

# ***Calidad de proceso de desarrollo***

## ***Conceptos fundamentales***

***Universidad Tecnológica Nacional – Facultad Regional Córdoba***

***Cátedra de Ingeniería de Software***

***Autores: Bosi, Nicolás Maximiliano (64606) – Cerruti, José Santiago (71614) - Latron, Romain (300329) – Mercado, Matías Gabriel (67768) – Morillas, Maximiliano David (64834) – Pucheta Moyano, Fernanda Noel (67356) – Valdeiglesias, Bruno Ezequiel (67713)***

***Resumen – En el siguiente documento se tratarán aspectos fundamentales e introductorios referidos a gestión de la calidad y su aplicación en procesos de desarrollo de software. Se analizará la diferencia entre control de calidad y aseguramiento de calidad; atributos de calidad de un producto software; la relación entre calidad de producto y calidad de proceso; estándares de calidad; plan de calidad, y métricas de software.***

***Abstract – The following document is about core aspects in the study of Quality Management and its application to software development. It is meant to be an introduction into the concepts of Software Process Quality Assurance. The aspects addressed ahead are: Difference between Quality Control and Quality Assurance; Product Quality, Process Quality, and how they are related; Quality Standards, Quality Planning; and Software metrics.***

### **INTRODUCCIÓN**

La calidad es una propiedad que posee un objeto que permite apreciarlo como igual, mejor o peor que otros objetos del mismo tipo. A lo largo de los años el concepto de calidad ha ido evolucionando y muchas filosofías acerca de qué es la calidad han aparecido ampliando el concepto de calidad hasta lo que se conoce hoy en día. Se debe entender a la calidad como algo subjetivo, y existen muchas aproximaciones hacia el significado de la palabra.

Muchas de estas perspectivas sobre qué es la calidad tienen mucho en común y comparten una idea general. Cuando se habla de calidad se debe tener en cuenta dos grandes campos:

- Conformidad con las especificaciones: es un tipo de calidad definido como el conjunto de características de un producto o servicio, se puede medir y evaluar si satisfacen un conjunto de especificaciones definidas.
- Conformidad con los clientes: es un tipo de calidad que define la capacidad que tiene un servicio o producto para satisfacer las necesidades y expectativas de un cliente.

Antes, el foco estaba en la calidad del producto pero poco a poco se fue dirigiendo la atención hacia los clientes. Hoy en día, la calidad no solo depende del producto o servicio ofrecido sino que también depende de los grupos de personas hacia los que está dirigido, quiénes son los interesados y quiénes lo consumen; ya que son los que finalmente determinan la calidad del producto.

Normalmente, cuando una persona quiere comprar algo, no solo busca que el producto cumpla con sus especificaciones sino que también quiere la garantía de que el producto tenga cierto nivel de calidad. Es por esto que las organizaciones buscan calidad en sus productos y/o servicios tomando como base la opinión de sus consumidores para satisfacer al máximo sus necesidades y expectativas, sobre todo considerando el hecho de que la sociedad es exigente y el mercado muy competitivo.

Una organización que entiende las necesidades de sus clientes, les brinda productos que se ajusten a las mismas y de esta manera aumenta la lealtad de

estos hacia la organización. Si una organización tiene clientes descontentos por la mala calidad del producto, es lógico pensar que buscarán alternativas diferentes.

Ofrecer productos de alta calidad y que cumplan con las expectativas de los grupos de interés es una inversión a largo plazo, ya que aumenta la reputación de la organización mediante las buenas críticas a los productos, incrementa la cantidad de clientes que compran el producto, disminuye los costos de producción de nuevos productos y permite mantener los precios de venta relativamente altos.

La calidad en una organización es principalmente importante para que puedan sobrevivir en el mercado y prosperar. Cuando las organizaciones quieren tener calidad en sus productos recurren a implementar modelos de calidad que les permite producir bienes y/o servicios de calidad y les trae muchos beneficios.

El hecho de que una organización implemente un modelo de calidad quiere decir que tiene la intención de crecer en el mercado y ser competitiva, ser más efectiva y eficiente para lograr sus metas, mejorar la cultura organizacional y por supuesto la calidad de vida de su personal, centrarse en las necesidades y expectativas de los clientes, y mejorar continuamente.

Los primeros conceptos de calidad surgen en el año 1946, referidos en este momento a procesos de producción industrial. Dos conceptos importantes que se definirán son el control de calidad y el aseguramiento de la calidad. [6 - 10]

## **I. CONTROL DE CALIDAD (QC) Y ASEGURAMIENTO DE LA CALIDAD (QA) EN CONTEXTO**

Lo central de estos conceptos es que son aplicados a productos tangibles. Bajo esta concepción los procesos de producción se analizan y se especifican. Cada proceso tiene una salida o componente con especificaciones definidas (medidas, peso, etc.), a las cuales el producto debería ajustarse para ser considerado de calidad.

Existen dos conceptos importantes a definir a la hora de hablar de calidad. Por un lado, el **control de calidad (QC)**, refiere a realizar mediciones sobre los componentes de un producto en cada etapa del proceso de manufactura. El control de calidad asegura que dichas mediciones estén comprendidas

dentro del rango de los valores especificados que el producto debería tener. Por otro lado, el **aseguramiento de la calidad (QA)**, no toma parte en el proceso propio de manufactura, sino que, se encarga de auditar el proceso para asegurar que las pautas y los estándares establecidos se siguen correctamente. Además, el aseguramiento de calidad toma métricas que sirven a la mejora continua del proceso.

Estos conceptos mencionados fueron exitosos al ser aplicados a procesos de manufactura, por lo que surgió la idea de también aplicarlo a procesos de desarrollo de software. Sin embargo, existen grandes diferencias entre estos procesos, y por ello es necesario hacer algunas distinciones importantes a la hora de definir **control de seguridad del software (SQC)** y **aseguramiento de la calidad del software (SQA)**. [11], [3]

## **II. CONTROL DE CALIDAD DEL SOFTWARE (SQC) Y ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE (SQA)**

Las diferencias entre un producto manufacturado y un producto de software surgen esencialmente del hecho de que el primero es tangible, mientras que el segundo, no. Por lo que, tanto sus funcionalidades, como sus costos y sus beneficios no se pueden medir fácilmente.

A continuación se resaltarán algunos problemas relacionados a aplicar el modelo de QA y QC de los procesos de manufactura, al desarrollo de software:

- Un producto manufacturado es una realización física de los requerimientos de un cliente.
- La funcionalidad del producto puede verificarse a partir de su realización física.
- Tanto los costos de manufactura como el valor del producto están previamente establecidos y son visibles.

Para identificar costos y beneficios de software se pueden definir un número de características, comúnmente llamadas **atributos de calidad** o **métricas de software**. La intención en este punto es desmenuzar el producto de software en atributos que puedan ser medibles en términos de beneficios y costos.

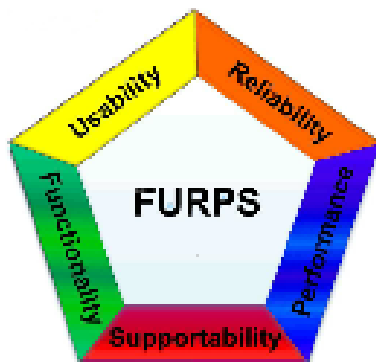
Hay muchas definiciones respecto a los atributos de calidad del software, pero una de las definiciones

más comunes es la provista por Robert Grady y se denomina modelo “FURPS+”. [11]

### III. MODELO FURPS+

“FURPS” es un acrónimo de los atributos Functionality, Usability, Reliability, Performance, Supportability; los cuales serán desarrollados a continuación:

- 1) Funcionalidad: refiere al conjunto completo de requerimientos funcionales que debe satisfacer el sistema. Esto representa las características principales del producto y debe responder qué debe hacer el sistema y no cómo debe hacerlo, ya que esto último responde más a cuestiones relacionadas al diseño e implementación. Es importante analizar en este punto el valor de negocio que el software le proporciona al cliente.
- 2) Usabilidad: involucra aspectos de experiencia de usuario, como por ejemplo, cuán natural es acceder a una determinada funcionalidad que el usuario requiera. También se consideran en este atributo aspectos relacionados a la estética visual de la interfaz.
- 3) Confiabilidad: incluye aspectos como ser, la disponibilidad, la precisión de los resultados obtenidos del software, y la capacidad de recuperación del mismo ante fallas diversas.
- 4) Performance: este atributo está relacionado al rendimiento general que posee el sistema, como ser, el tiempo de respuesta, de recuperación, de inicio, entre otros.
- 5) Matenibilidad: es un conjunto general de requisitos que se refieren a la forma de dar soporte del software. Esto quiere decir, que el sistema sea testeable, adaptable, integrable, escalable, fácil de configurar, etc.



*Figura 1. Modelo FURPS+*

El signo + en la sigla FURPS, hace referencia al conjunto de restricciones adicionales y particulares del producto, como por ejemplo, restricciones de diseño, implementación, interfaz, etc.

Es importante distinguir que las verificaciones de los atributos de calidad es responsabilidad del equipo de control de calidad de software, y no del equipo de aseguramiento de calidad del software.

Ni el personal de SQA ni el de SQC es responsable de colocar las características de FURPS+ en el producto. Una vez desarrollado el producto, SQC se encarga de controlar la presencia o ausencia de tales características en el mismo. Por su parte, SQA controla que se sigan los procedimientos que maximicen la probabilidad de que una vez finalizado el producto, el mismo posea tales características.

Sin embargo, a la hora de hablar de calidad del software hay que tener en cuenta que el mismo posee grandes diferencias con otros tipos de productos. Es por esto que su calidad también deberá ser medida de una manera diferente. [11]

### IV. CARACTERÍSTICAS PARTICULARES DEL SOFTWARE

No puede medirse la calidad del producto de software de la misma manera que se mide la calidad de un producto de manufactura, ya que el software tiene características particulares: [2]

- El software es un producto mental y creativo. No es tangible, es abstracto, y por lo tanto su calidad también lo es.
- El software se desarrolla, no se fabrica. Los costos están más relacionados con los recursos humanos que se necesitan en el proceso de desarrollo, que en la fabricación.
- El software no se va deteriorando con el tiempo ya que no se ve afectado por aspectos del entorno.
- El producto es construido a medida y esto dificulta más el control de su calidad. Cada producto de software es único y por consiguiente, su control de calidad debe adaptarse a esas características que lo hacen único.
- El mantenimiento del software se vuelve muy complejo, mucho más que el de hardware, ya que un cambio en el mismo implica cambios en muchas etapas (diseño, implementación, etc.).

- Se acepta que el software tenga errores ya que se asume que es inevitable la existencia de los mismos.

A lo largo del tiempo, varios autores publicaron artículos relacionados a la calidad del software. Es de interés del equipo de trabajo hacer mención a los “Seis principios de calidad del software”.

## V. SEIS PRINCIPIOS DE CALIDAD DEL SOFTWARE

Watts Humphrey fue considerado como el padre de la calidad del software debido a su dedicación a combatir los problemas relacionados con la crisis del software. Publicó muchos artículos, y entre ellos, al que se hará referencia en este documento es al de los 6 principios de calidad del software. Los mismos serán desarrollados a continuación.[1]

- 1) Si un cliente no demanda calidad, no la conseguirá
- 2) Para obtener calidad de manera constante, los desarrolladores deben gestionar la misma en su trabajo
- 3) Para poder gestionar la calidad es necesario que los desarrolladores la midan
- 4) La calidad de un producto está determinada por el proceso usado para desarrollarlo
- 5) Debido a que las pruebas solucionan solo una parte de los defectos, las pruebas deben tener calidad
- 6) La calidad sólo es producida por profesionales motivados y orgullosos de su trabajo

A la hora de definir la calidad del software es necesario realizar una diferencia entre la calidad del producto y la calidad del proceso de desarrollo. Sin embargo, los objetivos que se establezcan para obtener la calidad del producto de software van a determinar los objetivos a establecer para obtener la calidad del proceso. Esto implica que será más probable que un producto sea de calidad si el mismo fue desarrollado por un proceso de gestión de calidad.

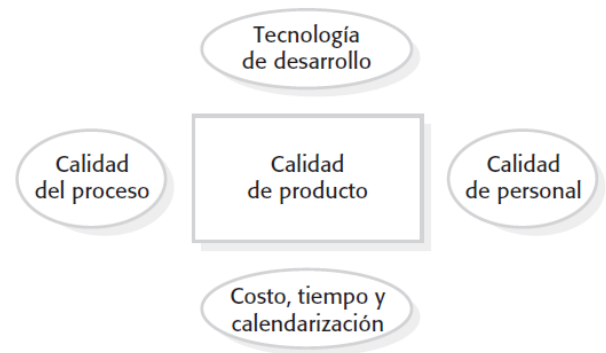
## VI. RELACIÓN CALIDAD DE PROCESO – CALIDAD DEL PRODUCTO

La relación entre la calidad del proceso y la calidad del producto es más compleja en el desarrollo del software, ya que el producto es intangible y depende de procesos intelectuales que no pueden ser automatizados. La calidad del software se encuentra

más bien influida por su proceso de diseño que por sus procesos de fabricación. El proceso que se utilice suele ser el factor más influyente en la calidad del producto. Sin embargo, cuando se realizan aplicaciones innovadoras, el personal que participa en el proceso tiene más influencia en la calidad, que el propio proceso que se utilice.

Para cualquier producto intelectual, donde la calidad depende del diseño, hay cuatro factores elementales, propuestos por Ian Sommerville, que determinan la calidad del producto. Estos son:

- Tecnología de desarrollo
- Calidad de personal
- Costo, tiempo y calendarización
- Calidad del proceso.



**Figura 2. Factores que determinan la calidad del producto**

Teniendo en cuenta los factores que determinan la calidad del producto en el proceso de desarrollo de software, si el mismo es un proceso definido, se puede incorporar en sus etapas iniciales la elaboración de un **plan de calidad**, a ser ejecutado a lo largo del mismo. [4]

## VII. PLAN DE CALIDAD

La planificación de calidad es el proceso de desarrollar un plan de calidad para un proyecto. El plan debe establecer las características que más interesan del software y describir qué tan importantes son. Sin esta definición, cada desarrollador involucrado puede hacer diferentes interpretaciones sobre cuáles atributos del producto reflejan las características de calidad más importantes. Los planes de calidad, que se desarrollan como parte del proceso de planificación general del proyecto, varían en detalle dependiendo del tamaño y tipo de sistema que se desarrolla.

El plan de calidad establece cuáles son los atributos de calidad más importantes para el software que se desarrollará. A lo mejor, la eficiencia sea crítica y tengan que sacrificarse otros factores para que se logre esto. Dicho plan debe incluir también una definición del proceso de valoración de la calidad. Ésta debe ser una forma acordada de valorar si determinado grado de calidad, como la mantenibilidad o robustez, está presente en el producto.

Algunas personas consideran que, si el proceso de desarrollo es de calidad, entonces el software final también lo será. Es aquí donde surgen distintos procesos establecidos, basados en *estándares* de organizaciones y procedimientos de calidad asociados, los cuales verifican el seguimiento de estos estándares mediante el equipo de desarrollo de software. [5]

## VIII. ESTÁNDARES DE CALIDAD

Los estándares proporcionan un marco para definir lo que se entiende por el término “calidad”. Como se describió previamente, la calidad del software es algo subjetivo, y el uso de estándares establece una base para decidir si se logró el nivel de calidad requerido. Una vez que los estándares han sido seleccionados para su uso, deben definirse los procesos específicos de proyecto para realizar un seguimiento en el uso de éstos y comprobar que se respetó dicho seguimiento. Los estándares deben entregar valor, en la forma de calidad aumentada del producto.

Los estándares de ingeniería de software se dividen en dos grandes tipos que se usan en la gestión de calidad del software:

- 1) Estándares del producto: se aplican al producto de software a desarrollar. Deben diseñarse de manera que puedan aplicarse y comprobarse de forma efectiva en lo que refiere a costos.
- 2) Estándares de proceso: establecen los procesos que deben seguirse durante el desarrollo del software. En estos se definen las buenas prácticas de desarrollo y deben incluir la definición de procesos que comprueben que se siguieron dichos estándares.

Se muestra a continuación una lista de ejemplos de estándares de proceso y de producto:

Estándares de producto	Estándares de proceso
------------------------	-----------------------

Formato de revisión de diseño	Realizar revisión de diseño
Estructura de documento de requerimientos	Enviar nuevo código para la construcción del sistema
Formato de encabezado	Proceso de liberación de versión
Estilo de programación Java	Proceso de aprobación del plan de proyecto
Formato de plan de proyecto	Proceso de control de cambio
Formato de solicitud de cambio	Proceso de registro de prueba

**Tabla 1. Ejemplos de estándares de proceso y producto**

Dependiendo del tipo de software a desarrollar, se utilizarán distintos procesos de desarrollo; es por ésta razón que los estándares deben ser adaptables.

No se debe forzar a trabajar de una manera determinada si esta es inadecuada para un proyecto. Cada administrador de proyecto debe tener la capacidad para poder modificar los estándares de proceso según lo que requiera en cada contexto particular. [5]

## IX. MÉTRICAS DE SOFTWARE

Una métrica de software es una característica de un sistema de software, documentación de sistema o proceso de desarrollo que puede medirse de manera objetiva. Algunas de estas métricas son el número de fallas detectadas en un producto de software liberado o la cantidad de horas-persona necesarias para desarrollar un componente.

Al comparar los valores medidos unos con otros, y con los estándares que se aplican a través de una organización, es posible sacar conclusiones sobre la calidad del software, o valorar la efectividad de los procesos, las herramientas y los métodos de software.

Las métricas de software se pueden clasificar en:

- Métricas de control (o métricas de proceso): cuyo objetivo es asistir a la dirección del proyecto; algunos ejemplos son el esfuerzo promedio y el tiempo requerido para reparar defectos.

- Métricas de predicción (o métricas de producto): se relacionan con el software en sí. Se puede mencionar como ejemplo la complejidad ciclomática de un módulo, la longitud promedio de los identificadores de un programa, y el número de atributos y operaciones asociados con las clases de objetos en un diseño.

Profundizando un poco en métricas de proceso, se las puede encuadrar en los siguientes tipos:

1. Aquellas que miden el tiempo que tarda en completarse un proceso, como puede ser el tiempo total dedicado al proceso, tiempo calendario, tiempo empleado en el proceso por ciertos roles, etcétera.
2. Aquellas que miden los recursos requeridos, como esfuerzo total en horas-persona, costos de transporte o recursos de procesamiento.
3. Aquellas que miden el número de ocurrencias de un evento particular, como ser la cantidad de defectos descubiertos durante la inspección del código, el número de cambios solicitados a los requerimientos o el número de componentes afectados ante un cambio de requerimientos.

Los primeros dos tipos de medición pueden usarse para descubrir si los cambios al proceso mejoraron la eficiencia de un proceso. Definiendo puntos fijos de avance a lo largo del proceso, se pueden métricas como el tiempo y esfuerzo entre tales puntos, y realizar mediciones en antes y después de introducir cambios al proceso. Al comparar las mediciones respectivas, se puede evaluar si los cambios en el proceso lograron mejorar los valores evaluados – es decir, si se redujo el tiempo o disminuyó el esfuerzo requerido.

Por su parte, las mediciones de la cantidad de eventos que se producen tienen una relación más directa con la calidad del software. Por ejemplo, si luego de implementar un cambio en el proceso se observa un crecimiento en la cantidad de defectos descubiertos durante las pruebas, esto muy probablemente indicará una mejora en la calidad del producto. [4-5]

## **X. CALIDAD EN LOS PROCESOS DE DESARROLLO – VENTAJAS**

Algunas de las ventajas que se destacan son:

- ✓ Incorporar un proceso de aseguramiento de calidad en el desarrollo de software incrementa significativamente la probabilidad de lograr aceptación del cliente respecto al producto logrado. Además, si el software desarrollado es lo suficientemente complejo, la experiencia histórica nos demuestra cómo se torna casi imposible el logro de ciertas características pretendidas del mismo (por ejemplo la fiabilidad, mantenibilidad, facilidad de reutilización) en plazos razonables sin llevar a cabo actividades de gestión de la calidad.
- ✓ La sabiduría acerca de buenas prácticas en el proceso que tiende a lograr calidad de producto, adquirida mediante la experiencia, puede ser plasmada en estándares de software. Los mismos se aplican al proceso y reflejan la sabiduría que es de valor para la organización. Mejor aún, una organización sin haber atravesado las dificultades que dieron origen a cierto estándar puede adoptarlo, reutilizando la experiencia de otros equipos de desarrollo.
- ✓ El conocimiento de parte de un cliente acerca de la aplicación de prácticas de gestión de calidad por parte de su proveedor de software, puede incrementar la confianza hacia el mismo. Aún más, algunos clientes demandan que sus proveedores de software tengan certificación en estándares de calidad divulgados (por ejemplo ISO 9001). Así, pueden estar seguros de que la organización que desarrolla el software tiene un sistema de gestión de calidad aprobado.

## **XI. CALIDAD EN LOS PROCESOS DE DESARROLLO – DESVENTAJAS**

Algunas de las desventajas que se destacan son:

- ✗ La aplicación de técnicas de QA en el desarrollo de software supone comprometer un conjunto de recursos y, por ende, un gasto adicional para la organización en los proyectos sobre los cuales se aplique. En organizaciones pequeñas, donde el presupuesto es muy limitado, se suele prescindir de las actividades de control del producto y, en caso de llevarlas a cabo, suele estar muy ligado el personal control al de desarrollo – lo cual, como se trató anteriormente, no es deseable.
- ✗ La medida en que las adaptaciones del proceso contribuyen a lograr mejoras en la calidad del producto depende de lo oportunas y adecuadas que sean tales adaptaciones al desarrollo del

producto. Por ello, si el equipo de QA no es lo suficientemente certero en el análisis realizado respecto a cómo lograr, mediante cambios en el proceso, los atributos de calidad deseados en el producto, las adaptaciones realizadas pueden ser contraproducentes.

## CONCLUSIÓN

A lo largo del documento se logró brindar una idea introductoria acerca de conceptos de calidad y la aplicación los mismos en el desarrollo de software. En sus inicios, estos conceptos estaban destinados a brindar calidad a productos físicos, resultantes de producción en masa. Como se explicó, las características intrínsecas del software hacen que éste difiera en una infinidad de aspectos a esos productos físicos, por lo que aplicar calidad a un producto de software involucra un trabajo sustancialmente diferente.

Como el software no se crea ni se construye, sino que se *desarrolla*, mayormente mediante procesos iterativos e incrementales (recordar los fundamentos tanto del Proceso Unificado de Desarrollo como de las Metodologías Ágiles), es imprescindible que, si se desea que como resultado se obtenga un producto de software con calidad, el proceso también sea de calidad.

Cabe destacar que existe una diferencia en las formas de aplicar calidad de proceso si se trata de procesos definidos o procesos empíricos. Conceptos tales como el plan de calidad y los estándares internacionales de calidad de procesos (Ej: ISO 9001) son aplicables a procesos definidos, los cuales buscan estandarizar procesos independientemente del proyecto en el cual se apliquen. Por su parte, en procesos empíricos la calidad del proceso no se aborda principalmente desde la certificación de estándares, sino desde la aplicación eficaz de los frameworks con los que se trabaje y el descubrimiento propio del equipo acerca de las mejores prácticas que llevan a la calidad de producto en cada proyecto realizado.

Por otra parte, es importante resaltar que la calidad del software es un concepto subjetivo, que depende de cada producto, de cada equipo de desarrollo y de las necesidades de cada usuario.

De acuerdo a los diferentes contextos, se definirán y priorizarán los atributos de calidad y se establecerá la forma en la que se medirá la calidad. La tarea anterior no es para nada sencilla, dado que

por ejemplo, se podrá definir que un atributo para medir la calidad será la mantenibilidad, pero en tal caso no se puede decidir si un sistema es realmente mantenible sin haberlo usado durante un largo periodo de tiempo. En todos estos casos, el equipo de gestión de calidad tiene que usar su juicio propio para decidir si se logró un nivel de calidad aceptable.

Consecuentemente, aún más complejo es decidir cómo las características del proceso influyen en dichos atributos. Si bien es cierto que existen estándares y modelos de calidad de proceso, la estructuración y las mediciones de calidad del proceso de desarrollo (como la cantidad de errores durante la codificación) pueden deteriorar la creatividad de los desarrolladores, lo que devendrá en disminuir la calidad del producto en lugar de aumentarla.

## BIBLIOGRAFÍA

- [1] Pablo Turmero. (2016). *Gestión de proyectos de software. El producto de software y su ciclo de vida*. <http://www.monografias.com/trabajos102/gestion-proyectos-software/gestion-proyectos-software.shtml#Comentarios>
- [2] Lic. Fernanda Scalone. (2006). *Estudio Comparativo de los Modelos y Estándares de Calidad del Software*. <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>
- [3] Dave Nielsen (2010). *CMM and Project Quality Management*. <https://pmhut.com/cmm-and-project-quality-management>
- [4] Sommerville, Ian - INGENIERÍA DE SOFTWARE - Novena Edición (Editorial Addison-Wesley 2011). Capítulo 26 “Mejora de procesos”
- [5] Sommerville, Ian - INGENIERÍA DE SOFTWARE - Novena Edición (Editorial Addison-Wesley 2011). Capítulo 24: “Gestión de calidad”.
- [6] Dankocs (2012). *Calidad del Software. Mejoramiento de la Calidad del Software*. <http://dankocs2012.blogspot.com.ar/2012/11/mod-elos-y-estandares-de-calidad-del.html>
- [7] Isotools. (2015) *¿Qué es el Aseguramiento de la Calidad y Cómo se Consigue?* <https://www.isotools.org/2015/03/20/que-es-el-aseguramiento-de-la-calidad-y-como-se-consigue/>

[8] CQI IRCA. (2015). *What is Quality?*  
<https://www.quality.org/article/what-quality>

[9] Ian Linton. (2016). *Why is Quality Important for a Business?*  
<http://smallbusiness.chron.com/quality-important-business-57470.html>

[10] Bob Napierala. (2012). *Five Important Factors in Total Quality Management*  
<http://aboutthree.com/blog/five-important-factors-in-total-quality-management/>

[11] Ian Fleming (año desconocido). *What is SQA?*  
<http://www.sqa.net/index.htm>