

LA CALIDAD DEL SOFTWARE Y EL FUTURO DE LOS DESARROLLOS COLOMBIANOS

Andrés Fernando Bolívar Figueroa*

RESUMEN

El término "calidad del software" cada vez se está haciendo más popular en nuestro territorio, pero raras veces se comprende en su totalidad. Que las empresas que desarrollan software en Colombia estén más pendientes no quiere decir que se satisfagan completamente los requisitos de calidad en la producción de un software. No obstante no pretendo mostrar aquí un compendio de las métricas empleadas para medir la calidad de un desarrollo de software, sino ir más allá de las "nociones" y conceptos, es decir ver el significado de calidad como algo más que mediciones, es tratar de mostrar de una manera simple que no sólo es un aspecto más de la ingeniería del software, sino que conlleva a responsabilidades mas internas por parte del desarrollador.

Palabras clave: Calidad del software, Cliente, Métricas, Ingeniería del software y Madurez del software.

INTRODUCCIÓN

Las organizaciones Colombianas, últimamente están más interesadas en generar productos software con mayor calidad, hemos comenzado a ver una nueva era donde el concepto de "calidad" ya no es simplemente una palabra de nuestro idioma, sino que se ha comenzado a "respetar" más su significado, y no quiero decir con esto que los productos y desarrollos de software tienen los más altos niveles en cuanto a calidad se refiere, sino que las organizaciones, los analistas, ingenieros y en general todas las partes que hacen posible un desarrollo y construcción de software están "poniéndose las pilas" y están viendo más allá de lo que hace algún tiempo ni se presentía.

En este orden de ideas, lo que se busca al integrar la calidad a los desarrollos nacionales, es llevar al software colombiano a un nivel más alto, incluir un valor agregado en lo que se refiere a la calidad no es verla como estándares a cumplir o normas que exige el gobierno, sino es entregarse más como ingenieros y adoptar medidas que lleven a cubrir las necesidades del

mundo real de una manera satisfactoria. Que cumpla con lo más importante en un producto, que es precisamente la excelencia y eficiencia que son realmente la firma de nuestros conocimientos.

No obstante, el concepto de calidad no debe solo estar asociado con métricas, estándares o normas, sino debe ser visto como parte del desarrollador, es decir que sea como un valor inculcado desde la infancia. Pero no solo es un valor individual, debe ser un valor colectivo, un valor organizacional que provea a los procesos de desarrollo y construcción de software las herramientas necesarias para la retroalimentación y sobre todo para estar "al día" en materia de estándares, metodologías y entrega para obtener así productos que realmente valgan la pena y no que den pena.

El problema es bastante claro, la globalización es un hecho, las barreras físicas se han ido perdiendo, el mundo va girando y adoptando nuevas estrategias de mercado y exigencias por parte de los consumidores

* Ingeniero de Sistemas U.A.C. Correos: ferfuac@yahoo.es, ferfuac@gmail.com

del software, que a su vez generan nuevos retos y nuevas situaciones que nos obligan a entrar y estar en la globalidad, generando así nuevas oportunidades de negocio que no se pueden desaprovechar. La mano "invisible" de la que alguna vez hablaba uno de los clásicos de la economía ha hecho incursión dando un empujón y haciendo que las fuerzas de los mercados actuales sean las que rijan los aspectos de calidad en los productos y que las exigencias de los consumidores hagan las veces de estándares y normas a cumplir.

Del mismo modo, la competencia hace que los productos tengan mayores prestaciones con el fin de obtener mejores resultados para retener al cliente, y este tire y afloje provee estándares de desarrollo de mayor calidad y que se adaptan a clientes cada vez menos tolerantes a los errores y sí más estrictos en materia de satisfacción y cumplimiento. En consecuencia, actualmente se puede apreciar que los consumidores de productos software ya no son los que se adaptan al producto, sino que buscan que el producto se adapte a ellos y que en el proceso fabricación de software se adopten medidas y estándares mundiales como normas ISO y metodologías como el proceso unificado de desarrollo de software, entre otras metodologías y normas, que realmente provean de calidad a los desarrollos que ellos adquieren, con el fin de que sus necesidades queden satisfechas y sus exigencias sean cumplidas al pie de la letra.

CALIDAD EN EL SOFTWARE

Tenemos que ser claros en definir que se entiende por la idea de calidad de los desarrollos de software y para esto se debe precisar como es que el concepto de calidad se inserta e integra en un producto de software. En primer lugar, es necesario aclarar que un producto software es distinto de otros productos por poseer ciertas características que lo diferencian frente a otros artículos, aunque se puede decir que es "fabricado" lo más adecuado será decir que el software se ensambla, en una serie de subprocesos, como lo hemos visto en los conceptos tomados de la ingeniería del software (análisis, diseño, implementación, pruebas... etc.) y no se desarrolla como en una línea de producción, sino que tiene que cumplir con las exigencias del cliente. "El software es algo intangible y difícil de medir o cuantificar [Luque, 1999]", es decir, no es palpable y no se daña con el uso normal dado por los usuarios, pero cumple con un proceso de vida útil, es decir "pasa de moda" o ya no cubre nuevas necesidades del cliente y su confiabilidad no se puede asegurar a un ciento

por ciento. Un ejemplo claro de ello son los sistemas operativos de la actualidad, que una vez salidos al mercado comienzan a presentar bastantes fallos y necesitan de actualizaciones y *Service pack*, para poder funcionar más o menos libres de errores. Análogamente a los sistemas operativos, el resto de los productos software también se construyen bajo especificaciones y requieren de actualizaciones o mejoras.

A continuación, se dará una idea de en qué podría radicar la calidad de los productos software, primero que todo, se debe tener claro que es lo que se entiende por calidad, sobre que se puede aplicar y de que forma puede ser relacionada con productos software.

La calidad, al buscar su significado en el diccionario se puede definir como: "conjunto de cualidades que constituyen la manera de ser de una persona o cosa"¹. De esta manera se podría decir que la calidad de los productos puede expresarse como una relación entre lo que el software puede hacer y sus cualidades que lo constituyen. Así, este concepto puede aplicarse a cualquier producto software. La calidad se puede medir realizando una comparación entre dos o más productos observando las diferencias que tengan y eligiendo la mejor opción, la frase popular dice que las comparaciones son "odiosas", pero la calidad parte de estas comparaciones, por ejemplo comparar versiones de sistemas operativos (Windows Xp con Windows Vista) o algo aún mejor comparar productos de empresas distintas por ejemplo (Linux con Windows).

De la misma manera, la calidad es hoy una de las mayores ventajas competitivas para las empresas desarrolladoras de software, no obstante, del mismo afán de competencia se van generando nuevas formas de mejorar y prestar altos niveles de calidad. El concepto de calidad es algo sin valor si no se asumen las necesidades existentes de los consumidores. De tal manera, ciertas definiciones de la misma son válidas en este contexto, por ejemplo algunas de ellas son: calidad es llevar a cabo los requerimientos del cliente, a su vez la calidad puede definirse como satisfacer al cliente o lo que el consumidor requiere y le plantea al desarrollador.

Siendo más formales, una definición más adecuada de calidad en el software sería:

"En general con respecto al software, queremos que funcione (y que lo haga siempre, no aleatoriamente o bajo ciertas condiciones difíciles de cumplir y no prefijadas), que tenga las funcionalidades que dice tener (y por las que lo hemos adquirido) y que po-

¹ Tomado de: diccionario enciclopédico Lexis 22, Círculo de Lectores, 1978.

damos usar estas funcionalidades de una forma natural (o que las podamos aprender de una forma reglada y adecuada) [Piattini, 2003, 4]”.

Otra definición de calidad en el software que se puede decir que es más acertada y expresa de una mejor manera lo que significa este concepto, sería:

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente [Pressman, 2002, 135].”

De acuerdo con las anteriores definiciones, se requiere llevar a cabo los siguientes criterios elementales: lo que el cliente espera debe tenerse muy en cuenta y generar con estas expectativas un modelo adecuado de requisitos para cumplir lo que el consumidor solicita; plantear un modelo y gestionarlo para desarrollar un procedimiento con el fin de satisfacer dichos requerimientos de usuario de una forma adecuada y predecible y ante todo fácil de llevar a cabo, como se mencionó antes, definir las obligaciones de cada integrante del equipo, con el fin de reducir al máximo la ocurrencia de errores en el proceso de construcción del software, contar con un modelo que permita medir de alguna forma la satisfacción del consumidor.

Del mismo modo, tenemos que tener en cuenta que la ingeniería del software apenas está surgiendo, que los errores están presentes todavía en muchos desarrollos (en la mayoría), la experiencia es pequeña en comparación con otras ingenierías que llevan más tiempo y por consiguiente se han ido perfeccionando con el paso de los años. De manera que nos tocó “bailar con la más fea” en el sentido que nosotros como desarrolladores tenemos la dura y penosa tarea de corregir los errores que vayamos descubriendo en esta disciplina cuya perspectiva de madurez está aun lejos, y que se caracteriza por la abundancia de normas, métodos y herramientas muchas veces difíciles de seguir y cumplir.

Aunque la Ingeniería del Software sea todavía muy joven, es preciso tener en cuenta normas y estándares que ya se han ido implantando con las experiencias pasadas, con el fin de minimizar la aparición de errores en el proceso del software.

La ingeniería del software que es la piedra angular en materia de calidad en los desarrollos de software, obedece en muchos de los aspectos a los equipos que desarrollan el producto, y depende en gran parte de la habilidad que estos tengan y a su vez de la experiencia que puedan aportar al desarrollo como tal.

La calidad en la ingeniería del software conlleva a una serie de maneras de establecer la calidad en los desarrollos algunas de ellas pueden ser [Piattini, 2003]:

Eficiencia, flexibilidad, corrección, confiabilidad, facilidad de funcionamiento, facilidad de mantenimiento, facilidad de uso, seguridad, integridad, entre otras.

Obviamente, lo que se busca no es más que el software funcione de una manera adecuada, que haga lo que se espera que haga, siendo muy específicas las funcionalidades que esperamos y que varían de un desarrollo a otro. Es decir, la calidad en la ingeniería del software debe ser el acatamiento de los requerimientos por parte del desarrollo del producto software, a su vez del cumplimiento de las normas y funcionalidades esperadas en el producto final.

Del mismo modo, si no se cumplen los requerimientos, o si el sistema falla (así sea un fallo pequeño) vemos que nuestra percepción de la calidad se ve disminuida y de la misma manera comenzamos a desconfiar del producto y de los que lo desarrollaron. Muchas veces vemos que un fallo en un producto software se puede percibir fácilmente por parte del cliente, sin que el daño sea muy evidente para el desarrollador.

La calidad, la van adquiriendo las organizaciones desarrolladoras de software optimizando el proceso como tal de elaboración del software, que va desde el análisis de requerimientos hasta la fase de pruebas, además de adquirirse también con el buen manejo del recurso humano, la estrategia y gestión del software. Con el fin de mejorar los procesos anteriormente mencionados y a su vez mejorar la calidad en los servicios y desarrollos se debe tener muy en cuenta al cliente, familiarizarse con él, estudiarlo predecir sus necesidades y proveer soluciones adecuadas a los requerimientos de usuario, además es preciso conocer también al sector y a las otras empresas desarrolladoras lo cual es fundamental para prestar mejores servicios y mejorar la gestión del modelo de la calidad establecido.

En este orden de ideas, el conocer al cliente y a la competencia otorgará un significativo incremento de la confiabilidad, generará buen nombre a la organización desarrolladora de software, reducirá la cantidad de “parches” y mejoras al producto, disminuirá la necesidad de realizar mantenimientos, incrementará la satisfacción de los clientes, mantendrá la incursión en el mercado, mejorará la gerencia de los proyectos, ayudará a identificar más rápido y eficientemente los errores y por supuesto con todo esto, la relación costo beneficio se verá favorecida. De la misma manera, también se mejorará la “madurez” en los procesos de de-

sarrollo de software y las metodologías destinadas para mejorarlos para poder adquirir una gestión adecuada de los progresos de dichos procedimientos.

Para establecer y garantizar la calidad en el proceso de software, será necesario tener claro en qué grado de madurez, está la empresa desarrolladora, con el fin de poder tener estrategias y a su vez determinar los pasos a seguir.

Vale la pena anotar, que una empresa desarrolladora que tenga carencia de madurez se identifica por efectuar procesos sin ningún tipo de control, en ocasiones procesos estructurados y bien pensados no son cumplidos a cabalidad ni llevados correctamente a buen término, la falta de madurez se ve reflejada también en los jefes de proyecto que pierden tiempo en asuntos a los que no se les debería dedicar demasiado tiempo, además se exceden los presupuestos destinados al proyecto y no se cumple el cronograma, debido a la falta de experiencia, razón por la cual la calidad se ve reducida.

En el mismo sentido, la inmadurez de la organización influye de manera significativa en la calidad del producto, debido a que por tratar de cumplir con el cronograma, se disminuye o prescinde realizar acciones de incremento de la calidad tales como: revisiones de casos de uso, revisiones de código, pruebas de caja blanca, pruebas de caja negra etc. Además de descuidar otros aspectos como estándares de procesos y normas internacionales. Con todo esto se puede apreciar que una empresa desarrolladora con un bajo grado de madurez, incurrirá en errores que comprometerán su buen funcionamiento y su buen nombre, y obviamente el producto se verá afectado significativamente.

La organización desarrolladora, en su proceso de creación de un software debe tener características adecuadas de madurez y fiabilidad, que conlleven a un excelente desarrollo, estable, ávido de calidad y fiel a todas las partes del proceso. Deberá tener unas bases sólidas en recursos humanos, que son la esencia elemental y definitiva para llevar a cabo cualquier desarrollo y a su vez garantizar la calidad del producto; es decir, cualquier evento o cambio en la empresa deberá ser realizado por el recurso humano y éste también será el encargado del cumplimiento de los procesos y de la preservación de la calidad. Además, deberán tener estrategias para la gestión de procesos y control de los mismos. También, es necesario cumplir con un proceso de verificación y medición constante, con el fin de proveer decisiones correctas ya sean detectivas, preventivas o correctivas para garantizar el éxito en la calidad.

El Instituto de Ingeniería del software (Software Engineering Institute) o SEI de la Universidad de Carnegie-Mellon, se fundamentó en la experiencia de la producción del software a nivel global, para establecer un método que permitiera evaluar el proceso de desarrollo de software y a su vez, valorar en que grado de madurez se puede encontrar la organización, dicho método es llamado: CMM (Capability Maturity Model), Modelo de Madurez de la Capacidad para desarrollos Software, este modelo, detalla las bases y fundamentos en los cuales una empresa desarrolladora pueda controlar, medir y gestionar su madurez en la producción de software.

A su vez, el CMM, se basa en una estructura que influye y establece una ruta para gestionar las mejoras que la empresa debe cumplir, para evolucionar de lo desordenado y extemporáneo a lo ordenado y oportuno, cumpliendo con estándares y normas establecidos para tal fin. Por una parte se centra en la exposición y análisis de los procesos de software, en contraste al interés por los productos software. Por ejemplo: definición y desarrollo de los requisitos del software; generación de datos de prueba; planificación de la instalación del software. Por otra parte, se fundamenta en que la "madurez" del proceso es un indicador para construir software de calidad. En el nivel de la estrategia, puede ayudar a identificar una matriz DOFA, es decir identificar debilidades, oportunidades, fortalezas y amenazas asociadas con las organizaciones desarrolladoras y su proceso de producción.

El CMM cuenta con cinco niveles de "madurez" con unas áreas claves del proceso, que tienen prácticas clave a llevar a cabo, que a su vez comprenden preguntas específicas enfocadas a la calidad del proceso de software que son independientes a las actividades del marco de trabajo y están presentes durante todo el desarrollo.

Este modelo, basado en investigaciones y encuestas realizadas por el SEI busca que las organizaciones desarrolladoras establezcan y se percaten en qué nivel se halla actualmente su proceso de software, cuáles deben ser los pasos a seguir y las correcciones a priori que se deben llevar a cabo y en qué particularidades o acciones vitales del proceso de creación del software, que sean fundamentales para satisfacer las condiciones necesarias para garantizar un desarrollo con alta calidad, que siga estándares y cumpla tiempos establecidos.

Con el fin de determinar en que nivel se encuentra la organización, se determina el grado según el modelo de capacidad de madurez (CMM), el cual ayuda a de-

finir que actividades del proceso de software son claves y que nivel tienen en el proceso de desarrollo

Para determinar el estado de madurez de un proceso el SEI provee una medida dada por cinco niveles [Pressman, 2002, 17], definidos a continuación:

Nivel 1: Inicial. El proceso del software se caracteriza según el caso, y ocasionalmente incluso de forma caótica. Se definen pocos procesos, y el éxito depende del esfuerzo individual.

Nivel 2: Repetible. Se establecen los procesos de gestión del proyecto para hacer seguimiento del coste, de la planificación y de la funcionalidad. Para repetir éxitos anteriores en proyectos con aplicaciones similares se aplica la disciplina necesaria para el proceso.

Nivel 3: Definido. El proceso del software de las actividades de gestión y de ingeniería se documenta, se estandariza y se integra dentro de un proceso de software de toda una organización. Todos los proyectos utilizan una versión documentada y aprobada del proceso de la organización para el desarrollo y mantenimiento del software. En este nivel se incluyen todas las características definidas para el nivel 2.

Nivel 4: Gestionado. Se recopilan medidas detalladas del proceso del software y de la calidad del producto. Mediante la utilización de medidas detalladas, se comprenden y se controlan cuantitativamente tanto los productos como el proceso del software. En este nivel se incluyen todas las características definidas para el nivel 3.

Nivel 5: Optimización. Mediante una retroalimentación cuantitativa del proceso, ideas y tecnologías innovadoras se posibilita una mejora del proceso. En este nivel se incluyen todas las características definidas para el nivel 4.

Para cumplir las metas de madurez descritas anteriormente, es aconsejable emplear los "pasos" escritos a continuación con el fin de cerciorarse el cumplimiento de la calidad del proceso de software y de los productos software.

Exponer claramente y de forma concreta las responsabilidades, productos a entregar, etc. con el fin de definir un proceso de desarrollo con políticas adecuadas.

Estar incentivando al equipo desarrollador, a su vez proveerle de la debida capacitación con el fin de que estén actualizados en materia de normas y estándares.

Desarrollar el producto, siguiendo siempre los requerimientos del cliente en todo el proceso de software, con el fin de evitar demoras ya sea en la corrección de errores o en mejoras innecesarias.

Estimar mediante métricas tales como puntos de función y/o el modelo COCOMO los recursos que se requieran: Recurso humano, Equipos de cómputo, software, Material de oficina, etc. Determinar las funciones de cada integrante del equipo de desarrollo, llevando un estricto control de cada módulo asignado con el fin de no salirse del presupuesto y estar dentro del cronograma establecido.

Cumplir rigurosamente los estándares dados por la ISO por ejemplo el ISO 9001 entre otras normas.

Llevar a cabo una documentación adecuada de los modelos de análisis y diseño, además de la correspondiente documentación de la estructura y el código, con el fin de evitar demoras en la capacitación del personal nuevo que se integre al equipo de desarrolladores.

Realizar los manuales del software, bien detallados y que cumplan realmente el objetivo sin excederse demasiado o ser muy escuetos.

De la misma manera, es pertinente conocer cuáles son las capacidades de la empresa y estructurar la forma de desarrollo para no comprometerse con proyectos que no se puedan llevar a cabo de una forma adecuada o cumpliendo con el tiempo y los recursos destinados para tales proyectos. Además será necesario contar con un equipo alterno que de soporte a los desarrollos y que ante todo gestione y controle el proceso de la elaboración del software.

Con el fin de satisfacer los referente a la calidad, las organizaciones desarrolladoras deben tener un recurso humano idóneo teniendo en cuenta son las capacidades reales de la empresa, "el núcleo de la fábrica de software es el equipo de desarrollo [Piattini, 2003, 5]", debido a que una organización con deficientes prestaciones de soporte de recursos humanos a la calidad, se verá obligada a salir del mercado por no satisfacer las necesidades y requerimientos que los usuarios le plantean, además pierde opciones de competencia con otras empresas mejor preparadas para cumplir con criterios de calidad.

En este orden de ideas, desarrollar productos software de calidad requiere de constancia, disciplina, estudio, capacitación, motivación, perseverancia, además de tener los pies sobre la tierra y no comprometerse con desarrollos ilógicos o que excedan las facultades reales de la organización. La experiencia tiene mucho que

ver en los desarrollos de calidad, se podría decir que a mayor experiencia mayor será la calidad, es decir, la calidad se va adquiriendo y generando paulatinamente a medida que se van elaborando proyectos y no se puede pretender tener alto grado de calidad en el primer proyecto elaborado.

Con el fin de dar cumplimiento a los requerimientos del usuario y satisfacer plenamente la calidad del desarrollo, en la organización debe estar presente una forma de reconocimiento de los requerimientos del cliente y de sus necesidades actuales. Al elaborar software sin que sea aprobado previamente por el cliente, sin la comunicación efectiva cliente-desarrollador, no se conseguirá tener la calidad requerida. Hacer lo que determine el cliente, es fundamental para tener satisfecho al mismo, pero si lo que el cliente exige está fuera del alcance del desarrollo, se le deberá orientar y mostrarle una forma más adecuada de llevar a cabo el proyecto.

Como se mencionaba anteriormente, la calidad no se adquiere de la noche a la mañana, debe ser construida mediante un constante proceso de evolución que conlleve al mejoramiento diario y continuo que sea duradero y que genere crecimiento y se adapte al medio, donde lo que realmente se lleve a cabo sea la realización plena de los requerimientos del cliente y el cubrimiento de las necesidades del mismo.

Los desarrolladores deben centrarse en la calidad, e implementarla en todos los desarrollos, debido a que esto implica la evolución de la organización, a su vez provee las herramientas necesarias para la administración de la calidad, que desplegada con una planeación estratégica lleve a producir una proyección de la calidad en los desarrollos futuros, a la creación de modelos de calidad y ante todo a la gestión y verificación de la calidad de las creaciones obtenidas en la producción del software.

El término de la calidad, está asociado al concepto de organización y secuenciación de los trabajos, debe estar orientado a resultados visibles de manera rápida y para conseguir esto es necesario generar un plan estratégico que incluya: la gestión de los procesos de calidad con intervalos de mejoramiento de corta duración, identificar los errores y reaccionar de manera inmediata y vertiginosa donde no haya cabida para el retraso y que por supuesto sea el que mejores resultados provea, que la estrategia esté ligada al objetivo de la organización, establecer únicamente las tácticas totalmente indispensables que aseguren mejoramiento de la capacitación del recurso humano, motivación,

entrega, gestión, control, modelos, metodologías y cumplimiento.

Si se aplica con sensatez, la metodología de cumplimiento de la calidad, el proyecto tendrá resultados óptimos y no fracasará, es obvio que si se cumplen los requerimientos del cliente y se desarrolla con estándares de calidad no habrá cabida a errores, pero no se deberá exceder la gestión de la calidad, el ¿por qué?, simplemente, al ser tan minuciosos el tiempo de construcción y los costos se dispararían y se perdería el objetivo de la ingeniería del software.

El objetivo principal en un equipo de desarrollo de software, debe ser el de originar constantemente software que tenga las características necesarias para ser de alta calidad y excelente desempeño. Con el fin de dar cumplimiento a este objetivo se deberá tener en cuenta los siguientes aspectos:

1. El recurso humano es el encargado de los desarrollos de software, es decir son creados por un equipo, el cual en su totalidad debe estar lo suficientemente enterado del problema a solucionar que presente el cliente, de la misma manera debe estar atento en todo momento para detectar fallos cometidos durante el proceso del software y generar una solución de manera oportuna.
2. Normalmente se relaciona el concepto de calidad con la elaboración de algún producto, pero no sólo se debe mirar así, la calidad debe ser una característica inherente al recurso humano; que sea como un valor inculcado desde la infancia. Entonces una organización que desee mejorar los aspectos de calidad del software, deberá abarcar a todo el personal que de una u otra manera tenga que ver con el proceso de producción. El concepto de calidad involucra muchos factores y el recurso humano es uno de los más importantes, debiendo ponerse atención extrema al control ejercido al personal.

En síntesis, un equipo de desarrollo en el que la calidad sea su responsabilidad principal, se debe ser el que ejecuta y posee lo siguiente:

- Crea y ejecuta medidas para asegurar la interacción adecuada con el cliente, con el fin de satisfacer las necesidades que éste requiere de forma apropiada.
- Capacita y gerencia al recurso humano, generando así una cultura de cumplimiento y que por supuesto conlleve a tener desarrollos de calidad.

- Se desempeña con objetivos claros, siguiendo las especificaciones del cliente, teniendo muy en cuenta los estándares y normas de calidad en los productos de software.
- Genera una estructura organizacional, con sentido de eficiencia y efectividad, sin descuidar el tener un adecuado nivel de comunicación entre las personas del equipo.
- Aplica métricas para controlar y supervisar los desarrollos y encaminarlos hacia la calidad.
- Tiene muy en cuenta los aportes que haga el cliente y lo guía para obtener los resultados esperados, evitando errores de análisis y diseño.
- Debe ser un equipo de desarrollo sólido y unido que fortalezca a la organización impidiendo que se lleven a cabo productos fuera de tiempo o con baja calidad.
- Implanta el respeto por el recurso humano, incentivando a los desarrolladores para el trabajo en equipo y creando un buen clima organizacional.
- Está en constante retroalimentación, aprendiendo de los errores cometidos en ocasiones pasadas, llevando un registro y documentación de los mismos.

La casa de desarrollo, debe generar estrategias para garantizar productos de calidad, y ésta tarea es en buena parte responsabilidad de la parte administrativa y gerencial, desde donde se "inculca" el cumplimiento de la calidad en toda la organización. La calidad debe estar presente en todas las etapas y procesos de producción del software, empezando por el análisis y el diseño; la organización debe mantener siempre un nivel alto de la calidad en todas las fases de desarrollo, pruebas, comercialización y mantenimiento.

Para finalizar, los conceptos anteriormente descritos pueden resumirse en cumplimiento, organización y satisfacción por parte de los clientes. Pero cualquier proceso que pretenda implantar un modelo de calidad debe ser instaurado de manera paulatina, porque la aceptación de un modelo de calidad de software generará resistencia al cambio. Es preciso que la organización entienda la importancia de la gestión de calidad y para qué sirve y esto únicamente se logra con esfuerzo, seguimiento de estándares y cumplimiento a los clientes enfocando los desarrollos hacia la calidad de los productos.

FACTORES PARA EL CUMPLIMIENTO DE LA CALIDAD

De acuerdo con lo anterior, la calidad que pueden conseguir los productos software, y cualquiera de los desarrollos informáticos realizado bajo requerimientos de usuario, está subyugada a como se desarrolla cada una de las fases de la existencia y ciclo de vida del producto, partiendo por la definición de la representación conceptual del producto hasta el desarrollo y mantenimiento del mismo.

De esta forma, darle calidad a un producto contempla los siguientes factores entre otros:

- Gestión de la calidad, asegurando disminuir las diferencias entre los recursos determinados previamente y los recursos realmente empleados en las distintas partes del desarrollo. Dichos recursos incluyen el costo de oficina, el equipamiento y tiempo de desarrollo del producto de software.
- Aplicación de conocimientos de Ingeniería de Software, teniendo en cuenta técnicas de desarrollo, metodologías e instrumentos que faciliten el diseño.
- Utilización de técnicas formales a lo largo de todo el desarrollo, tales como el proceso unificado de desarrollo entre otros.
- Reducción de las modificaciones entre los productos, disminuyendo las incompatibilidades y fallas entre distintas versiones del producto.
- Aplicación de pruebas en todas las etapas del desarrollo.
- Llevar a cabo documentación de cada etapa, tanto de apoyo al desarrollo como la entregada al usuario final.
- Prestar un apoyo técnico correcto y proveer de mantenimiento preventivo y correctivo al usuario final.
- Flexibilidad.
- Reusabilidad.
- Fiabilidad
- Eficiencia
- Interoperatividad.
- Satisfacción del usuario.
- Aporte y solución a las necesidades del consumidor.

Algunos de los anteriores factores los propone McCall, proporcionando las siguientes descripciones [Pressman, 2002, 324]:

- Corrección.** Hasta dónde satisface un programa su especificación y logra los objetivos propuestos por el cliente.
- Fiabilidad.** Hasta dónde se puede esperar que un programa lleve a cabo su función con la exactitud requerida.
- Usabilidad** (facilidad de manejo). El esfuerzo necesario para aprender a operar con el sistema, preparar los datos de entrada e interpretar las salidas (resultados) de un programa.
- Facilidad de mantenimiento.** El esfuerzo necesario para localizar y arreglar un error en un programa.
- Flexibilidad.** El esfuerzo necesario para modificar un programa que ya está en funcionamiento.
- Facilidad de prueba.** El esfuerzo necesario para probar un programa y asegurarse de que realiza correctamente su función.
- Portabilidad.** El esfuerzo necesario para transferir el programa de un entorno hardware/software a otro entorno diferente.
- Reusabilidad** (capacidad de reutilización). Hasta dónde se puede volver a emplear un programa (o partes de un programa) en otras aplicaciones, en relación al empaquetamiento y alcance de las funciones que realiza el programa.
- Interoperatividad.** El esfuerzo necesario para acoplar un sistema con otro.

va aplicando la gestión de la calidad en el proceso de software, tales como el análisis, diseño, codificación, pruebas y en el producto final.

En la fase de análisis, la calidad se va implantando con el desarrollo de un buen proceso de recolección de los requisitos del cliente, de esta manera, el equipo desarrollador tendrá plena comprensión de lo que el cliente desea en el software, además podrá identificar necesidades y la forma de satisfacerlas. Es importante tener una buena gestión de la calidad en ésta fase de análisis, debido a que la misma es la base de cualquier producto software.

En este mismo sentido, es fundamental que haya un estricto control de la calidad en el análisis debido a que es completamente indispensable empezar con la documentación precisa de los requisitos a desarrollar, además, cumpliendo con la gestión de calidad de puede descomponer el problema de una mejor manera, entendiendo más y descubriendo detalles que en un proceso de análisis desordenado serían imposibles de encontrar. Para ello se emplean una serie de métricas que presentan una ayuda fundamental para garantizar la gestión de la calidad tales como: métricas basadas en la función, que ayudan a estimar el tamaño del desarrollo; métrica bang, que al igual que la anterior proporciona una indicación para establecer el tamaño de software; métricas de calidad de la especificación, que son una serie de características para evaluar la calidad en el modelo de análisis y la respectiva especificación de requisitos.

En la fase de diseño, la calidad debe estar presente, debido a que esta etapa se compone de varios pasos cuya finalidad es la de proveer la estructura de datos necesaria para la elaboración del producto, elegir y establecer la arquitectura de software, elaboración de bocetos de interfaz y realizar diagramas de diseño que provean la guía necesaria para la codificación. En este orden de ideas, el proceso de diseño tiene como finalidad refinar lo realizado en la fase de análisis, preparando características esenciales para llevar a cabo el desarrollo del software con el fin de proveer las herramientas necesarias para comenzar a escribir código.

Además, el proceso de diseño se debe efectuar cumpliendo una serie de estándares y métricas de desarrollo, lo que significa que la calidad estará implícita en el desarrollo, para ello se debe seguir una de las normas internacionales de calidad, una de las cuales puede ser el estándar **ISO 9001**, empleando las reglas **ISO 9000-3:1993** que proporcionan una guía para las organizaciones de software que requieren un registro en mencionado estándar. En esta fase el ni-

La calidad en el software se debe a un buen desarrollo durante todo el proceso de software, es decir la calidad se forja en cada una de las partes del proceso de creación de un producto software. De manera global, se puede identificar la intervención de la calidad en las fases de desarrollo. A grosso modo se verá como se

vel de calidad se ve aumentado, debido a que con el seguimiento de las normas se pueden identificar errores más fácilmente.

Para las empresas desarrolladoras de productos software, los estándares internacionales de mayor importancia para la fase de desarrollo son [Pressman, 2002, 146]:

1. **ISO 9001.** Quality Systems-Model for Quality Assurance in Design, Development, Production, Installation and Servicing. Este es un estándar que describe el sistema de, calidad utilizado para mantener el desarrollo de un producto que implique diseño.
2. **ISO 9000-3.** Guidelines for Application of ISO 9001 to the Development, Supply and Maintenance of ISO 9004-2. Quality Management and Quality System Elements, Part 2. Este documento proporciona las directrices para el servicio de facilidades del software como soporte de usuarios.

Además de emplear estándares de desarrollo para garantizar la calidad, se deberán adoptar métricas que hagan de la fase de diseño un fuerte eslabón de la cadena de desarrollo, las cuales pueden ser: métricas del diseño arquitectónico: para establecer la arquitectura del producto software, métricas de diseño a nivel de componentes: las cuales se usan para medir la cohesión, acoplamiento y la complejidad y las métricas para el diseño de la interfaz: para especificar como se debe elaborar la interfaz de usuario.

Cuando el software se está codificando, la calidad aparece en cuanto se cumpla con las especificaciones de la fase de diseño en el momento de comenzar la codificación del mismo. De manera tal, si los requerimientos de usuario están bien determinados y detallados, la obtención de la calidad es producto de las especificaciones llevadas a cabo en la programación.

La calidad en el producto software es la parte más importante, como se mencionó antes, lo más importante es la plena satisfacción del cliente y que el producto cumpla con los objetivos deseados. De manera tal, se puede afirmar que la calidad en la producción es consecuencia directa de la gestión de calidad de las fases anteriores y es el factor más importante en la producción de un software. A su vez, existen también métricas para medir las características de calidad del software ya producido tales como [Somerville, 2002, 602]:

Métricas dinámicas, que son recogidas por las mediciones hechas en un programa en ejecución.

Métricas estáticas, que son recogidas por las mediciones hechas en las representaciones del sistema tales como el diseño, el programa o la documentación.

Por otro lado, controlar la calidad es fundamental y consiste en efectuar una vigilancia permanente de la realización de las actividades que logran brindar una calidad neutral a la forma en como se está desarrollando un proyecto de software, debe haber revisión constante y un estricto control al transcurso del proceso de software, con el fin de garantizar que se están siguiendo los estándares y normas que garanticen la calidad. Este objetivo se logrará, a medida que se vaya revisando paso a paso y aplicando las metodologías de desarrollo de software, aplicando métricas y pruebas a los productos en su última fase.

El apropiado control de la calidad, comprueba que las entregas del software cumplan con los estándares definidos, a su vez, permite identificar los errores y efectuar las correcciones apropiadas por parte del equipo de desarrollo en el proceso de construcción, evitando de esta manera que el software se produzca con errores prevenibles. Esto implica que las mejoras han sido implementadas gracias al continuo proceso de revisión y control de calidad en las fases de desarrollo y a la experiencia del equipo desarrollador.

Uno de los factores fundamentales en el control de calidad, es el de la creación constante de documentación, ya sea en la etapa de análisis y diseño o en la producción como tal. Dicha documentación permite mejorar la forma en que se entiende el desarrollo, además permite mejorar el tiempo de capacitación del equipo (en caso de mejora o ampliación del producto), con lo cual la calidad se va viendo reflejada. También, facilita la incorporación de nuevos miembros al equipo de desarrollo, debido a que no tendrán que descifrar código sino revisar y estudiar la documentación del mismo.

Mencionado control de calidad, se apoya además en tecnología no solo para la realización eficiente del proyecto como tal, sino como fuente de herramientas que permitan controlar mejor el manejo de errores y a su vez disminuir costos en las deficiencias o fallas detectadas.

Con el propósito de comprobar la calidad en el desarrollo de un proyecto software, se proponen los siguientes dos enfoques [Somerville, 2002, 597]:

Revisiones de la calidad, donde el software, su documentación y los procesos utilizados en su desarrollo son revisados por un grupo de personas. Se encargan de revisar que se han seguido los estándares del pro-

yecto y del software y los documentos concuerdan con estos estándares. Se toma nota de las desviaciones de los estándares y se comunican al gestor del proyecto.

Valoración automática del software en la que el software y los documentos producidos se procesan por algún programa y se comparan con los estándares que se aplican a ese proyecto de desarrollo en particular. Esta valoración automática comprende una medida cuantitativa de algunos atributos del software.

En resumen, el control de calidad es un elemento que genera unas bases sólidas durante el proceso de software, y como se sabe una base sólida da como resultado una estructura fuerte, además que reduce el costo de desarrollo al identificar con antelación errores que pueden ser catastróficos en el producto terminado.

CALIDAD EN LOS DESARROLLOS COLOMBIANOS

De acuerdo a las definiciones de calidad vistas anteriormente, lo que se pretende al adquirir un software es que éste haga lo que se supone que debe hacer, de una manera correcta y sin errores, que esté dentro del tiempo que está estipulado para su desarrollo, que tenga un ciclo de vida largo (o por lo menos suficiente, para que con su uso se recupere la inversión) y que la organización que lo adquirió se sienta satisfecha y vea que se están cubriendo las necesidades por las cuales obtuvo el software. En este orden de ideas, la calidad de software producido en Colombia debe cumplir con lo anteriormente expuesto, además debe estar suficientemente soportado (técnicamente), con sus respectivos manuales, documentación y ante todo cumpliendo con los estándares de desarrollo, con el fin de generar excelencia de producción y confianza en los clientes que lo adquieran.

Ahora bien, ¿Será que los desarrollos de software producidos en nuestra nación con el fin de satisfacer necesidades de clientes exigentes, cumplen con los estándares necesarios, normas y métricas, que les den el calificativo de productos con calidad óptima?

Desafortunadamente, la percepción que hay es que los desarrollos nacionales no parece que tengan ningún control de calidad, luego, pienso que Colombia no cuenta con desarrollos de buena calidad, no se trata solo si el aplicativo software, funciona o no, o si tiene control de concurrencia o cosas por el estilo, se trata de también de prever que tan ampliable o modificable puede ser en el futuro con el fin de cubrir nuevas necesidades del cliente, que tan satisfecho se siente el consumidor con el producto software y si su funcio-

namiento y ciclo de vida, aportan a las necesidades de las organizaciones que los adquieren.

Nuestros productos software, tienen la desventaja de no ser planeados de una manera correcta, debido a que se dedica bastante tiempo a mejorar errores y a extender funcionalidades faltantes, que por falta de comunicación entre el cliente y el desarrollador se pasaron por alto. Este tiempo perdido dedicado a mejorar el producto se podría evitar si en el proceso de análisis y diseño se tuvieran en cuenta los aportes del cliente, a su vez que el equipo encargado del análisis, tuviera una visión más holística y previera las futuras necesidades del cliente. Otro factor fundamental para "ahorrar" tiempo invertido en mejoras es dedicarle mayor tiempo al análisis, para que de esta manera se obtenga un resultado mejor y más cercano a lo que el consumidor desea.

El problema en los desarrollos colombianos actuales, radica en la falta de implementación de técnicas que produzcan resultados óptimos, es decir falta regirse de una manera adecuada y llevar procesos estandarizados. Si en las empresas desarrolladoras de software colombianas, se siguiera una metodología, cumpliendo a cabalidad cada uno de sus pasos, seguramente se obtendría una mejora significativa en materia de calidad y eficiencia, asegurando así que el producto va a cumplir con ciertas características que lo harán ser un software con calidad garantizada y que seguramente cumpla con los requerimientos del cliente y lo satisfaga plenamente.

Llevar a cabo una metodología, implica cumplirla al pie de la letra, con un equipo preparado para su ejecución y seguimiento, a su vez llevar correctamente una metodología ayuda a ir cumpliendo los requerimientos de usuario y así poder ir revisando con ayuda del mismo, cada uno de los avances de los módulos entregables, para poder seguir adelante y pasar a la siguiente fase, una vez sea aprobada por el mismo cliente. De esta manera habrá una comunicación adecuada entre el equipo encargado del análisis y el usuario, lo que genera orden y confianza en ambas partes.

La falta de metodología en los desarrollos colombianos, es realmente la piedra en el zapato en materia de mejoramiento de la calidad en los productos. Se sabe que el recurso humano colombiano tiene gran demanda en el mundo, precisamente por la inteligencia que tenemos y que desafortunadamente no aprovechamos para mejorar los procesos y generar investigación que conlleve a ser un poco más estrictos y metódicos. Es algo evidente que nosotros los colombianos hacemos todo a última hora, de afán, creando estrategias para tratar de cumplir a tiempo. Si fuéramos más discipli-

nados seguro que Colombia sería potencia mundial en desarrollar software de calidad, porque poseemos las capacidades intelectuales y empeño requeridos.

De la misma manera, al hacer todo de afán se nos olvida practicar las correspondientes pruebas a los productos que desarrollamos, y como consecuencia se presenta una lista de errores que se hubieran podido evitar al ser más disciplinados y al llevar correctamente una metodología adecuada.

Otro factor fundamental en el rezago del desarrollo colombiano, es precisamente la falta de capacitación permanente al equipo de desarrollo que conlleve al mejoramiento de las técnicas y métodos de creación de software y que actualice permanentemente al recurso humano con el fin de estar al día en materia de métricas normas y estándares aplicables a cualquier parte del proceso de software.

Además, el cumplimiento de políticas de la organización es otro aspecto fundamental y clave para tener productos con mayor calidad, es indispensable seguir patrones de diseño, estándares ISO, métricas de software, cumplir con el cronograma etc. Todo esto, con el fin de tratar de proveer aspectos de calidad para que se mejoren nuestras creaciones.

Para finalizar, es indispensable que el software elaborado en nuestras casas de producción criollas, tenga un nivel de madurez adecuado, que provea de herramientas al cliente, que sea ampliable, que tenga en

cuenta futuras necesidades del usuario, con el fin de prestar mejores funcionalidades, que el ciclo de vida del programa no se acabe antes de que genere las "ganancias" necesarias para cubrir su costo, además que los usuarios de nuestras organizaciones se sientan satisfechos con los resultados obtenidos en los productos desarrollados en las empresas nacionales de "fabricación" de software.

CONCLUSIONES

Finalmente, se puede observar en el contenido de este artículo que la calidad de productos software es el resultado de muchos procesos, los cuales al tener garantizada la calidad proveen la misma al producto final, además para el usuario final, es decir el cliente que va a hacer uso del software, la calidad está dada por el buen funcionamiento del producto y por el cubrimiento dado a las necesidades que este requiere.

La calidad en los desarrollos colombianos está aún muy precoz, debido a que tenemos la particularidad de dejar todo para el final, sin dar cubrimiento real a los requerimientos de usuario y por tanto tener desarrollos deficientes y de baja calidad; no obstante si nos empeñamos en cubrir factores clave que garanticen la calidad de nuestros desarrollos podremos despegar en materia de creación de software y así lograr ser la próxima potencia mundial en desarrollo, en ingenieros de sistemas y por supuesto en productos con calidad.

BIBLIOGRAFÍA

Luque Ruíz Irene, INGENIERÍA DEL SOFTWARE: FUNDAMENTOS PARA EL DESARROLLO DE SISTEMAS INFORMÁTICOS, editorial Alfaomega Ra-ma, 1999

Mario G. Piattini Velthius, CALIDAD EN EL DESARROLLO Y MANTENIMIENTO DEL SOFTWARE., editorial Alfaomega Ra-ma, 2003

Roger Pressman, INGENIERÍA DEL SOFTWARE. Un enfoque práctico. (Quinta Edición), editorial McGraw-Hill, 2002.

Sommerville Ian, INGENIERÍA DEL SOFTWARE (Séptima Edición), editorial Pearson Addison Wesley, 2002.

