ISSN: 1692-7257 - Volumen 2 - Número 24- 2014

Revista Colombiana de
Tecnologías de Avanzada

Recibido: 10 de Mayol de 2014
Aceptado: 4 de Julio de 2014

REVIEW OF METHODOLOGIES FOR RISK MANAGEMENT IN BUILDING A SOFTWARE METHODOLOGY PROJECTS FOCUSED ON TESTING STAGE

REVISIÓN DE METODOLOGÍAS DE GESTIÓN DE RIESGOS PARA CONSTRUCCIÓN DE UNA METODOLOGÍA EN PROYECTOS SOFTWARE ENFOCADA EN LA ETAPA DE PRUEBAS

Ing. Sandra Patricia Manrique Mesa, PhD Ricardo Llamosa Villalba

Universidad Industrial de Santander
Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software
E-mail: Sandramanrique31@gmail.com, nrllamos@gmail.com

Abstract: To achieve compliance with software projects in enterprises presents risk management at different stages to identify and control potential adverse events that may prevent the successful and timely delivery of projects. But there are projects where risks are identified only but were not making the management and monitoring these. It is for this reason that this paper addresses the issue of risk management in the different stages of a software project and since in the literature review is not a methodology that focuses on the testing stage and detail the steps at this stage given that it is a stage that has high importance and need to address the issues promptly to halt the delivery of the project because it is the last stage in software projects.

Keywords: Software projects, risk management, software testing, software companies.

Resumen: Para lograr el cumplimiento de los proyectos software en las empresas se presenta la gestión de riesgos en las diferentes etapas para identificar y controlar posibles eventos negativos que pueden evitar el éxito y entrega a tiempo de los proyectos. Sin embargo existen proyectos donde sólo se identifican los riesgos pero no se les realiza la gestión y el seguimiento a estos. Es por esta razón que en el presente trabajo se trata el tema de gestión de riesgos en las diferentes etapas de un proyecto software y dado que en la revisión bibliográfica no se presenta una metodología que se enfoque en la etapa de pruebas y detalle los pasos en esta etapa teniendo en cuenta que es una etapa que tiene alta importancia y necesidad de solucionar con prontitud los temas que detengan la entrega del proyecto debido a que es la última etapa en los proyectos Software.

Palabras clave: Proyectos Software, gestión de riesgos, pruebas software, empresas de software**.**

1. INTRODUCCIÓN

Con la inclusión de las TI (Tecnologías de Información) en el progreso de la sociedad se generó un gran cambio en las diferentes formas de trabajo. Los computadores remplazaron a las máquinas de escribir, las TI fueron la solución para muchas cosas, pero también tenían dificultades, los usuarios pedían solución y los desarrolladores se las ofrecían, sin embargo seguían teniendo diferencias, debido a esto resulto la necesidad de mejorar en muchos aspectos y por esto nacen los métodos como el ciclo de vida de desarrollo software, es en ese momento que se tienen en cuenta los riesgos que se pueden tener con los nuevos inventos como lo son las TI y por esta

razón se originó la necesidad de las pruebas, el aseguramiento de la calidad y la gestión de riesgos

La ausencia o desatención de los riesgos[1] en todo tipo de procesos ha sido, frecuentemente, una constante. Por ejemplo, en la industria software, sólo el 25% de sus proyectos cumple con el plazo y los costos programados (Charette, 2005). Estos escenarios, plantean como necesario, el establecer estrategias para predecir la ocurrencia de las situaciones positivas o negativas que puedan afectar los procesos y los proyectos. Las estrategias que se sugieran deben instaurarse ideológicamente sustentadas en líneas base epistemológica (cognitiva o de conocimiento), ontológica (con sustento real y objetivo) y metodológica (con acciones sistémicas y sistemáticas) para lograr pronosticar el riesgo de la ocurrencia en niveles de probabilidad, con lo cual, será posible desarrollar planes de acción preventivos, de mitigación o de recuperación ante los riesgos negativos o planes de explotación de las acciones que deben hacerse para aprovechar las oportunidades brindadas ante los riesgos positivos, para cuando se hagan realidad los eventos inciertos, se garantice la continuidad o la mejora a los procesos y los proyectos.

Considerar la gestión de riesgos como estrategia para la mitigación, recuperación, aceptación o fortalecimiento de la continuidad del negocio gubernamental, empresarial o social, es esencial para todas las organizaciones. Como ejemplos, se pueden encontrar: el Capabiliy Maturity Model Integration CMMI, producto de una alianza estratégica de la NASA y la Universidad de Carnegie Mellon, para crear modelos en el que se consideran los procesos de gestión de riesgos de contratación, desarrollo o prestación de servicios tecnológicos; las guías o estándares del PMBOK potenciadas en certificaciones profesionales de prácticas de gestión de riesgos en proyectos; o el estándar AS/NZS 4360:1999 del gobierno australiano, que define las prácticas de gestión de riesgos para cualquier actividad.

Ahora, el mejor resultado de un proyecto software se determina por la calidad del proceso software y dentro de este proceso se encuentra la etapa de pruebas al cual no se le da la importancia que se le debe proporcionar, dado que para la solución de los problemas relacionados con la etapa de pruebas se debe tener un proceso bien definido y gestionar y controlar los riesgos que se presentan en la construcción de las pruebas, la cual inicia desde la fase de definición de requisitos, teniendo en cuenta los riesgos a partir de los requisitos y se irá desarrollando en forma paralela al desarrollo del software.

Este trabajo de investigación propone integrar y enfocar en el proceso de pruebas de software la gestión de riesgos para mejorar el desarrollo y la entrega de los proyectos software buscando comprometer conscientemente, individuos y colectivos, en la planeación, el aseguramiento, el control y el desarrollo del proceso con calidad y en el tiempo que el cliente lo solicita.

El artículo está organizado de la siguiente manera: en la sección 1 se presenta la fundamentación teórica de la gestión de riesgos, la sección 2 se describe el proceso de las pruebas de desarrollo software y la sección 3 se refiere a la metodología.

2. RIESGOS Y GESTIÓN DE RIESGOS

En la literatura clásica, el riesgo se ve como la variación en la distribución de probabilidad asociada a la ocurrencia de eventos negativos o positivos que pueden afectar el desarrollo de un proceso. Frecuentemente (Charette, 1999) comenta que el riesgo se reduce al análisis de los eventos que impactan negativamente a un proceso sin considerar el ciclo de vida de sus productos o servicios. Por otra parte (Boehm y Covenin, 1995) especifican al riesgo como una exposición de los procesos a factores, que representados en amenazas, pueden afectar los resultados esperados por la potencial pérdida económica causada en el proceso por la ocurrencia de eventos no deseados. La norma AS/NZS 4360:1999, El Project Management Institute (PMI y Silberfich, 2009) concluyen que el riesgo está asociado a las condiciones que desencadenan un evento no previsto e incierto, produciendo un efecto positivo o negativo en los objetivos de un proyecto que causan impacto en la organización y que deberían predecirse con estimaciones y suposiciones con la información limite que se posee del contexto. En síntesis, el riesgo implica dos dimensiones: La Incertidumbre, establecida en la posibilidad de ocurrencia de eventos no previstos y el efecto o consecuencia, por la realidad de lo que ocurra cuando un evento no previsto se presenta.

[1] El riesgo es la vulnerabilidad que presentan los bienes, productos o servicios ante un potencial perjuicio o daño. En los procesos, se relaciona con la ocurrencia de eventos no previstos que afectan la continuidad de evolución de su planeación, aseguramiento, control y ejecución. (PMBOK, 2004).

Los riesgos se clasifican en tres grupos (Pressman, 2002). (1) Los riesgos de proyecto que afectan el plazo de suministro de recursos, aspecto que afecta la planificación, costo y la calidad del proyecto, con problemas potenciales que se desprenden de presupuesto, agenda, personal e infraestructura. 2) Los riesgos del producto, que afectan la calidad o el desempeño del servicio en desarrollo y se identifican con los posibles problemas de incertidumbre, especificación de requisitos de sistema, proceso, diseño, implementación, verificación y mantenimiento. 3) Los riesgos del negocio que afectan a la organización que desarrolla o suministra un producto o servicio y que amenazan la viabilidad y estrategia por el uso del producto/servicio en el mercado.

Para transformar, relacionar y aplicar los riesgos pragmáticamente implica la existencia de una metodología para definir cómo gestionar los riesgos en los procesos o proyectos (Boehm, 1990), (Thayer, 2003), (NTC5254, 2005) y en la figura No. 1 se establecen un conjunto de principios y prácticas que están estructuradas en un proceso que conducen a la planificación, identificación, análisis, respuesta, seguimiento, tratamiento y control de los factores de riesgo y los eventos que podrían hacer fracasar el alcance (objetivos), costo y tiempos en los proyectos o procesos. Un factor importante en la identificación de riesgos es la ingeniería de requisitos ya que cada requisito tiene un riesgo asociado con la capacidad que debe poseer un sistema, producto, servicio o componente para satisfacer un contrato, un estándar o una especificación (Jackson, 2001), estos requisitos pueden ser de producto y de proyecto (proceso). Los requisitos de producto incluyen los requisitos técnicos, de seguridad y de desempeño. Los requisitos de proyecto (proceso) incluyen los requisitos organizacionales (de la empresa), de dirección y de entrega del producto.

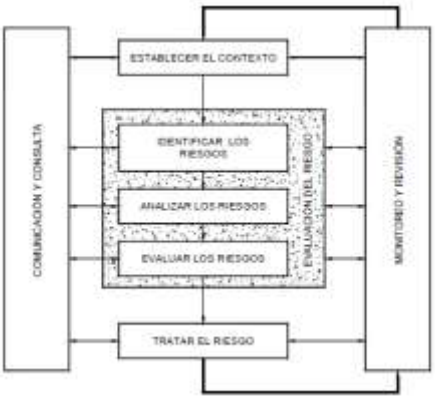

Fig. 1. Proceso de gestión de riesgos.

Continuando con los modelos de gestión de riesgos (Rodenes & Torralba, 2003; Pérez y Donoso, 2010) los caracterizan por generaciones: a) La generación tradicional, b) la reactiva y c) la causal o emergente. La primera se presenta en 1970 y se limitaba a la identificación de riesgos en los proyectos con técnicas como cuestionarios, entrevistas y lluvia de ideas de los miembros del equipo del proyecto; la segunda analiza los riesgos en los proyectos y eso asocia a la planificación para prevenir situaciones negativas durante el avance del proyecto, esta generación nace en 1980 y la última es la generación de 1990 la cual establece causalidad explicativa y predictiva entre los factores de riesgo.

Como referentes metodológicos de gestión de riesgos, merecen citarse NIST[2], Risk TI, Octave, Magerit, entre otros. Todos tienen un objetivo en común, el integrar buenas prácticas para facilitar el análisis de riesgos y servir de guía en la implementación de la gestión de riesgos (Gómez, 2010). Estos marcos de trabajo, estándares y metodologías contienen estrategias diferentes, estas pueden ser reactivas[3] o proactivas[4] y concuerdan que la aplicación de estrategias proactivas son las más indicadas para utilizar y en la necesidad de la realización de los análisis temprana de riesgos sistémica, sistemática y formalmente.

Con la creciente dependencia con las Tecnologías de la Información y las comunicaciones (TIC) (Guerrero, 2010); (Boehm, 1990); (Charette, 2005) se ha experimentado una transformación en el crecimiento con la aparición de facilidades de estructuración y comunicación de la información, lo cual ha desencadenado cambios en los sistemas de información que han llevado a desarrollar nuevos enfoques de gestión de riesgos que deben estar en conformidad el análisis, diseño y desarrollo de sistemas, en este caso los sistemas software, planteando por lo tanto nuevas metodologías que ayudan a las organizaciones con sus proyectos. Teniendo en cuenta el campo de investigación de este trabajo a continuación se describe las metodologías de gestión de riesgos para las empresas de software.

Iniciando con la metodología (ITIL, 1980) (Biblioteca de Infraestructura de Tecnologías de la Información) la cual gestiona operaciones y

[2] National Institute of Standards and Technology.
[3] Evaluar las consecuencias del riesgo cuando este ya se ha producido.
[4] Aplicar el método de evaluación previa de los riesgos y sus posibles consecuencias.

servicios de Tecnologías de información, se definió en 1980 por la oficina de comercio del reino unido. El objetivo es alinear el negocio y tecnologías de la Información en las organizaciones. Cubre temas, desde el cableado hasta la gestión de la continuidad del negocio. Una de sus áreas de trabajo es la gestión de incidentes para detectar alteraciones en los servicios de Tecnologías de Información. El problema de este marco de trabajo es el no considerar las fases de desarrollo de software ni la gestión de proyectos asociados a la construcción de activos software (ITIL, 1980).

Así mismo el Centro de Investigación en Seguridad en Internet del software *Engineering Institute* generó el método (OCTAVE) (*Operationally Critical Threat, Asset and Vulnerability Evaluation*) el cual analiza los riesgos orientado a activos organizacionales. Incluye personas, hardware, software, información y sistemas. Útil en organizaciones pequeñas. El objetivo es facilitar la evaluación de riesgos en una organización. Es un método complicado de entender ya que se requiere tener conocimiento profundo. (OCTAVE, 1999).

El software Engineering Insitute también desarrollo el modelo (CMMI, 2002) (Capability Maturity Model Integrated) el cual es un estándar mundial para la medición de la calidad de los procesos de desarrollo de software. Tiene 22 áreas de proceso. La gestión de riesgo es un área de proceso que identifica problemas antes de que ocurran para planificar actividades de tratamiento de riesgos y lograr los objetivos del proyecto. Este modelo puede ser muy detallado para algunas organizaciones. Es prescriptivo, requiere de inversión para implementarse (CMMI, 2002).

Por otra parte la (NTC5254, 2006) Norma Técnica Colombiana es una guía genérica de gestión de riesgos se aplica en actividades, decisiones u operaciones de cualquier empresa, grupos o individuos. Es una base rigurosa y confiable en la toma de decisiones y la planificación, identifica oportunidades y amenazas; es una estrategia proactiva que asigna recursos para la gestión de incidentes y la reducción en las perdidas. Asume que el riesgo es manejado por un estilo de riesgo operativo de grupo, y que la organización cuenta con conocimientos adecuados y grupos de gestión de riesgo para tratar los riesgos.

En el año 2007 el Instituto de Gobierno plantea el método (COBIT, 2007) como un conjunto de herramientas de soporte para gerenciar los requerimientos de control, temas técnicos y riesgos de negocio. Permite desarrollar políticas prácticas para control de TI. Integra las mejores prácticas de TI en un marco de gobierno que ayuda a comprender y administrar los riesgos y beneficios asociados con las tecnologías de información. La estructura de procesos brinda una visión completa de TI, la toma de decisiones, la alineación de las estrategias de las tecnologías con la estrategia del negocio y logra la estandarización para la mitigación de los riesgos. Requiere profundidad en el estudio, provee de guías de auditorías que por dificultades económicas y de gestión no se pueden obtener y se enfoca más en el control que en la ejecución.

El Project Management Institute en el 2004 presenta el (PMBOK, 2008) Project Management Body of Knowledge con su guía de los fundamentos de la dirección de proyectos donde uno de sus capítulos, incluye los procesos relacionados con la planificación, la identificación, el análisis, la respuesta, el monitoreo y control de riesgos en un proyecto. La gestión de riesgos en un proyecto establece aumentar la probabilidad y el impacto de eventos positivos para disminuir la probabilidad y el impacto de eventos negativos.

La Metodología de análisis y gestión de riesgos de TI (MAGERIT) desarrollada por el Consejo Superior de Administración Electrónica y publicada por el Ministerio de Administraciones Publicas de España es una metodología de análisis y gestión de riesgos de TI. Está orientada a los activos de la organización y su objetivo es descubrir los riesgos expuestos y concientizar a los responsables de los sistemas de información de la existencia de los riesgos y la necesidad de impedirlos a tiempo y ayudar a descubrir y planificar los planes oportunos para mantener los riesgos bajo control. Aplica a organizaciones que trabajan con información digital y sistemas informáticos. El problema de esta metodología es que es larga y compleja, maneja claves propias que necesitan memorizarse para identificar los riesgos.

El SEI es un Método Continuos Risk Management el cual está compuesto por una guía de principios, conceptos y funciones para la toma de decisiones en torno a los riesgos que deben ser evaluados continuamente. Permite tomar decisiones en cuanto a la gestión de riesgos de un proyecto en todas sus etapas.

Y por último la IEEE establece una norma para el desarrollo de planes de gestión del riesgo que se constituye por el uso de formatos. Aconseja que

cada organización debiera desarrollar un conjunto de prácticas y procedimientos destinados a la preparación y ejecución de planes de gestión de riesgos.

3. GESTIÓN DE RIESGOS EN PRUEBAS DE SOFTWARE

Dentro del ciclo de vida del software el cual es el proceso que se sigue en el desarrollo de un proyecto en las empresas de software se encuentra la etapa de pruebas software el cual por ser el último proceso del ciclo tiene mayor riesgo de retrasos y de recorte de alcance en los proyectos y la mayoría de las veces no se le da la importancia requerida (Richard, 2010). El ciclo de vida de desarrollo software se define como un proceso iterativo el cual representa los pasos que se debe seguir para el desarrollo de un software, este ciclo de vida está compuesto por las siguientes etapas principales: Definición de necesidades, Análisis, Diseño, Codificación, Pruebas, Validación y Mantenimiento, las cuales se visualizan en la fig. 2.

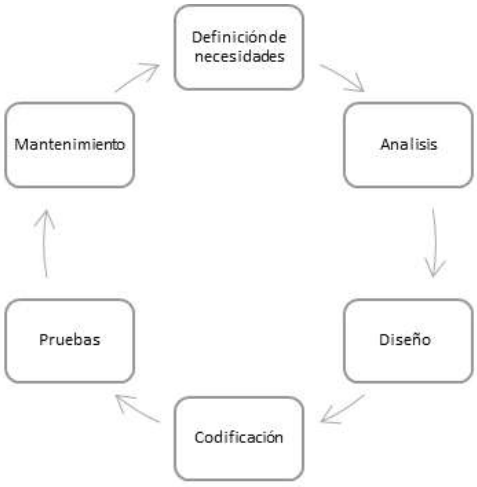

Fig. 2. Ciclo de vida del Software.

Dado que los errores humanos dentro de la programación son varios y pueden aumentar la dificultad del problema por esta razón es necesario realizar las pruebas las cuales garantizan un correcto funcionamiento. Las pruebas consisten en ejecutar el software con determinados datos de entrada y producir resultados que luego serán comparados con los teóricos. Según el diccionario de la IEEE define las pruebas como una actividad en la cual un sistema o uno de sus componentes se ejecutan en dos o más circunstancias previamente especificadas, los resultados se observan, se registran y se realiza una evaluación. También se le conoce a las pruebas de software como verificación y validación y está orientada a determinar si los productos desarrollados cumplen los requisitos y si el software satisface las necesidades del usuario (Pargas, 1999)

Por otro lado (Myers, 2004) define el proceso de pruebas como: "El proceso de ejecutar un programa con la intención de encontrar errores" y El Comité Internacional de Certificación de Pruebas de Software (*International Software Testing Qualifications Board* - ISTQB) define tres objetivos a perseguir con la realización de las pruebas, definir los defectos, ganar confianza en calidad y proporcionar información y prevenir defectos. (Müller, 2005).

El proceso de pruebas está compuesto por etapas como son: planificación, control, diseño, ejecución y evaluación, según el centro de ensayos de Software (CES, 2007), en la etapa de planeación se determina el alcance, el cronograma, los objetivos de la prueba, la estrategia de la prueba y se preparan las entradas y salidas para la prueba, el control de la prueba es transversal a las demás etapas del proceso teniendo en cuenta cuantas pruebas completaron y cuantas fallaron, el diseño revisa la base de las pruebas, analiza los riesgos del producto, los requerimientos, la arquitectura, las interfaces, brindando una lista de lo que se interesa probar; en la ejecución se establece la prioridad en los casos de prueba, se crean los datos que se necesitan, se asegura que el entorno de la prueba este instalado correctamente y se ejecutan los casos de prueba en conjunto, luego de esta ejecución se comparan los resultados actuales con los esperados y por último esta la fase de evaluación donde se revisan los resultados teniendo en cuenta los criterios de aceptación y se escribe un reporte técnico con los resultados de las pruebas.

El objetivo de las pruebas es validar si el comportamiento del software cumple o no con las especificaciones. Las funciones son probadas ingresando las entradas y examinando las salidas, el propósito que se busca con las pruebas funcionales es mostrar las diferencias con las especificaciones y no demostrar que el programa cumple su especificación (Myers, 2004). También existen las pruebas de regresión para verificar que ocurrió en la calidad del producto luego de realizar un cambio, asegurando que esos cambios no introducen un comportamiento no deseado, es decir luego de encontrar los defectos pasarlos a que realicen cambios en el desarrollo se re ejecuta

alguna o todas las pruebas realizadas anteriormente (Müller, 2005).

Al día de hoy se calcula que el proceso de pruebas constituye más de la mitad del costo de un desarrollo, por esta razón demanda un tiempo similar al de la programación lo cual lleva a un alto costo económico, de este modo si el proceso de pruebas requiere más tiempo y dinero entonces necesita la importancia de una metodología que exige herramientas y conocimientos destinados a optimizar la tarea de pruebas.

En la etapa de pruebas de software por lo general tratan de implementar mejoras en los procesos, lo cual no siempre es la mejor forma de complementar el proceso. Existe la gestión de pruebas esbeltas la cual se ve como un buen complemento para los modelos de mejora de pruebas dado que estos modelos se apoyan en la comparación de la organización y se centran en el proceso en sí, en el caso de las organizaciones es necesario considerar las mejoras necesarias, es decir no solo centrarse en el proceso de pruebas si no también los procesos que rodean las pruebas y sus interacciones (Elsheikh, 2008).

No es frecuente encontrar en el exterior estudios sobre la práctica de las pruebas, sin embargo si se han revisado algunas aportaciones a los factores que incluyen el tema de las pruebas, diferente a lo que ya han contribuido modelos como el CMMI, Iso15504 y otras. (Baar, 2008) menciona que la etapa de pruebas es la más difícil en el ciclo de vida del software una de las causas es la estimación de esfuerzo, problemas que causa el proceso de prueba que no está optimizado, la disponibilidad de recursos y herramientas y la formación y actitud del personal. En Colombia existen factores que hacen que las pruebas de software sean una etapa complicada en el desarrollo, estos factores son: las decisiones en la organización, la carrera profesional en la disciplina de las pruebas, la formación profesional en pruebas, la actitud de los profesionales hacia las pruebas, las estrategias y técnicas utilizadas en la etapa de pruebas y la situación del mercado software. Se realizó una encuesta para determinar los factores que son causas que las pruebas de software no alcancen un nivel apropiado de eficacia y eficiencia y como resultados los factores más sobresalientes en este estudio fueron: la ausencia de la carrera profesional en cuanto al área de pruebas, la realización de las pruebas como última fase en el desarrollo de software y tener que recortar en calidad por problemas de presupuesto.

Es por esto que la importancia de integrar las actividades de gestión de riesgos en las fases del ciclo de vida de desarrollo de software y con los problemas mencionados que se tienen en la etapa de pruebas es necesario que en esta etapa sean llevados lo riesgos de igual manera que en las primera etapas. Sin embargo esto es un reto para los líderes de proyecto, para insertar las actividades de gestión de riesgos en cada una de las etapas del desarrollo del proyecto sin alterar dichas etapas logrando minimizar los costos y la asociación con la gestión del proyecto, es decir la gestión de riesgos no puede aislarse del desarrollo de software (Boehm, 1991).

Un estudio realizado donde aplican un modelo de gestión de riesgos en diferentes proyectos de software y en distintas fases del ciclo de vida, en el cual se identificaron debilidades y bondades, en forma general los proyectos presentaron problemas en la planificación y gestión, dando como resultado que el proceso de planificación del riesgo no se le dio la importancia requerida. El costo del ciclo de vida de desarrollo asociado a las fallas de un producto supera el 10% de la facturación anual de una empresa y el factor importante que contribuye a esta pérdida es el desempeño insuficiente de la etapa de pruebas. (Engel, 2003).

El Centro Experimental de Ingeniería del Software definen como practicas ausentes del desarrollo de software la falta de medición en el proceso, la inexistencia del registro de datos históricos, estimaciones y costos imprecisos, el mal uso de las herramientas para la planificación y estimación, presión en los calendarios, crecimiento excesivo de los requerimientos, deficiencia en el monitoreo en el proceso del ciclo de vida de desarrollo, y la no formalización de la gestión de riesgos. Por estas malas prácticas es común que en el proceso de pruebas se tengan retrasos. Según Anaya se incumple con la entrega de productos o se entregan productos defectuosos donde el costo de la corrección de errores es alto, esto se debe a que se realiza un mayor énfasis en las pruebas finales que en revisiones intermedias, es decir no se centran a realizar pruebas desde los requisitos y diseño.

El problema de investigación que se deducen de la literatura analizada plantea que: A pesar que existen numerosas herramientas y técnicas para la gestión de riesgos, en la literatura no se evidencia un modelo de gestión de riesgos que se enfoque en el marco de la etapa de pruebas de software.

La gestión de riesgos es importante en las pruebas dado que estas se realizan al final del ciclo de vida del proyecto, en el momento que el proyecto llega a su fin el impacto de los riesgos es más alto (Wideman, 1992). Algunos proyectos pueden tratar de mitigar los riesgos en todo el ciclo de vida, sin embargo al final del proyecto muchos de los riesgos se convierten en problemas, por esta razón es necesario identificar los riesgos en la etapa de pruebas. (Pressman, 2002; Cardoso, 2001; Sommerville, 2000) Afirman que el proceso de pruebas debe ser considerado durante todo el ciclo de vida de un proyecto para así obtener productos de calidad, es por esto que la identificación de riesgos y el seguimiento temprano de ellos durante todo el proyecto permitirán que los planes de contingencia se efectúen con éxito. Dado que el 10% de la facturación anual de las empresas está asociado a fallas en el producto y esto se debe al bajo desempeño del sistema de pruebas de software (Barad, 2003).

En la literatura se encuentra que la gestión de riesgos se queda muchas veces solamente en la investigación y no pasa a tener un correcto funcionamiento en la práctica; como resultado de esto se tienen los fracasos en los proyectos ya sea por costo, tiempo o cualquier factor que afecte el proceso de un proyecto con éxito (Ropponen, 1999; kwak, 2004; Stoddard, 2004; Morris, 1996; Pfleeger, 1998; Stoddard, 2008) De la brecha identificada en la revisión de la literatura se selecciona como aspectos principales a tratar la integración y enfoque de la gestión de riesgos en la etapa de pruebas software durante todo el ciclo de vida de desarrollo.

Encuestas realizadas en la industria indica que sólo un 25% de los proyectos de software se completa según lo programado, presupuestado y especificado (Charette y Johnson, 2006) mencionan que este es un problema mundial teniendo impacto en las organizaciones. En Australia se presentan casos de estudio en las agencias de gobierno en 17 proyectos de 17 agencias que reflejan que las organizaciones trabajan solamente la primera etapa del proceso de la gestión de riesgos, es decir identifican los riesgos en los proyectos y después de esto no existe una planificación, seguimiento y control. El estudio que se realizó a nivel Nacional en la Zona centro, zona cafetera y zona oriente del País, por medio de la red Colombiana de calidad de Software (RCCS) en 19 pymes Colombianas del sector software, arrojó los siguientes resultados: A manera general el 78.13% de las empresas no presentan una metodología clara para la gestión de riesgos, donde se define la identificación y clasificación de los riesgos, así como los planes o acción para reaccionar ante el riesgo, como tampoco los mecanismos de seguimiento y un 59.38% de las empresas no tiene un proceso institucionalizado para gestión de riesgos.

La metodología a seguir para el desarrollo del trabajo de investigación corresponde a una investigación cualitativa la cual es un campo interdisciplinar, atraviesa las humanidades, las ciencias sociales y las físicas, implica un enfoque interpretativo y naturalista hacia sus objetos de estudio (Rodriguez, 1996). Respecto a las fases de la investigación cualitativa, se diferencian cinco fases de trabajo, a) definición del problema; b) diseño del trabajo; c) recogida de datos; d) análisis de datos y e) el informe y validación de la información. Y dentro de la investigación cualitativa este trabajo se enfoca en la modalidad de investigación- acción participativa.

4. METODOLOGÍA DE INVESTIGACIÓN

La modalidad de investigación - acción participativa tiene que ver con la necesidad de aprender orientada a la posibilidad de realizar un trabajo sistemático de elicitación, registro y análisis de las percepciones, juicios y comprensiones que son aportados por los que intervienen en las distintas fases de investigación, desde el diseño, hasta el uso pasando por la implementación (Casilimas, 2002). A continuación se describen las fases:

Estudio del problema: En esta fase se realiza una caracterización de las pymes del sector software con el propósito de revisar el contexto nacional, se examinan estudios realizados en la industria software colombiana y estudios a nivel regional. (Fedesoft y Proexport, 2009,2008). En la construcción de la caracterización de las empresas de software en Colombia se realiza la investigación teniendo como referencia el estudio de (fedesoft, 2012). Dado el modelo de referencia la caracterización contempla algunos factores y variables. Se inicia con la descripción del factor software con sus respectivas variables, también se tienen en cuenta la segmentación dependiendo del alcance demográfico en el País.

La selección de estándares y modelos para establecer un análisis formal esta soportado por el método de estudio de similitud entre modelos y estándares (MSSS) se adapta este método para

analizar los estándares de la gestión de riesgos y el proceso de software (Manzano et al) Las fases del modelo se describen en la tabla 1.

Tabla 1: Paso del método MSSS

Paso	Actividad
Seleccionar estandares	Se tienen en cuenta los estandares y modelos de la gestión de riesgos y proceso de pruebas que tengan mas utilizacion en estos ambitos y que la información este disponible.
Definir aspectos a analizar	Se revisa el alcance y enfoque del estandar, los principios basicos y parametros del estandar.
Identificar procesos a analilizar	Los procesos que se analizan se determinan por el ambito de la gestión de riesgos y ciclo de vida de pruebas y se estudian los elementos de los estandares con el fin de determinar aspectos de cada uno.
Establecer objetivo del analisis	Determinar caracteristicas de los estandares, se comparan los estandares y se presentan tablas donde se comparan las caracteristicas mas relevantes, esto permite detrminar cual estandar se considera relevante en el area de estudio que se esta tratando.
Definir estructura para presentar analisis	Alcance del estandar, principios basicos y parametros del estandar, estructura de los pasos que componen el estandar .
Identificar similitudes, sintetizar información y recolectar resultados	La forma de presentar los resultados es por medio de una tabla comparativa con las metodologías de gestión de riesgos y del ciclo de vida de pruebas para determinar el modelo actual de la gestión de riesgos y el proceso de pruebas en proyectos software con el fin de identificar estos procesos como están hoy en día.

El producto de esta actividad implicará determinar las variables, las técnicas y herramientas y la evaluación del modelo de procesos entorno a la gestión de riesgos y de la etapa de pruebas software. Para realizar esta fase se consultan diferentes bases de datos como son: IEEE, Proquest, Elsevier, Emerald entre otras; a partir de esta búsqueda se seleccionan artículos relevantes en el tema de investigación para apoyar la caracterización del ciclo de vida de gestión de riesgos y la etapa de pruebas.

Propuesta de la solución: Busca seleccionar las fases, técnicas y herramientas que serán integradas en la metodología a proponer. Para la selección de las técnicas y herramientas se utilizará el procedimiento de análisis y toma de decisiones bajo evaluación formal por medio de puntuaciones, esta técnica permite comparar diversas alternativas y seleccionar las técnicas y herramientas que cumplen con ciertos criterios establecidos. El objetivo de esta técnica es facilitar, soportar y orientar un proceso formal de evaluación para la toma de decisiones y la búsqueda de una solución. Las actividades de este proceso son:

a) Análisis de la información y enfoque de la situación: Se debe analizar la información actual y existente sobre la situación y en base a esto se realiza un análisis de causas que genere claridad sobre la misma y justificar la solución.

b) Generación de la pree-valuación: Se realiza una selección previa de dos o más alternativas posibles a implementarse, donde el principal propósito es presentar una propuesta de evaluación formal. Para esto implica la definición de los siguientes aspectos: Criterios de evaluación, alternativas de solución, y método de evaluación. La definición de los criterios de evaluación consiste en la selección de los criterios pertinentes para tener en cuenta al momento de tomar la decisión de la solución, consiste en asignar un peso porcentual a cada uno de los criterios, la suma porcentual de los criterios debe ser igual al 100%. La definición de alternativas de solución: teniendo los criterios definidos se deberá identificar entre dos y cuatro alternativas de solución para cada situación teniendo en cuenta: maximizar (generar la mejor solución posible), satisfacer (elegir la primera opción que sea mínimamente aceptable para satisfacer la situación presentada y optimizar (la solución que genere el mejor equilibrio posible entre las distintas identificadas. La definición del método de evaluación: teniendo las alternativas de solución identificadas y los criterios elegidos y el análisis de sus diferentes condiciones, el método que se utiliza para la evaluación de las alternativas es el método Delphi o panel de expertos, luego de implementar este método se continua con:

c) la evaluación formal: Se compone de los siguientes pasos: Los integrantes del panel de expertos deben aprobar o rechazar los criterios y

pesos, las alternativas de solución, el método de evaluación y la solución elegida. Si los pasos anteriores son aprobados por el panel de expertos se entiende que se da por aprobado la implementación de la solución elegida.

d) Implementación de la solución: La solución definida por el panel de expertos debe ser planeada, implementada, seguida y evaluada.

Y como tercera fase se describe la aplicación de la solución la cual se valorará en una prueba piloto que se realizará en un proyecto de software en una organización o institución. Con el proceso de validación de la metodología se busca, poder observar la funcionalidad de las herramientas y técnicas que contiene la metodología definida. En el momento en que la metodología ha sido aplicada se tendrá lecciones aprendidas acerca de la funcionalidad y se identificará si se deben realizar mejoras y ajustes.

5. CONCLUSIONES

Se identificaron las diferentes Metodología de gestión de riegos que existen para los proyectos Software y proyectos en general.

Se identificó que es claro que las organizaciones hoy en día son conscientes de la necesidad de identificar los riegos asociados a TI y al no aplicar una metodología adecuada a cada negocio, es decir se debe entender su cultura organizacional y sus procesos, es imposible lograr que las metodologías estudiadas alcancen el objetivo de minimizar los riesgos. Vale la pena destacar que una evaluación de riesgos es particular para cada organización y que no es bueno desarrollar una evaluación de riesgos de una empresa a partir de los resultados obtenidos por una organización diferente (Gómez et al)

Se encontró en el estudio de la etapa de pruebas que estas quedan atrapadas al final del ciclo de vida del Software y muchas veces con un calendario corto. Dado que por lo general sufren los retrasos de las fases anteriores de desarrollo y la etapa de pruebas no puede retrasar su final ya que lo siguiente es la entrega a cliente. Algunos autores como (Sanz et al) justifican que las pruebas son la actividad probablemente más difícil dentro del desarrollo de Software.

Con estas conclusiones, es posible identificar las fases y actividades principales de una metodología de gestión de riesgos para la etapa de pruebas software para diseñarla e implementarla en empresas de software para ayudar a realizar el proceso de gestión de riesgos con sus diferentes fases.

REFERENCIAS

Charette, Why software fails. IEEE Spectrum, 42 (9), 42–49., 2005

Charette, Software Engineering Risk Analysis and Management. McGraw-Hill, New York. , 1989

Boehm, B., & Covenin. (1991; 1995) Software risk management principles and practices. IEEE Software 8 (1), 32–41. .

PMI, & Silberfich. (2008; 2009). PMBOK Project Management Body of Knowledge, Project Management Institute. 4ta Edición.

Pressman, R. (2002). Ingeniería del Software: Un enfoque Práctico. McGraw Hill.

Boehm, Software Risk Management: Principles and Practices 1990

Thayer; Software Engineering Project Management; 2003

NTC5254 Norma ISO 27001:2005 NORMA TÉCNICA COLOMBIANA 5254 esta norma es una adopción idéntica (IDT) de la AS/NZ 4360:2004. IEEE Software January; IEEE Computer Society.

Gómez R, P. D. (2010). Metodología y gobierno de la gestión de riesgos de tecnologías de la información. Revista de ingeniería Universidad de los Andes.

Guerrero, Tesis gestión de riesgos y controles en sistemas información; 2010

Boehm, Software Risk Management: Principles and Practices; 1990

Charette Why software fails. IEEE Software January, 2005 IEEE Spectrum 42 (9), 42–49.

ITIL. (1980). Biblioteca de Infraestructura de Tecnologías de la Información. Oficina de comercio del reino unido.

OCTAVE. (1999). Operationally Critical Threat, Asset and Vulnerability Evaluation. Software Engineering Institute.

CMMI. (2002). Guía para la integración de procesos y la mejora de productos. Mary Beth Chrissis Mike Konrad Sandy Shrum. (ISBN: 9788478290963) publicado por Pearson Educación, S.A.

NTC5254. (2006). Norma ISO 27001:2005 Norma Técnica Colombiana 5254.

COBIT. (2007). Governance Institute.

PMBOK. (2008). Project Management Body of Knowledge, Project Management Institute (PMI), 4ta Edición.

MAGERIT. (n.d.). Comité Técnico de Seguridad de los Sistemas de Información y Tratamiento Automatizado de Datos Personales. SSITAD, del Consejo Superior de Informática.

Richard A, C. A. (2010). CERT Resilience Management Model. A maturity Model for managing operational resilience, White CERT RMM Versiòn 1.1 SEI Series a cert Book.

Pargas, R. P. (1999). Test data generation using genetic algorithms, The journal of software testing, verification and reliability.

Myers, G. J. (2004). The art of Software Testing, 2da Ed. John Wiley & Sons, Inc, New Jersey, USA, pp. 6.

Müller, T. &. (2005). Certified Tester Foundation Level Syllabus V2007, Actualizado el 12 de abril de 2007, ISTQB, pp. 12-28.

Elsheikh, A. &. (2008). Linear Mathematical Driver for the Future of Software Testing Process. Frontiers in Artificial Intelligence and Applications. 182, pp. 51-59.

Boehm, B. (1991). Software risk management principles and practices. IEEE Software 8 (1), 32–41.

Engel, A. B. (2003). A methodology for modeling VVT risks and costs. . Systems Engineering Journal 6 (3), 35–151, Wiley InterScience, Online ISSN: 1520-6858, Print ISSN: 1098–1241.

Pressman, R. S. (2002). Ingeniería del Software: Un enfoque Práctico.

Cardoso; (2001). Pruebas del Software.

Sommerville. (2000). Software Engineering. Mérida, Venezuela: McGraw Hill; Pearson Education.

Barad, E. &. (2003). A methodology for modeling VVT risks and costs. Systems Engineering Journal 6 (3), pp. 135–151, Wiley InterScience.

Ropponen (1999). Risk assessment and management practices in software development. In: Willcocks, L.P., Lester, S. (Eds.),

Willcocks, L. P., Stephanie Lester, S. (1999). Beyond the IT Productivity Paradox. John Wiley & Sons, Chichester

Stoddard; (2004). Project risk management: lessons learned from software development envi. pp. 247–266.

Pfleeger. (1998). Software Engineering.

Ropponen. Software Engineering; Risk assessment and management practices in software development. In: Willcocks, L.P., Lester, S. (Eds.), Beyond the IT Productivity Paradox. John Wiley & Sons, Chichester, pp. 247–266. 1999

Kwak, Y. H. (2004). Project risk management: lessons learned from software development environment, Technovation.

Stoddard, J., & Bannerman. (2008). Risk and risk management in software projects: A reassessment.

Charette, & Johnson. (2006). Why software fails. IEEE Spectrum 42 (9), 42–49.

Rodriguez Gomez G (1996). Metodología de la investigación cualitativa, Bilbao, Universidad de Deusto.

Casilimas, C. S. (2002). Programa de especialización en Teoria, metodos y tecnicas de investigación social. ARFO Editores e impresores Ltda.

FEDESOFT (2009). Sector de ti en colombia. Technical report, Federacion Colombiana de la Industria de Software.

ESI. (2008). Industria de software en colombia. Technical report, European Software Institute-Tecnalia.

Castellanos, F., Mayerly, A., and S., L. (2007). Estudio de previsión tecnológica industrial para la industria del software y servicios asociados. Technical report, Universidad Nacional de Colombia.

DNP. (2007). Agenda interna para la productividad y la competitividad. Documento

Sectorial software. Technical report, Departamento Nacional de Planeaci_on.

FEDESOFT. (2008). Descripción del sector del software. Technical report, Federación Colombiana de la Industria de Software

PROEXPORT. (2008). Industria de tecnologías de informacion. Technical report, Proexport Colombia.

Palomino, K. (2011). Estudio del comportamiento de la Industria del software en Colombia ante escenarios de capacidades de innovacion y ventajas comparativas por medio de dinamica de sistemas; pp 16.

FEDESOFT. (2012). Estudio de la caracterización de productos y servicios de la Industria de Software y servicios asociados.

Alberts, C. (1999) Operationally Critical Threat, Asset and Vulnerability Evaluation SM (OCTAVESM) Framework, Versión 1.0. Technical Report. CMU/SEI- 99- TR-017. Londres.

Calvo-Manzano, J. A. et al. (2009). Process Similarity Study: Case Study on Project Planning Practices Based on CMMI-DEV v1.2.