



Scientia Et Technica

ISSN: 0122-1701

scientia@utp.edu.co

Universidad Tecnológica de Pereira
Colombia

Castro Bermúdez, Y.V.; Solarte Martínez, G. R.; Muñoz Guerrero, L. E.
Planificación, Gestión y Control de la Calidad del Software
Scientia Et Technica, vol. 24, núm. 4, 2019, Julio-Septiembre, pp. 611-617
Universidad Tecnológica de Pereira
Colombia

Disponible en: <https://www.redalyc.org/articulo.oa?id=84961238009>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en [redalyc.org](https://www.redalyc.org)

redalyc.org
UAEM

Sistema de Información Científica Redalyc
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Planning, Management and Control of Software Quality

Planificación, Gestión y Control de la Calidad del Software

Y. V. Castro Bermúdez , G. R. Solarte Martínez , L. E. Muñoz Guerrero 

Abstract—At present, software companies are part of an important economic sector worldwide, because they are more lucrative and well-paid every day, generating significant transformations at a social and economic level, in the same way with the progress of the business and knowledge management; At present, software companies are part of an important economic sector worldwide, because they are more lucrative and well-paid every day, generating significant transformations at a social and economic level, in the same way with the progress of companies and knowledge management; in order to obtain quality software, it implies the use of agile methodologies, or international standardized models of ISO quality for the analysis, design, programming and testing of software that allow unifying the work philosophy, in order to achieve greater “reliability, maintainability and analysis”, while raising productivity, both for development work and for software quality control; In this research, a study of the latest scientific and bibliographic publications is carried out, such as: Standards, policies, methodologies or procedures for the analysis, design, development, programming and control tests of software established for its elaboration, as evidenced in this article; The policies established for its elaboration are related to the definition of standards, in order to develop competitive software at the regional and international level it is required to be connected to the ISO / IEC 12207 „, ISO / IEC 15504 SPICE, ISO / IEC 14598 certification systems and ISO 25000, this to consider if a company is mature. As a result, Colombia could be one of the interpreters of the new digital economy, through coordinated actions, if the concept of software quality is consolidated to become an exporter of this service, in order to develop competitive software in the international arena it is required to be connected to the certification systems, this is because today it is considered a poorly mature industry that from some points of view looks like "a craft rather than an industry"

Index Terms—Control, Estimation, Planning Quality, Technical.

Este manuscrito fue enviado el 01 de mayo de 2018 y aceptado el 26 de junio de 2019. Este trabajo fue soportado por la Universidad Tecnológica de Pereira. Facultad de Ingeniería de Sistemas, Universidad Tecnológica de Pereira, Risaralda Colombia es un trabajo derivado del proyecto de investigación, estrategia aprendizaje a través del TIC, tipo de trabajo es, de investigación, sobre y Control de Calidad del Software

Yanci Viviana Castro Bermúdez, Ingeniera de Sistema y Computación yvcastro@utp.edu.co. Profesora de universidad tecnológica de Pereira.

Guillermo Roberto Solarte Martínez, Doctor en Informática de la Universidad Pontificia de Salamanca con sede Madrid España Suficiencia

Resumen— En la actualidad las empresas de software son parte de un sector económico importante a nivel mundial, debido a que cada día son más lucrativas y bien remuneradas, generando transformaciones significativas a nivel social y económico, de la misma manera con el progreso de las empresas y la administración del conocimiento; En la actualidad las empresas de software son parte de un sector económico importante a nivel mundial, debido a que cada día son más lucrativas y bien remuneradas, generando transformaciones significativas a nivel social y económico, de la misma manera con el progreso de las empresas y la administración del conocimiento; para lograr obtener un software de calidad, implica la utilización de metodologías agiles, o modelos estandarizados internacionales de calidad ISO para el análisis, diseño, programación y pruebas de software que permitan unificarla filosofía de trabajo, en aras de lograr una mayor “confiabilidad, mantenibilidad y análisis”, a la vez que elevan la productividad, tanto para la labor de desarrollo como para el control de la calidad del software; en esta investigación se realiza un estudio de las últimas publicaciones científicas y bibliográficas como: Estándares, políticas, metodologías o procedimientos para el análisis, diseño, desarrolló, programación y pruebas de control de software establecidas para su elaboración, como se evidencia en este artículo; las política establecidas para su elaboración esta relacionadas con la definición de estándares, para desarrollar software competitivo en el ámbito regional e internacional se requiere estar conectado a los sistemas de certificación ISO/IEC 12207,, ISO/IEC 15504 SPICE ,ISO/IEC 14598 y ISO 25000, esto para considerar si una empresa es madura. Como resultado Colombia podría ser uno de los intérpretes de la nueva economía digital, mediante acciones coordinadas, si se consolida el concepto de calidad de software para convertirse en exportador de este servicio, para desarrollar software competitivo en el ámbito internacional se requiere estar conectado a los sistemas de certificación, esto porque hoy es considerada una industria pobremente madura que desde algunos puntos de vista se parece a “una artesanía más que a una industria”

Palabras claves—Calidad, Control, Estimación, Planificación, Técnicas.

investigativa, D.E. A Universidad Pontificia de Salamanca con sede Madrid España, Magíster en Investigación de Operativa y Estadística de la Universidad Tecnológica de Pereira roberto@utp.edu.co.

Luis Eduardo Muñoz Guerrero, Msc. En Ingeniería de Sistemas. Universidad Nacional de Bogotá, profesor Asociado, de planta, del programa en ingeniería de Sistemas y Computación

I. INTRODUCCIÓN

SEmpieza con un conjunto de actividades que en grupo se llaman PLANIFICACION DEL PROYECTO [1]. Antes de que el proyecto comience, el gestor del proyecto y el equipo de software deben estimar el trabajo que habrá de realizarse, los recursos que se requerirán y el tiempo que transcurrirá desde el principio hasta el final, con base en los diferentes planes que intervienen; teniendo en primer orden la calidad, luego la validación, la gestión de configuración, no se debe olvidar el mantenimiento y como elemento envolvente el plan de desarrollo personal que contempla las habilidades y experiencias del equipo del proyecto.

Una vez que se completen estas actividades, el equipo de software debe establecer un Plan De Proyecto, que defina las tareas y fechas clave de la Ingeniería del software, que identifique quién es el responsable de dirigir cada tarea y especifique las dependencias entre tareas que pueden ser determinantes del progreso.

II. PLANIFICACIÓN Y ESTIMACIÓN DE UN PROYECTO

La planificación requiere que los gestores técnicos y los miembros del equipo de software establezcan un compromiso inicial y para dar un soporte integral e integrado han de tener cuenta algunas destrezas adquiridas tales como las menciona [2] en su libro “Gestión Integrada de Proyectos”, pagina 300. Aunque la estimación es tanto un arte como una ciencia, esta importante actividad NO condiciona realizarse en una forma improvisada. Puesto que la estimación coloca los cimientos para las demás actividades de planificación del proyecto y ésta proporciona la ruta para la ingeniería del software exitosa, se estaría mal aconsejando si se embarcara sin ella, pero como se sabe en algunos países aún se inician cualquier cantidad de mal llamados Proyectos de Software, a veces son mejores de lo que parece los de aula que los de este tipo; sin proyecciones, sin manejo o gestión de riesgos, la calidad empírica y otros males venideros en las grietas del futuro software [3].

En una actividad de ingeniería de software, más o menos gestionada, con algún grado de seriedad en este aspecto, el riesgo de valoración se mide por el grado de incertidumbre que proviene de las estimaciones cuantitativas establecidas para recursos, costos y programa de trabajo, la administración ayuda en anticipar los posibles riesgos que afectan a la programación del proyecto o a la calidad [4] del mismo. También al desarrollar u organizar acciones que impidan que los riesgos se materialicen. En pocas palabras la gestión de riesgos del proyecto beneficia la minimización de los posibles tropiezos, tratando de llevar a cabo todas las tareas y en el tiempo requerido con la calidad provista.

A. Técnicas y métodos para la planificación y estimación.

Como menciona [5], “La estimación se define como el proceso que proporciona un valor a un conjunto de variables para la realización de un trabajo, dentro de un rango aceptable

de tolerancia. Podemos definirlo también como la predicción de personal, del esfuerzo, de los costes y de la planificación que se requerirá para realizar todas las actividades y construir todos los productos asociados con el proyecto”. Siendo una de las primeras tareas la estimación de costo y esfuerzo. Sin embargo, la estimación general del proyecto se puede transformar de una práctica transparente o en otros casos oscura en observancia a una serie de pasos sistemáticos que proporcionan estimaciones con riesgo aceptable, en la trilogía calidad-costo-tiempo.

Para conocer la estimación de un proyecto de software, se pueden revisar dos técnicas ampliamente conocidas que son:

B. Técnica de descomposición

Harold S Geneen, dice: “Lees un libro de principio a fin. Usted tiene un negocio de la manera opuesta. Se empieza por el final, y después de hacer todo lo imprescindible para llegar a él”. En el caso de proyectos de ingeniería de software, se puede pensar de forma semejada, para la realización, empezar por ver cuáles son los objetivos que se han de alcanzar y luego abstraer que cosas tenemos que hacer para alcanzar estos fines.

Esta descomposición pasará por identificar las fases de nuestro proyecto y el esfuerzo a aplicar en cada una de ellas, a su vez estas fases se descompondrán en tareas. También tendremos que marcar unos puntos de control que algunos denominan hitos, estos permiten saber si el proyecto va de acuerdo con lo previsto.

La estimación del proyecto de software es una forma de resolver problemas, en la mayoría de los casos complejos como para considerar una sola pieza, por ello, se descompone el problema.

De las técnicas recientes existen las WBS (Works Break Down Structure) [6], en las cuales todavía hay técnicas que se explican a nivel de pregrado y postgrado, como puntos de función, métricas asociadas a la tarea, el proceso o la función.

C. Modelos empíricos de estimación

Un proceso de explotación de información puede definirse como un conjunto de tareas relacionadas lógicamente, que se ejecutan para extraer conocimiento no-trivial que reside de manera implícita en los datos disponibles en distintas fuentes de información. Una metodología de explotación de información permite gestionar la complejidad de estos procesos de manera uniforme. Entre estas metodologías, la comunidad científica considera aprobada la metodología CRISP-DM [7]. Resultados previos señalan la necesidad de abordar la estimación empírica de proyectos de explotación con base en proyectos [8] realizados para pequeños y medianos emprendimientos.

Los datos empíricos que apoyan la mayoría de los modelos

de estimación proceden de una manera limitada de proyectos, por esta razón ningún modelo de estimación es apropiado para todas las clases de software ni en todos los entornos de desarrollo.

III. ESTIÓN DE LA CALIDAD DEL SOFTWARE

La calidad del software según es una metodología que ayuda a recopilar evidencias del uso de un producto o software, para garantizar esta calidad se utilizan directrices para medir el control de la calidad de este, además de prolongar la vida del software. La correlación entre ocupación de la auditoría, garantía de calidad, control de calidad y prueba de software frecuentemente se complica. La garantía de calidad es el conjunto de tareas que sirven como soporte para proveer una adecuada seguridad donde cada uno de los procesos creados es permanentemente optimizado, tratando que los productos cumplan con los requerimientos y especificaciones del producto o software [9], además es necesario que cumplan con las condiciones ideales para su uso. El encargado de comparar la calidad del producto con las políticas adoptadas y las medidas establecidas de no conformidad cuando se detecta una inconsistencia se conoce como control de calidad. [9] La auditoría es la encargada de la evaluación y verificación del seguimiento de las normas, procedimientos y planes.

IV. MARCO NORMATIVO RELACIONADO CON LA CALIDAD

Actualmente existen en el mercado “normativo” varias opciones, entre las que se destacan:

ISO 9001 Es el cargador de medir la importancia del software y los procesos productivos de la organización. Por ejemplo, en la entrega y mantenimiento del software.

ISO/IEC 9003 Este modelo está orientado a la Ingeniería de software, sirve como indicador de la norma ISO 9001:2000, es un conjunto de tareas y procedimientos que han tenido éxito en el concepto de software sobre los procesos de la organización.

ISO/IEC 12207 *Information Technology / Software Life Cycle Processes*, son normas que se utilizan para el ciclo de vida del proceso de software de una organización, además sirve de apoyo para ISO 15504-SPICE.

ISO/IEC 15504 SPICE - Software Process Improvement and Assurance Standards Capability Determination). Este estándar está conformado por 7 modelos que sirven de base para la implantación, mejoramiento (capacidad) y madurez de los procesos de las organizaciones, se utilizan para la evaluación de la calidad de los procesos. La descripción de los métodos se realiza teniendo en cuenta la norma ISO/IEC 12207. La familia4 de normas 15504 espera que la nueva ISO 29110 sea publicada para crear definitivamente el esquema internacional de certificación, que actualmente está creado con procesos de calidad en las entidades de certificación (realizando evaluaciones externas sobre ISO/IEC 15504-2 e ISO/IEC TR 15504-7:2008.

A. Capability Maturity Model Integration (CMMI)

Un modelo es una representación simplificada del mundo. Modelos como CMM, contienen los elementos esenciales de los procesos efectivos de uno o más cuerpos de conocimiento. Estos elementos se basan en los conceptos desarrollados por Crosby, Deming, Juran y Humphrey [Crosby 79, 88 Juran, Deming 86, Humphrey 89]. [FM108.T101] Al igual que otros modelos CMM, Capability Maturity Model Integration (CMMI) [10], permiten orientar y ser utilizados en el desarrollo de los procesos. Los Modelos CMMI no son procesos o descripciones de procesos. [11]

Los procesos reales utilizados en una organización dependen de muchos factores, incluyendo el dominio de aplicación(es), de la estructura y del tamaño de la organización(es). En particular, las áreas de proceso de un modelo CMMI, típicamente no es una correspondencia de uno a uno con los procesos utilizados en su organización. Por otro lado, hoy en día el modelo CMMI [12] es una exigencia casi en todo el mundo para poder acceder a la exportación de servicios de software. Esta guía que nos permite crear estrategias de calidad y mejoramiento de los procesos de una organización que tiene como función el desarrollo, mantenimiento de software, además cuenta con un programa de certificación para empresa privada (no ISO).

ISO/IEC 9126. Software engineering – Product quality, creada entre 1991 y 2001 Esta norma está dividida en 4 etapas, la primera consiste en un conjunto de normas ISO/IEC 9126 que tiene que ver con la calidad del producto de software, la segunda y tercera etapa trabaja con la medición (métricas internas y externas), la cuarta está dedicada hacia la calidad de uso, de acuerdo con condiciones particulares.

ISO/IEC 14598. Software product evaluation, creada entre 1999 y 2001. Está conformada por 6 partes que están interrelacionadas con la familia ISO 9126, esta es la encargada de la parte de evaluación de producto de software

ISO 25000. Hace parte del modelo de calidad para el producto de software, está conformado por 5 fases que se encuentran en desarrollo, fue creada para reemplazar la ISO 9126 e ISO 14598 ya que desde 2001 no se publicaron nuevas versiones.

SCRUM. Es una metodología muy fácil y práctica de aprender está encargada de la parte de gestión de los proyectos y deja de lado la parte de desarrollo de software. Está conformada por 3 etapas fundamentales, la primera etapa tiene que ver con la planificación para tener una visión general del proyecto (estimación y viabilidad del proyecto), la otra etapa se dedica a la construcción de backlog y distribución de tareas utilizando Sprint, la última etapa consiste en informar al grupo de trabajo un recuento del éxito o fracaso logrado.

V. FACTORES Y MODELOS DE CALIDAD

Se clasifican en: Operaciones del producto: Para medir la calidad un producto debe cumplir con las siguientes características operativas:

Corrección: Es la capacidad que tiene un programa para satisfacer los requerimientos y objetivos planteados por el usuario. ¿Hace lo que se le pide?

Fiabilidad: Cuando un programa (todas sus funciones) generan resultados con la precisión requerida ¿Lo hace de forma fiable todo el tiempo?

Eficiencia: Cuando un programa ejecuta sus funciones con tiempo de respuesta adecuado, ¿Qué recursos hardware y software necesito?

Integridad: Es la capacidad de restricción que tiene un programa para acceder a los datos por usuarios no autorizados, para eso se utilizan tipos de usuarios (administrador, empleado, usuario) ¿Puedo controlar su uso?

Facilidad de uso: Es el grado de usabilidad que tiene el programa, como por ejemplo que sea fácil usar, aprender y de utilizar (¿Es fácil y cómodo de manejar?).

A. Revisión del producto: capacidad para soportar cambios.

Facilidad de mantenimiento: La capacidad de encontrar y reparar errores en programas, los programas deben ser flexibles, permitir la reutilización y portabilidad ¿Puedo localizar los fallos?

Flexibilidad: Es la capacidad que tiene un programa en permitir introducir o modificar código de un programa ¿Puedo añadir nuevas opciones?

Facilidad de prueba: el grado de dificultad que presenta un programa para realizar pruebas de funcionamiento. ¿Puedo probar todas las opciones?

Transición del producto: adaptabilidad a nuevos entornos.

Portabilidad: Es la capacidad de transferir un programa a otro entorno HW y/o SW a otro. ¿Podré usarlo en otra máquina?

Reusabilidad: Es la capacidad de reutilizar un programa o componente SW en otras aplicaciones. ¿Podré utilizar alguna parte del software en otra aplicación?

Interoperabilidad: grado de acoplamiento que tiene un sistema o aplicación con otro ¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?

B. Modelo.

Desde el principio de la ingeniería de software, se observó que la calidad es una serie de combinaciones de diferentes

metodologías. Por lo tanto, un modelo de calidad se puede representar mediante la descripción de sus características y sus relaciones entre sí.

Modelo de McCall. Este modelo se dio a conocer en el año de 1977 y está dirigido hacia el producto final, se dedica a la identificación de atributos claves a partir de la perspectiva del usuario, estos atributos se conocen como factores de calidad, estos comúnmente son factores externos, pero en algunos casos también se tienen en cuenta factores internos.

Modelo de Boehm [13]. Se dio a conocer por Barry Boehm en el año 1978, uno de los aspectos más importantes es que introduce características de alto nivel, intermedio y primitivas donde cada uno aporta al nivel general de calidad.

Modelo FURPS. Tiene las características de los modelos anteriores, pero incluyen otros aspectos como funcionalidad, usabilidad, confianza, performance, soportabilidad.

Modelos ad-hoc [14]. Este modelo sirve para el monitoreo de la calidad del software, existen dos caminos:

Adoptar un camino fijo. Adoptar un modelo fijo, se supone que todos los factores de calidad importantes son un subconjunto de los de un modelo publicado, se acepta el conjunto de criterios y métricas asociados al modelo.

Desarrollar un modelo propio de calidad. Se acepta que esté compuesta por varios atributos, pero no se adopta lo impuesto por modelos existentes. En este último caso, se debe consensuar el modelo con los clientes antes de empezar el proyecto, se deciden cuáles atributos son importantes para el producto y las medidas específicas que los compone.

C. Metodología para el aseguramiento de la calidad.

La metodología descompone cada uno de los procesos en actividades y éstas a su vez en tareas. Se describen sus principales elementos:

- Planificación de Sistemas de Información (PSI)
- Desarrollo de Sistemas de Información
- Estudio de Viabilidad del Sistema (EVS)
- Análisis del Sistema de Información (ASI)
- Diseño del Sistema de Información (DSI)
- Construcción del Sistema de Información (CSI)
- Implantación y Aceptación del Sistema (IAS)
- Mantenimiento de Sistemas de Información (MSI)

D. Gestión y control de riesgo

Los riesgos son amenazas para el proyecto, ya sea en menor o mayor grado de afectación para el software que está en desarrollo para la organización, a continuación, se describen algunas maneras o métodos de plantearlos, sin ser exhaustivos en esta apreciación, algunas vigentes por estándares, otras por

experiencias adquiridas y otras por debido proceso y cumplimiento de las bases legales:

Clasificaciones fallas básicas:

Naturaleza Incendio, Inundación, Terremoto, tiene que ver con desastres naturales.

Hardware: Daño del computador, la unidad daña el disco, daño de la impresora, daño del equipo de telecomunicaciones, daño en la transmisión entre otros.

Humanas Error de: Reporte, Transcriptor, Transmisión, Programador, Operador, otros.

Humanas Intencionales: Saqueos, Sabotajes con o sin violencia, Violación de la privacidad, Fraude, otros.

En el contexto estratégico del "decreto 1599 del 20 de mayo de 2005" para la administración del riesgo, establece: "Para la formulación y operacionalización de la política de administración del riesgo es fundamental tener claridad de la misión institucional, sus objetivos y tener una visión sistemática de la gestión de manera que no se perciba esta herramienta gerencial como algo aislado del mismo accionar administrativo. Por ende, el diseño se establece a partir de la identificación de los factores internos o externos a la entidad que pueden generar riesgos que afecten el cumplimiento de sus objetivos".

De lo que se describe en el punto de identificación de riesgos presentes en diferentes metodologías se presenta este cuadro básico ver Tabla I:

TABLA I.
BÁSICA DE IDENTIFICACIÓN DE RIESGOS.

Tipo	Categorías	Subcategorías	Ejemplos
Internos	Proyecto	Genéricos Específicos Impredecible	Retraso de firmas en documentos Fallas del equipo de trabajo, no se puede predecir
	Software	G/E/I	Cambio de fecha de pruebas
	Negocio	G/E/I	Cambio de fechas cierre. Licencias
	técnicos	G/E/I	
Externos	Entorno	Genéricos Específicos Impredecibles	Falta de inversión Cambio regulatorio y tributarios, no se puede predecir

Fuente: Naim Jhon Cruzado Paredes [15]

Ya mencionado el Análisis cualitativo de los riesgos, este método evalúa la prioridad de los riesgos de acuerdo con la prioridad de ocurrencia, impacto de ocurrencia y alcance del proyecto como costo, cronograma y calidad ver Tabla II:

TABLA II.
ANÁLISIS CUALITATIVO DE LOS RIESGOS

Amenaza	Prob * impac to	Ri es go	Causas	Estrat egia	Respu sta de riesgo	Riesgo Owaer	Cu mp lió
Falta de experiencia del programador en java	0,4	Técnico	Poca disponibilidad de un experto java	Mitigar	Capacitación al programador	Analista funcional	8
Entrega de componentes después de la fecha prevista	0,2	Software	Demoras entrega de componentes	Transferir	Penalizar al proveedor	Gerente del proyecto	5
Estimación de costos totales del proyecto por debajo de los costos reales	0,2	Proyecto	Falta de experiencia, no contar con la información necesaria al momento de la estimación	Mitigar	Reserva del presupuesto	Gerente del proyecto	5

Fuente Naim Jhon Cruzado Paredes: [15]

Existen varias definiciones sobre riesgo de software, la gran mayoría está de acuerdo que el riesgo implica dos características:

Incertidumbre: Suceso que caracteriza al riesgo puede o no ocurrir, por ejemplo, no hay riesgos de un 100 por ciento de probabilidad.

Pérdida: Es cuando un riesgo se convierte en realidad y produce consecuencias no deseadas o pérdidas económicas.

Un aspecto importante a tener en cuenta en los riesgos es medir el nivel de incertidumbre y el grado de pérdidas relacionado con cada riesgo. Para hacerlo, se consideran diferentes categorías de riesgos.

E. Tipos de riesgos asociados al software.

Los tipos de riesgo más comunes son:

Tamaño del producto: riesgo relacionado con el tamaño general del software a construir o a modificar.

Impacto en el negocio: riesgo relacionado con las restricciones asignadas por la gestión o por el mercado.

Características del cliente: riesgo relacionado con el cliente y la habilidad del desarrollador para comunicarse con el cliente en los momentos oportunos.

Definición del proceso: el software y su seguimiento por la organización de desarrollo.

Entorno de desarrollo: riesgos relacionados con la

disponibilidad y calidad de las herramientas que se van a emplear en la construcción del producto.

Tecnología para construir: riesgo relacionado con la complejidad del sistema a construir y la tecnología de punta que contiene el sistema.

Tamaño y experiencia de la plantilla: riesgo relacionado con la experiencia técnica y proyectos de los ingenieros de software que van a realizar el trabajo.

Control y supervisión de riesgos.

Una vez se haya identificado, analizado y priorizado los riesgos del proyecto se pueden someter a control.

El control de riesgos cubre dos actividades: resolución de riesgos y supervisión (monitorización).

Supervisión. Consiste en realizar una búsqueda de los riesgos durante la ejecución del proyecto. Un buen método es realizar seguimientos periódicos (semanales, por ejemplo) de los riesgos más significativos.

Resolución. Qué hacer en este punto depende mucho del tipo de riesgo. Sin embargo, las acciones o estrategias que se pueden seguir son las siguientes:

Evitar el riesgo: es decir, no realizar actividades arriesgadas cambiando el plan del proyecto.

Trasladar el riesgo a un tercero (habitual en riesgos económicos) o trasladarlo a otra parte menos crítica del sistema.

Conseguir información acerca del riesgo cuando éste no es muy conocido.

Mitigar el riesgo. No se elimina el origen, pero su cambia el plan para que su exposición sea menor.

Comunicar el riesgo al resto del equipo, al cliente y a la dirección, para que estén prevenidos.

Controlar/Aceptar el riesgo. Aceptar que puede ocurrir y hacer un plan de contingencia para minimizar su exposición.

Recordar el riesgo para planes futuros.

VI. CONCLUSIONES

El éxito en la producción de software se obtiene si se hace con calidad y se demuestra el grado de ésta, calificado como buena. Esto sólo es posible con la implantación de un Sistema para el Aseguramiento de la Calidad del Software directamente relacionado con la política establecida para su elaboración y que esté en correspondencia con la definición internacional ISO de calidad, ampliamente aceptada por los estándares del grupo ISO.

Recomendación se sugiere la aplicación de normativas establecidas a nivel internacional, mediante la aplicación de métodos de calidad al software a desarrollar. Con ello se logra garantizar la calidad de este y el logro del objetivo.

Actualmente, una de las empresas más lucrativas del mundo es una de software, situación no pensable hace unos 40 años, cuando las empresas que integraban esas listas eran otras muy aparte de este sector. En ese sentido, el desarrollo de software constituye un sector de cardinal importancia mundial, se encuentra en el centro de todas las grandes transformaciones; sobre todo si se considera que los grandes temas del momento, como lo son la economía digital [16], la evolución de las empresas y la administración del conocimiento, se resuelven con software.

La industria del software se interpone en todos los procesos que habilitan a la que se le considera una industria clara o blanda que no contamina y que genera fuentes de trabajo generalmente bien remuneradas o por lo menos es la creencia. Actualmente hay movimientos hacia otros mercados fuera de los nacionales. Colombia podría ser uno de los intérpretes de la nueva economía digital, mediante acciones coordinadas, si se consolida el concepto de calidad de software para convertirse en exportador de este servicio. Para desarrollar software competitivo en el ámbito internacional se requiere estar conectado a los sistemas de certificación, esto porque hoy es considerada una industria pobremente madura que desde algunos puntos de vista se parece a “una artesanía más que a una industria” [17]

REFERENCIAS

- [1] J. Molina, R. Gimeno, M. S. Figueroa y J. Freixa, “Gestión Integrada de Proyectos,” *Universidad Politécnica de Catalunya*, vol. 31, p. 427, 2006. Editorial Virtuales. <http://sedici.unlp.edu.ar/handle/10915/19326>
- [2] R. S. Pressman. Somerville, I. “Ingeniería del software 7/e.” *Editorial Pearson Educación*, pagina 88. 2010. http://artemisa.unicauca.edu.co/~cardila/Libro_Pressman_7.pdf
- [3] R. S. Pressman, Ingeniería del Software. “Un Enfoque Práctico Séptima”, *editorial McGraw-Hill*, México: 2010. <http://cotana.informatica.edu.bo/downloads/Id-Ingenieria.de.software.enfoque.practico.7ed.Pressman.pdf>.
- [4] G. Mario, V. Piattini, “Calidad del Producto y Proceso Software,” editorial Ra-Ma, EAN9788478979615, ISBN 978-84-7897-961, p.665. Edición1, fecha publicación 11-02-2010 https://www.ra-ma.es/libro/calidad-del-producto-y-proceso-software_49157/
- [5] D. Rodríguez, P. Cattaneo, P. V. María Florencia Britos y G. M. Ramón “Estimación empírica de carga de trabajo en proyectos de explotación de información. In XVI Congreso Argentino de Ciencias de la Computación,” 2010. ISBN: 978-950-9474-49-9. <http://sedici.unlp.edu.ar/handle/10915/19326>
- [6] M. Alcívar Loor, “Estructura de Desglose de Trabajo como herramienta para la Planificación de Proyectos”, revista RIEMAT Vol. 1 Núm.2 Julio 2016, DOI: 10.33936/riemat.v1i2.919.
- [7] I. Lamarca Orozco, J. García Minguez, R. Rodríguez José Ramón Rodríguez “Gestión de proyectos informáticos: métodos, herramientas y casos”, vol. 4 nº 2 p.2-18, 2018 ISBN: 9788497882927, <http://www.editorialuoc.com/gestion-de-proyectos-informaticos-metodos-herramientas-y-casos>.
- [8] G. Garita, F. Lizano, “Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO”, editorial UNICIENCIA Vol. 32, No. 1, pp. 118-133. Enero-junio, 2018. DOI: 10.15359/ru.32-1.8
- [9] CMMI, para desarrollo, “Equipo del producto CMMI mejora de los procesos para el desarrollo de mejores productos y servicios,” noviembre 2010. Technical Report, Software Engineering Process Management

- Program. <https://www.coursehero.com/file/17419284/CMMI-Dev-13-Espa%C3%B1ol/>
- [10] M. Rodríguez, M. Piattini "Revisión sistemática sobre la certificación del producto software", editorial Computar Science and Engineering Special Issue: Pag.16-24 Julio 2012. DOI: 10.5923/j.computer.20120001.03
- [11] M. Palacios, V. Gisbert Soler "Metodología de aplicación de CMMI servicios en pequeñas y medianas empresas", editorial 3C Empresa, Pag.92 – 100 m, diciembre 2017, DOI:10.17993/3cemp.2017.especial.92-100
- [12] [12] CMMI, para Desarrollo, Versión 1.3 CMMI-DEV, V1.3 "Equipo del Producto CMMI Mejora de los procesos para el desarrollo de mejores productos y servicios," noviembre 2016. <https://www.coursehero.com/file/17419284/CMMI-Dev-13-Espa%C3%B1ol/>
- [13] M. Callejas-Cuervo, A. C. Alarcón y A. M. Álvarez-Carreño, "Modelos de calidad del software, un estado del arte Vol. 13 No. 1, 2017 enero - junio," Editorial Entramado, vol. 13, nº 1, p. 10, 2017. <http://www.scielo.org.co/pdf/entra/v13n1/1900-3803-entra-13-01-00236.pdf>.
- [14] P. Elinbaum., "Planes fuera del sistema. Instrumentos ad hoc para la ordenación y gestión de las áreas urbanas plurimunicipales," editorial EURE (Santiago), vol. 42, nº 127, p. 12, 5 09 2016. DOI: 10.4067/S0250-71612016000300002
- [15] Cruzado, Paredes John, "Dirección de Proyectos bajo el enfoque de PMI, Perú, 2010. http://naimcruzado.blogspot.com/2010_04_01_archive.html
- [16] L. S. Valencia, P. A. Villa a, C. A. Ocampo, "La calidad de software, la academia y la empresa," editorial Scientia et Technica Año XVI, No 48, agosto 2011. Universidad Tecnológica de Pereira. ISSN 0122-170, 20011. DOI: <http://dx.doi.org/10.22517/23447214.1253>
- [17] R. L. Herrera González, A. Hidalgo Nuchera "Dinámica de la gestión de la innovación de servicios y co-creación en empresas del sector economía digital," editorial Contad. Adm vol.64 no. spe1 México 2019. DOI: 10.22201/fca.24488410e.2018.1802



Luis Eduardo Muñoz Guerrero, (Pasto, Nariño, Colombia 1975). Msc. En Ingeniería de Sistemas Universidad Nacional de Bogotá, profesor Asociado, de planta, del programa en ingeniería de Sistemas y Computación. Ha publicado artículos en revistas especializadas nacionales e internacionales. Pertenece al grupo de investigación GIA: grupo de inteligencia artificial ORCID: <https://orcid.org/0000-0002-9414-6187>



Yanci Viviana Castro Bermúdez Plata, Antioquia -Colombia 1989). Ingeniera de sistemas y computación de la Universidad Tecnológica de Pereira. Ha publicado artículos en revistas nacionales especializadas.

ORCID: <https://orcid.org/0000-0003-2078-2460>



Guillermo Roberto Solarte Martínez, Doctor en Informática de la Universidad Pontificia de Salamanca con sede Madrid España Suficiencia investigativa, D.E.A Universidad Pontificia de Salamanca con sede Madrid España, Magíster en Investigación de Operativa y Estadística de la Universidad Tecnológica de Pereira. Ingeniero de Sistemas de la Universidad Cooperativa de Colombia. Profesor ISC-UTP, Transitorio Titular en la Facultad de Ingenierías. Ha publicado artículos en revistas nacionales especializadas. Libros publicados: Guía didáctica de estructuras de datos, Pertenece a los grupos de investigación Grupo Grande Grupo de Avanzada en Desarrollo de Software, Grupo GIA - Grupo de investigación en Inteligencia Artificial, Semillero de Investigación GNTO Grupo de Nuevas técnicas de búsqueda y de optimización.

ORCID: <https://orcid.org/0000-0001-5147-7798>