencias con otras empresas de la competencia dedicadas al desarrollo del software. La idea a José le parece muy buena, y acuerdan informarse con más detalle sobre este nuevo modelo para valorar su adopción.



Ya hemos presentado a CMMI (**Capability Maturity Model Integration**), ahora haremos una breve descripción del mismo.

CMMI es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Fue publicado en su primera versión en 2002 y se basa en otros modelos como el Modelo de Capacidad y



Madurez en la Ingeniería de Sistemas (SW-CMM Software Engineering-Capability Maturity Model) y el SPICE antes comentado. El modelo se integra perfectamente con RUP (Rational Unified Process) y UML (Lenguaje de Modelado Unificado) por lo que está muy extendido en la Ingeniería del Software a nivel internacional.

CMMI crea un marco conceptual basado en las mejores prácticas de ingeniería del software para ayudar a la **organización** en la:

- **Caracterización de la madurez de sus procesos:** según el modelo CMMI, las organizaciones que desarrollan software presentan un atributo denominado madurez, cuya medida es proporcional a los niveles de capacidad e institucionalización de los procesos que emplean en su trabajo. Se modelo se centra en aumentar dicha madurez como indicador de la calidad.
- **Establecimiento de objetivos de mejora de procesos,** ya que la calidad de un producto o sistema es consecuencia directa de los procesos empleados en su desarrollo.
- **Establecimiento de prioridades de acción inmediata,** obliga a una revisión constante.
- **Introducción de una cultura de ingeniería del software de excelencia,** ya que es un modelo para la mejora de las organizaciones.

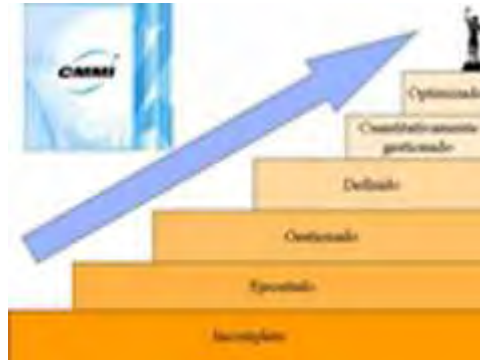


Este modelo **CMMI** es general y puede aplicarse a cuatro **áreas** distintas, pero nosotros podemos aplicarlas todas o elegir aquellas en las que queramos centrarnos para mejorar aspectos específicos, como por ejemplo la Ingeniería del Software:

- **Ingeniería de Sistema:** Cubre la construcción de un sistema con o sin software
- **Ingeniería de Software:** Cubre la construcción de soluciones software
- **Integración de productos** y procesos de desarrollo: Cubre la relación a largo plazo con el **cliente**.
- **Relación con proveedores:** Cubre los procesos relacionados con la subcontratación de partes del sistema

Hemos hablado de la **madurez de los procesos**, ¿en qué consiste?

La **madurez** de un proceso es un indicador de la capacidad para construir un software de calidad. Hay que estudiar los procesos de desarrollo de software de una organización para producir una evaluación de la madurez de la organización según una escala de niveles. Hay dos **representaciones de esta escala de niveles** que son equivalentes y podemos utilizar una u otra en función de las características y prioridades de mejora de nuestra empresa:



■ Escala continua:

Analiza la capacidad de cada proceso por separado, mostrando la representación del nivel de capacidad de cada una de las áreas de proceso del modelo. Se introducen 25 áreas de proceso agrupadas en 4 **categorías** según su finalidad: **Gestión de proyectos, Ingeniería, Gestión de procesos y Soporte a las otras categorías**. Igual que ocurría en SPICE, se proponen 6 **niveles** que miden la capacidad de cada proceso:

- Incompleto (0),
- Ejecutado (1),
- Gestionado (2),
- Definido (3),
- Cuantitativamente gestionado (4), y
- Optimizado (5).

CMMI: Niveles de la capacidad de un proceso

■ Escala escalonada:

La visión escalonada definirá a la organización dándole en su conjunto un nivel de madurez del 1 al 5, en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería. Se proponen 5 **niveles** de madurez que se van alcanzando conforme vamos implantando una serie de requisitos y buenas prácticas que se proponen:



- Inicial (1),
- Repetible (2),
- Definido (3),
- Gestionado (4), y
- Optimizado (5).

Modelo CMMI: Niveles de madurez de la empresa

Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. La consecución de estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables. Existen **herramientas** software que nos ayudan a implantar y verificar el modelo de niveles de CMMI. Una de esas herramientas es CMM-Quest.

Para saber más

Podemos profundizar en el conocimiento de CMMI visitando esta página de la Wikipedia donde se recogen múltiples enlaces a las páginas oficiales relacionadas:

Wikipedia: CMMI

<http://es.wikipedia.org/wiki/CMMI> [versión en cache]

Las herramientas CASE de apoyo a la implantación de sistemas de calidad son imprescindibles. Aquí podemos ver una de las más utilizadas, disponible en inglés y alemán:

CMM-Quest

<http://www.cmm-quest.com/english/> [versión en cache]

Calidad del software

Auditoría informática

Caso.

Si Andalucía ha apostado fuerte por la implantación de un sistema de gestión de calidad en el desarrollo del software, y ha decidido que era imprescindible de cara a sus clientes que ese sistema estuviera certificado por una entidad externa, como **AENOR**. Para ello, auditores cualificados de esta empresa han llevado a cabo una auditoría informática sobre SI Andalucía, y **José** ha estado muy ocupado atendiendo al auditor. En ella, han realizado una evaluación de todos los sistemas de la empresa, equipos informáticos, sistemas de información, procedimientos, controles, bases de datos, seguridad, rentabilidad, documentación, planes de calidad... El resultado ha sido más que satisfactorio en todos los aspectos. Por tanto, han recibido el sello correspondiente de calidad certificada, y podrán usarlo en toda la documentación de la empresa.



En la unidad anterior comentamos el uso de auditorías como mecanismos para realizar verificaciones y validaciones del software. La **auditoría** también es una herramienta útil para evaluar la calidad de los proyectos y la empresa. Podemos realizar **auditorías internas** realizadas por personal de la propia empresa, o **auditorías externas** en las que un servicio o empresa ajena a nuestra organización realiza la auditoría. En este apartado vamos a explicar los aspectos fundamentales de las auditorías informáticas.



Empecemos por definir el **concepto de auditoría informática**:

Es el conjunto de técnicas, actividades y procedimientos, destinados a analizar, evaluar, verificar y recomendar en asuntos relativos a la

planificación, control eficacia, seguridad y adecuación del servicio informático en la empresa, por lo que comprende un examen metódico, puntual y discontinuo del servicio informático, con vistas a mejorar en:

- **Rentabilidad**
- **Seguridad**
- **Eficacia**



En definitiva, se trata de comprobar si la empresa está haciendo las cosas tal y como dice que las hace, y además, comprobar si los desarrollos que realiza se ajustan a los estándares de calidad necesarios para satisfacer las necesidades de los clientes.

La **auditoría informática** deberá comprender la evaluación de todos los sistemas de la empresa, equipos informáticos, sistemas de información, procedimientos, controles, bases de datos, seguridad, documentación, planes de calidad, etc. La auditoría del Sistema de Información en la empresa, a través de la evaluación y control que realiza, tiene como objetivo fundamental mejorar la rentabilidad, la seguridad, la eficacia y la calidad de la empresa u organización.



Las auditorías informáticas deben seguir una **metodología** preestablecida a partir de un plan elaborado en conjunto por la empresa auditora y la empresa que se audita. El **desarrollo de la auditoría** tendrá una fase de diagnóstico con entrevistas, cuestionarios, observaciones directas, estudio de documentos, etc. y una fase en la que se preparará un plan de mejoras. Como conclusión, podemos decir, que la auditoría es un mecanismo de evaluación y control que ayuda a asegurar la calidad de las empresas.

Autoevaluación

Indica cuál de estas afirmaciones es cierta:

- a) Las auditorías internas las realiza una empresa consultora contratada.
- b) Las auditorías externas las realiza una empresa consultora contratada.
- c) Las auditorías externas las realiza el personal de la propia empresa.
- d) Las auditorías internas deben realizarse por personas ajenas a la empresa.

[Comprobar](#)

Calidad del software

Caso.



Cuando implantaron el sistema de gestión de calidad, en **SI Andalucía** todos pensaban que sólo era algo que se definía y organizaba en la empresa, pero pronto se dieron cuenta de que cada **proyecto** que desarrollaban tenía sus



peculiaridades, y que era necesario hacer un plan de calidad adaptado a cada proyecto, detallado con una serie de puntos preestablecidos. Además se dieron cuenta que en cada proyecto no era sólo necesario establecer el plan de calidad, sino que era necesario incidir sobre determinados factores para asegurar que esa calidad se mantiene. Esos **factores** están formados por datos medibles y cuantificables, de forma que permitan en cada momento saber en qué nivel de calidad está el proyecto y proponer planes de mejora. Tras aplicar esos planes y volver a medir esos valores, se podrá establecer el grado de mejora real que se ha producido. Por tanto, para cada uno de los proyectos que se emprenden en SI Andalucía, se deben registrar los valores de una serie de factores de calidad, que comprenden características operativas, características sobre revisión del producto y sobre la transición del producto a nuevos sistemas. **Carmen y Víctor**, que se encargan de medir esas características en todos los proyectos en los que participan, se quejan mucho del trabajo que supone. José les dice que ese trabajo no es ni mucho menos gratuito, y que produce sus frutos, porque para saber si las cosas se están haciendo con calidad, medir es imprescindible, y usar la información medida para decidir posibles mejoras es igualmente imprescindible. Y claro está, tomar decisiones es lo que les permite hacer las cosas bien. José les comenta, a modo de ejemplo, que el hecho de haber trabajado con criterios de calidad les ha permitido en más de una ocasión reutilizar software de unos proyectos en otros nuevos, como cuando hicieron una aplicación para una biblioteca municipal usando un porcentaje alto del código de la aplicación de gestión de un videoclub que ya tenían.



En los **modelos de calidad** que hemos descrito hemos podido ver que todos incluyen acciones que deben ser desarrolladas a nivel de toda la empresa y acciones que deben **concretarse** para cada uno de los proyectos que vayamos a desarrollar, como podemos observar en la figura adjunta. Debido a las exigencias de normas del cliente (por ejemplo dar soporte a su modelo de calidad ISO) o a condiciones especiales del producto a desarrollar (por ejemplo, en un proyecto para la administración pública hay que seguir su



normativa), los documentos generales deben adaptarse a las necesidades de los desarrollos al pasar del nivel de empresa al nivel de proyecto.

Para cada nuevo desarrollo es necesario elaborar un **Plan de Calidad del Proyecto** que deberá recoger información sobre:

- **Descripción general del desarrollo:** con indicaciones del producto a obtener.
- **Planificación de la gestión del proyecto:** con estimaciones de tiempo y costes, recursos necesarios, financiación, viabilidad, análisis de riesgos, etc.
- **Descripción del desarrollo:** metodologías de desarrollo, modelo de ciclo de vida, documentación a generar, herramientas CASE, estrategias, técnicas y herramientas a utilizar.
- **Plan de aseguramiento de la calidad:** se realiza a nivel del proyecto e incluirá todas las medidas y estrategias que debemos aplicar durante el desarrollo para asegurarnos de que vamos a mantener los niveles de calidad acordes con nuestro modelo de calidad y con las necesidades y expectativas concretas de nuestro cliente. Este plan debe incluir tres grandes acciones:
 - El plan de **verificación y validación** del software con las actividades de pruebas, revisiones y auditorías a realizar, este tema se vio en la unidad anterior.
 - Las **métricas** del software para la evaluación del proyecto y sus procesos, lo podremos ver más adelante en el apartado de **métricas del software**, en esta misma unidad.
 - La **gestión de la configuración** del software. Como se ha ido viendo a lo largo de todo este módulo profesional, la aplicación de un modelo de desarrollo del software es una pieza clave para obtener software de calidad. La gestión de su desarrollo y el control de todos sus procesos será clave. La gestión de la configuración establece, en un momento dado del desarrollo, la configuración necesaria para que el producto obtenido satisfaga las necesidades del cliente. Por ejemplo, una tarea de la gestión de configuración, es el etiquetado y control de las versiones de los documentos elaborados, evitando confusiones al equipo de desarrollo.

Para asegurarnos la **calidad del proyecto** siempre es necesario que se cumplan ciertos estándares de calidad que debemos medir a través de las métricas. Aspectos como la mantenibilidad, fiabilidad, portabilidad o usabilidad del software son clave en un desarrollo de calidad, veremos estas cuestiones en el apartado siguiente. Por otro lado, en lo referente a la calidad del código un aspecto fundamental es la documentación de los programas, por ello le dedicamos también un apartado en esta unidad.

Autoevaluación

¿Qué elementos son clave para un plan de aseguramiento de la calidad?

- a) Un plan de verificación y validación, la aplicación de métricas y la gestión de la configuración software.
- b) Un Manual de calidad y un plan de uso de herramientas. CASE

- c) Un plan de verificación y validación y la aplicación de métricas.
- d) Un plan de verificación y validación, la aplicación de métricas y un plan de calidad.

[Comprobar](#)

Calidad del software

Factores para el aseguramiento de la calidad

En la anterior unidad vimos cómo las pruebas y revisiones estaban orientadas a la detección de errores del software, pero en esta unidad hemos visto que la **calidad** es algo más que evitar fallos. Debemos cumplir las especificaciones del cliente, y hacer desarrollos competitivos y repetibles en el futuro. Para ello, hemos visto distintos **modelos** que proponen estrategias y medidas para aplicar una gestión de calidad a nuestros desarrollos y productos. Pero todos estos modelos exigen realizar evaluaciones del producto. Si hablamos de coches es relativamente fácil establecer unos parámetros de calidad que deben cumplirse: rendimiento del motor, consumo, seguridad mediante test de accidentes, medición del espacio habitable, etc. Sin embargo, para el **software**, por sus propias características, es difícil medir su bondad o calidad. Para solucionar el problema, los diferentes modelos han descompuesto el concepto genérico de **calidad del software** en propiedades que puedan ser medibles. En ese sentido, para asegurarnos la calidad del software y buscar la excelencia, deberemos buscar que nuestro desarrollo proporcione altos valores en estas propiedades, veamos cuáles son esas propiedades.



Existen diferentes **modelos de calidad del software** pero nos basaremos en el de **McCall** por ser de los más conocidos.

Este **modelo** descompone el concepto de calidad del software en tres **capacidades** del software desde el punto de vista del usuario. Cada capacidad se descompone en unos **factores** que determinan la calidad. Estos factores pueden implicar la existencia de varias **propiedades** o criterios que deben satisfacerse. Lo que propone este modelo es evaluar estas propiedades mediante una métrica que las mide, y a partir de ahí podremos tomar decisiones de mejora. Es decir, si queremos proporcionar el aseguramiento de la calidad del software, deberemos asegurarnos de que estas propiedades están presentes en nuestro desarrollo, y si la métrica indica que no lo están, introduciremos cambios para alcanzarlos.



A continuación indicamos el desglose de estos **factores de la calidad del software** clasificados por capacidades definidas desde el punto de vista del usuario:

■ **Características operativas.**

- **Corrección y funcionalidad.** ¿Hace lo que se le pide?: Grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente
- **Fiabilidad y robustez.** ¿Lo hace de forma fiable todo el tiempo?: Grado que se puede esperar de una aplicación para que lleve a cabo las operaciones especificadas y con la precisión requerida
- **Eficiencia.** ¿Qué recursos hardware y software necesito?: Cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados
- **Seguridad e integridad.** ¿Puedo controlar su uso?: Grado con que se puede controlar el acceso al software o a los datos a personal no autorizado
- **Facilidad de uso.** ¿Es fácil y cómodo de manejar?: Esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella y conseguir resultados

■ **Revisión del producto:** capacidad para soportar cambios

- **Facilidad de mantenimiento.** ¿Puedo localizar los fallos y corregirlos?: Esfuerzo requerido para localizar y reparar errores.
- **Flexibilidad y extensibilidad.** ¿Puedo modificarlo o añadir nuevas opciones?: Esfuerzo requerido para modificar, ampliar o adaptar a nuevas funcionalidades una aplicación
- **Facilidad de prueba.** ¿Puedo probar todas las opciones?: Esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.



■ **Transición del producto:** adaptabilidad a nuevos entornos.

- **Portabilidad y compatibilidad.** ¿Podré usarlo en otra máquina?: Esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
- **Reusabilidad y reutilización.** ¿Podré reutilizar alguna parte del software?: Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones, es un factor crítico en el software actual y de futuro.
- **Interoperabilidad.** ¿Podré hacerlo interactuar con otro sistema?: Esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos



Lo que importa es satisfacer estos **factores externos** de la aplicación (los que puede percibir el usuario de la aplicación), pero para lograrlo tendremos que cumplir con otros **factores internos** (los que están implementados en el código, que sólo podrán ser comprobados por otros desarrolladores del software). Estos factores serán los que permitirán conseguir los parámetros externos de calidad que puede percibir el usuario. Algunos de esos factores externos son:

- Modularidad,
- Legibilidad,
- Transparencia,
- Abstracción,

- Ocultación de la implementación,
- Complejidad ciclomática,
- Líneas de código,
- etc.

Estos factores los deberemos medir con la métrica adecuada. En todo este tema, juega un papel fundamental para el éxito, la utilización de **metodologías de desarrollo** proporcionadas por la Ingeniería del software y el uso de **herramientas CASE** de ayuda y apoyo.

Autoevaluación

¿Qué factores de calidad corresponden con las características operativas del software?

- a) Portabilidad, flexibilidad, robustez.
- b) Eficiencia, integridad, fiabilidad, corrección
- c) Facilidad de uso, facilidad de prueba, facilidad de mantenimiento
- d) Reutilización, compatibilidad, robustez e integridad

Comprobar

Calidad del software

Documentación de los programas

A lo largo del **ciclo de vida** del desarrollo del software hemos estado viendo que se obtienen diferentes productos, algunos de los cuales son documentos. Todos ellos participarán de las especificaciones marcadas para el aseguramiento de la calidad. En esta unidad hemos visto algunos de los factores que marcan **la calidad del software**, algunos de ellos estaban ligados a la actividad de la programación. Para alcanzar un buen nivel de calidad en factores como la mantenibilidad, reusabilidad o facilidad de uso del software es importante que los programas estén bien documentados. En este apartado vamos a ver los aspectos que deben tenerse en cuenta para realizar una **documentación de calidad del código**. No vamos a hablar de la documentación relativa al plan de pruebas y al manual del usuario porque ya han sido tratados en otras unidades y forman parte de otros procesos del desarrollo del software.

La documentación nos va a facilitar la **detección de errores** y las **modificaciones** futuras del código. Esta documentación será útil:

- para el propio programador que lo ha realizado,
- para otros programadores que lo usen en el futuro,
- para otros miembros del equipo de desarrollo (por ejemplo para los encargados de las bases de datos), y
- para el usuario del software.



Documentar el código de un programa es añadir información para explicar lo que hace, de forma que no sólo los ordenadores sepan interpretarlo, sino que además los humanos entiendan qué está haciendo y por qué.

La documentación de los programas tiene dos **partes**, una **interna**, que está incluida en el propio código, y otra **externa**, que acompañará al código:



- **Documentación interna:** esta documentación forma parte del propio código como comentarios y cubre los aspectos de legibilidad del código fuente relativos a la sintaxis del lenguaje. Deben considerarse los siguientes aspectos:
 - **Cabecera** del programa o subprogramas: nombre del programador, fecha de la versión actual, breve descripción de la función del programa, etc.
 - **Nombres significativos** para describir **identificadores**, variables, atributos, etc
 - **Comentarios** relativos a la **función del programa** como un todo, así como los paquetes, módulos, clases y métodos que lo componen.
 - **Claridad de estilo y formato** (una sentencia por línea, líneas en blanco para separar módulos, etc.).
 - **Comentarios significativos** obtenidos a través de la **verificación** del programa.
 - Uso de **comentarios con un formato definido que permita generar** a partir de ellos **la documentación externa** en un formato flexible, y de forma automática. Ejemplo: Comentarios javadoc en java, que permiten generar automáticamente la documentación de las clases en formato HTML, listo para su publicación en la web.
- **Documentación externa:** documentación ajena al programa fuente, que se suele incluir en un manual que acompaña al programa. Debe incluir:
 - **Especificación del programa:** documento que define el propósito y modo de funcionamiento del programa e incluye esta información:
 - Nombre del programa.
 - Proceso al que pertenece.
 - Sistema al que pertenece.
 - Lenguaje de programación.
 - Descripción de la función general del programa con indicación de las tareas a realizar y del algoritmo de resolución en forma de diagrama de flujo, tablas de decisión, etc.
 - Diagrama de estructuras o diagramas de bloques.
 - **Listado** del programa fuente, mapas de memoria, referencias cruzadas, etc.
 - Especificaciones de **fórmulas complejas**.
 - Especificación de los **datos** a procesar: archivos o bases de datos externas incluyendo el formato de las estructuras de los registros, campos, etc.
 - Descripción de entradas/salidas y **formato de pantallas** utilizadas para interactuar con los usuarios.
 - Cualquier indicación que pueda servir a los programadores que deben **mantener** el programa.



Para saber más

El paquete de desarrollo Java incluye una herramienta, javadoc, para generar un conjunto de páginas web a partir del código. Esta herramienta toma en consideración algunos comentarios del código para generar una documentación de clases y componentes de clases (variables y métodos). Aunque javadoc no ayuda a la comprensión de los detalles de código, sí ayuda a la comprensión de la arquitectura del programa. La mayoría de los IDE (entornos integrados de desarrollo) para Java, además, tienen integrada la generación de documentación a partir de los comentarios javadoc, de forma integrada y automática. Puedes ver más información de javadoc visitando la página web de la compañía Sun:

Sun: javadoc

<http://java.sun.com/j2se/javadoc/index.jsp> [versión en cache]

Calidad del software**Caso.**

Víctor y Carmen, cuando se encargan de recoger los datos medibles sobre los **factores de calidad** que hay que registrar en cada proyecto, no lo hacen de cualquier manera. Siguen para ello una serie de **reglas** bien definidas, y establecidas claramente en una **metodología de métrica del software** determinada.



Eso resulta imprescindible por varios motivos.

Uno es tener una sistemática de trabajo que permita reducir los tiempos necesarios para realizar esta tarea.

Otra, asegurar que se ha medido correctamente todo lo que debe ser medido, y que se han extraído las conclusiones acertadas sobre esas mediciones.

Un tercero, y no menos importantes, es que de esta forma, cualquier otra persona podrá entender correctamente los resultados que ellos obtengan, incluido, por supuesto, el auditor que se encargue de hacer las auditorías de certificación, tanto iniciales como de renovación.

Como ya sabemos, para llevar a cabo el **aseguramiento de la calidad** del software es necesario completar tres aspectos: las verificaciones y validaciones del software, la gestión de la configuración y las métricas.

Todos los modelos de calidad que hemos explicado a lo largo de esta unidad hacen uso de las métricas, por todo ello vamos a describirlas y presentar algunas de las técnicas asociadas más utilizadas.



Comenzamos por hacer una **definición formal de Métrica**:

Es una asignación de un valor (cuantitativo o cualitativo) a un atributo (tiempo de ejecución, complejidad, etc.) o un proceso (pruebas, diseño, etc.), según unas reglas bien definidas.

En la unidad 5 de este módulo ya vimos propuestas de medición como la estimación de costes por medio del modelo COCOMO. Ahora veremos aquí algunas métricas con el objetivo del aseguramiento de la calidad.

Las **mediciones** pueden ser:

- **dinámicas** y hacerse en tiempo ejecución, o
- **estáticas** y hacerse sobre el diseño, código del programa y su documentación.

Un **ejemplo** de medida dinámica sería medir lo que tarda un programa en procesar un volumen determinado de datos, así podemos calcular la eficiencia de diferentes implementaciones. O podemos **medir el tiempo** transcurrido entre fallos del sistema para comprobar la fiabilidad del programa. Como vemos, las métricas dinámicas tienen una correspondencia directa con los atributos de calidad del software (eficiencia, fiabilidad, etc).

Las métricas estáticas tienen una relación indirecta con los atributos de la calidad. Vamos a ver algunos **ejemplos** de estas métricas:

- **Longitud del código**: mide el tamaño del programa, en general cuanto mayor sea el programa más posibilidades habrá de errores. Se pueden contar las líneas de código o conocer la proporción de operandos, operadores, etc.
- **Medidas de los comentarios**: se puede medir el número de líneas de comentario o la proporción de comentarios sobre el total de líneas.
- **Estudio de los grafos de control**: por ejemplo a través de la complejidad ciclomática de McCabe que vimos en la unidad anterior. Valores altos pueden indicar un código demasiado complejo que debe refactorizarse para simplificarlo.
- **Nombres de los identificadores**: nombres largos pueden ayudar a entender mejor el código.
- **Acoplamiento entre módulos**: hay dos medidas básicas.:
 - El **Fan-in** de una función mide el número de módulos o funciones que llaman a dicha función y los parámetros utilizados. Un valor alto de fan-in significa que la función está fuertemente acoplada con los otros módulos y su cambio provocará efectos colaterales.
 - El **Fan-out** mide el número de funciones que son llamadas por la función y los datos actualizados. Un valor alto de fan-out puede indicar un exceso de complejidad por la lógica de control necesaria para coordinar los componentes llamados.



Para la **programación orientada a objetos** se han creado nuevas métricas ya que algunas de las anteriores están muy ligadas a la programación estructurada. Veamos algunas:



- **Medida fan-in y fan-out:** similar a la anterior pero referida a los métodos y distinguiendo las llamadas realizadas dentro del objeto y las realizadas desde objetos distintos.
- **Complejidad de los métodos:** se contaría el número de métodos complejos de un objeto, valores altos indicarían que la clase podría ser difícil de mantener y quizás habría que rediseñarla.
- **Proporción de métodos y atributos ocultos:** valores altos indican una buena encapsulación.
- **Proporción de métodos y atributos heredados:** valores altos pueden implicar un exceso de herencia que dificultará su mantenimiento.
- **Cohesión en los métodos:** mide el grado en que los métodos hacen referencia al mismo conjunto de atributos. Un valor alto indicaría un diseño con un buen nivel de abstracción.
- **Relación de herencia:** mide el número de operaciones de una superclase que son anuladas en la subclase. Un valor alto indica que la superclase no es la clase madre adecuada para la subclase.
- **Profundidad del árbol de herencia:** mide el número de niveles del árbol de herencia de las clases. Un valor alto indica un exceso de complejidad, normalmente no deben superarse los 6 niveles.

En cuanto a las métricas para comprobar la **seguridad** del sistema tenemos:

- Número de palabras **claves** de acceso caducadas.
- Número de **usuarios** externos al sistema.
- Frecuencia de las **actualizaciones** contra **virus**.
- Número de **virus** detectados en un período de tiempo.
- Número de **intrusiones** detectadas en la red en un período de tiempo.

Todas estas métricas deben utilizarse a partir de los objetivos marcados y dentro de un plan completo de calidad para que tengan utilidad. Además, debemos ayudarnos de **herramientas software** que automaticen las mediciones y gestionen su control.

Autoevaluación

¿Qué tipo de métrica es la medida de la profundidad del árbol de herencia?

- a) Dinámica.
- b) Estática para la programación estructurada.
- c) Estática para la programación orientada a objetos.
- d) Dinámica para la programación orientada a objetos.

Comprobar

Calidad del software