



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

IDENTIFICACIÓN DE RIESGOS POTENCIALES EN PROYECTOS DE
SOFTWARE WEB UTILIZANDO MODELOS DE GESTIÓN DE RIESGOS

CORREA ELIZALDES KEVIN DAVID
INGENIERO DE SISTEMAS

MACHALA
2019



UTMACH

FACULTAD DE INGENIERÍA CIVIL
CARRERA DE INGENIERÍA DE SISTEMAS

IDENTIFICACIÓN DE RIESGOS POTENCIALES EN PROYECTOS
DE SOFTWARE WEB UTILIZANDO MODELOS DE GESTIÓN DE
RIESGOS

CORREA ELIZALDES KEVIN DAVID
INGENIERO DE SISTEMAS

MACHALA
2019



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

IDENTIFICACIÓN DE RIESGOS POTENCIALES EN PROYECTOS DE SOFTWARE
WEB UTILIZANDO MODELOS DE GESTIÓN DE RIESGOS

CORREA ELIZALDES KEVIN DAVID
INGENIERO DE SISTEMAS

HONORES TAPIA JOOFRE ANTONIO

MACHALA, 22 DE AGOSTO DE 2019

MACHALA
22 de agosto de 2019

Nota de aceptación:

Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado Identificación de riesgos potenciales en proyectos de software web utilizando modelos de gestión de riesgos, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



HONORES TAPIA JOOFRE ANTONIO
0704811751
TUTOR - ESPECIALISTA 1



MOLINA RIOS JIMMY ROLANDO
0703691980
ESPECIALISTA 2



CARDENAS VILLA VICENCIO OSCAR EFEREN
0703935312
ESPECIALISTA 3

Fecha de impresión: jueves 22 de agosto de 2019 - 14:14

Urkund Analysis Result

Analysed Document: InformeCompexivo - Correa Kevin.docx (D54802886)
Submitted: 8/13/2019 5:43:00 PM
Submitted By: kdcorrea_est@utmachala.edu.ec
Significance: 0 %

Sources included in the report:

Instances where selected sources appear:

0

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, CORREA ELIZALDES KEVIN DAVID, en calidad de autor del siguiente trabajo escrito titulado Identificación de riesgos potenciales en proyectos de software web utilizando modelos de gestión de riesgos, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 22 de agosto de 2019



CORREA ELIZALDES KEVIN DAVID
0705890598

DEDICATORIA

Este trabajo de titulación en primera instancia va dirigido para Dios, por ser quien puso en mi los conocimientos, habilidades, destrezas y perseverancia necesaria para llevar a cabo este proceso de educación superior.

A mi madre, por ser la persona incondicional que me apoya completamente en toda actividad que realizo, dándome motivación y aconsejándome sobre cada decisión en vida.

A mi familia y amigos, por ser quienes formaron parte de mi vida durante todo este proceso universitario, animándome a continuar esforzándome por mis sueños y metas propuestos.

Sr. Correa Elizaldes Kevin David

AGRADECIMIENTO

Agradezco a Dios por ser el motor principal de mi vida y por llenarme de bendiciones, sabiduría, persistencia, alegrías y tristezas que me ayudaron a formar y madurar como persona para así poder alcanzar todas mis metas.

A mi madre, por ser la persona más importante en mi vida y ser quien me aconsejaba en todo momento confiando en mi potencial para cumplir mis metas y con mucho esfuerzo me ayudo con los recursos necesarios para lograr mi sueño.

A mi tutor el Ing. Honores Tapia Joofre Antonio, por brindarme sus conocimientos profesionales tanto en las asignaturas que son parte currículo académico como también guiarme para realizar la parte práctica del examen completo.

Sr. Correa Elizaldes Kevin David

RESUMEN

IDENTIFICACIÓN DE RIESGOS POTENCIALES EN PROYECTOS DE SOFTWARE WEB UTILIZANDO MODELOS DE GESTIÓN DE RIESGOS

Correa Elizaldes Kevin David, 0705890598

El desarrollo de software mantiene un gran crecimiento a nivel mundial, debido a que la mayor cantidad de tareas y actividades se encuentran automatizadas en la actualidad mejorando así los aspectos primordiales en acciones tradicionales. Los proyectos encaminados al desarrollo de sistemas web tienen gran aceptación por parte de los usuarios y en consecuencia los desarrolladores han optado con crear más proyectos de este tipo, pero al igual que cualquier software deben enfrentarse a diversos riesgos que pueden detener o suspender el proyecto si no se toman medidas estratégicas para mitigarlo. La gestión y control de los riesgos en el desarrollo de software debe ser un tema crítico e importante que debe ser tomado en cuenta por los desarrolladores, puesto que si se crea un sistema que ofrece funcionalidad, rendimiento y compatibilidad pero presenta fallas o errores puede afectar negativamente la imagen organizacional así como también la información de usuarios que hacen uso del sistema, por esta razón el personal involucrado en el desarrollo del proyecto deben identificar los riesgos potenciales antes y durante la creación del sistema, esta medida permitirá crear un plan sistemático y estratégico para reducir su impacto perjudicial. El objetivo de este trabajo de titulación es identificar los riesgos potenciales presentes en el desarrollo de proyectos de software web a través de modelos de gestión de riesgos para la determinación de aspectos que el equipo de trabajo debe tener en cuenta durante el desarrollo, mitigando así los riesgos y creando un producto final de mayor calidad.

PALABRAS CLAVES: Software web, riesgos, modelos de gestión de riesgos, proyecto, gestión de riesgos.

ABSTRACT

IDENTIFICATION OF POTENTIAL RISKS IN WEB SOFTWARE PROJECTS USING RISK MANAGEMENT MODELS

Correa Elizaldes Kevin David, 0705890598

The software development maintains a great growth worldwide, due to the fact that the greater amount of tasks and activities are currently automated, thus improving the fundamental aspects of traditional actions. Projects aimed at developing web systems are widely accepted by users and, consequently, developers have chosen to create more projects of this type, but just like any software they must face several risks that can stop or suspend the project if no strategic measures are taken to mitigate it. The management and control of risks in software development should be a critical and important issue that should be taken into account by developers, since if a system is created that offers functionality, performance and compatibility but has failures or errors, it can affect negatively the organizational image as well as the information of users who make use of the system, for this reason the personnel involved in the development of the project must identify the potential risks before and during the creation of the system, this measure will allow to create a systematic and strategic plan to reduce its harmful impact. The objective of this titling work is to identify the potential risks present in the development of web software projects through risk management models for the determination of aspects that the work team must take into account during the development, thus mitigating the risks and creating a higher quality final product.

KEY WORDS: Web software, risks, risk management models, project, risk management.

CONTENIDO

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN	3
ABSTRACT	4
CONTENIDO	5
CONTENIDO DE FIGURAS	6
CONTENIDO DE TABLAS	6
1. INTRODUCCIÓN	7
1.1 Marco Contextual	8
1.2 Problema	8
1.3 Objetivo General	9
2. DESARROLLO	9
2.1. Marco Teórico	9
2.1.1 Proyecto de desarrollo de software	9
2.1.2 Software web	9
2.1.3 Etapas del ciclo de vida del software	9
2.1.4 Metodologías de desarrollo de software web	10
2.1.5 Riesgos de proyecto	10
2.1.6 Gestión de riesgos	10
2.1.7 Modelos de gestión de riesgos	10
2.2. Marco Metodológico	11
2.3. Resultados	12
3. CONCLUSIONES	17
BIBLIOGRAFÍA	18

CONTENIDO DE FIGURAS

Figura 1 Ciclo de vida del software.....	9
Figura 2 Fases principales del modelo de gestión de riesgos	12
Figura 3 Taxonomía de riesgos de desarrollo de software.....	13
Figura 4 Riesgos identificados por cada fase del ciclo de vida del software	16

CONTENIDO DE TABLAS

Tabla 1 Comparativa de modelos y <i>estándares</i> de gestión de riesgos.....	12
Tabla 2 Riesgos comunes en un producto de software.....	13
Tabla 3 Riesgos potenciales en el desarrollo de software en base al ciclo de vida del desarrollo de software	15
Tabla 4 Riesgos potenciales sobre navegación y equipo de desarrollo del sistema.....	16

1. INTRODUCCIÓN

Con el auge de la tecnología; el uso de sistemas informáticos se puede categorizar como una necesidad latente por parte de la población en virtud a la cantidad de beneficios que estos otorgan en términos de tiempo, eficiencia y eficacia, que a su vez son notorios si los comparamos con acciones realizadas de forma manual. Con el transcurso de los años y con la nueva era de tecnología creciente los sistemas informáticos tradicionales fueron progresando, a fin de mejorar los aspectos tales como portabilidad, disponibilidad y accesibilidad [1]; según eso los desarrolladores optaron por emplear nuevas tendencias como sistemas basados en la web los cuales para su funcionamiento establecen una arquitectura denominada cliente-servidor, pero aun así su utilización no asegura que los riesgos de desarrollo sean eliminados gradualmente, por su parte se crean nuevas vulnerabilidades encaminadas al tipo de arquitectura, hardware, software, o personal empleado.

Los riesgos dentro del desarrollo de software son considerados como características nocivas que pueden provocar afectaciones directas en el sistema, criterio similar al propuesto por varios desarrolladores y profesionales dentro del área de la ingeniería de software, estos mencionan que los riesgos están presentes antes, durante y después de la creación de un proyecto de software web y que a su vez inducen un impacto perjudicial al rendimiento del sistema [2], con la finalidad de reducir estas afectaciones contraproducentes el equipo de desarrollo debe determinar estos riesgos, puesto que basándose en esa identificación se pueden establecer acciones para prevenir los posibles fallos (acciones preventivas) o corregir los errores identificados previamente (acciones correctivas), esto a través de un plan de riesgos que sirva de soporte en la toma de decisiones por parte del jefe de proyectos en conjunto con el grupo de trabajo, esto con la finalidad de disminuir el riesgo presente en el proyecto de desarrollo. La utilización de modelos de gestión de riesgos, tales como ISO 21500 ,ISO 31000, APM, CMMI o Guía PMBOK sirven como guía para el equipo de trabajo ya que especifican pautas estructurales y analíticas que pueden prevenir la suspensión del proyecto o el aumento en el presupuesto del mismo [2].

El objetivo de este trabajo es identificar los riesgos potenciales presentes en el desarrollo de proyectos de software web a través de modelos de gestión de riesgos para la determinación de aspectos que el equipo de trabajo debe tener en cuenta en el desarrollo de proyecto de software web, mitigando así los riesgos y creando un producto final de mayor calidad.

Como parte de la estructura general de este trabajo de titulación se encuentra distribuida en tres diferentes capítulos los cuales contienen la siguiente información:

Capítulo I: Esta sección se orienta a la presentación en contexto del marco conceptual de la investigación, la problemática y el objetivo propuesto para solventarla.

Capitulo II: Este capítulo se presenta el marco bibliográfico que establece la teoría que permite dar fundamentación a la investigación.

Capitulo III: Este apartado indica las conclusiones del trabajo realizado.

1.1 Marco Contextual

Las nuevas tecnologías empleadas para la automatización de procesos se encuentra en constante crecimiento en los diversos ámbitos tales como comercial, tecnológico, operacional, social o de aprendizaje, dentro de estas áreas se emplean sistemas informáticos encargados de realizar sus operaciones y tareas organizacionales con mayor efectividad, pero con el crecimiento de tendencias informáticas los diversos usuarios ven como mejor opción utilizar sistemas orientados a la web, ya que ofrece aún más beneficios en términos de robustez, tiempo de respuesta, disponibilidad, accesibilidad, instalación y conectividad esto frente a las aplicaciones de escritorio.

Dentro del contexto, el desarrollo de software web es riesgoso y el vínculo que mantiene con el éxito del proyecto es crucial, ya que la calidad del mismo puede verse afectada por los diversos riesgos que si no se identifican y toman estrategias decisivas que permitan gestionarlos o controlarlos, pero para ello es necesario que el equipo de desarrollo liderado por el jefe de proyectos tenga conocimientos claros de las fuentes en donde se origina el riesgo con la finalidad de tomar decisiones para reducir considerablemente su impacto y a su vez no afecten aspectos como el presupuesto, tiempo, recursos, personal o imagen corporativa, basándose en lo dicho, se puede indicar que la gestión de riesgos es un proceso esencial en el desarrollo de proyecto de software web.

1.2 Problema

La falta de un plan de riesgos y la identificación de los mismo en el proyecto de desarrollo de software web puede producir daños perjudiciales en el producto final lo que a su vez puede ocasionar un demora en los tiempos de entrega del sistema, los costos o afectar las características de un buen sistema como la efectividad, disponibilidad, mantenibilidad y sobre todo la seguridad del sistema, estos problemas en cadena generan como resultado una reducción significativa en la calidad del sistema; es por eso que garantizar la existencia de un plan de riesgos en el cual se encuentre especificado

el riesgo, la fuente de donde se origina su impacto y la estrategia que se debe tomar en caso se suscite, son acciones esenciales en todo proyecto de desarrollo.

1.3 Objetivo General

Identificar los riesgos potenciales presentes en el desarrollo de proyectos de software web a través de modelos de gestión de riesgos para la determinación de aspectos que el equipo de trabajo debe tener en cuenta durante el desarrollo, mitigando así los riesgos y creando un producto final de mayor calidad.

2. DESARROLLO

2.1. Marco Teórico

2.1.1 Proyecto de desarrollo de software

2.1.2 Software web

Un software orientado a la web podría definirse como un programa de tipo informático que puede ser accesible a través de Internet por medio de páginas web [3]. Este tipo de sistema emplea la arquitectura cliente-servidor lo cual permite que el cliente efectúe una solicitud al servidor web y este responda dicha petición de manera eficaz lo que proporciona ventajas en tiempo de respuesta, accesibilidad y disponibilidad de información [4].

2.1.3 Etapas del ciclo de vida del software

Una estructura en la que se establecen aspectos puntuales para la construcción de un sistema, es la finalidad del ciclo de vida destinado al desarrollo de software; a través de este conjunto de procesos el equipo de trabajo puede guiarse y mantener una organización en las tareas a realizarse durante el desarrollo del proyecto, estas etapas según Molina, Honores y Zea [5] se componen por lo siguiente procesos:

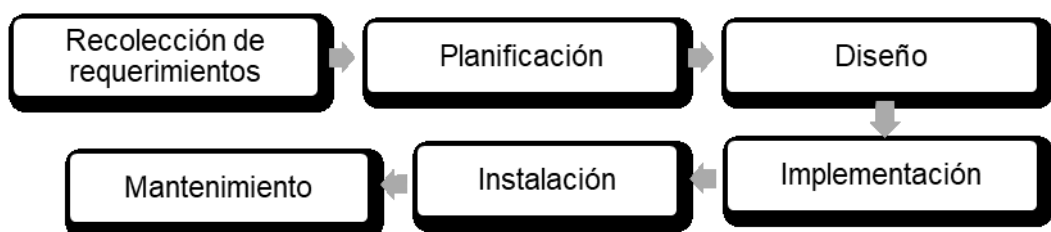


Figura 1 Ciclo de vida del software

Los procesos presentados en la figura 1 expresan las etapas clásicas que están presente en todo proyecto de desarrollo de un sistema. La primera etapa (requerimientos) consiste en la recolección de requisitos necesarios para el sistema; en el diseño se realiza el modelado del sistema basado en las especificaciones del cliente; la implementación es la tercera fase y es en la cual se construye el software según los

requisitos y diseño efectuado en las etapas anteriores; la verificación es una etapa de pruebas del sistema [6] y su finalidad es comprobar que el sistema cumpla con el objetivo para el cual se está desarrollando; si esta etapa cumple con todo lo previsto se procede a poner en marcha el sistema, si después de eso se detectan errores o fallos del sistema pueden ser corregidos en la última etapa denominada mantenimiento.

2.1.4 Metodologías de desarrollo de software web

Las metodologías de desarrollo dentro de la ingeniería de software establecen una estructura con aspectos puntuales que permiten al equipo de trabajo crear un marco de desarrollado que permitirá estructurar, planificar, gestionar y controlar el proyecto [7].

Según Molina y Zea [8] existen varias metodologías orientadas a la web entre las que encontramos:

- Modelo de diseño de hipertexto (HDM)
- Metodología de diseño de hipermedia orientada a objetos basada en escenarios (SOHDM)
- Metodología de diseño de sitio web (WSDM)
- Metodología de gestión de relaciones (RMM)
- Modelo de diseño de hipermedia orientado a objetos (OOHDM)
- Método de desarrollo orientado al servicio (SOD-M)

2.1.5 Riesgos de proyecto

Los riesgos son posibilidades de que se suscite una dificultad en el proyecto que a corto o largo plazo afecta de manera negativa el desarrollo de un producto de software [9] y que a su vez alteran el objetivo y resultados del mismo [10], estos riesgos deben ser controlados por el equipo de trabajo para que no generen una disminución en la calidad del sistema considerada por la ingeniería de software como uno de los aspectos primordiales que debe llevar todo sistema [11].

2.1.6 Gestión de riesgos

La gestión de riesgos puede definirse como procesos a realizarse para efectuar un manejo de errores [12], además, a través de la gestión de riesgos se puede establecer estrategias que permitan mitigarlos a fin de aumentar el éxito de un proyecto de software [13].

2.1.7 Modelos de gestión de riesgos

Los modelos de gestión de riesgos forman parte de las prácticas de la ingeniería de software y su objetivo fundamental es proporcionar un marco sistemático con pautas puntuales como la identificación de riesgos, estimación de riesgos, mitigación y

monitoreo [14], aspectos que permiten tomar acciones necesarias para tomar decisiones estratégicas y así evitar que estos hechos ocasionen afectaciones en el proyecto de software.

2.2. Marco Metodológico

La ISO (Organización Internacional de Normalización) estableció dos estándares internacionales relacionado a riesgos, la ISO 21500 y la ISO 31000 que se encuentran orientados a la gestión de riesgos de un proyecto de desarrollo y de proyectos organizacionales respectivamente, pero ambos tiene como finalidad proporcionar al grupo de trabajo los procesos que se deben tomar en cuenta en la gestión de un proyecto con referencia al control y manejo de riesgos [15][16], por su parte la asociación para la gestión de proyectos (APM por sus siglas en inglés) es también un grupo profesional que mantiene una finalidad similar a los estándares mencionados anteriormente, facilitar un marco de trabajo para que los desarrolladores efectúen un sistema con la menor cantidad de errores causados por peligros durante el desarrollo [17].

La guía de los fundamentos para la dirección de proyectos (guía PMBOK sexta edición) establece dentro las áreas del conocimiento la gestión de los riesgos del proyecto [9] en la cual se indican de manera detallada los procesos necesarios para gestionar riesgos del proyecto de software. El Modelo de Madurez de Capacidades de Integración (CMMI) [18] dentro de su área de gestión de riesgos de proyecto propone una estructura para la identificación temprana de amenazas potenciales y a su vez el tratamiento de los mismos, esto con el objetivo de prevenir daños en el proyecto de desarrollo y a todas las partes interesadas.

Con base a una investigación bibliográfica se estableció varios modelos orientados a la gestión de riesgos en proyectos de software los mismos que son presentados en la tabla 1 con cada proceso que forma parte de su estructura en comparativa con otros modelos y estándares.

Características estructurales	Estándares o modelos				
	ISO 21500 [15]	ISO 31000 [16]	APM [17]	CMMI [18]	Guía PMBOK [9]
Planificar la gestión del riesgo		✓	✓	✓	✓
Identificar	✓	✓	✓	✓	✓
Analizar	✓	✓	✓	✓	✓
Evaluar	✓	✓			

Planificar la respuesta		✓	✓	✓	✓
Controlar	✓	✓			✓

Tabla 1 Comparativa de modelos y estándares de gestión de riesgos

Fuente: Autor

En base a los aspectos propuestos en la tabla antes presentada se puede establecer un conjunto de fases que la mayor cantidad de modelos y estándares de gestión de riesgos indican dentro de su estructura, las cuales son presentadas en la figura 2.

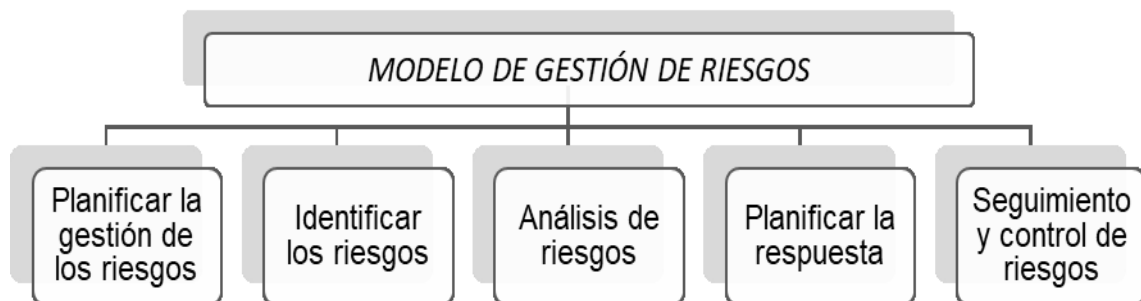


Figura 2 Fases principales del modelo de gestión de riesgos

Fuente: Autor

Para la identificación de los riesgos según indica la Guía Pmbok [9] se deben contar con información preliminar, tales como, plan de gestión de riesgos, cronograma de trabajo, estimación preliminar de costos de desarrollo, contrato con las partes interesadas, factores ambientales de la organización y planificación de los recursos de software y hardware.

2.3. Resultados

La existencia de una taxonomía de riesgos encaminados en el desarrollo de proyectos de software web es una pieza fundamental para conocer la estructura sistemática general de estos riesgos, además la taxonomía se encuentra orientada en el tradicional ciclo de vida de desarrollo de software[19]. En la figura 3 se presenta la taxonomía de riesgos de desarrollo de proyectos basado en la información propuesta por Ahmadvand y Pretschner [20].

Los riesgos en los proyectos de desarrollo de software pueden estar enmarcados en varias etapas del ciclo de vida del software[21], estas amenazas pueden ser internas o externas al grupo del proyecto de desarrollo.



Figura 3 Taxonomía de riesgos de desarrollo de software

Dentro de la clasificación de riesgos presentes en el desarrollo de sistemas informáticos se pueden encontrar peligros potenciales a los que se enfrentan el grupo de desarrollo y que pueden afectar el proyecto de forma global, estos riesgos son presentados en la tabla 2 basado en la investigación de [22].

N°	Riesgo identificado	Causa posible
R1	Virus informáticos	El servidor no posee un protocolo de seguridad frente ataques de virus maliciosos que puedan alterar el funcionamiento del sistema
R2	Hackeo informático	Accesos de sistema no protegidos
R3	Desastres naturales	Situaciones provocadas por efectos de la naturaleza

Tabla 2 Riesgos comunes en un producto de software

A continuación, se presenta en la tabla 3 los riesgos organizados en cada fase del ciclo de vida del software establecidos en diferentes estudios tales como [15][23][24][25].

N°	Riesgo identificado	Causa posible
RECOLECCIÓN Y ANÁLISIS REQUERIMIENTOS		
R4	Entendimiento errado	El grupo de trabajo deduce los posibles requisitos en base a lo que piensa
R5	Requerimientos incompletos	El cliente no especifica de manera adecuada el objetivo del sistema
R6	Requerimientos no especificados	El cliente no menciona todas las actividades que el sistema debe realizar
R7	Cambios incorrectos en los requerimientos	El grupo de desarrollo cambia requerimientos sin comunicar a todas las partes interesadas
R8	Ambigüedad de los requisitos	Los requisitos no fueron expresados o entendidos correctamente

R9	Requerimientos no validos	Requisitos que no están acorde a los valores éticos y morales del profesional.
PLANIFICACIÓN		
R10	Cronograma irrealista	El grupo de desarrollo planifica el cronograma con una holgura insuficiente
R11	Presupuesto insuficiente	Se establece un presupuesto inicial sin tomar en cuenta todos los elementos (software y hardware) necesarios para el proyecto
R12	Planificación deficiente	No se planifica todos los aspectos que forman parte del proyecto
R13	Falta de una metodología	El grupo de trabajo no emplea una metodología de desarrollo de software para el proyecto
R14	Menos comunicación	Los miembros del grupo de desarrollo no mantienen comunicación sobre cómo se llevará a cabo la ejecución del proyecto
R15	Ausencia de datos históricos	No se recolecta información suficiente para el cumplimiento del objetivo del software
R16	Falta de especificación de roles y responsabilidades al equipo	No se especifican todos los roles y las responsabilidades que tienen a cargo cada miembro del equipo
DISEÑO		
R17	Diseño de la interfaz de usuario incorrecto	El equipo de trabajo desconoce de la teoría de colores o utilización de ventanas y botones
R18	Diseño incompleto	No se traduce en los diagramas del sistema todas las necesidades del cliente
R19	Falta de integración del equipo de desarrollo de software	Las partes interesadas del proyecto no forman parte sobre la estructuración del diseño de software
R20	Desconocimiento del objetivo del sistema	Los diseñadores no conocen o tiene una mala interpretación del objetivo por el que se desarrollara el software
R21	Falta de diseño de requisitos	Se excluye requisitos funcionales en la diagramación
R22	Inapropiada o faltante documentación	No se establece una documentación en la que se presente todo el diseño del sistema
R23	Estructura lógica inadecuada	No se establece una adecuada conexión entre los diferentes módulos del sistema
CODIFICACIÓN		
R24	Falta de estándares de programación	No se aplica estándares de programación por parte del equipo de trabajo
R25	Cronograma modificado	Debido a la complejidad del sistema o retrasos de los módulos del software

R26	Inapropiada seguridad del sistema	El grupo de desarrollo no especifica medidas de seguridad que protejan la información de usuarios
R27	Complejidad	La arquitectura del sistema presenta un alto grado de complejidad para su desarrollo
R28	Conocimiento inadecuado	El grupo de desarrollo no conoce el lenguaje de programación en el que se desarrollará el sistema
R29	Insuficiencia de herramientas	No contar con los recursos de hardware suficientes para completar el proyecto
R30	Integración fallida del sistema	Se produce por un mal diseño del software Requerimientos contradictorios que no fueron analizados correctamente
R31	Escasez de personal	No contar con el personal indispensable para el desarrollo y que permita cumplir con los plazos de tiempo establecidos
R32	Cambio de tecnología empleada para el proyecto	Forzoso cambio de tecnologías que están fuera del cronograma
R33	validaciones de sistema	No se efectúan las validaciones necesarias para el correcto funcionamiento del software
R34	Documentación	Por recursos de tiempo no se establece la documentación del código fuente indispensable para futuros cambios
R35	Copias de seguridad	Perdida de la última versión del sistema
MANTENIMIENTO		
R36	Escasez de recursos	El cliente no dispone de los recursos de hardware necesarios para la implantación del software
R37	Documentos del usuario incompletas	No se realiza entrega de manuales de usuario para entendimiento del sistema
R38	complejidad	No se realizaron capacitaciones para educar al personal del funcionamiento del sistema
R39	Falta de pruebas	No se realizaron suficientes pruebas de funcionamiento del sistema antes de su implementación

Tabla 3 Riesgos potenciales en el desarrollo de software en base al ciclo de vida del desarrollo de software

Otros aspectos a considerar dentro del software web son los presentados por Kendall [26] y Elbanna [27] enmarcados en la temática de investigación.

NAVEGACIÓN		
R40	Difícil navegación	Las paginas no son diseñadas adecuadamente para dirigirnos a ellas. Falta de capacitación al personal del sistema.
R41	Volumen de información	Contiene gran cantidad de información redundante dentro de sus páginas.
EQUIPO DE DESARROLLO		

R42	Falta de utilización de metodologías	Los desarrolladores no poseen los conocimientos necesario sobre una metodología de desarrollo de software.
R43	Conflictos	El equipo de trabajo mantiene conflictos laborales durante el desarrollo
R44	Retiro imprevisto	Los integrantes deciden no continuar trabajando en el proyecto por motivos externos o internos en la organización.

Tabla 4 Riesgos potenciales sobre navegación y equipo de desarrollo del sistema

En la figura 4 se presenta de manera gráfica la cantidad de riesgos identificados para cada fase de desarrollo de software, para lo cual se puede apreciar que en la fase de codificación es donde se pueden identificar la mayor cantidad de riesgos potenciales que pueden afectar negativamente el proyecto.

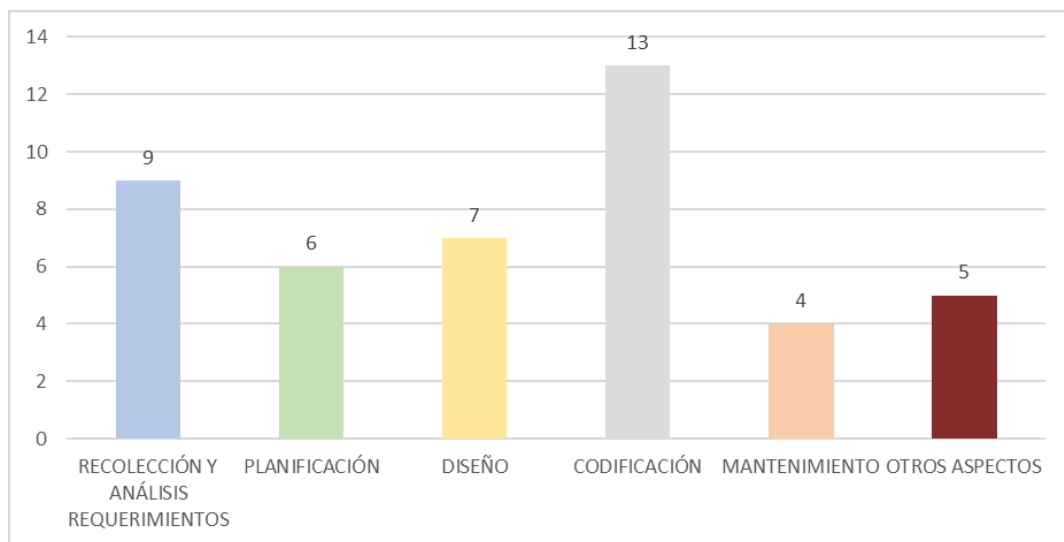


Figura 4 Riesgos identificados por cada fase del ciclo de vida del software

3. CONCLUSIONES

En base a la investigación realizada se logró identificar una gran cantidad de riesgos potenciales que están presentes en el desarrollo de software web especificando las posibles causas que lo pueden provocar, con la finalidad de mitigar estos peligros de desarrollo es necesario emplear un modelo de gestión de riesgos en el cual se establezcan putas estructurales que le permitirán establecer estrategias para prevenir una posible afectación en sistema.

Una mayor cantidad de riesgos se encuentran en la cuarta fase del ciclo de vida del software (codificación), siendo esta etapa en la que mayor cantidad de estrategias deben emplearse, por su parte la utilización de metodologías ágiles es un de más medidas recomendables de utilizar en un proyecto de software web, esto debido a que permite mayor adaptabilidad a los diferentes cambios que pueden presentarse en cualquiera de las etapas de desarrollo de software.

El uso de modelos o estándares para la gestión de riesgos es indispensable en todo proyecto ya que estos modelos dentro de su estructura establecen marcos logísticos y procesos sistemáticos que se deben ser tomados en cuenta a fin de aumentar el índice de probabilidad del logro efectivo de un proyecto, entre estos modelos podemos encontrar APM, CMMI y la guía PMBOK y los estándares ISO 21500 y 31000. La guía PMBOK es el modelo de gestión de riesgo en el desarrollo de un proyecto de software que cuenta con más estructuración en su modelo con referencia a las fases del manejo de riesgos: planificar la gestión del riesgo, identificar, analizar, planificar la respuesta y controlar los riesgos.

BIBLIOGRAFÍA

- [1] J. R. Molina Ríos, J. A. Honores Tapia, M. P. Zea Ordóñez, and A. S. Gómez Moreno, "Analysis Methodologies Web Application Development," *Int. J. Appl. Eng. Res.*, vol. 11, no. 16, pp. 9070–9078, 2016.
- [2] J. R. Molina Ríos, V. P. Milton, and M. P. Zea Ordóñez, *Diseño de sistemas*, UTMACH. Machala, 2015.
- [3] J. R. Molina Ríos and M. P. Zea Ordóñez, "SNAIL a hybrid model for the management of agile web software development processes," *Int. J. Eng. Res. Technol.*, vol. 11, no. 7, pp. 1067–1083, 2018.
- [4] F. F. Redrován Castillo, N. M. Loja Mora, K. D. Correa Elizaldes, and J. I. Piña Orozco, "Comparación de métricas de calidad para el desarrollo de aplicaciones web," *3C Tecnol.*, vol. 7, no. 3, pp. 94–113, 2018.
- [5] J. R. Molina Ríos, J. Honores Tapia, and M. P. Zea Ordóñez, *Nociones de ingeniería de software*. Machala, 2015.
- [6] P. Morrison, D. Moye, R. Pandita, and L. Williams, "Mapping the Field of Software Life Cycle Security Metrics," *Inf. Softw. Technol.*, vol. 1, p. 20, 2018.
- [7] G. Papadopoulos, "Moving from traditional to agile software development methodologies also on large , distributed projects .," *Procedia - Soc. Behav. Sci.*, vol. 175, pp. 455–463, 2015.
- [8] J. R. Molina Ríos and M. P. Zea Ordóñez, "Metodologías de desarrollo en aplicaciones web," *ARJÉ*, vol. 11, no. 21, pp. 245–270, 2017.
- [9] Project Management Institute, *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK Sexta edición)*. Newtown Square, 2017.
- [10] A. Olechowski, J. Oehmen, W. Seering, and M. Ben-daya, "The professionalization of risk management: What role can the ISO 31000 risk management principles play ?," *Int. J. Proj. Manag.*, vol. 34, no. 8, pp. 1568–1578, 2016.
- [11] J. R. Molina Ríos, N. M. Loja Mora, M. P. Zea Ordóñez, and E. L. Loaiza Sojos, "Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python," *Rev. Latinoam. Ing. Softw.*, vol. 4, no. 4, pp. 201–207, 2016.
- [12] J. J. Hao Song, "Risks Identification in Embedded Software Development: Evidence from MVBC Project Survey," *Inf. Technol. Quant. Manag.*, vol. 91, pp. 798–806, 2016.
- [13] C. F. Oduoza, O. Odimabo, and A. Tamparapoulos, "Framework for Risk Management Software System for SMEs in the Engineering Construction Sector," *Procedia Manuf.*, vol. 11, no. June, pp. 1231–1238, 2017.
- [14] M. Wanderley, J. Menezes, C. Gusmão, and F. Lima, "Proposal of risk management metrics for multiple project software development," *Procedia - Procedia Comput. Sci.*, vol. 64, pp. 1001–1009, 2015.
- [15] A. Calderón, M. Ruiz, and R. V. O. Connor, "Computer Standards & Interfaces A serious game to support the ISO 21500 standard education in the context of software project management," *Comput. Stand. Interfaces*, vol. 5, pp. 1–13, 2018.
- [16] F. Javier and A. Sánchez, "Expert system for predicting buildings service life under ISO 31000 standard. Application in architectural heritage," *Cult. Herit.*, vol. 18, pp.

209–218, 2016.

- [17] Association for Project Management, *Project risk analysis and management*. United Kingdom: APM, 2018.
- [18] H. Rahmani, A. Sami, and A. Khalili, “CIP-UQIM: A unified model for quality improvement in software SME’s based on CMMI level 2 and 3,” *Inf. Softw. Technol.*, vol. 71, pp. 27–57, 2016.
- [19] N. Saher, F. Baharom, and O. Ghazali, “Requirement Change Taxonomy and Categorization in Agile Software Development,” in *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, 2017, pp. 1–6.
- [20] M. Ahmadvand, A. Pretschner, and F. Kelbert, “A Taxonomy of Software Integrity Protection Techniques,” in *Advances in Computers*, 1st ed., Elsevier Inc., 2018, pp. 1–73.
- [21] A. Aslam *et al.*, “Decision Support System for Risk Assessment and Management Strategies in Distributed Software Development,” *IEEE Access*, vol. 5, pp. 20349–20373, 2017.
- [22] M. Xiaojuan, “Research and Implementation of Computer Data Security Management System,” *Procedia Eng.*, vol. 174, pp. 1371–1379, 2017.
- [23] M. Pasha, G. Qaiser, and U. Pasha, “A Critical Analysis of Software Risk Management Techniques in Large Scale Systems,” *IEEE Access*, vol. 6, pp. 12412–12424, 2018.
- [24] A. Kumar, O. Williams, and X. Li, “Towards an efficient risk assessment in software projects – Fuzzy reinforcement paradigm R,” *Comput. Electr. Eng.*, pp. 1–14, 2017.
- [25] S. Kim and H. Lee, “Software systems at risk: An empirical study of cloned vulnerabilities in practice,” *Comput. Secur.*, vol. 77, pp. 720–736, 2018.
- [26] R. P. Kendall *et al.*, “Risk-Based software development practices for create multiphysics HPC software applications,” *Comput. Sci. Eng.*, vol. 18, no. 6, pp. 35–46, 2016.
- [27] A. Elbanna and R. Holloway, “Risks of Agile Software Development: Learning from Adopters,” *IEEE Softw.*, vol. 33, no. 4, pp. 1–15, 2016.