

Examen de los factores más influyentes en el aseguramiento de la calidad del *software* en la industria costarricense

Miguel Marchena, Mauricio Vargas

Escuela de Ingeniería,
Universidad Latinoamericana de Ciencia y Tecnología,
ULACIT, Urbanización Tournón, 10235-1000
San José, Costa Rica
jmarchenam048, evargasc457]@ulacit.ed.cr
<http://www.ulacit.ac.cr>

Resumen La industria de *software* costarricense produce un 1,3 por ciento del producto interno bruto del país, esto es más de lo que produce la exportación de café, lo cual deja en claro la relevancia de la industria del *software* para el país. Dicha industria es altamente competitiva a nivel internacional, por lo que producir un *software* de calidad se ha convertido en una necesidad para los desarrolladores nacionales. Existen diversos estándares de aseguramiento de la calidad, cada uno presenta una serie de pasos y herramientas determinadas, sin embargo implementar estos estándares puede ser costoso tanto en tiempo como en esfuerzo, y ninguno de ellos puede garantizar un *software* libre de defectos. Teniendo esto en claro, el presente artículo plantea examinar los factores más influyentes en el aseguramiento de la calidad del *software* en la industria costarricense. Con el fin de lograr esto, se identifican dichos factores con ayuda de expertos en la materia, para luego exponer la relevancia de estos factores, basándose principalmente en estándares de la IEEE y la ISO, por ser dos de las organizaciones de estandarización más respetadas a nivel mundial.

Keywords: gestión de software, calidad de software, calidad

1. Introducción

Al ver la televisión, navegar por internet o simplemente dar un paseo por el centro comercial, es muy probable toparse con la palabra calidad en reiteradas ocasiones. Dicho término figura con frecuencia en esta época competitiva, en la que las compañías aseguran vender productos y servicios de “calidad”, aspecto que brinda mayor valor agregado a su clientela. Sería adecuado empezar por explorar su raíz etimológica: los orígenes de esta palabra se remontan al término griego “Kalos”, que significa “lo bueno” o “lo apto”. También, a “qualitatem”, proveniente del latín y utilizado para representar “calidad.” “propiedad”. En este sentido, calidad es de naturaleza subjetiva. Nace de la apreciación que cada individuo define según sus experiencias y expectativas.

Aunque se podría pensar que la calidad es un concepto moderno, el ser humano se ha visto interesado en ella desde la antigüedad. Los primeros estudios acerca de esta materia se realizaron durante los años treinta, antes de la Segunda Guerra Mundial, en Estados Unidos. El doctor W. A. Sheward, de Bell Laboratories, en el año 1933, aplicó el concepto de control estadístico de proceso por primera vez con propósitos industriales.

Desde entonces, el concepto de calidad ha evolucionado con el pasar de los años, hasta su forma actual de “Calidad Total”. La definición del concepto de calidad total, fue desarrollado por el Dr. Armand Feigenbaun. Este considera que la calidad total, consiste en un tipo de estrategia de calidad que tiene por objetivo inculcar una conciencia de calidad en todos los procesos vinculados con el desarrollo de los productos o servicios dentro de una organización, creando una filosofía de mejora perpetua.(Feigenbaun, 1984)

El concepto de calidad ha calado hondo en múltiples ramas de conocimiento y, como es de esperarse, en cada una de ellas existen parámetros diferentes que la definen y/o regulan. En términos de desarrollo de *software*, se define como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, y a la utilización de los estándares de desarrollo propuestos. Es decir, que el proyecto de desarrollo se apegue al plan establecido, con base en la documentación hecha con anterioridad. Según se expone, un *software* debe cumplir con una serie de requisitos a lo largo de su ciclo de desarrollo, incluso como producto final. Este esfuerzo se realiza para poder alcanzar la definición de *software* de calidad, y dio paso a un tema que paulatinamente creó toda una nueva disciplina dentro de la computación: el aseguramiento de la calidad de *software*, o SQA (*Software Quality Assurance*, en inglés).

Existen muchos estándares que exponen una serie de pasos y herramientas para crear un proceso estructurado de aseguramiento de la calidad para el desarrollo de *software*. Entre ellos, se encuentra el estándar 730-2014, del Instituto de Ingenieros Eléctricos y Electrónicos (“IEEE”, por sus siglas en inglés), el cual define SQA como “Un conjunto de actividades que definen y evalúan la idoneidad de los procesos de *software* para brindar evidencia que confirme que los mismos son apropiados para producir *software* de calidad, acorde con las necesidades propuestas.”(IEEE., 2014).

Actualmente, el mundo vive un auge tecnológico y Costa Rica no es la excepción; ahora, la mayoría de sistemas de cómputo se encuentran en línea e intercambian información. Algunos de estos sistemas son críticos, por ejemplo, sistemas de control aéreo, bancarios, médicos y muchos otros. Todos estos ejemplos de sistemas de cómputo, tener como único denominador común su dependencia de un componente de *software* para su funcionamiento e interacción con los usuarios. Es innegable la importancia del *software* en la vida cotidiana y por ende el ejercicio de un desarrollo de calidad y la debida implementación del proceso de aseguramiento de dicha calidad que lo respalda y contribuye a obtenerlo.

Según el periódico El Financiero, en la actualidad los servicios de *software* en Costa Rica, representan el 1,3 por ciento de la producción local, lo cual

significa que la economía de Costa Rica genera más servicios de *software* que exportaciones de café. Por el mencionado auge del desarrollo de *software* en el país, el aseguramiento de la calidad surge como un tema de relevancia para la industria de *software* costarricense. Dada la alta competitividad internacional, resulta fundamental implementar metodologías o esquemas de aseguramiento de la calidad, pues promueven la implementación de una serie de buenas prácticas y con ello se puede aumentar la competitividad, la eficiencia y brinda prestigio al *software* tico.

Por lo expuesto anteriormente, el presente artículo pretende examinar cuales son los factores más influyentes en el aseguramiento de la calidad del *software* en la industria costarricense, según el criterio experto de profesionales en el área de calidad de *software*, del sector público y privado. Se examinarán estos factores desde el punto de vista académico, apoyándose principalmente en los estándares de la IEEE y la ISO, en relación con el aseguramiento de la calidad de *software* con el fin de comprender ¿Cuál es la relevancia de estos factores en el aseguramiento de calidad de dicho *software*?

2. Marco Teórico

Debido a que el concepto de la calidad resulta ambiguo para muchos, es importante primero abarcar qué implica en un contexto general para así poder especificar paulatinamente su alcance en la rama informática. A menudo, los profesionales de diversas áreas de conocimiento, orientan su esfuerzo hacia cómo mejorar o garantizar la calidad, de acuerdo con su disciplina, sin conocer los fundamentos universales y fallan en el intento. Por tanto, se abordarán varios términos relacionados con la calidad para aclarar la temática y ahondar a nivel técnico.

2.1. Calidad

La calidad se puede definir como el grado de aceptación o satisfacción que proporciona un producto o servicio a las necesidades y expectativas del cliente. Las norma ISO 9000 define la calidad como “Grado.^{en} el que un conjunto de características inherentes cumple con los requisitos” comprendiendo como requisito “expectativa o necesidad, comúnmente obligatoria o implícita”. (ISO, 2015). Las dos definiciones anteriores hacen evidente que la calidad gira en torno al cumplimiento de ciertos requisitos, por lo que se podría decir que el concepto de calidad puede variar según el cliente, la industria o hasta la cultura. Lo señalado permite afirmar que la calidad puede ser más una condición psicológica o de percepción que una cuestión técnica, la calidad es un concepto totalmente relativo. Otra definición popular de calidad es la expuesta por Joseph Moses Juran, quien la define calidad como “La aptitud para el uso”, refiriéndose a la idoneidad de un producto o servicio para el uso para el que fue concebido. El concepto de calidad puede ser aplicado en diversos tipos de actividades. En el ámbito del desarrollo de *software*, la búsqueda de la calidad implica la implementación de

procedimientos. Este considera, entre otros aspectos, metodologías o estándares para controlar el análisis, diseño, codificación y prueba del *software*. Lo expuesto permite establecer la filosofía de trabajo, en pro de obtener un mayor grado de confiabilidad, mantenibilidad y facilidad de prueba, mientras se eleven la productividad de la labor de desarrollo y el control de la calidad del *software*. La IEEE plantea una definición más pragmática del concepto de calidad de *software*, estrechamente relacionado con el concepto amplio de la calidad. Está planteada la calidad de *software* como “El grado en que un producto de *software* cumple con los requisitos establecidos; sin embargo, la calidad depende del grado en que los requisitos establecidos representan con exactitud las necesidades, deseos y expectativas de las partes interesadas” (IEEE., 2014). La calidad de *software* propone el cumplimiento de expectativas y necesidades, además, considera la implementación de procedimientos, estándares o metodologías para controlar y establecer los procesos que guíen el desarrollo del *software* a su definición de calidad. Lograr esta definición de calidad no es tarea fácil; algunas organizaciones no se molestan en alcanzarla, desconocen cómo hacerlo o no tienen el apoyo suficiente de las jefaturas para promoverla y, sin embargo, es de suma importancia. De acuerdo con el informe “The CHAOS Manifesto”, del Standish Group, sólo el treinta y nueve por ciento de los proyectos de software termina dentro del tiempo planificado, con los recursos establecidos y con una calidad aceptable.(Group, 2013)

2.2. Aseguramiento de la calidad

Teniendo clara la definición de calidad y su aplicación en el contexto del desarrollo de *software*, es posible tratar con soltura el tema central del presente artículo, el aseguramiento de la calidad del *software*. Según el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), el aseguramiento de la calidad se define como “ Un conjunto de actividades que definen y evalúan la idoneidad de los procesos de *software* para brindar evidencia que confirme que los mismos son apropiados para producir dicho *software* de calidad acorde con las necesidades propuestas. Un atributo clave de SQA es su objetividad, con respecto al proyecto. La función SQA puede estar organizada independientemente del proyecto; es decir, aislada de presiones técnicas, gerenciales y financieras del proyecto.”(IEEE., 2014).” , En síntesis, el aseguramiento de la calidad es la parte de la gestión de la calidad enfocada a proporcionar confianza en que se cumplan los requisitos de dicha calidad. Como se mencionó anteriormente, la calidad del *software* lleva a la implementación de prácticas, estándares y metodologías, con el fin de lograr obtener la calidad del mismo. Así, el aseguramiento de la calidad se enfoca en definir y evaluar dichos procesos para determinar si son los idóneos y vigilar que los mismos se ejecuten respetando todos los pasos y controles previamente establecidos. También incluye una serie de actividades o factores por tomar en cuenta que tienen como objetivo poner en práctica el proceso como tal; aunque existen varios factores que aseguran la calidad y cada autor propone más o menos factores, el presente artículo tomará como principales, los siguientes:

- Control de la calidad
- Gestión de la configuración
- Pruebas

2.3. Control de calidad

Según la IEEE, el control de la calidad de *software* es definido como “Un modelo planificado y sistemático de todas las acciones necesarias para proporcionar la confianza adecuada de que un artículo o producto se ajusta a los requisitos técnicos establecidos. Un conjunto de actividades diseñadas para evaluar el proceso por el cual los productos son desarrollados o fabricados.” (IEEE., 2014).

El control de la calidad del *software* especifica técnicas y acciones de carácter operativo, que se utilizan para verificar los requisitos relativos a la calidad, se centran en mantener bajo control el proceso de desarrollo y eliminar las causas de los defectos en las diferentes fases de la confección del producto.

Es una estructura que organiza evaluaciones, inspecciones, auditorías y revisiones que asegura el cumplimiento de las responsabilidades asignadas. Garantiza que se utilicen eficientemente los recursos y se logre el cumplimiento de los objetivos del producto. Tiene como meta, mantener bajo control un proceso y eliminar las causas de los defectos en las diferentes fases del ciclo de vida del *software*.

El control de la calidad del *software* está centrado en dos objetivos fundamentales:

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

En general, se puede decir que el control de la calidad del *software* considera las actividades o acciones operativas que se realizan para evaluar la calidad de los productos desarrollados. (Cueva Lovelle, 2011)

2.4. Gestión de la configuración

La gestión de la configuración en el desarrollo de *software*, es el proceso de identificar y limitar los elementos en un sistema determinado, controlando el cambio a lo largo de su ciclo de vida. También, registra y reporta su estado, las solicitudes de cambio, y verifica que los elementos estén completos y que sean los correctos a lo largo del ciclo de vida del *software*.

La finalidad de la gestión de la configuración de *software* es definir y mantener la integridad de los productos por medio del ciclo de vida del proceso de *software*. Para realizar este proceso, se requiere identificar la configuración de dicho *software* en puntos estratégicos en el transcurso del tiempo, además, de sistematizar los cambios mientras se mantiene la integridad y trazabilidad.

La información creada en los procesos de configuración de *software*, se puede definir como parte de los elementos del proceso de ingeniería, más específicamente como un elemento de configuración del *software* o ECS. Además de la información creada en este proceso, se pueden citar otros ECS, tales como el

plan del proyecto, manuales de usuario, operación, e instalación, especificaciones del sistema y diseños, listados del código fuente, plan y procedimiento de pruebas, ejecutables y la información relacionada con peticiones u órdenes de mantenimiento.

2.5. Pruebas de *software*

Las pruebas de *software* son una de las partes más importantes del aseguramiento de la calidad y una de las acciones más comunes, pueden ser definidas como un conjunto de actividades planificadas y sistemáticas que proporciona confianza en que los productos y servicios cumplirán los requerimientos especificados y las necesidades del usuario. El aseguramiento de la calidad aplicado a *software* define y evalúa el proceso mediante el cual este se desarrolla (Kaner, 2013). Estas pruebas pueden conducirnos a algún objetivo en concreto, tales como la observación de fallos o un comportamiento erróneo, y sirven para disminuir el riesgo en la etapa de su implementación.

Existen diferentes metodologías para asegurar la calidad de un *software* mediante el tratamiento de pruebas que dependiendo de su complejidad se dividen en pruebas funcionales (black box) y no funcionales (White box). Las pruebas funcionales verifican el correcto funcionamiento de las funciones externas o el comportamiento esperado por las especificaciones del usuario. Las pruebas no funcionales o estructurales se enfocan en verificar la observación de fallos internos, las estructuras y las relaciones entre ellos. Se pueden aplicar muchas técnicas diferentes de pruebas de *software* en distintas fases de su ciclo, en varios tipos de productos y en muchos ambientes. Por esta razón las pruebas se pueden enfocar en lo que específicamente se necesita, cuál es el modelo que se debe utilizar, si son aplicables a otras partes y si se pueden combinar las técnicas para aumentar la efectividad de las pruebas.

3. Metodología

El presente artículo pretende explorar cuáles son algunos de los factores más influyentes en el proceso de aseguramiento de la calidad de *software* en la industria costarricense, y exponer la relevancia de dichos factores. Para satisfacer este objetivo, se aplicó un cuestionario a un grupo de expertos en el aseguramiento de la calidad del *software*, los cuales laboran en empresas líderes en materia de desarrollo del mismo, tanto en la industria privada como en el ámbito gubernamental. A partir de los resultados obtenidos, se determinaron cuáles son los tres factores más relevantes en el aseguramiento de la calidad de *software* en la industria costarricense según el criterio de expertos en la materia. Una vez identificados estos factores, se realiza una revisión bibliográfica, mayormente enfocada a la revisión de estándares de aseguramiento de la calidad de la ISO y la IEEE. Lo anteriormente indicado, con el fin de otorgar sustento teórico a la investigación y validar la trascendencia de los factores identificados.



Figura 1. Factores de calidad identificados

4. Resultados

Luego de analizar e interpretar las respuestas de los cuestionarios, se identificaron los factores mencionados sin omisión, con el fin de determinar un criterio común, para finalmente seleccionar los tres factores más relevantes según el énfasis que se dio a cada uno de ellos.

Dentro de las respuestas de los cuestionarios, se mencionan diversos factores que afectan el aseguramiento de la calidad. A continuación se listan algunos de los factores propuestos en las encuestas, pero que no fueron seleccionados dentro de los tres principales factores que afectan la calidad del *software*:

- Políticas de garantía de calidad
- Buenas prácticas de gestión de proyectos
- Cultura general de los profesionales con pensamiento progresista
- Adecuada documentación de requisitos

Estos y otros factores, no fueron mencionados por unanimidad por los encuestados, ni se enfatizaron de igual manera que los tres factores seleccionados, por lo que fueron descartados de la lista de factores más influyentes. Dentro de

las preguntas realizadas en el cuestionario, se pueden destacar algunas cuyos resultados fueron respondidos en concordancia entre todos los encuestados. Entre ellas se destaca: ¿por qué se debe asegurar la calidad, cuál es la intención de hacerlo? Para dicha pregunta los encuestados coincidieron en que el aseguramiento de calidad lleva a la mejora de los servicios y productos, aumenta la eficiencia del proceso de desarrollo y brinda prestigio a la organización. ¿El aseguramiento de la calidad debería ser implementado en todas las empresas, o es más característico de empresas de mayor tamaño? La respuesta a esta pregunta fue unánimemente que el proceso de aseguramiento de calidad de *software* debería implementarse en todas las empresas, sin importar el tamaño de las mismas.

Finalmente, al preguntar directamente por los factores más influyentes que afectan el aseguramiento de la calidad de *software*, los tres factores mencionados con unanimidad y en los que se hizo énfasis son:

- Configuración de la calidad
- Pruebas de *software*
- Control de la calidad

Los expertos encuestados coincidieron de manera unánime, en que la adecuada ejecución de estos tres procesos en el proceso de aseguramiento de la calidad es crucial para obtener un *software* de calidad. Aunque muchos son los factores que influyen en la obtención de un *software* de calidad, estos tres factores forman la base para conseguir dicho *software*. No aplicar cualquiera de ellos, dificultaría la obtención de un *software* de calidad. A continuación, se tratarán más a fondo los factores seleccionados con el fin de comprender mejor su relevancia y función en el proceso de aseguramiento de la calidad.

4.1. Control de la calidad

El control es un factor esencial en la gestión de la calidad. Este adquiere relevancia cuando se alcanza un modelo en el cual el sistema corresponde a un diseño altamente eficiente, fácil de entender y mantener. Su importancia se centra, en que las métricas de este procedimiento son evidencias que valoran la existencia de los atributos de un proyecto de *software*. Por ejemplo, se puede citar las listas de comprobación que califican el producto desarrollado para una condición en particular, directrices de inspección que evalúan el *software*, su documentación para atributos específicos. Además, se considera la información o medidas cuantitativas de características tales como el tamaño, la complejidad, la independencia, el uso de datos, entre otras. El control de calidad permite establecer requerimientos para iniciar, planificar, controlar y ejecutar una garantía de calidad adecuada y armonizada a un ciclo de vida, y conjuntamente si se basa en estándares internacionales de renombre como ISO/IEEE se puede aumentar su eficiencia y eficacia. Además, este control debe procurar siempre proporcionar los requisitos aceptables, uniformes y mínimos para los procedimientos de calidad (QA) en apoyo a un proyecto de *software*. (Board, 2014)

Por lo tanto, el factor de control de calidad ayuda a identificar y definir parámetros, indicadores o criterios que se van a utilizar en la medición para

controlar la calidad del *software*. Una vez seleccionados estos parámetros, se obtienen beneficios como:

- Clasificación del *software*: tipo, ámbito de aplicación, complejidad, entre otros, acorde con los estándares implantados para el desarrollo del *software*.
- Selección de medidas que puedan ser utilizadas al objeto de control.
- Establecimiento de los métodos de valoración: técnicas manuales como, cuestionarios o encuestas.
- Especificaciones o regulaciones del personal que realiza el control: ¿Cuándo se efectúa el control de la calidad, quiénes participan y qué documentación se revisa y entrega?.

4.2. Gestión de la configuración

La gestión de la configuración debe procurar tener una metodología que afirme un desarrollo y mantenimiento integral del *software*, se debe tratar la integridad del producto antes y después de su entrega a producción. Con respecto a esto, se observa que el factor de “Configuración del *software*” es uno de los elementos indispensables para el desarrollo de gestión de la calidad. En un estudio realizado en la Universidad Estatal a Distancia (UNED), ubicada en Costa Rica, se plantea que este factor influye en la creación de empresas desarrolladoras de *software* en Costa Rica. En dicho estudio se menciona que, entre las limitaciones que los administradores de proyectos enfrentan dentro de la cultura costarricense en el ámbito de la producción de *software*, es que, en algunas micro y pequeñas empresas el mismo se produce de manera artesanal, lo cual influye en su calidad, en su mayoría por escasez de recursos humanos y tecnológicos. Además, algunos desarrolladores se resisten al trabajo relacionado con la gestión de la configuración como, medición de la productividad, control de cambios, etc., refiriéndose así a que esto se debe a una práctica deficiente de gestión de la configuración de *software*. (Sánchez, 2011)

Por lo anteriormente expuesto, una buena gestión de la configuración trae como beneficios la identificación de elementos que conjuntamente definen la configuración del *software*. Además, se obtiene la buena administración de los cambios a uno o más de dichos elementos, y facilita la construcción de diferentes versiones de una aplicación. Debido a esto, su buena práctica garantiza que la calidad del *software* se conserva conforme la configuración evoluciona a lo largo del tiempo.

Como resultado de este examen, se concreta que el factor de gestión de la configuración toma relevancia en la administración de la calidad porque define y mantiene la integridad de los productos, durante el ciclo de vida del *software* y busca un mejoramiento continuo de su eficacia de acuerdo con los requisitos de normas internacionales. En estos términos los expertos recomiendan incluir este elemento como un factor influyente dentro de la gestión de calidad. (Gabriela Salazar Bermúdez, 2008)

4.3. Pruebas de *software*

Aplicar pruebas de *software* es un factor fundamental en el proceso de aseguramiento de la calidad, pues su principal objetivo es el de encontrar errores introducidos sin intención en el proceso de desarrollo del producto de *software*. Una correcta ejecución de pruebas mencionadas, aporta beneficios al proceso de desarrollo tales como:

- Encontrar defectos
- Ganar confianza acerca del nivel de calidad
- Proveer información para toma de decisiones
- Prevenir defectos

Las pruebas de *software* a menudo representa la acción de mayor esfuerzo del proyecto. Si se lleva a cabo sin orden ni planeación, se cae, sufren una serie de consecuencias como pérdida de tiempo, el incurrir en esfuerzos innecesarios, y lo peor, los errores se cuelan durante el proceso sin ser detectados. Por esto, el establecer una estrategia sistemática de pruebas del mismo, es una tarea de vital importancia para asegurar la calidad.

Frecuentemente, se considera que las pruebas de *software* solo se ejecutan al final del desarrollo, o en el mejor de los casos de manera continua durante el proyecto, cada vez que el equipo de desarrollo entrega una unidad del *software* para ser sometida a pruebas, lo cual en caso de aplicar un buen proceso de pruebas no sería del todo cierto. El proceso de ejecución de pruebas inicia desde las etapas tempranas del proyecto, planeando los escenarios de pruebas desde antes que inicie el proceso de codificación del *software*, revisando documentos del proyecto que son de requerimientos, y haciendo aportes tempranos al proceso de aseguramiento de la calidad, en forma de comentarios y recomendaciones en caso de que sean descubiertos incidentes o errores.

The art of software testing aborda el proceso de pruebas de *software* a un nivel psicológico, dejando clara la relevancia y la complejidad de la adecuada ejecución de pruebas del mismo. En él se dice que “los seres humanos tienden a ser altamente orientado a los objetivos, y el establecimiento de la meta apropiada tiene un importante efecto psicológico en ellos.” Esto con el fin de establecer el correcto enfoque que se debe dar a la planificación de las pruebas, el cual es “Encontrar y corregir errores” en vez de “Comprobar que el *software* funciona correctamente” error en el que muchas veces se incurre y puede llegar a sesgar las pruebas, crear inconscientemente escenarios para comprobar el correcto funcionamiento del *software* en vez de tratar de encontrar los errores en el mismo. (Myers, 1979)

Las pruebas son beneficiosas, ya que se genera ganancia en tiempo y dinero. Asimismo, la planeación de pruebas ayuda a prevenir los errores, formalizar y ejecutar actividades de pruebas disminuye los riesgos, lo cual ayuda en gran medida a lograr un *software* de calidad y cumplir con el proceso de aseguramiento de la calidad.

5. Conclusiones

Para ser competitivo en el mercado del desarrollo de *software* se debe obtener un producto de calidad, pues el mundo actual exige calidad. Por lo cual el aseguramiento de la calidad del *software* viene a ser un proceso indispensable en la industria costarricense y en general. Muchos son los factores que influyen en el aseguramiento de la calidad del *software*, y varían según el marco de referencia o la metodología que se consulte. Además, es de suma importancia tener en claro que no existe un *software* totalmente libre de errores, y también se debe mantener en mente que la calidad es algo subjetivo, por lo tanto no existe metodología o marco de referencia que dé como resultado un producto perfecto. Sin embargo, sí se eligen bien los factores que se deben tomar en cuenta para el proceso de aseguramiento de la calidad, y se les da la relevancia y atención pertinente, el desarrollo del *software* tendrá mejores posibilidades de cumplir con la definición de calidad establecida por el cliente y la empresa.

Según el criterio de los expertos consultados, los factores más relevantes para lograr un *software* de calidad son, la configuración de la calidad, las pruebas que se realicen del mismo y el control de la calidad. Estos tres factores tratan temas fundamentales en el proceso de desarrollo de *software*. Las pruebas de *software* mitigan los riesgos de que el mismo presente defectos, el control de la calidad se encarga de las metodologías utilizadas y los procesos que se deben seguir para lograr un producto de calidad, mientras que la gestión de la configuración identifica los componentes de *software* y controla la evolución de los mismos en el tiempo. Los tres factores cubren puntos fundamentales de su desarrollo, pero por sí mismos no son suficientes si no se complementan con normas, buenas prácticas y especificaciones bien definidas. Por lo tanto, aunque estos factores identificados son de gran relevancia, al igual que cualquier otro marco de referencia o metodología, no aseguran del todo la calidad de esta importante herramienta.

Referencias

- Board, I.-S. S. (2014). Ieee standard for software quality assurance processes. pages 8
- Cueva Lovelle, J. M. (2011). Calidad del software. *Universidad de Oviedo*, 1, 108. pages 5
- Feigenbaum, A. (1984). Total quality management. , 1. pages 2
- Gabriela Salazar Bermúdez, V. L. V., Fulvio Lizano Madriz. (2008). Propuesta de un sistema de gestión de la calidad para un laboratorio de pruebas. , 1. Descargado de http://citic.ucr.ac.cr/sites/default/files/recursos/propuesta_de_un_sgc_para_un_laboratorio_de_pruebas.pdf pages 9
- Group, S. (2013). Chaos manifesto. , 1. Descargado de <https://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf> pages 4

- IEEE. (2014). Standard for software quality assurance processes. *Public Administration Review*, 1(1), 700–706. Descargado de <http://ieeecommons.ieee.org/> pages 2, 4, 5
- ISO. (2015, 9). Iso 9000:2015. , 51. Descargado de http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=03&ics2=120&ics3=10&csnumber=45481 pages 3
- Kaner, C. (2013, febrero). *Software testing as a social science*. pages 6
- Myers, G. (1979). The art of software testing. , 1. pages 10
- Sánchez, J. C. (2011). Estudio de factibilidad para la creación de una pequeña empresa desarrolladora de software en costa rica. *UNED*, 1, 333. Descargado de <http://unpan1.un.org/intradoc/groups/public/documents/icap/unpan047695.pdf> pages 9