



El Estándar ISO y su Aportación al Proceso de Calidad del Desarrollo de Software

Francisco Raúl Calderón Macías
Ingeniería Técnica en Informática de Gestión
TFG - Gestión de Proyectos

Xavier Martínez Munné
Carme Martín Escofet

Junio 2016

Copyright

© (Francisco Raúl Calderón Macías)

Reservados todos los derechos. Está prohibida la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual por un permiso o bajo los términos acordado con la organización adecuada de los derechos de reprografía. Las consultas relativas a la reproducción fuera del alcance de lo anterior deben ser enviadas al autor. No debe circular este trabajo de ninguna otra forma y que debe imponer esta misma condición en un adquirente. Los enlaces a sitios web de terceros son proporcionados por el autor de buena fe y sólo a título informativo. El autor se exime de cualquier responsabilidad por los materiales contenidos en páginas web de terceros que se hace referencia en este trabajo.

Copyright

© (Francisco Raúl Calderón Macías)

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior permission in writing of the author, or as expressly permitted by law, by licence or under terms agreed with the appropriate reprographics rights organization. Enquiries concerning reproduction outside the scope of the above should be sent to the author. You must not circulate this work in any other form and you must impose this same condition on any acquirer. Links to third party websites are provided by the autor in good faith and for information only. The author disclaims any responsibility for the materials contained in any third party website referenced in this work.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>El Estándar ISO y su Aportación al Proceso de Calidad del Desarrollo de Software</i>
Nombre del autor:	<i>Francisco Raúl Calderón Macías</i>
Nombre del consultor/a:	<i>Xavier Martínez Munné</i>
Nombre del PRA:	<i>Atanasi Daradoumis Haralabus</i>
Fecha de entrega (mm/aaaa):	06/2016
Titulación::	<i>Ingeniería Técnica en Informática de Gestión</i>
Área del Trabajo Final:	<i>TFC - Gestión de Proyectos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<p>Estándar ISO Criterios de Calidad Desarrollo de Software Aplicaciones Software Personalizadas Calidad del Software</p> <p>ISO Standard Quality Criteria Software Development Custom-designed Software Applications Software Quality</p>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Actualmente el desarrollo de aplicaciones y sistemas de software ha crecido de manera sustancial, esto es así, hasta el punto que nadie entendería que cualquier empresa o negocio, del tamaño que sea, prescindiera del uso de la informática y de aplicaciones de software diseñadas a medida.</p> <p>Si el desarrollo del software no va en concordancia con el cumplimiento de unos criterios de calidad óptimos, que garanticen la creación de aplicaciones de una eficacia notable, estaremos perdiendo efectividad y versatilidad en el uso de las tecnologías.</p> <p>Hoy en día existen una gran cantidad de estándares que proponen criterios y pautas para que esta producción de software alcance un valioso nivel de</p>	

calidad, para que en definitiva, redunde en obtener un buen producto, de lo cual se beneficiará el usuario final en su uso diario.

Uno de los estándares más conocidos en todos los ámbitos, no solo en el mundo de la informática, con unos criterios y procedimientos de aplicación exigentes, y con mayor reputación, es el Estándar ISO.

Este “Estándar ISO” es el que deseamos analizar, viendo cómo actúa en cada fase o etapa del desarrollo software, analizando sus fortalezas y sus debilidades, y decidir, de acuerdo con los resultados obtenidos, qué se puede implementar, modificar y/u omitir, para hacer de este estándar ISO un arma casi infalible para el desarrollo de un software de calidad.

Abstract (in English, 250 words or less):

Currently, the development of software applications and systems, has increased considerably, to the extent that nobody would understand that any company or business, whatever its size, to operate without using computing and custom-designed software applications.

If there is no concordance between the development of software, does not meet the highest quality criteria, that guarantee the creation of highly efficacious applications, we will be losing effectiveness and versatility in the use of technologies.

These days there are a great number of standards that put forward criteria and guidelines in order that this software production reaches a useful quality standard, so that, in general, it results in achieving a good product, from which the final user will benefit from daily.

One of the best known standards in every field, not exclusively in the world of computer science, with very demanding criteria and application procedures, and with the highest reputation, is the ISO standard.

Our aim is to analyse this ISO standard, examining how it works in every stage of software development, as well as strengths and weaknesses. According to the results obtained, we will decide what can be implemented, modified and/or omitted, in order to make the ISO standard an infallible powerful tool for the high-quality software developments.

Índice

1. Introducción	8
1.1 Contexto y justificación del Trabajo.....	8
1.2 Objetivos del Trabajo.....	9
1.3 Enfoque y método seguido.....	10
1.4 Planificación del Trabajo.....	11
1.5 Breve resumen de productos obtenidos.....	17
1.6 Breve descripción de los otros capítulos de la memoria.....	17
2. Antecedentes históricos y evolución	18
2.1 Concepto de Calidad y Evolución.....	18
3. Modelos de Calidad del Software	19
3.1 Introducción.....	19
3.2 Modelos de Calidad del Software.....	22
3.2.1 Modelo CMM.....	22
3.2.2 Modelo CMMI.....	24
3.2.3 AENOR.....	26
3.3 Metodología ISO.....	27
3.3.1 Normas ISO 9000, ISO 9001 y 9004.....	29
3.3.2 ISO/IEC 9126.....	30
3.3.3 ISO/IEC 12207.....	35
3.3.4 ISO/IEC 15288.....	37
3.3.5 ISO/IEC 15504.....	38
3.3.6 ISO 20000-1.....	40
3.3.7 ISO 25000-1.....	41
4. Análisis del Estudio Realizado	45
4.1 Ventajas e inconvenientes del modelo <i>CMM</i>	45
4.2 Ventajas e inconvenientes del modelo <i>CMMI</i>	46
4.3 Ventajas e inconvenientes del modelo <i>AENOR</i>	47
4.4 Ventajas e inconvenientes de las normativas ISO 9000, 9001 y 9004....	48
4.5 Ventajas e inconvenientes de las normativas ISO 12207 y 15504.....	49
4.6 Ventajas e inconvenientes de la normativa ISO 15288.....	50
4.7 Ventajas e inconvenientes de la normativa ISO 20000.....	50
4.8 Ventajas e inconvenientes de la normativa ISO 25000.....	52
5. Conclusiones del Estudio	53
5.1 Características de la Nueva Norma <i>ISO</i> para Proyectos de Pequeño y Mediano Volumen.....	54
5.2 Características de la Nueva Norma <i>ISO</i> para Proyectos de Gran Volumen.....	56
6. Glosario	58
7. Bibliografía y Referencias	60

Lista de figuras

Figura 1. Diagrama de Gantt para el hito 1.....	página 12
Figura 2. Diagrama de Gantt para el hito 2.....	página 13
Figura 3. Diagrama de Gantt para el hito 3.....	página 13
Figura 4. Diagrama de Gantt para el hito 4.....	página 14
Figura 5. Diagrama de Gantt para el hito 5.....	página 16
Figura 6. Gráfico de informe <i>Chaos Report</i>	página 20
Figura 7. Gráfico de los niveles del modelo <i>CMM</i>	página 23
Figura 8. Gráfico de los niveles del modelo <i>CMMI</i>	página 25

1. Introducción

1.1 Contexto y justificación del Trabajo

Es evidente que el mundo de la informática ha monopolizado en la actualidad cualquier ámbito de nuestra sociedad, siempre con la mejor intención, que no es otra que facilitar la labor de sus usuarios.

Nadie entendería, que cualquier empresa o negocio, del tamaño que sea, grande, mediana o pequeña, prescindiera del uso de la informática, ni del uso de aplicaciones de software diseñadas a medida.

Si los métodos de desarrollo de software no siguen unos estándares y criterios que garanticen que el proceso de creación de la programación aglutine unos valores de calidad exigentes, estaremos perdiendo efectividad y fluidez en el uso de las tecnologías.

Existen gran cantidad de estándares que proponen criterios para que ese desarrollo de software sea de alto nivel y de gran calidad, y que ello redunde en el usuario final y su gestión.

Uno de los estándares más conocidos y con mayor reputación en todos los ámbitos, no solo el informático, es el Estándar ISO.

Este es el que deseamos analizar, en cada una de sus versiones y evoluciones, viendo cómo actúa en cada fase del desarrollo de software. Se estudiarán los actuales y principales estándares de calidad y las normativas que se aplican en los procesos de desarrollo de software para garantizar un producto final de la más alta calidad.

Se profundizará en las fases de los modelos de construcción de software, y cómo actúa el estándar ISO en cada una de ellas, así como cuáles son sus criterios para garantizar un proceso de calidad.

Se valorarán sus fortalezas, sus debilidades, y se planteará qué se puede implementar, modificar y/o corregir, para hacer de este estándar un arma casi infalible del desarrollo de software de calidad. Se analizará si se está realizando de una manera adecuada, o si existe margen de mejora.

Todo lo anterior servirá para obtener un resultado de calidad y robustez.

1.2 Objetivos del Trabajo

El objetivo principal del presente trabajo será realizar un análisis que evalúe las fases del desarrollo de software, estudiando cómo actúa la aplicación de los actuales estándares y normativas para garantizar la calidad del producto generado, principalmente en trabajos de grandes dimensiones, y contemplar si pudiera ser necesario modificar, resumir o simplificar cualquier pequeña labor que conlleve el ahorro de mucho tiempo y recursos, siempre y cuando no afecte a la pérdida de calidad.

Aunque, por otra parte, no se trata de prescindir ni recortar actuaciones, ya que incluso, si el análisis del presente trabajo lo contempla, puede ser que se necesite añadir ciertos aspectos que redunden en beneficio de la excelencia del producto final.

Se trata de disponer de un proceso lo suficientemente refinado y concreto, cuya aplicación en cada una de las fases consiga garantizar calidad.

Desglosaremos el presente apartado en: Objetivos generales y Objetivos específicos.

1. Objetivos Generales: Son aquellos que mostrarán a grandes rasgos las ideas generales que se quieren acometer.

En nuestro caso, se pueden mencionar los siguientes:

- 1- Se explicará al lector qué son los estándares de calidad, informando para qué sirven y cómo actúan. Enumeraremos los principales estándares de calidad actuales: ISO, AENOR, IEC, etc., en el ámbito de la informática en general.
- 2- Se analizarán las metodologías de desarrollo del software, examinando su funcionamiento, sus fases de diseño e implementación en el desarrollo del software, contemplando sus puntos fuertes y débiles.

2. Objetivos Específicos: En este apartado se profundizará con más detalle en los objetivos anteriormente mencionados, por ejemplo:

- 3- Se sintetizará la información de las principales normativas ISO, AENOR, IEC, etc., sus historias, versiones y evoluciones en el tiempo.
- 4- Concretaremos el trabajo en el estándar ISO, analizando cómo trabaja en cada fase del desarrollo de software, desde el punto de vista de la calidad del software.

- 5- Extraeremos las fortalezas y debilidades del estándar ISO aplicado al proceso de desarrollo de software.
- 6- Los resultados obtenidos serán analizados, proponiéndose la implementación, modificación y/o restricción, de acciones generales o puntuales que puedan reforzar la actual normativa, para hacer de este estándar ISO un método casi infalible en el proceso de desarrollo de software, para poder obtener programas una calidad óptima.

1.3 Enfoque y método seguido

En el presente trabajo el método a seguir consistirá en repartir el trabajo en dos partes diferenciadas, pero no aisladas.

En una primera parte, que coincidirá con la primera entrega, se realizará un arduo trabajo de recopilación de información de los conceptos que el proyecto abarca.

La segunda fase, y que se finalizará con la segunda entrega, se dedicará práctica y exclusivamente a realizar los trabajos de análisis, evaluación y valoración de toda la información extraída, para que con ello podamos obtener unas conclusiones objetivas y meditadas, viendo cómo y cuáles son los criterios aplicados, y qué acciones se podrían tomar para poder acrecentar la garantía de calidad en el desarrollo de software.

Este trabajo no se podría entender si ambas fases no estuviesen relacionadas, puesto que toda la información recopilada durante la primera fase, influirá directamente en la etapa de evaluación. De la misma manera, mientras se trabaja en la fase de análisis y conclusiones de datos, esto implicará reflejar ciertos aspectos en la fase previa. Ambas fases estarán “vivas” y en constante actualización durante todo el proceso de realización del presente trabajo.

Evidentemente, este trabajo tratará de dar un enfoque de estudio y análisis de los estándares actuales existentes, no pretenderá crear un nuevo estándar, sino poder aportar ideas y/o conclusiones que implementen a las ya existentes, con el único objetivo de ser tenidas en cuenta para fortalecer su aplicación.

1.4 Planificación del Trabajo

En este apartado se hará un desglose de todo el trabajo global que el presente proyecto propondrá. Se iniciará con la entrega del Plan de Trabajo, pasando por sus distintas etapas, que irán coincidiendo con las entregas a realizar, hasta llegar a las fases finales, donde se realizará la preparación de las diapositivas para su exposición durante la defensa.

El trabajo se divide en 5 etapas:

- **Hito 1:** Elaboración del Plan de Trabajo.

En este apartado, se elaborará un Plan de Trabajo que recogerá el diseño y planificación que se seguirá durante el resto de la elaboración del proyecto.

Este apartado se realizará entre el 2 de marzo al 15 de marzo, ambos inclusive.

Algunas de las tareas que se realizarán son:

- Lectura del Plan Docente y los módulos teóricos de la asignatura.
- Recopilación de lluvia de ideas de la información.
- Descripción del Trabajo Fin de Carrera.
- Contexto y Justificación del Trabajo.
- Objetivos del Trabajo: separado en Objetivos Generales y Objetivos Específicos.
- Exposición del Enfoque y Método a Seguir.
- Planificación del Trabajo: Hitos, Temporización General y Elaboración del Diagrama de Gantt.
- Elaboración del Sumario de Productos Obtenidos.
- Descripción de Otros Capítulos de la Memoria.
- Elaboración del Tipo de Licencia del Trabajo Final de Carrera.

Y todo ello finalizará el 15 de marzo, con la entrega de la PAC 1.

Plasmando todo lo anterior en un Diagrama de Gantt centrado únicamente en esta primera etapa, quedaría de la siguiente manera:



Figura 1. Diagrama de Gantt para el hito 1.

- **Hito 2:** PAC 2.

La ejecución de esta entrega se realizará entre el 16 de marzo al 19 de abril, ambos inclusive, y corresponde a la etapa en la cual se entregará la primera fase de la ejecución del plan de trabajo.

En esta etapa se trabajarán los siguientes conceptos:

- Fase de Recopilación de Información y Documentación.
- Estudio y Compilación de Información sobre los Distintos Estándares de Calidad.
- Estudio en Profundidad del Estándar ISO aplicado al Desarrollo del Software.
- Estudio de las Metodologías de Desarrollo de Software.
- Preparación de Capítulos y Documentación para Entrega.

La entrega de la 1ª Fase del Trabajo Fin de Carrera deberá cumplir con el Diagrama de Gantt del presente hito, que se muestra a continuación:

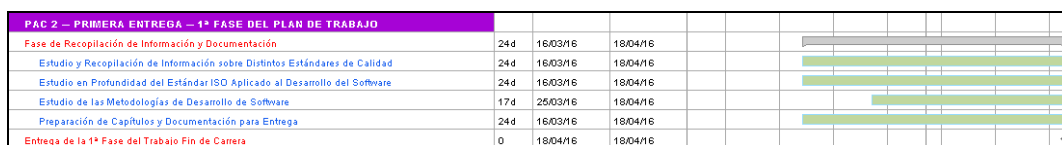


Figura 2. Diagrama de Gantt para el hito 2.

- **Hito 3:** PAC 3.

El trabajo continuará avanzando, y comenzarán las labores de realización de la fase de evaluación y análisis de toda la información que se ha recogido durante la entrega anterior.

Por ejemplo:

- Estudio y Recopilación de Información sobre los Distintos Estándares de Calidad.
- Realización de Análisis.
- Presentación de Conclusiones.
- Muestra de Fortalezas.
- Muestra de Debilidades.
- Propuestas del Autor para Mejora del Estándar ISO en el Desarrollo de Software.

Esto se deberá realizar entre el 20 de abril y el 24 de mayo, pudiendo de esta manera, dar por finalizada la segunda entrega de la ejecución del Plan de Trabajo.

Se representa en el siguiente Diagrama de Gantt:

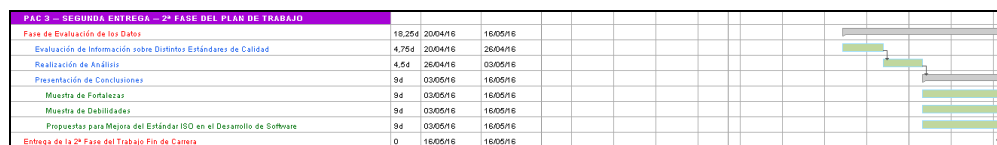


Figura 3. Diagrama de Gantt para el hito 3.

Hasta aquí llegarían las fases principales de estudio y análisis, pero aún restarían dos entregas más, que corresponderían a:

- **Hito 4:** Entrega Final del Trabajo.

Para concluir, se realizará un repaso general de toda la documentación hasta ahora aportada.

Se elaborará una memoria, así como la preparación de una presentación en soporte informático, que se utilizará como resumen y apoyo para ayudar a la comprensión de Trabajo Fin de Carrera por parte del Tribunal.

Este apartado deberá recopilar información relativa a:

- Enfoque y método seguido.
- Breve sumario de las conclusiones obtenidas.
- Descripción de otros capítulos de la memoria.
- Conclusiones.
- Glosario.
- Bibliografía.
- Anexos.
- Documentación de la memoria.
- Preparación de la Defensa.
- Autoinforme del Trabajo.
- Entrega de la Memoria y la Defensa.

La representación gráfica en un Diagrama de Gantt sería la siguiente:

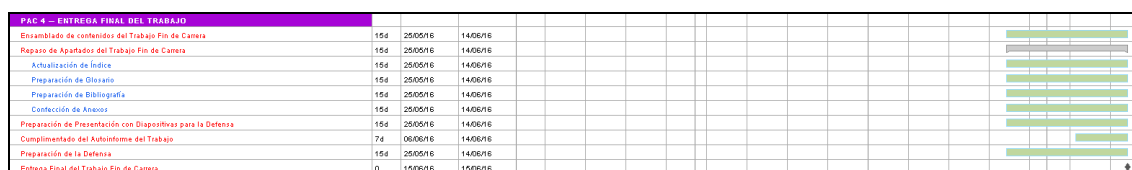


Figura 4. Diagrama de Gantt para el hito 4.

- **Hito 5:** Defensa del Trabajo Fin de Carrera ante el Tribunal.

Se trata de la última etapa del Trabajo Fin de Carrera, y que consistirá en explicar al Tribunal cuáles han sido las labores realizadas, en qué orden, con qué objetivos, así como las conclusiones y propuestas que se realizan en función a los datos obtenidos en las evaluaciones efectuadas.

Se utilizará una presentación con diapositivas como apoyo a la explicación y como soporte gráfico de comprensión de algunos de los conceptos principales que se desea trasladar al Tribunal. Esta misma presentación puede ser de ayuda para los futuros lectores o personas que deseen consultar el Trabajo Fin de Carrera.

En cuanto al aspecto de la dedicación temporal, se pretende trabajar todos los días laborales unas 2 horas aproximadamente, ya que por motivos laborales se hace prácticamente imposible dedicar más tiempo. Se debe tener en cuenta que el Trabajo Fin de Carrera se ejecutará a la vez que se cursan otras asignaturas y por este motivo se deberán dividir los tiempos de trabajo. Los días festivos y fines de semana se tratará de mantener la dedicación de 2 horas diarias.

Para el periodo de Semana Santa y la semana siguiente será imposible poder avanzar el trabajo por motivos personales, cuestión que ha complicará el avance deseado en la extensión de los puntos del Trabajo Fin de Carrera, aunque los puntos previstos en el hito 2 se incluirán totalmente.

Para finalizar, se mostrará mediante un gráfico cómo será el desglose temporal de las tareas que se realizarán, las cuales se reflejarán en un Diagrama de Gantt.

En él se mostrarán las tareas desempeñadas, las cuales han sido divididas en los hitos que anteriormente se han detallado.

El Diagrama de Gantt mostrará la duración de cada una de esas tareas, con sus fechas de inicio y fin programadas, así como las relaciones de dependencia entre ellas, hasta llegar a ser completadas, con la consecución del hito.

Si bien hay que decir, que la planificación indicada en el Diagrama de Gantt es flexible, esto quiere decir, que si algunas de las tareas necesitasen más tiempo o su orden se viese alterado, acabarían afectado a la propuesta inicial. No obstante, las fechas de entrega serán fijas e inamovibles, con lo cual se estima que las desviaciones sean mínimas.

En la página siguiente se verá una representación gráfica del Diagrama de Gantt completo.

Diagrama de Gantt

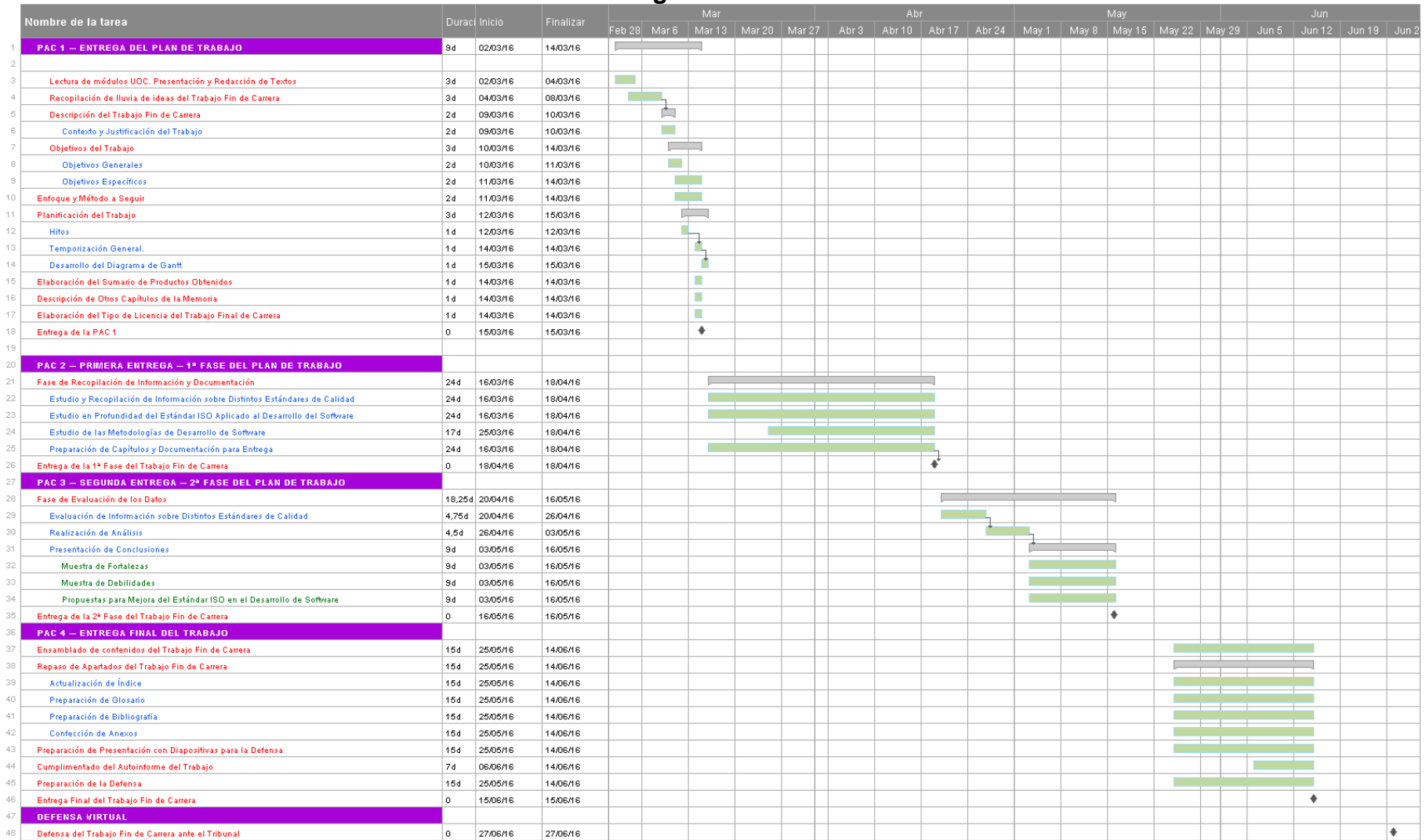


Figura 5. Diagrama de Gantt completo del Trabajo Fin de Carrera.

1.5 Breve resumen de productos obtenidos

El principal resultado de este trabajo será el de obtener un estudio que aportará, desde la mayor objetividad posible, un análisis de la evolución histórica del estándar ISO, sus actualizaciones, y ver si su aplicación en el apartado de calidad en el desarrollo del software pudiera conllevar algún tipo de acción, realizando posibles implementaciones o modificaciones.

El trabajo además, aportará otras documentaciones adicionales durante su realización, desde su inicio, en las cuales se podrá recoger una gran cantidad de información sobre los estándares de calidad, contribuyendo ello a introducir al lector en saber qué son los estándares de calidad, cómo se aplican, así como, una comparativa entre dichos estándares.

También se emitirán las conclusiones obtenidas respecto a cómo actúa el estándar ISO en el proceso de desarrollo de un software de calidad, apoyado en la aplicación de las metodologías del proceso de generación de aplicaciones informáticas.

En cada etapa se irán realizando análisis, comparativas, valoraciones, para finalmente poder emitir las conclusiones a las que nos llevarán todos estos estudios.

De este modo, también el lector podrá interpretar y captar los puntos fuertes y las debilidades en cada estándar, metodología, y obtener sus propias conclusiones.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos que irán conformando el presente trabajo se basarán en los siguientes contenidos generales, que posteriormente se irán ampliando y actualizando con la información que se irá recabando:

- Estándares actuales utilizados: qué son, inicios, historia, evolución, actualizaciones y últimas versiones activas que están en funcionamiento.
- Metodologías más comunes y utilizadas para el desarrollo de software.
- Comparativa entre las diversas metodologías existentes.
- Estándar ISO aplicado a la informática y en el desarrollo del software de calidad.

- Análisis de los resultados obtenidos del estudio del estándar ISO en el proceso de desarrollo del software.
- Conclusiones del trabajo.
- Glosario.
- Anexos.
- Bibliografía.
- Presentación.

Todos estos apartados se irán actualizando y perfeccionando a lo largo del trabajo, para que aporten el mayor contenido de información posible, y que sirvan de guía al lector y le ayuden a entender los análisis y conclusiones a los que se pretende llegar con el estudio.

2. Antecedentes históricos y evolución

2.1 Concepto de Calidad y Evolución

Antes de dar el primer paso de partida, será fundamental aclarar qué se entiende por *calidad*. Si se acude a la definición que aparece en una de las acepciones del diccionario de la Real Academia Española, la define como: “*Adecuación de un producto o servicio a las características especificadas*”.

Aparentemente esta definición no nos dice mucho, puesto que si extrapolásemos la definición de calidad al campo del desarrollo de software, no aportaría mucha ayuda, ya que cualquier producto que fuese el resultado de haber pasado sucesivamente por las fases de diseño e implementación y que funcionase, ya cumpliría la definición.

El presente proyecto pretende ir mucho más allá, no se trata de finalizar y obtener una aplicación, sino que esta se guie por un cumplimiento en todas las fases del ciclo de vida que garantice que el objetivo ha superado una serie de controles y expectativas con amplitud suficiente como para ser lanzado con prestigio y garantía al mercado.

Históricamente, y desde el crecimiento de la producción de desarrollo del software, se lanzan infinidad de productos constantemente al mercado, sin embargo, la inmensa mayoría de ellos carecían de unos mínimos controles de calidad. Actualmente es cuando más se está incidiendo en desarrollar software bajo protocolos y evaluaciones de calidad, que todo

producto de prestigio, debería pasar en todas sus fases, diseño, planificación, desarrollo, etc.

Desde el punto de vista empresarial siempre ha surgido el conflicto entre la rapidez de desarrollar software y la calidad. Se daba por supuesto, que tener que dotar al producto de unos niveles de calidad trascendentales supondría implícitamente una pérdida de tiempo y esfuerzo importantes. Es decir, finalizar un software que hubiese superado los controles de calidad exigibles, emplearía el mismo tiempo y recursos que poder ejecutar dos o más proyectos software. Esto visto desde un punto de vista empresarial no era lo deseable.

Sin embargo, la gran demanda de desarrollo de software, y el florecimiento de una multitud de empresas generadoras, han ayudado a comprender que obtener el fruto deseable pasa por priorizar la calidad por encima de la cantidad, y que esto redundará en un beneficio del software creado y sobre todo en el usuario final.

No se debería caer tampoco en ningún falso prejuicio, como pudiera ser que para desarrollar software de poca envergadura se podría evitar el proceso de calidad, y que este es únicamente para proyectos de gran volumen. Desde nuestro punto de vista la calidad no es ni será discutible, independientemente del volumen del trabajo a desarrollar, y que todo el tiempo que se emplee en optimizar y mejorar los procesos de evaluación de calidad es tiempo que a medio y largo plazo supondrá un beneficio a la propia empresa que lo desarrolla. Se trata de una filosofía que debe ser impuesta como un canon de funcionamiento obligatorio.

Es por ello, que si procediésemos a analizar la evolución de los sistemas de calidad del software existirán multitud de metodologías y procesos, sin embargo, y haciendo un pequeño resumen de aquellos más significativos, no deberíamos olvidarnos de estudiar los siguientes métodos y filosofías: CMM, CMMI, AENOR e ISO-IEC, de las cuales se hará un pequeño resumen, análisis y evolución histórica.

3. Modelos de Calidad del Software

3.1 Introducción

A continuación, se mencionarán una serie de normas y modelos utilizados como referencia para la valoración de la calidad de desarrollo del software.

Algunos de los principales modelos serán: CMM, CMMI, AENOR e ISO/IEC. Existen muchísimos más, pero simplemente se enumerarán algunos de los más conocidos y utilizados, haciendo a su vez, una breve

explicación de cada uno de ellos, para finalmente, en capítulos posteriores, centrarse en el modelo ISO que es el objeto de estudio del presente Trabajo Fin de Carrera.

Una de las cuestiones que deberíamos tener en cuenta es que un proceso que no actúa ni valora la calidad en su proceso de generación de software tiene una gran probabilidad de concebir un producto carente de calidad, así pues, se podrá decir, que la calidad en el proceso de desarrollo de software es directamente proporcional a la calidad del producto generado.

Un ejemplo muy gráfico de cómo se ha evolucionado en el desarrollo del software es el informe *Caos Report* realizado por *Standish Group*.

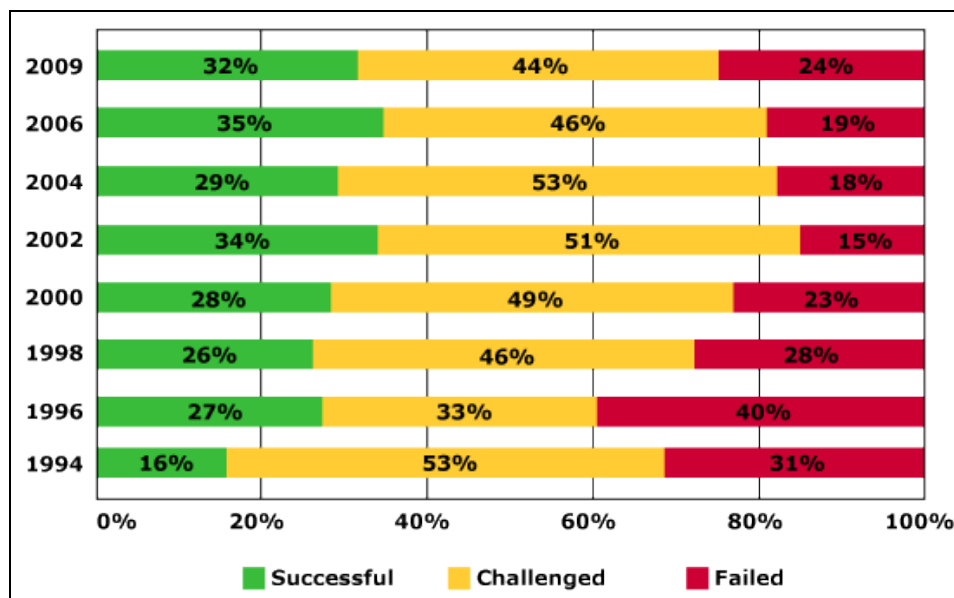


Figura 6. Gráfico de los resultados del informe *Caos Report* en diversos años.

En 1994, tal y como se observa en el gráfico, los resultados del estudio realizado a miles de proyectos de desarrollo software, y los resultados obtenidos eran los siguientes:

- El 31% de los proyectos fueron cancelados durante alguna de las fases de creación.
- Un 53% fueron terminados fuera del tiempo estimado y además con un sobrecoste que en algunos casos llegó al 189%.
- Y un 16% de los proyectos se realizaron según la planificación propuesta.

El estudio *Caos Report* se ha ido realizando paulatinamente durante los años 1996, 1998, 2000, ..., y diez años más tarde, en 2004, los resultados obtenidos fueron los siguientes:

- El 18% de los proyectos fueron cancelados durante alguna de las fases de creación.
- Un 53% fueron terminados fuera del tiempo estimado y además con un sobrecoste que en algunos casos llegó al 189%.
- Y un 29% de los proyectos se realizaron según la planificación propuesta.

Analizando los datos, podremos ver como se duplicó el número de proyectos finalizados según la planificación original, un dato positivo, y que los proyectos que se descartaron o no llegaron a su fin fueron la mitad que diez años atrás. Por último, se mantuvo prácticamente constante el número de proyectos que se terminaron fuera de coste y/o sufrieron algún sobrecoste.

Aparentemente, y viendo que se finalizaron en tiempo y forma el doble de proyectos, los datos parecen sustancialmente mejores, sin embargo, si consideramos que la suma de los proyectos finalizados fuera del tiempo programado y los no finalizados alcanzaban un 71% de todos los proyectos, no se pueden considerar en absoluto unos datos mínimamente aceptables.

Según nos vamos acercamos a la actualidad, y tal y como nos muestran los gráficos que aparecen en la web <http://www.evergreenpm.com/standish-group-2015-chaos-report/>, podremos ver los datos correspondientes entre los años 2011 al 2015, ambos inclusive. Los valores se han mantenido durante estos años prácticamente constantes, obteniendo en 2015 los siguientes resultados:

- El 19 % de los proyectos fueron cancelados durante alguna de las fases de creación.
- Un 52 % fueron terminados fuera del tiempo estimado y además con un sobrecoste que en algunos casos llegó al 189%.
- Y un 29% de los proyectos se realizaron según la planificación propuesta.

Esto nos llevaría a realizar dos conclusiones:

1. Los datos desde el año 2004 hasta la actualidad se han mantenido constantes, con solamente un 29% de proyectos exitosos.

2. ¿Qué está fallando y qué se debe mejorar para revertir esta tendencia?

Pues es aquí donde se encuentra la fundamentación al presente Trabajo Fin de Carrera, que no es otro que una vez se analicen los datos, se realizarán aportaciones que nos lleven a la mejora de la calidad del desarrollo del software.

3.2 Modelos de Calidad del Software

3.2.1 Modelo CMM

El problema de la falta de agilidad y destreza en la generación de software implicará tener que buscar metodologías que mejoren cada fase del trabajo de una manera evidente. Por este motivo surgieron metodologías como *CMM*, *Capability Maturity Model for Software*, es decir, *Modelo de Madurez de Capacidad*, que ayudan a la mejora del control sobre estos procesos. *CMM* aún está en vías desarrollo, y es un estándar que la industria acepta para evaluar y garantizar la calidad y madurez de sus aplicaciones.

Esta metodología, creada por *Software Engineering Institute (SEI)*, tiene como objetivo describir los elementos principales para llevar a cabo los procesos de software de una forma efectiva, proponiendo actuar a nivel de planificación, ingeniería y administración del desarrollo y mantenimiento de software. Siguiendo estas prácticas aumentará la habilidad con la que una organización podrá maximizar las metas en aspectos como el coste, la programación, la funcionalidad y calidad de producto. Se tratará de visualizar cuales son los aspectos prioritarios del proceso para mejorar con ello la calidad. *CMM* es un método aplicado a casos reales, y además tiene a su favor que está documentado y tiene acceso público.

CMM se desarrolló en cinco etapas evolutivas hasta llegar a la implementación de prácticas de calidad. Durante estos pasos *CMM* se utiliza como una destreza para conseguir mejores resultados, además, consigue detectar las deficiencias organizativas, y también ser un modelo a seguir para evolucionar hacia la eficiencia y la calidad. *CMM* es un método de aplicación gradual, con un continuo potencial de crecimiento.

Los niveles del *CMM* son cinco, y están planteados para ayudar a una organización a mejorar sus procesos y conseguir reducir los costes. Una evolución, pasando del nivel 1 al nivel 3, puede reducir los costes económicos entre un 50-60%, justificando claramente la dedicación temporal que exige.

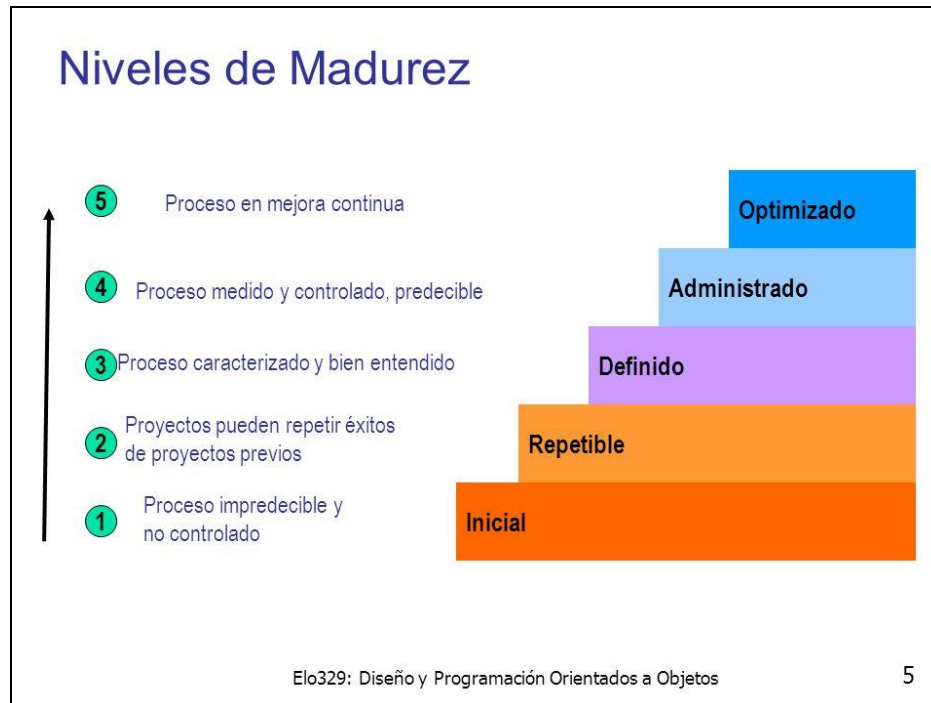


Figura 7. Gráfico de los niveles del modelo CMM.

Vamos a enumerar de manera resumida los cinco niveles del CMM, que son:

1 – Inicial. Se encuentran en este nivel aquellas organizaciones que no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Existe falta de planificación, y ello conllevará a retrasos temporales y sobrecostos económicos.

2 – Repetible. En este nivel, en las organizaciones existe algún tipo de práctica en la gestión de proyectos, con métricas básicas y un razonable seguimiento de la calidad.

3 – Definido. Localizaremos una buena gestión de proyectos, con buenos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos.

4 – Gestionado. Alcanzan este nivel las organizaciones que disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.

5 – Optimizado. La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Beneficios de la implantación del modelo CMM

- Alta efectividad para detectar errores durante el ciclo de vida del desarrollo del software, consiguiendo grandes resultados en la disminución de las deficiencias y del riesgo. Consigue un beneficio en los plazos de respuesta al cliente.
- Incrementa la comunicación en las organizaciones y la comunicación entre sus grupos.
- Disminución de los desfases temporales y sobrecostes económicos de los proyectos.
- Gran tolerancia para los cambios, incrementos, adopción y adaptación a las tecnologías actuales.

Implementación en la Organización

Una organización que decida implementar el modelo *CMM*, dará un paso tanto en la calidad de la organización como en la implicación de un proceso mejora continuado, evolucionando hacia una empresa flexible con la integración de nuevas metodologías.

3.2.2 Modelo CMMI

CMMI (Capability Maturity Model Integration), Integración de Modelos de Madurez de las Capacidades, es una evolución de la metodología *CMM*, y surgió como un síntoma que demuestra que cada vez más organizaciones están interesadas en implantar modelos y procesos de calidad en el desarrollo del software cada vez mejores.

La finalidad principal de *CMMI* es mejorar y evaluar la madurez de los procesos de desarrollo de software, de tal manera que su previsión, planificación y presupuesto fuesen mucho más predecibles, con lo que se reducirían los tiempos y los sobrecostes futuros.

CMMI integra diferentes modelos de calidad como:

- Capability Maturity Model for Software.
- Electronic Industries Alliance Interim Standard.
- Integrated Product Development Capability Maturity Model.

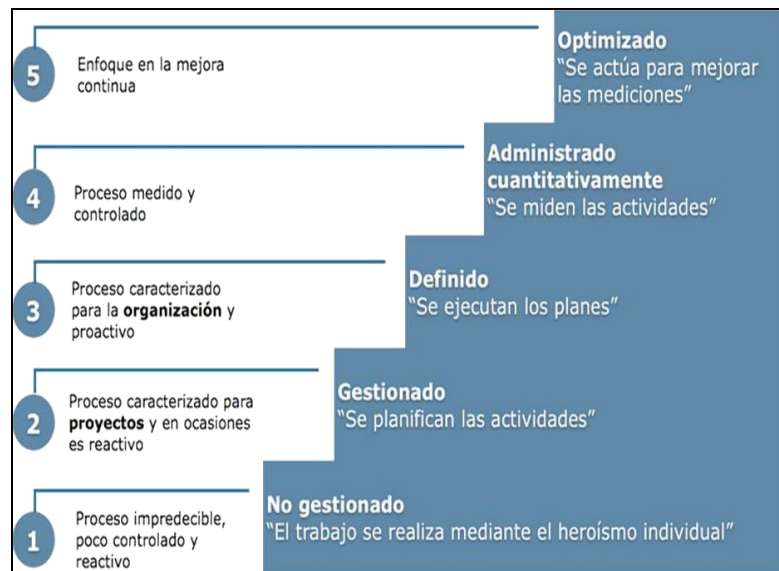


Figura 8. Gráfico de los niveles del modelo CMMI.

Esta metodología puede ser aplicada a diferentes disciplinas, pero este estudio se centrará en lo relativo al desarrollo de software. Para ello *CMMI* plantea cinco niveles de madurez en las organizaciones:

- **Inicial** – En este nivel las organizaciones carecen de procedimientos y metodologías de evaluación. Estas organizaciones tienden a no cumplir sus compromisos, con una probabilidad casi absoluta al abandono del proceso.
- **Gestionado** – Existen un conjunto de buenas prácticas basadas en la experiencia adquirida. Se encuentran definidos los objetivos en la evaluación de los trabajos.
- **Definido** – Toda la organización se involucra totalmente en el análisis y en la evaluación de la calidad del proceso de desarrollo del software. Existe una metodología perfectamente definida y documentada.
- **Cuantitativamente Gestionado** – La evaluación de los proyectos se podrá realizar mediante valores numéricos. Todas estas valoraciones quedarán almacenadas y servirán de ayuda para los futuros proyectos de la organización.
- **Optimizado** – Utilizando todos los datos almacenados, las evaluaciones y valoraciones obtenidas a lo largo del tiempo, se podrán estimar los futuros sobrecostos y las posibles desviaciones temporales que afecten al proyecto, con el fin de minimizar sus impactos y maximizar y optimizar el trabajo.

Estos niveles de madurez *CMMI* se pueden representar en dos modalidades: por etapas (*staged*) y continuo (*continuous*), según el objetivo que la organización se plantee, y aunque la forma de presentarlo y organizarlo es independiente en cada caso, el contenido será similar en ambos casos.

Conclusiones

- Resulta fundamental que toda organización adquiera a la mayor brevedad un procedimiento, con un nivel mínimo 2 de madurez, con unos objetivos definidos y haciendo uso de las experiencias adquiridas, esto ayudará a poder alcanzar el nivel 3 de madurez.
- El personal de la organización deberá asumir la disciplina y sentirse partícipe del procedimiento a seguir, como una parte más del trabajo, esto evitaría el miedo a los cambios y el tener que hacer una imposición de las jerarquías superiores.
- Las metodologías *CMM* e *CMMI* tienen una base común muy similar, de hecho, se puede definir la metodología *CMMI* como una evolución de *CMM*.

3.2.3 AENOR

Historia

La *Asociación Española de Normalización y Certificación (AENOR)*, es una entidad privada sin fines lucrativos creada en 1986. Su actividad contribuye a mejorar la calidad y competitividad de las empresas, sus productos y servicios, mediante el desarrollo de normas técnicas y certificaciones.

Es el organismo legalmente responsable del desarrollo y difusión de las normas técnicas en España. Estas normas indican cómo debería ser un producto o cómo debería funcionar un servicio para que sea seguro y responda a lo que el consumidor espera de él.

AENOR comenzó en 1989 a certificar sistemas de gestión, según la *norma UNE-EN ISO 9001*, y tiene a su disposición uno amplio catálogo, con más de 28.900 documentos normativos que contienen soluciones eficaces. Esto implicará que los certificados emitidos por *AENOR* se consideren de un alto valor, tanto en España como a nivel internacional,

habiendo emitido certificados en más de 60 países. *AENOR* se sitúa entre las 10 certificadoras más importantes del mundo.

El objetivo de *AENOR* es difundir la cultura de la calidad entre el tejido productivo español para así mejorar su competitividad. Actualmente, son más de 200 los comités técnicos de normalización en los que participan cerca de 6.000 expertos. Su trabajo es reconocido internacionalmente, como lo demuestra que cada vez sean más las normas desarrolladas en España que sirven de referencia en la elaboración de normas europeas e internacionales.

En la década de los 90 esta certificación era fundamentalmente demandada por las organizaciones puramente industriales, pero a partir del año 2000, con la publicación ese año de una nueva versión de la *Norma ISO 9001*, se abre el abanico a las organizaciones de servicios y a las pymes.

Actualmente, treinta años después de su creación, *AENOR* cuenta con 200 Comités Técnicos de Normalización, el catálogo de normas técnicas español supera las 28.000 normas, más de 90.000 productos exhiben la Marca N y se han emitido más de 60.000 certificados de sistemas de gestión. Con todos estos datos, *AENOR* se sitúa entre las 10 certificadoras más importantes del mundo.

Una vez vistas las metodologías *CMM*, *CMMI* y *AENOR*, nos quedaría hacer la presentación de la *metodología ISO*, que será la metodología sobre la que gira el presente Trabajo Fin de Carrera, la cual vamos a estudiar a continuación.

3.3 Metodología ISO

ISO (International Organization for Standardization), Organización Internacional de Normalización, es una organización internacional independiente, no gubernamental, de la que forman parte actualmente 163 organismos nacionales de normalización.

Esta compuesta por un miembro por país, y su sede y la Secretaría Central se encuentran en Ginebra (Suiza) que es el lugar desde el cual se coordina el sistema.

Está compuesta por delegaciones gubernamentales y no gubernamentales, subdivididos en una serie de subcomités encargados de desarrollar las guías que contribuirán al mejoramiento.

Se trata de la organización que se encarga de promover el desarrollo de normas internacionales de fabricación, bien sean productos o servicios, comercio y comunicación para todas las ramas industriales.

Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones, públicas o privadas, a nivel internacional.

A través de sus miembros, que reúne a expertos para compartir conocimientos y desarrollar estrategias basadas en el consenso, el mercado Normas Internacionales voluntarias y relevantes que apoyan la innovación y aportar soluciones a los retos globales.

Las normas desarrolladas por *ISO* son voluntarias, comprendiendo que *ISO* es un organismo no gubernamental y no depende de ningún otro organismo internacional, por lo tanto, no tiene autoridad para imponer sus normas a ningún país.

El contenido de los estándares está protegido por derechos de copyright y para acceder a ellos el público corriente debe comprar cada documento.

La Organización está compuesta por representantes de los organismos de normalización nacionales (ON), que produce diferentes normas internacionales industriales y comerciales.

Dichas normas se conocen como “*normas ISO*” y su finalidad es la de coordinar las normas nacionales, en consonancia con el Acta Final de la Organización Mundial del Comercio, con el propósito de facilitar el comercio, el intercambio de información y contribuir con normas comunes al desarrollo y a la transferencia de tecnologías.

La organización *ISO* está compuesta por tres tipos de miembros:

- Miembros simples, uno por país, recayendo la representación en el organismo nacional más representativo.
- Miembros correspondientes de los organismos de países en vías de desarrollo y que todavía no poseen un comité nacional de normalización. No toman parte activa en el proceso de normalización pero están puntualmente informados acerca de los trabajos que les interesen.
- Miembros suscritos, de países con reducidas economías a los que se les exige el pago de tasas menores que a los correspondientes.

Normalización

Según la Real Academia Española, se define normalizar como: “*tipificar, ajustar a un tipo o norma*”. Podemos por tanto, definir la *normalización*

como el “proceso de elaborar, aplicar y mejorar las normas que se aplican a distintas actividades científicas, industriales o económicas con el fin de ordenarlas y mejorarlas”.

Según ISO, la normalización es “la actividad que tiene por objeto establecer, ante problemas reales o potenciales, disposiciones destinadas a usos comunes y repetidos, con el fin de obtener un nivel de ordenamiento óptimo en un contexto dado, que puede ser tecnológico, político o económico”.

La normalización persigue fundamentalmente tres objetivos:

- **Simplificación**: con el fin de disminuir el número de modelos, con la idea de seleccionar aquel o aquellos más necesarios.
- **Unificación**: con objetivo de unirlos para facilitar el intercambio a nivel internacional.
- **Especificación**: con la idea de alejar los fallos y errores en la identificación, lo cual llevará a general un lenguaje mucho más claro y preciso.

Un gran número de los considerados países desarrollados emplean grandes inversiones económicas en organismos de normalización, tanto nacionales como internacionales. Esto es un claro síntoma de que se comienza a dar importancia y valorar como se merece a la normalización.

A continuación se van a enumerar y desglosar las principales normas ISO en vigor y que afectan directamente al desarrollo software de calidad.

3.3.1 Normas ISO 9000, ISO 9001 y 9004

Todo el compendio de Normas ISO 9000 surgieron con el objetivo de asistir a todo tipo de organizaciones que deseen instaurar sistemas de gestión de la calidad eficaces.

La Norma ISO 9000 describe los fundamentos de los sistemas de gestión de la calidad y especifica la terminología para los sistemas de gestión de la calidad.

Por otro lado, ISO 9001, recoge las exigencias necesarias para los sistemas de gestión de calidad para aquellas organizaciones que quieren disponer de un proceso adecuado para la generación de

soluciones de eficiencia, con lo cual ofrecerán productos de calidad al cliente final.

La Norma *ISO 9004* incluye pautas para proporcionar eficacia y eficiencia del sistema de gestión de la calidad. El objetivo final de la norma será mejorar los procesos de la organización y la satisfacción de los usuarios finales.

Como se podrá comprobar, todas estas normas inciden en conceptos como calidad, eficiencia y mejora en los procesos, tanto dentro de las organizaciones, como para los usuarios finales.

A continuación se verán otras normas *ISO*, cada vez más específicas y centradas ya en el desarrollo del software.

3.3.2 ISO/IEC 9126

La norma *ISO 9126* se define como un estándar internacional, se publicó en 1992, y define los propósitos para la evaluación de la calidad de software, como la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software.

Este estándar está fraccionado en cuatro partes, las cuales se encargarán de dirigir, las métricas externas, las métricas internas, la calidad en las métricas de uso y expendido.

Los modelos de calidad para el software se describen así:

- **Calidad interna y externa:** Pormenoriza los detalles mediante la aplicación de 6 características para calidad interna y externa. Cuando el software se utiliza como una parte de un sistema informático global, estas divisiones se aprecian externamente, y se presentan como el resultado de atributos internos de software.
- **Calidad en uso:** es el resultado final que contemplará el cliente, después de aplicar las 6 características de la calidad interna y externa del software. También se especifican 4 características para la calidad en uso.

Al agrupar calidad interna, calidad externa y la calidad en uso se conseguirá un modelo para la evaluación mucho más robusto y eficaz.

Pero no se debería equiparar la usabilidad del modelo de calidad externa e interna con el modelo de calidad en uso, ya que, la usabilidad es la forma como los profesionales entienden la funcionabilidad del software, mientras que la calidad en uso se podrá interpretar como la forma en la cual lo entiende el usuario final.

Al fusionarse los dos modelos, el resultado será que se mantendrán los seis indicadores del primer modelo como atributos, y en cuanto al modelo de calidad, sus 4 indicadores pasarán a actuar como sus atributos.

Se establecerán categorías para las cualidades de la calidad externa e interna, así como para la calidad en uso del software. Si se tienen en cuenta los 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), los cuales se desglosarían a la vez en varios indicadores; estos se podrían medir por métrica interna o externa.

Las definiciones se verían para cada característica y subcaracterística de calidad del software que influye en la calidad. Por cada característica y subcaracterística, la capacidad del software se determinaría por el total de los atributos internos que pueden ser medidos. Idénticamente, las características y subcaracterísticas se podrán medir externamente por la capacidad del sistema que contiene el software.

El estándar describe 6 características generales, las cuales se detallarán a continuación:

- Funcionalidad.
- Confiabilidad.
- Usabilidad.
- Eficiencia.
- Mantenibilidad.
- Portabilidad.

○ **Funcionalidad.**

Es la capacidad que tiene el software para cumplir y dotar de las funciones necesarias para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. Incluye las siguientes subcaracterísticas:

- Adecuación. Propiedades que tiene el software y están relacionadas con la presencia y aptitud de un conjunto de funciones para tareas especificadas.
- Exactitud. Atributos del software relacionados con la disposición de resultados o correctos.

- Interoperabilidad. Atributos del software que se relacionan con su pericia para la interacción con sistemas especificados.
- Seguridad. Propiedades del software relacionadas con su capacidad para advertir y evitar acceso no autorizados, deliberados o fortuitos, a aplicaciones o bases de datos.
- Cumplimiento funcional.

○ **Fiabilidad.**

Es el conjunto de atributos que tienen relación con la capacidad del software para mantener su nivel de prestación durante un período establecido.

Sus subcaracterísticas son:

- Madurez. Capacidad que presentará el sistema y que conseguirá disminuir la probabilidad de sufrir fallos.
- Recuperabilidad. Atributos relacionados con la magnitud de restablecer el nivel de continuar trabajando y recuperación del estado y de los datos afectados en caso de avería.
- Tolerancia a fallos. Capacidad de mantener una cota de garantía y estabilidad adecuada en el caso de sufrir inconvenientes no esperados.
- Cumplimiento de Fiabilidad. Posibilidad de aglutinar normativas y legislación que tengan como finalidad garantizar la fiabilidad del sistema.

○ **Usabilidad.**

Es el conjunto de propiedades de un software, que facilita a todos los roles de usuario final, poder trabajar y gestionarlo con la suficiente destreza e intuición.

Sus subcaracterísticas son:

- Aprendizaje. Cuantifica la cantidad de tiempo necesaria que necesita un usuario para poder conocer con cierta soltura potencial de las funciones y recursos que ofrece el software.

- Comprensión. Mide el nivel de rapidez y facilidad con el que el los usuarios pueden conocer el potencial de las funciones del software.
- Operatividad. Facilidad con la que el usuario puede realizar las operaciones que desea en cada momento para ejecutar una acción.
- Atractividad. Valora la presentación gráfica del software, así como la muestra de las acciones.

○ **Eficiencia.**

Se refiere a la valoración y cuantificación entre la cantidad de recursos que ofrece la aplicación, y que puede ser en relación con el tiempo de resolución de las operaciones, como por el número de recursos que ofrece.

Existen dos alternativas:

- Comportamiento en el tiempo. Cantidad de tiempo de respuesta y procesamiento necesarios para ejecutar una acción con el software.
- Comportamiento de recursos. Cantidad y número de tipos de recursos necesarios por la aplicación para resolver determinadas tareas.

○ **Mantenibilidad.**

Magnitud que cuantifica la facilidad que presenta la aplicación para ser ampliada, alterada o depurar posibles fallos que se localicen.

Sus subcaracterísticas serían:

- Estabilidad. Fortaleza con la que el sistema responde ante posibles riesgos o situaciones inesperadas.
- Facilidad de análisis. Rapidez del sistema para diagnosticar y mostrar los resultados de posibles fallos o circunstancias excepcionales.
- Facilidad de cambio. Versatilidad con la que el software puede ser modificado por cuestiones de actualizaciones o imprevistos sufridos.

- Facilidad de pruebas. Medida con la que se valora la simplicidad con la que la aplicación ejecuta distintas funciones.

- **Portabilidad.**

Es la capacidad y potencia de la que dispone una aplicación o sistema software para poder ser trasladado de unas plataformas a otras.

- Capacidad de instalación. Esfuerzo que se necesita realizar para proceder a la instalación de una aplicación a un entorno concreto.
- Capacidad de reemplazamiento. Esfuerzo necesario para trabajar con una aplicación diferente a otro software previo en un entorno concreto.

- **Calidad de uso.**

Son el conjunto de propiedades que envuelven a la seguridad del sistema y la aprobación por el cliente del software creado.

- Eficacia. Capacidad del software para resolver los procesos que el cliente le encarga.
- Productividad. Cuantificación del rendimiento con el que el sistema resuelve las labores que le encomienda el usuario.
- Seguridad. Capacidad de evitar riesgos e intrusiones, voluntarias o involuntarias, de acceso no autorizado al sistema.
- Satisfacción. Grado de conformidad que presenta el cliente del software encargado en origen y del producto obtenido.

El atributo “*Conformidad*” no se detallada anteriormente porque se especificó en todas las características.

Se ha procedido a detallar cada una de las subcaracterísticas en un conjunto de atributos. Un atributo se puede definir como un concepto que se puede evaluar y calcular en la aplicación desarrollada. Estos atributos variarán entre cada uno de los productos software.

Un proyecto software englobará documentación muy variada, como archivos ejecutables, código fuente, descripciones de arquitectura, y demás. Todo ello afectará a lo que podremos considerar también usuarios de roles como operadores y programadores, quienes harán uso y reutilizarán partes de código que se encuentran disponibles en las bibliotecas software.

Las “*métricas internas*”, son aquellas que son independientes a la ejecución del software, y las “*métricas externas*”, por el contrario, son las que se aplicarán al programa que está en funcionamiento.

Para un correcto uso, la calidad en las métricas, solo estarán realmente disponibles, cuando el software obtenido se prueba en situaciones reales. Del mismo modo, la calidad interna no implicará calidad externa.

3.3.3 ISO/IEC 12207

La norma *ISO 12207* es una metodología que se utiliza para la gestión del ciclo de vida del software. Este ciclo de vida del software se define como el proceso mediante el cual, en función de una serie de entradas, que se procesarán, se obtendrán unas salidas.

ISO 12207 se trata de un estándar elaborado de una manera flexible y modular, de tal modo que pueda adaptarse al usuario que lo utilice.

El *estándar 12207* se apoya en dos pilares principales: modularidad y responsabilidad. La modularidad es la capacidad de que el sistema tenga el máximo de afinidad y empaque con el menor esfuerzo posible. La responsabilidad tratará de que en cada etapa del proceso exista un encargado, de tal modo que en la organización, el trabajo final obtenido, se vea representado en cada fase por un responsable del desarrollo software, dando de este modo cohesión y unidad al proyecto global.

Objetivos

El objetivo principal sería que *ISO 12207* se pudiese instaurar en una organización de una manera natural, de acuerdo al grado de madurez que cada organización disponga, así como disponer de unos cimientos adecuados, y que ellos redunden en mejoras en los procesos del desarrollo software.

Por tanto, se pudiera decir, que el propósito de *ISO 12207*, fuese el de establecer un marco con el cual poder desarrollar un software de una calidad adecuada, aplicando para ello, metodologías de desarrollo de software y metodologías de mantenimiento, lo cual provocaría una evidente mejora en la calidad del producto, siempre bajo un estricto control en los proyectos.

Por otro lado, *ISO 12207*, tiene claro que persigue lo que se acaba de describir, sin embargo, no dice cómo se debe hacer, ni cuales son las herramientas a utilizar, ni a qué lenguajes o entornos afecta, ni tampoco es un estándar que aplica métricas de valoración de la calidad del software.

Procesos

Los procesos se clasifican en tres tipos:

- Procesos principales.
- Procesos de soporte.
- Procesos de la organización.

Los *procesos principales* no se aplicarán a nivel global, sino con total independencia del proyecto software que se desee desarrollar.

Sin embargo, los *procesos de soporte y de organización*, deben guiarse por una metodología impuesta, genérica, sin tener ninguna vinculación con la aplicación a desarrollar o de la propia empresa que lo realiza.

Los *procesos principales* implican:

- Adquisición.
- Suministro.
- Desarrollo.
- Operación.
- Mantenimiento.

Los procesos de soporte:

- Documentación
- Gestión de la configuración.
- Aseguramiento de calidad.
- Verificación.
- Validación.
- Revisión conjunta.

- Auditoría.
- Resolución de problemas.

Y los procesos de la organización:

- 1- Gestión.
- 2- Infraestructura.
- 3- Mejora.
- 4- Recursos Humanos.

Resumen

En definitiva, el método *ISO 12207*, nos dice *qué* es lo que deberíamos intentar conseguir, pero no nos dice *cómo*, lo cual por una parte ayudaría, pero sin embargo parece una metodología que estudiada de una manera global parece estar coja.

3.3.4 ISO/IEC 15288

ISO 15288 crea un marco común para detallar el modo de ejecución de los procesos de descripción del ciclo de vida para la generación de desarrollo de software a medida. La planificación inicial de la norma ISO comenzó en 1994, cuando se reconoció la necesidad de un marco común de proceso de Ingeniería de Sistemas.

Estos procesos se pueden aplicar en cualquier nivel de la jerarquía, y en cualquiera de las fases de gestión y elaboración del diseño de un sistema. Para ello, cada parte o sección de la organización se deberá implicar, con la intención final de alcanzar una aplicación de total aceptación por parte del usuario final.

ISO 15288, también aportará una serie de indicadores orientados a la organización, para ser aplicados en cada etapa del proceso de vida del desarrollo del software, para su mejora y eficiencia.

ISO 15288 está claramente enfocado para el desarrollo de software a medida, tanto a nivel hardware, software, bases de datos, etc.

La norma divide los procesos en cuatro categorías:

- Técnico.
- Proyecto.
- Acuerdo.
- Empresa.

Cada proceso persigue un objetivo, un resultado y una actividad.

ISO 15288 abarca en total 25 procesos diferentes, que recogen 123 resultados derivados de 403 actividades.

Entre otras, se verán afectados apartados como los procesos de diseño, de aplicación, de análisis de requerimientos, procesos de verificación y validación, etc.

3.3.5 ISO/IEC 15504

Es conocido también como “*Software Process Improvement Capability Determination*”, *SPICE*, “*Determinación de la Capacidad de Mejora del Proceso de Software*”, y es un modelo para la mejora, evaluación de los procesos de desarrollo, mantenimiento de sistemas de información y productos de software.

Se trata de una norma de la cual no se ha realizado una implantación adecuada, en relación con su importancia en el área de la evaluación de la calidad del desarrollo del software.

El *proyecto SPICE* se generó mediante el Comité Internacional de estándares de Ingeniería de Software y Sistemas, con la intención de generar un modelo que pudiera evaluar los desarrollos software mediante la creación de un modelo que aglutinaba lo que se consideraba lo mejor de cada método de evaluación hasta la fecha existente.

La *ISO 15504* también se utilizará para poder optimizar la madurez de los procesos que se utilizan en las organizaciones *ISO 15504*, genéricamente, tiene el potencial de poder evaluar de una manera global cualquier modelo de procesos, no solo de software.

Los objetivos principales de *SPICE* son tres:

- Generar una guía de aplicación para la evaluación de procesos de desarrollo de software.
- Elaborar pruebas prácticas de dicha norma.

- Promover la transferencia de tecnología de la evaluación de procesos de software a la industria del software a nivel mundial.

Se trata de una norma que reúne las buenas prácticas que permitirán generar un procedimiento práctico destinado a las organizaciones dedicadas a la generación de software con calidad.

Cuando se haga uso de la norma *ISO 15504*, se deberá elegir un modelo de procesos. Por defecto, la *ISO 15504* propone algunos modelos de cómo ser aplicada con el modelo de procesos *ISO 12207*.

Dimensiones de ISO 15504.

Se trata de una metodología que está basada en dos dimensiones: *de proceso* y *de capacidad de proceso*.

Define que todo modelo de evaluación de procesos debe definir:

- la dimensión de procesos: el modelo de procesos de referencia (dimensión de las abscisas).
- la dimensión de la capacidad: niveles de capacidad y atributos de los procesos.

Los niveles de capacidad para todo modelo de evaluación de procesos podrá tener desde el 0 y al menos hasta el nivel 1 de los siguientes niveles de capacidad estándar:

- Nivel 0: Incompleto.
- Nivel 1: Realizado.
- Nivel 2: Gestionado.
- Nivel 3: Establecido.
- Nivel 4: Predecible
- Nivel 5: En optimización.

Para cada nivel existirán unos atributos de procesos estándar que ayudan a evaluar los niveles de capacidad.

Las características del modelo ISO 15504 son:

- Establecer un entorno, con una serie de requerimientos, útiles en cualquier etapa del proceso de evaluación del proceso de desarrollo de software.
- Ofrece formalidades para aplicar en las organizaciones y sus modelos utilizados en la evaluación de organizaciones.
- Proporcionar guías para la definir las competencias del encargado de realizar el proceso de evaluación, el cual estará compuesto de 10 partes, de las cuales, de la 1 hasta la 7 se encontrarán finalizadas, y el resto, desde la 8 hasta la 10, actualmente se están desarrollando.
- *ISO 15504*, dispone de métodos tanto para evaluar los procesos, como para su mejora y determinación de la capacidad.
- Presenta, en su apartado 5^o, modelos de evaluación de procesos y sus fases del ciclo de vida, ya definidos en el estándar *ISO/IEC 12207*, con las fases del ciclo de vida del desarrollo, mantenimiento y operación de los proyectos software.
- Proporciona, en su parte 6, un el procedimiento de evaluación de procesos para el ciclo de vida, definidos en el estándar *ISO/IEC 15288*.
- Aportará, en su sección 8^a, un modelo de evaluación de procesos para los procesos de servicios *TIC*, definidos en el estándar *ISO/IEC 20000*.
- *ISO* forma parte del modelo *CMMI*, siendo ambos totalmente compatibles con *15504*.

3.3.6 ISO 20000-1

La norma *ISO 20000-1* es utilizada para la certificación de los sistemas para la gestión de servicios de *TI (tecnología de la información)*, estando reconocida a nivel internacional.

ISO 20000-1 incluye también el denominado Código de Buenas Prácticas para la gestión de servicios de *TI* y ofrecer un servicio de calidad.

Entre otras aportaciones, incluye las actuaciones necesarias para configurar y gestionar los procesos que recogen algún tipo de problema

o anomalía, con la finalidad de realizar las mejoras oportunas en los servicios de TI.

Alineación con otras normas de sistemas de gestión

La *ISO 20000* se complementa con otras normativas del ámbito de los sistemas de gestión, lo cual supone un fundamento de fortaleza que ayuda en la implementación y funcionamiento en las organizaciones que harán uso de aplicaciones integradas.

Esto genera las siguientes consecuencias:

- Conjunción con otro tipo de normativas dentro del campo de los sistemas de gestión, como es la *ISO 9001*.
- Búsqueda continua de realización de mejoras en los procesos de gestión de servicios *TI*.
- Especificación de los requisitos mínimos en la planificación, documentación y registros.

Una buena aplicación en las *TI* repercutirá en un beneficio de la propia organización, consiguiendo una mejora de ingresos, reducción de costes y mejorar la imagen de la empresa.

Resumiendo, un estándar para la obtención de una valoración de una alta calidad en gestión de servicios de *TI* resulta indispensable para alcanzar el éxito en una organización de desarrollo del software, algo a lo que se puede llegar con un informe favorable *ISO 20000*.

3.3.7 ISO 25000-1

La norma *ISO 25000-1*, es la más reciente de todas las normativas *ISO* que afectan al desarrollo del software de calidad.

Calidad del software y calidad en el proceso de su desarrollo, deben ir acompañados en cualquier organización que actualmente aspire a unos estándares de calidad óptimos.

Es por ello que surge *ISO/IEC 25000*, como un nuevo compendio de actuaciones, de lo que se ha denominado *Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE - System and Software Quality Requirements and Evaluation)*.

ISO/IEC 25000 surge de la agrupación de actuaciones recogidas en las normativas *ISO/IEC 9126* y *ISO/IEC 14598*, con el principal objetivo de guiar el desarrollo de los productos de software.

La norma ISO 25000 la conforman varias divisiones:

▪ **ISO/IEC 2500n – División de Gestión de Calidad**

Todas las normas incluidas en este apartado tratarán de definir los modelos, terminología y aspectos que son comunes por todas demás normas de la familia 25000.

Esta división se encuentra formada por:

- *ISO/IEC 25000 - Guide to SQuaRE*: contiene el modelo de la arquitectura de SQuaRE, la terminología de la familia, un resumen de las partes, los usuarios previstos y las partes asociadas, así como los modelos de referencia.
- *ISO/IEC 25001 - Planning and Management*: recoge los requisitos, consejos y orientaciones para gestionar la evaluación y especificación de los requisitos del producto software.

▪ **ISO/IEC 2501n – División de Modelo de Calidad**

Este apartado recoge los modelos para la calidad interna, externa y en uso del producto software.

Esta división se encuentra formada por:

- *ISO/IEC 25010 - System and software quality models*: hace la descripción del modelo de calidad para los productos de software y para la calidad en uso.
- *ISO/IEC 25012 - Data Quality model*: define el modelo general para evaluar la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.

▪ ISO/IEC 2502n – División de Medición de Calidad

Este apartado incluye los modelos de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación.

La actual división se encuentra formada por:

- ISO/IEC 25020 - *Measurement reference model and guide*: presenta una explicación introductoria y un modelo de referencia común a los elementos de medición de la calidad. También proporciona una guía para que los usuarios seleccionen o desarrollen y apliquen medidas propuestas por normas ISO.
- ISO/IEC 25021 - *Quality measure elements*: define y especifica un conjunto recomendado de métricas base y derivadas que puedan ser usadas a lo largo de todo el ciclo de vida del desarrollo software.
- ISO/IEC 25022 - *Measurement of quality in use*: define específicamente las métricas para realizar la medición de la calidad en uso del producto.
- ISO/IEC 25023 - *Measurement of system and software product quality*: define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software.
- ISO/IEC 25024 - *Measurement of data quality*: define específicamente las métricas para realizar la medición de la calidad de datos.

▪ ISO/IEC 2503n – División de Requisitos de Calidad

Las normas que forman este apartado ayudarán a especificar requisitos de calidad que puedan ser utilizados en el proceso de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación. Para ello, este apartado se compone de:

- ISO/IEC 25030 - *Quality requirements*: provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.

- **ISO/IEC 2504n – División de Evaluación de Calidad**

Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software. Esta división se encuentra formada por:

- ISO/IEC 25040 - *Evaluation reference model and guide*: propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.
 - ISO/IEC 25041 - *Evaluation guide for developers, acquirers and independent evaluators*: describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.
 - ISO/IEC 25042 - *Evaluation modules*: define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
 - ISO/IEC 25045 - *Evaluation module for recoverability*: define un módulo para la evaluación de la subcaracterística Recuperabilidad (Recoverability).

La división de extensión de SQuaRE (ISO/IEC 25050 a ISO/IEC 25099) quedará reservada para normas o informes técnicos que aborden dominios de aplicación específicos o que puedan ser utilizados para complementar otras normas de la familia SQuaRE.

4. Análisis del Estudio Realizado

Una vez que se han estudiado los principales modelos y criterios para desarrollo de software de calidad: *CMM*, *CMMI*, *AENOR*, *ISO 9000*, *ISO 9001*, *ISO 9004*, *ISO 9126*, *ISO 12207*, *ISO 15504*, *ISO 2000* e *ISO 25000*, se comenzará un análisis de ventajas y fortalezas de cada modelo, así como de sus inconvenientes y debilidades.

4.1 Ventajas e inconvenientes del modelo *CMM*

El modelo *CMM* presenta las siguientes ventajas y fortalezas:

- Mayor efectividad a la hora de detectar errores durante el ciclo de vida del desarrollo del software, consiguiendo disminuir la cantidad de deficiencias.
- Capacidad de minimizar las desviaciones del tiempo de ejecución de los proyectos.
- Mayor tolerancia al cambio e incremento de la capacidad de adopción y adaptación de nuevas Tecnologías.
- Mejora en la rapidez y efectividad de respuesta ante exigencias del negocio.
- Mejora en la colaboración y comunicación.
- Mitigación de Riesgo.
- Reducción de los costes de los proyectos.
- Su implementación en una organización implica interés por el proceso de mejora por parte de la organización.
- Su implantación supone flexibilidad a la hora de integrar nuevos procesos.

Por el contrario, se han localizado los siguientes inconvenientes o debilidades:

- Falta de especificación en el momento de definir los procesos, para guiar las organizaciones y mejorar sus procesos. Orienta sobre *qué* actividades se deben realizar, pero nada sobre *cómo* hacerlo.

- No ofrece un patrón definido sobre las métricas que son adecuadas.
- Implica una labor ingente de burocracia administrativa y papeleo, lo cual da la apariencia de un sistema de control impuesto por la dirección a los trabajadores.

4.2 Ventajas e inconvenientes del modelo *CMMI*

Las ventajas que presenta el modelo *CMMI* son:

- Reducción en los costes de desarrollo.
- Agilidad en la localización y resolución de defectos y errores.
- Mejora la fiabilidad del proceso de planificación, en los aspectos de dedicación y calendario.
- Aumento de la productividad.
- Rebaja en los trabajos de correcciones en las fases de pruebas.
- Incremento de la efectividad en la planificación.
- Mejora de la calidad del producto.
- Minimiza el número de defectos en las fases iniciales del ciclo de vida.
- Mejora de la Imagen de Marca.
- Cuando el proceso se instaura en una organización produce beneficios en la mejora de la visibilidad de los proyectos, de la comunicación, y de la planificación. Todo ello redundará en mejorar la calidad del producto.

Los inconvenientes del modelo *CMMI* son:

- Falta de adecuación en el sector de las *TI*, en todos los ámbitos de la actividad, así como un alto esfuerzo para su implantación.
- Costoso proceso de evaluación por el elevado esfuerzo en tiempo.

- Complejidad de evaluación para mantener los objetivos de madurez.

4.3 Ventajas e inconvenientes del modelo *AENOR*

La aplicación del procedimiento *AENOR* supondrá beneficios como los que a continuación se detallan:

Desde el punto de vista de los usuarios:

- Las normas, al ser un instrumento contrastado, servirán de apoyo a los usuarios finales, a las Administraciones Públicas, y demás, en materia de simplificación reglamentaria.

Desde el enfoque de las empresas:

- Las normas elaboradas en los Comités de Normalización y publicadas, facilitan el acceso a otros mercados, es lo que se viene a denominar *Internacionalización*.
- Mejorar la competitividad, optimizando los recursos que se desean gestionar, lo que se verá reflejado en una minimización en los conflictos relacionados con las organizaciones.
- Supone un manual de buenas prácticas para las empresas, donde se enumeran una serie de criterios reconocidos, que permitirán la obtención de un producto o servicio óptimo.
- Mención aparte merecen las *PyME* (pequeñas y medianas empresas), para quienes este modelo permitirá un incremento en su productividad, posibilitándoles llegar a mercados mucho más amplios. Pudiendo a su vez, participar con propuestas en el proceso de normalización.

Por el contrario, se ha detectado algunos aspectos que pueden ser considerados como negativos para algunas empresas, como es:

- Que a pesar de que las metodologías se crearon con el propósito de mejora, como en todo ámbito de la vida, supone una aceptación por parte de las empresas y sus empleados para poder llevarlas a cabo. Esto también sucede con *AENOR*.

A partir de este momento se procederá al estudio de normativa únicamente de la metodología *ISO*.

Para ello, se agrupará el análisis en tres bloques, uno recogerá los puntos fuertes y débiles de las normativas *ISO 9000*, *9001*, *9004* y *9126*; otro bloque estudiará las normativas *ISO 12207* y *15504*; y por último se estudiarán las normativas *ISO 20000* y *25000*.

4.4 Ventajas e inconvenientes de las normativas ISO 9000, 9001 y 9004

Se mostrarán las principales ventajas de la aplicación de las normativas *ISO 9000*, *9001* y *9004*:

- Consiguen estandarizar las labores de todo el personal que trabaja dentro de una misma organización mediante la elaboración de documentación.
- Generarán un incremento de la satisfacción del cliente final, puesto que seguir estas metodologías ayudará a garantizar la calidad de productos o servicios final, mediante la aplicación de sus procedimientos.
- Mide y comprueba el desempeño de cada proceso.
- Incrementa la eficiencia en la organización para lograr los objetivos.
- Reduce errores e inconvenientes en el desarrollo del producto.
- Permite a la organización convertirse en un fuerte competidor en el mercado al ser aplicada.
- Consigue disminuir el gasto en tiempo, costes y recursos.
- Reduce la generación y aparición de errores.
- Aumenta el compromiso del personal de la empresa utilizando procesos internos más eficientes.
- Hace una labor de depuración en varias características y subcaracterísticas de cada componente de la calidad del software. Se trata pues de una labor de llegar al detalle en cada apartado, tratando de profundizar lo máximo posible.

Por otra parte, muchas veces las desventajas de estos modelos suelen coincidir y son:

- Los esfuerzos y el coste para elaborar la documentación y la instauración de la metodología en la organización.
- En ocasiones, la posible falta de compromiso por los altos cargos de la organización, parte de la plantilla o de los mandos intermedios en la organización.
- Carencia de planificación en la instauración de las metodologías.
- Déficit en la formación y capacitación de los trabajadores para conocer suficientemente el proceso de la metodología, o bien por falta de recursos.
- Posible intervención de consultores externos que no estén debidamente cualificados.
- Exige una dedicación e implicación exhaustiva por parte de todos los niveles de jerarquía que forman la organización. Esto nunca es sencillo de conseguir y coordinar.

4.5 Ventajas e inconvenientes de las normativas ISO 12207 y 15504

A continuación se presentan las ventajas del uso de las metodologías *ISO 12207* y *15504*:

- Permiten monitorizar los principales procesos.
- Su aplicación aporta efectividad al proceso de desarrollo de la calidad software.
- Permiten mantener registros de gestión, procesos y procedimientos.
- Ayudan a mejorar el producto final y con ello la satisfacción de clientes o usuarios finales.
- Reducen las incidencias y errores en la producción.
- Aglutinan labores como la evaluación de procesos, determinan sus capacidades y ayudan en la mejora de procesos.

- Definen los procesos del ciclo de vida del desarrollo de software, su mantenimiento y operatividad con los sistemas de software.

Por el contrario, también presentan ciertas debilidades o inconvenientes:

- Presentan compatibilidad con metodologías *CMMI* y otras normativas *ISO*, pero sin embargo *CMMI* aún no es conforme a *SPICE - ISO 15504*.

4.6 Ventajas e inconvenientes de la normativa ISO 15288

Los principales beneficios de la normativa *ISO 15288* son:

- Se puede aplicar en cualquier nivel de la jerarquía, en cualquiera de las fases de gestión y elaboración del diseño de un sistema.
- Cada sección de la organización se deberá implicar, con la intención final de alcanzar una aplicación de total aceptación por parte del usuario final.
- Aportará indicadores orientados a la organización, para ser aplicados en cada etapa del proceso de vida del desarrollo del software, para su mejora y eficiencia.
- Está claramente enfocado para el desarrollo de software a medida, tanto a nivel hardware, software, bases de datos, etc.

Los inconvenientes más destacados son:

- Existencia de un catálogo de procesos demasiado amplio, con 25 procesos diferentes, que recogen 123 resultados derivados de 403 actividades, esto puede ser demasiado laborioso para cualquier organización y sus empleados.

4.7 Ventajas e inconvenientes de la normativa ISO 20000

La normativa *ISO 20000* presenta las siguientes fortalezas:

- Posee mecanismos de control eficientes para generar aplicaciones de alta calidad a un coste efectivo.

- Las empresas desarrolladoras de software se han vuelto mucho más responsables desde el punto de vista de la calidad que proporcionan.
- Los proveedores externos de servicios pueden usar la certificación como un elemento diferenciador y acceder a nuevos clientes, ya que esto cada vez más se convierte en una exigencia contractual.
- Permite seleccionar, gestionar y proporcionar un servicio externo más efectivo.
- Mejora la eficiencia, fiabilidad y consistencia de sus productos y servicios que impactan directamente en la rentabilidad de los costes.

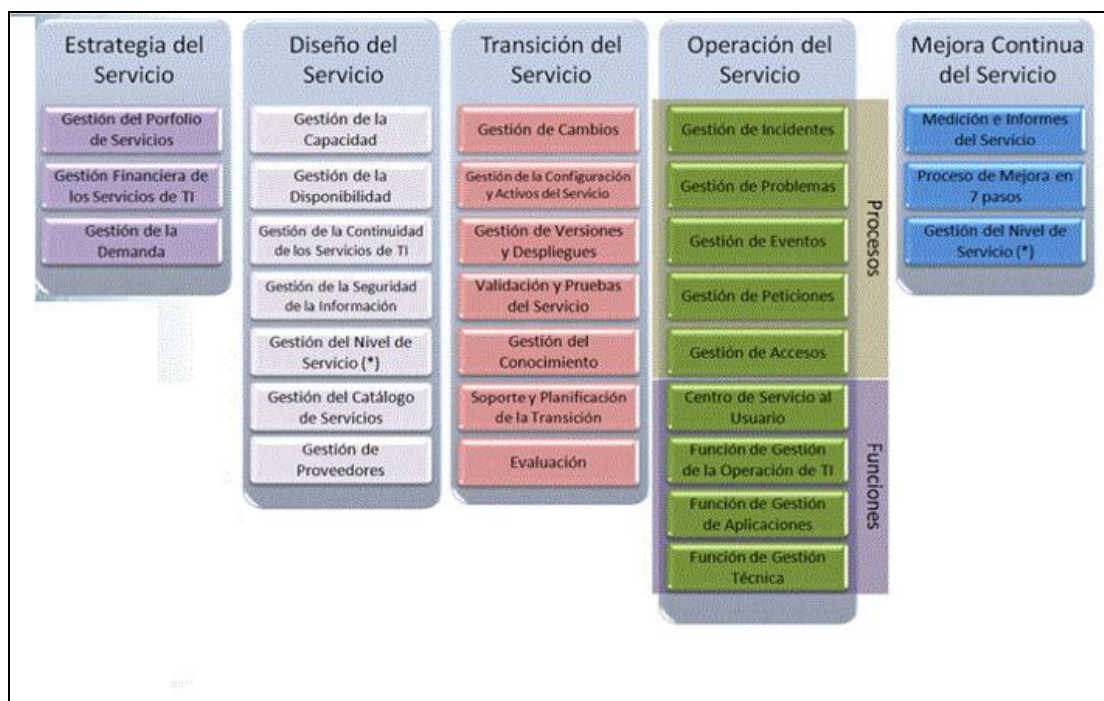


Figura Esquema de la normativa ISO 20000.
 Obtenido en <http://www.ingenia.es/en/servicio/itiliso-20000-0>

Por otro lado, sus mayores debilidades son:

- La certificación tiene una validez de tres años.
- Aparentemente la normativa parece enfocada para grandes organizaciones por su nivel de exigencia y dedicación.

4.8 Ventajas e inconvenientes de la normativa ISO 25000

Las ventajas que presenta se pueden desglosar en dos apartados:

Ventajas para la organización

- Detecta los objetivos del software con las necesidades reales y efectivas que solicita el cliente final.
- Evita ineficiencias y maximiza la rentabilidad y calidad del producto de software.
- Cumple los requisitos contractuales y demuestra a los clientes que la calidad del software es primordial.
- El proceso de evaluaciones periódicas ayuda a supervisar continuamente el rendimiento y la mejora.

Ventajas para los usuarios finales

- Demuestra el compromiso de la organización con la calidad del software.

Por contra, sus puntos más negativos son:

- No establece los niveles de calidad deseables para cada proyecto.
- No menciona “*un número de referencia a lograr*”, o el umbral que debe cumplir una métrica.
- Sería irreal fijar un valor, o valores, únicos de referencia para toda la industria.

5. Conclusiones del Estudio

Una vez que se han estudiado y analizado los modelos y normas anteriores, objeto del presente estudio, y que tienen que ver con los aspectos de control respecto a la calidad del desarrollo del software, se observa lo siguiente:

1. No existe una normativa *ISO* actual que reúna conjuntamente todos los criterios y aspecto necesarios para garantizar la calidad del desarrollo de software en función del tamaño del proyecto que se desea desarrollar.
2. Ante esta situación, el estudio que se realiza plantea la creación de dos nuevas versiones de normativa *ISO*, las cuales, utilizando los aspectos y criterios más convenientes, recogidos de las normativas actuales *ISO* y otras estudiadas, como *CMM*, *CMMI* y *AENOR*, y que en función del tamaño del proyecto o aplicación que se desea realizar, faciliten las fases de trabajo para la evaluación y un seguimiento correcto de la calidad del desarrollo del software.

Cabe destacar, que proponer dos nuevas normas no implicará dos niveles de calidad distintos, todo lo contrario, la calidad siempre será la referencia. Simplemente, lo que se propone son metodologías o normas adaptadas al tamaño del trabajo a desarrollar, puesto que aplicar una metodología exageradamente complicada a un pequeño proyecto software, no beneficia en absoluto al seguimiento de la evaluación de la calidad, sino que sucederá todo lo contrario, al no disponer de recursos ni personal adecuados para ejecutarla, fomentará el abandono y fracaso del proyecto.

Del mismo modo, si el proyecto a desarrollar es de gran dimensión, aplicar métodos de evaluación no acordes a la importancia, debilitará la robustez del producto final, puesto que partirá de una base debilitada en cada etapa que se desarrolle, pudiendo no llegar a alcanzar la finalización del trabajo.

Se pasa a detallar cuáles serían las características de cada una de las nuevas normas *ISO*.

5.1 Características de la Nueva Norma *ISO* para Proyectos de Pequeño y Mediano Volumen

La nueva norma *ISO* que se plantea para organizaciones de pequeño y tamaño, sería un compendio de las siguientes normas:

1. Normas de la “*Familia ISO 9000 (9000, 9001 y 9004)*” aportan un buen número de características para los que se denominan pequeños y medianos proyectos, lo cual facilitará la implantación y seguimiento desde el inicio, lo cual fortalecerá el desarrollo de software de calidad.

Las características que aportaría la “*Familia ISO 9000*” son:

- Son normas muy prácticas en aspectos como la detección de errores y su extensión para ser aplicadas en organizaciones.
- Al ser mucho más sencillas de implantar en organizaciones, ello implicará que todo el personal que trabaja las conozca, con el simple hecho de elaborar una documentación. Esto también supondría un aumento del compromiso de los trabajadores.
- Su labor de depuración en características y subcaracterísticas para los componentes de la calidad del software, consigue llegar al máximo detalle y profundizar al máximo en cada apartado.
- Aportaría gran flexibilidad a la hora de integrar nuevos procesos.

2. Las Aportaciones que efectuaría la norma *AENOR* son:

- Alta competitividad, puesto que la emisión de sus certificados supondría un gran prestigio para aquellas organizaciones que superen la evaluación de calidad.
- Permitiría un incremento en la productividad en pequeños y medianos proyectos, posibilitando que el proyecto llegue a mercados más amplios.

3. De la normativa *ISO 20000* la nueva norma *ISO* recogería:

- Los mecanismos de control eficientes para generar aplicaciones de alta calidad a un coste efectivo y adecuado al proyecto.

- La eficiencia, fiabilidad y consistencia del software obtenido impactaría directamente en la rentabilidad de los costes.

En este apartado, se debe mencionar, que para que sean efectivas las anteriores características de *ISO 20000*, debería ser eliminada la caducidad de la duración del certificado, actualmente de 3 años.

4. Aportaciones de la actual norma *ISO 25000*:

- Detecta los objetivos del software con las necesidades reales y efectivas que solicita el cliente final.
- Evitaría ineficiencias y maximizaría la rentabilidad y calidad del producto de software.
- Cumpliría los requisitos contractuales y demostraría a los clientes que la calidad del software es primordial.

5. Por último, también se plantearía recoger ciertos aspectos y características de la metodología *CMMI* (ya que al ser una evolución de la metodología *CMM* ya está implícita):

- Mejoraría la fiabilidad del proceso de planificación, en los aspectos de dedicación y calendario.
- Aumentaría de la productividad.
- Rebajaría los trabajos de correcciones en las fases de pruebas.
- Mejoraría la Imagen de Marca.

Si observamos, quedan excluidas las aportaciones de las normativas *ISO 12207*, *15288*, *15504* para los pequeños y medianos proyectos, puesto que se ha considerado que no encajan, quedando para ser utilizadas por organizaciones que desarrollan proyectos de mucho más volumen.

Si se consiguiese aglutinar todas estas características y actuaciones de las normas, se obtendría una norma ágil, flexible, fácil de instaurar y de implementar en el desarrollo de trabajos de pequeño o mediano volumen.

5.2 Características de la Nueva Norma ISO para Proyectos de Gran Volumen

Se empezará diciendo, que una organización que desarrolla trabajos de gran volumen, podrá disponer de muchos más recursos y mayor número de personal específicamente dedicado a la implantación, desarrollo, control y evaluación de la calidad del software que desarrolla.

Dentro del estudio realizado con el presente Trabajo Fin de Carrera, y más concretamente, dentro las normativas y metodologías estudiadas, existen una serie de normas que están muy enfocadas para ser desarrolladas en organizaciones que ejecutan grandes proyectos, como son por ejemplo, las normativas *ISO 12207*, *15288*, *15504*, a las que habrá que añadir también las normas *ISO 20000* e *ISO 25000*.

Debe entenderse también, que el resto de metodologías y normas que se han enfocada mayoritariamente para trabajos de pequeño y medio volumen, no quedan excluidas para las grandes organizaciones, todo lo contrario, sino que simplemente se sobreentiende que las normas orientadas a grandes proyectos englobarán a las anteriores.

Se detallarán a continuación las aportaciones de las metodologías y normativas:

1. De las normativas *ISO 12207* y *15504* se deberían incluir las siguientes actuaciones y características:
 - Monitorización de los principales procesos.
 - Mantenimiento de registros de gestión, procesos y procedimientos.
 - Realización de labores de evaluación de procesos, para determinar capacidades de ayuda y mejora de los procesos.
 - Definición de procesos del ciclo de vida del desarrollo de software, mantenimiento y operatividad de los sistemas de software.

2. La normativa *ISO 15288* aportaría:
 - Implicación para alcanzar la total aceptación por el usuario final.
 - Aportaría indicadores orientados a la organización, para ser aplicados en cada etapa del proceso de vida del desarrollo del software, para su mejora y eficiencia.

- Enfocada para el desarrollo de software a medida, tanto a nivel hardware, software, bases de datos, etc.

3. La norma *ISO 20000* añadiría:

- Mecanismos de control eficientes para generar aplicaciones de alta calidad a un coste efectivo.
- Responsabilidad desde el punto de vista de la calidad.
- Seleccionaría, gestionaría y proporcionaría un servicio externo más efectivo.

4. Y la norma *ISO 25000* complementarían la nueva norma con:

- Procesos de evaluaciones periódicas para supervisar continuamente el rendimiento y la mejora.
- Detección de los objetivos del software con las necesidades reales y efectivas que solicita el cliente final.

Se finalizarán las conclusiones, diciendo que nunca es fácil ajustar un proyectos y catalogarlo como de pequeño, mediano o de gran volumen, al igual que no es sencillo encajar totalmente una normativa en un tipo de organización, ya que existen muchos parámetros, características o subcaracterísticas que se solapan, pero lejos de ser un inconveniente, simplemente se ha tenido en cuenta como una ventaja.

La idea que ha perseguido el presente estudio es la de difundir la importancia de calidad del software y su proceso de desarrollo en las distintas fases de trabajo, son fundamentales, puesto que nunca debe entenderse que la inversión en calidad es una pérdida de tiempo o de recursos, sino todo lo contrario, puesto que supondrá una ayuda y mejorar continuamente los trabajos, pero sobre todo, debería ser entendida como una inversión que garantizará un producto final óptimo, algo que repercutirá positivamente en la organización a corto, medio y largo plazo.

6. Glosario

AENOR: *Asociación Española de Normalización y Certificación*, Se trata de una entidad dedicada al desarrollo de la normalización y la certificación todos los sectores, tanto Industriales como de servicios.

Atributo: Característica física o abstracta cuantificable.

Calidad: Nivel de medición de la habilidad para satisfacer las demandas solicitadas por parte de un usuario, físico o jurídico, en referencia a un servicio (en nuestro caso, una aplicación software).

Calidad externa: Extensión por la cual un producto satisface necesidades explícitas e implícitas cuando es usado bajo condiciones específicas.

Calidad interna: Suma total de atributos del producto que determinará su capacidad de agradar las necesidades propias e implícitas bajo condiciones específicas.

Calificación: Acción de evaluar el valor medido al nivel de calificación adecuado.

CMMI: *Integración de modelos de madurez de capacidades* o *Capability Maturity Model Integration (CMMI)*. Modelo para la mejora y evaluación de los procesos de desarrollo, mantenimiento y operación de sistemas de software.

Defecto: Un paso, proceso o definición de dato incorrecto en un programa de computadora.

Desarrollador: Persona u organización que se dedica a generar productos mediante el desarrollo software, en las distintas etapas del ciclo de vida del software.

Escala: Conjunto de valores con propiedades definidas.

Fallo: Finalización anómala o errónea de un producto software y de sus funciones.

ISO: (*International Organization for Standardization*), Organización Internacional de Normalización Es un organismo encargado de promover el desarrollo de normas internacionales (de productos y de servicios), con el objetivo principal de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones (públicas o privadas) a nivel internacional, con el objetivo de alcanzar los mayores niveles de calidad del producto final.

Medición: Actividad que persigue obtener una valoración métrica de los valores de un producto.

Medida: Número o categoría cuantificable de un atributo de una entidad o producto que se obtiene mediante una medición.

Medida directa: Una medida de un atributo que no depende del valor de ningún otro atributo.

Métrica: Es un método definido de valoración y su escala de valoración.

Medida externa: Una medida indirecta de un producto derivada de las medidas del comportamiento del sistema del que es parte.

Modelo cualitativo: Características que conforman la base de los requerimientos cualitativos específicos y la valoración cualitativa.

Módulo de evaluación: Técnica de evaluación para una característica o subcaracterística de calidad de un software específico.

Servicio: Producto o final que ofrece una organización a un usuario con el fin de realizar una labor o trabajo.

Sistema: Una composición integrada que consiste en uno o más procesos, hardware, software, instalaciones y personas, que proveen una capacidad para satisfacer una necesidad establecida o un objetivo.

Software: Todo o parte de los programas, procedimientos, reglas y documentación asociada a un sistema de procesamiento de información.

Usuario: Un individuo que utiliza el producto de software para realizar una función específica.

Valoración: Emplear una métrica para asignar uno de los valores de una escala (el mismo que puede ser un número o categoría) al atributo de una entidad.

Validación: Confirmación por inspección y provisión de evidencia objetiva de que los requerimientos particulares para un uso específico son alcanzados.

Verificación: Confirmación por examen y provisión de evidencia objetiva que los requerimientos específicos han sido alcanzados.

7. Bibliografía y Referencias

<http://www.rae.es/>

<http://www.evergreenpm.com/standish-group-2015-chaos-report/>

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/capitulo_5.pdf

<http://www.softqanetwork.com/%C2%BFque-es-el-modelo-cmm>

<http://www.adictosaltrabajo.com/tutoriales/cmmi/>

https://es.wikipedia.org/wiki/Modelo_de_Capacidad_y_Madurez

<http://www.adictosaltrabajo.com/tutoriales/cmmi/>

<http://www.aenor.es/aenor/inicio/home/home.asp>

<http://www.javiergarzas.com/2008/02/cuantos-proyectos-software-fallan-html.html>

<http://www.javiergarzas.com/procesos-software>

<http://www.iso.org/iso/home.html>

<http://www.softqanetwork.com/%C2%BFque-es-el-modelo-cmm>

<http://es.slideshare.net/nipingromanos/ventajas-y-desventajas-de-cmmi>

http://www.aenor.es/aenor/normas/normas/beneficios_normas.asp#.V0Bb42Ylw1w

https://es.wikipedia.org/wiki/Normas_ISO_9000

<http://www.bsigroup.com/es-ES/Gestion-de-Calidad-ISO-9001/>

<http://www.isotools.com.co/inconvenientes-obtener-certificacion-iso-9001/>

<http://www.intedya.com/internacional/96/consultoria-calidad-en-el-software-iso-25000.html#submenuhome>