

***GESTIÓN DEL CONOCIMIENTO EN LA PLANIFICACIÓN DE PROYECTOS
SOFTWARE EN GRUPOS DE INVESTIGACIÓN.***

LUIS FERNANDO SIERRA JOYA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD FISICOMECHANICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
MAESTRÍA EN INGENIERÍA INDUSTRIAL
BUCARAMANGA
2013**

**GESTIÓN DEL CONOCIMIENTO EN LA PLANIFICACIÓN DE PROYECTOS
SOFTWARE EN GRUPOS DE INVESTIGACIÓN.**

LUIS FERNANDO SIERRA JOYA

***Trabajo de investigación realizado para optar por el título de Magister en
Ingeniería Industrial***

Director:

MSc. LUIS CARLOS GOMEZ FLÓREZ

Codirector:

MSc. LUIS EDUARDO BECERRA ARDILA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD FISICOMECHANICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
MAESTRÍA EN INGENIERÍA INDUSTRIAL
BUCARAMANGA
2013**

CONTENIDO

INTRODUCCIÓN.....	10
1. FUNDAMENTACIÓN TEÓRICA	13
1.1 Gestión del conocimiento	13
1.2 Modelos de gestión de conocimiento.....	15
1.3 Planificación de proyectos software	22
1.4 Gestión de conocimiento en el desarrollo de software.....	31
1.5 Revisión Sistemática de literatura	33
1.6 Metodología de Investigación-acción.....	35
2. OBJETIVOS	38
2.1 Objetivo General	38
2.2 Objetivos Específicos	38
3. METODOLOGÍA.....	39
4. RESULTADOS	44
4.1 Caracterización del proceso de desarrollo Software en grupos de investigación	44
4.2 Conocimientos y procesos relevantes para la planificación de proyectos software en grupos de investigación universitarios.....	52
4.3 Modelo de Gestión de conocimiento en la planificación de proyectos software .	54
4.4 Prácticas y Herramientas para la Gestión de conocimiento aplicables en grupos de investigación que desarrollan software.....	55
4.5 Gestión del alcance de los proyectos de desarrollo software en grupos de investigación universitarios	70
4.6 Gestión del tiempo en proyectos de desarrollo software en grupos de investigación universitarios	72
5. CONCLUSIONES	74
BIBLIOGRAFÍA.....	76

LISTA DE TABLAS

Tabla 1.	Taxonomías de conocimiento y ejemplos.....	14
Tabla 2.	Clasificación de los modelos del capital intelectual.	15
Tabla 3.	Herramientas en los procesos de creación de conocimiento.....	20
Tabla 4.	Fases genéricas de desarrollo de software	25
Tabla 5.	Áreas del PMBOK	28
Tabla 6.	Palabras claves.....	42
Tabla 7.	Producto Software como resultado de investigación, requerimientos de existencia, categorías y puntajes relativos	44
Tabla 8.	Tipos, subtipos de integrantes y los requisitos respectivos.	46
Tabla 9.	Características del proceso de desarrollo Software en grupos de investigación	49
Tabla 10.	Clasificación de Herramientas y prácticas de gestión del conocimiento.....	56
Tabla 11.	Prácticas de gestión de conocimiento centradas en la gestión del conocimiento explícito.....	57
Tabla 12.	Prácticas de gestión de conocimiento centradas en la gestión del conocimiento tácito	60
Tabla 13.	Prácticas y herramientas de apoyo a la gestión de conocimiento	63
Tabla 14.	Herramientas de gestión conocimiento.....	69
Tabla 15.	Herramientas de Requerimientos.	71

LISTA DE FIGURAS

Figura 1.	Espiral de creación de conocimiento organizativo	18
Figura 2.	El ciclo de Gestión conocimiento.....	19
Figura 3.	Modelo Integrado Situacional de Gestión de conocimiento	21
Figura 4.	El ciclo de la investigación-acción	37
Figura 5.	Modelo de Gestión de conocimiento en la planificación de proyectos software	55

RESUMEN EN ESPAÑOL

TÍTULO: GESTIÓN DEL CONOCIMIENTO EN LA PLANIFICACIÓN DE PROYECTOS SOFTWARE EN GRUPOS DE INVESTIGACIÓN.¹

AUTOR: LUIS FERNANDO SIERRA JOYA

PALABRAS CLAVES: Gestión de conocimiento, planificación de proyectos software, Herramientas gestión conocimiento, grupos de investigación

DESCRIPCIÓN

En la actualidad las organizaciones han dado gran importancia al conocimiento generado dentro y fuera de la organización, considerándolo un activo valioso en la obtención de ventajas competitivas sostenibles y su influencia en la mejora de sus procesos, lo cual hace necesario crear estrategias direccionadas a una adecuada gestión del conocimiento; en particular en las organizaciones desarrolladores del software, la gestión del conocimiento juega un rol importante dado que permite descubrir nuevo conocimiento, capturarlo, almacenarlo, recuperarlo, compartirlo y entenderlo en pro de mejora de los proyectos desarrollados en términos de reducción de costos, mayor precisión en la definición del alcance y del tiempo y mejora en la calidad del proyecto. Este trabajo de investigación busca, mediante una metodología de investigación acción, identificar prácticas, herramientas y factores que contribuyan a la gestión de conocimiento en el proceso de planificación de proyectos de desarrollo Software en grupos de investigación universitarios con el fin de proponer un modelo guía en la incorporación de la gestión de conocimiento en la planificación de proyectos software, tomando como caso de estudio el grupo de investigación STI. Entre los principales resultados se encuentran conocimientos relevantes en las áreas de gestión de alcance, la gestión del tiempo y el seguimiento y control a los proyectos, las cuales se pueden mejorar con la implementación de prácticas y herramientas de gestión de conocimiento como: sistemas colaborativos, sistemas para la gestión de documentos, capacitaciones, sistemas para el manejo de lecciones aprendidas, entre otros.

¹Facultad Fisicomecánicas. Escuela de Estudios industriales y empresariales. Director Luis Carlos Gómez Flórez. Codirector Luis Eduardo Becerra Ardila

ABSTRACT

TITLE: KNOWLEDGE MANAGEMENT IN THE PROJECT PLANNING SOFTWARE IN RESEARCH GROUP.².

AUTHOR: LUIS FERNANDO SIERRA JOYA

KEYWORDS: Knowledge management, Project planning software, Knowledge management tools, Research groups

DESCRIPTION

Currently, the organizations have attached great importance to knowledge generated inside and outside the organization, considering it a valuable asset in obtaining sustainable competitive advantages and its influence on the improvement of its processes, therefore is necessary to create strategies delivered directly to the proper management of knowledge in organizations particularly in software developer's organizations. The knowledge management plays an important role since it allows discovering new knowledge, capture, store, retrieve, share and understanding towards improving the projects in terms of reduced costs, greater precision in defining the scope and time and improving the quality of the project. This research seeks, through action research methodology, identify practices, tools and factors that contribute to knowledge management in the planning process in software development projects in university research groups in order to propose a model to guide the incorporation of knowledge management in software project planning, taking as a case study research group STI. The main results are relevant expertise in the areas of scope management, time management and monitoring and control of projects, which can be improved with the implementation of practices and knowledge management tools such as collaborative systems, systems for document management, training, management systems for lessons learned, among others.

²Facultad Fisicomecánicas. Escuela de Estudios industriales y empresariales. Director Luis Carlos Gómez Floréz Codirector. Luis Eduardo Becerra Ardila.

INTRODUCCIÓN

Las organizaciones dan gran importancia al conocimiento generado por las personas, considerándolo como un activo valioso para la obtención de ventajas competitivas principalmente por su carácter intangible, lo cual lo hace difícil de imitar. Esto ocasiona que la gestión del conocimiento se convierta en un asunto primordial para muchas organizaciones, puesto que gestionar el conocimiento incrementa la capacidad de la organización para aprender de su entorno e incorporar nuevo conocimiento en sus procesos de negocios (Nonaka, Krogh, & Voelpel, 2006; Nonaka & Takeuchi, 1995). Por lo tanto, las nuevas características de las organizaciones orientadas al conocimiento requieren nuevas estructuras, cambios culturales profundos, nuevas plataformas tecnológicas y modelos para la estructuración del conocimiento y de los procesos que permiten gestionarlo (Capote et al., 2008).

En particular, las organizaciones en el área de ingeniería de software requieren un uso intensivo de conocimientos en el negocio, actividades, técnicas, conocimiento de nuevas tecnologías, lecciones aprendidas y manejo adecuado de los tiempos en la ejecución de proyectos software (Aurum, Daneshgar, & Ward, 2008; Dingsøyr & Conradi, 2002; Ioana Rus & Lindvall, 2002). Estos conocimientos no son estáticos en la organización sino que se encuentran en constante crecimiento a medida que se desarrollan diversos proyectos, razón por la cual, a menudo, las organizaciones tienen problemas para identificar el contenido, la ubicación y uso del conocimiento. A su vez, la invención de software requiere un esfuerzo amplio en el levantamiento de requerimientos, análisis, validación y diseño; estos muchas veces se ven afectados por no tener en cuenta los errores cometidos en el pasado, haciendo más costosa la implementación de estos proyectos (Corbin, Dunbar, & Zhu, 2007).

Ahora bien, las carencias o deficiencias en la planificación de proyectos, conducen a problemas como: proyectos que se abandonan, llevan más tiempo del esperado y presentan mayores costos generados en comparación con los costos planificados. De igual forma, la calidad del producto generado puede verse comprometida y el cliente puede tener una insatisfacción del producto desarrollado, ya que no cumple con los requisitos de una manera adecuada o no cumplen con el propósito con el que fue desarrollado (Bjornson & Dingsøyr, 2008; Ioana Rus & Lindvall, 2002).

En cuanto al desarrollo software enmarcado en actividades de investigación, en el interior de grupos de investigación, es realizado por equipos de desarrollo poco maduros, los cuales están conformados por un alto porcentaje por estudiantes que en ocasiones se enfrentan a sus primeros desarrollos presentándose problemas en la definición del alcance y en la gestión de tiempos del proyecto, así mismo al terminar el proyecto se van del grupo llevándose consigo parte del conocimiento adquirido en la elaboración del proyecto, el cual no queda explícito en los productos resultantes del proyecto, por tanto este conocimiento no puede ser aprovechado para el beneficio del grupo y sus futuros integrantes (Gaviria, Mejía, & Henao, 2007); lo anterior genera que el conocimiento obtenido no permanezca en la organización, y los nuevos integrantes deben planificar sus proyectos de desarrollo sin tener en cuenta el conocimiento generado por otros y repitiendo errores del pasado. A su vez, existe falta de aprovechamiento de conocimientos acerca de las temáticas de dominio en otras disciplinas para las cuales se han realizado desarrollos de software (Bjornson & Dingsøyr, 2008; Corbin et al., 2007). Según León, Gómez, & Pimentel (2011) definir el alcance del proyecto en grupos de investigación, puede ser una tarea bastante compleja ya que los estudiantes se están enfrentando en la mayoría de los casos con sus primeros desarrollos y el manejo inadecuado de los cambios que ocurren en el proyecto puede modificar radicalmente la finalización del proyecto.

Algunos autores como (Aurum et al., 2008; Bjørnson, 2007; Corbin et al., 2007; Desouza & Awazu, 2006; Dingsøyr & Conradi, 2002; Iversen, Mathiassen, & Nielsen, 2004; Ioana Rus & Lindvall, 2002) destacan la importancia de incorporar la gestión de conocimiento en la ingeniería del software, principalmente por la variedad y gran proporción de conocimientos. Corbin et al. (2007, p.3) destaca “la gestión de conocimiento permite en los proyectos de desarrollo de software, descubrir nuevo conocimiento, capturarlo, almacenarlo, recuperarlo, compartirlo y entenderlo, y eso sucede en el desarrollo de proyectos de ingeniería de software”. Por su parte (Aurum et al., 2008; Ward & Aurum, 2004) destaca: la gestión de conocimiento contribuye a reducir los costos, cumplir con los tiempos planificados y mejorar la calidad del proyecto. A su vez, las organizaciones desarrolladoras de software pueden ver la gestión del conocimiento como el aprovechamiento del conocimiento de los individuos en beneficio de la organización.

Sin embargo pese a la importancia de gestionar el conocimiento en proyectos de desarrollos software, todavía son pocos los estudios destinados a la creación, transferencia y aplicación de conocimiento en la organizaciones desarrolladoras de software (Bjornson & Dingsøyr, 2008), del mismo modo Aurum et al., (2008) identifica que los métodos, técnicas y herramientas empleados actualmente para

abordar la gestión de conocimiento en organizaciones de desarrollo de software son insuficientes. Por otra parte, en la literatura revisada se evidenció que son pocos los estudios enfocados en la formulación de modelos, metodologías y marcos de trabajo diseñados para guiar la incorporación de la gestión de conocimiento en la gestión de proyectos software en grupos de investigación, especialmente en el contexto colombiano.

La presente investigación pretende abordar, bajo un enfoque cualitativo y se va a seguir el diseño metodológico de investigación–acción (R. L. Baskerville, 1999; Checkland & Scholes, 1990; Sampieri, Fernández, & Baptista, 2010), el estudio sobre gestión del conocimiento en el proceso de desarrollo de software enfocándose en la etapa de planificación de los proyectos en grupos de investigación universitarios, con el fin de que los conocimientos aprendidos, sean aprovechados en beneficio de la organización y que futuros integrantes puedan hacer uso de este conocimiento.

Este proyecto de investigación tiene como objetivo general: “Proponer un modelo para la gestión de conocimiento en grupos de investigación que desarrollan software, que contribuya a la planificación de sus proyectos, tomando como caso de estudio el Grupo de Investigación en Sistemas y Tecnologías de Información (STI), adscrito a la Escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander”. Los resultados de este proyecto pueden servir como punto de referencia a futuras investigaciones cuyo propósito sea la formulación de modelos, metodologías y marcos de trabajo para gestionar conocimientos en el proceso de desarrollo software en grupos de investigación.

Este informe se estructura de la siguiente manera: a continuación la fundamentación teórica de la investigación en donde se da una contextualización de la gestión del conocimiento y su influencia en la gestión de proyectos software, posteriormente se presenta la forma en cómo se abordó la investigación, en tercera instancia se presentan los resultados del proyecto dividido en tres partes. Caracterización del proceso de desarrollo software en grupos de investigación, conocimiento relevantes y modelo de gestión de conocimiento propuesto; finalmente se presentan las conclusiones y las referencias.

1. FUNDAMENTACIÓN TEÓRICA

1.1 Gestión del conocimiento

A través del tiempo, la comunidad científica ha estado interesada en investigar acerca de la fuente de la ventaja competitiva para las organizaciones; en un principio se consideró la cantidad de los recursos (activos tangibles) disponibles en la organización, denominado enfoque basado en los recursos (Amit & Shoemaker, 1993; Collis & Montgomery, 1995; Grant, 1991). Más adelante las organizaciones empiezan a prestar mayor atención al conocimiento de las personas (activos intangibles), considerándose desde entonces el conocimiento como la principal fuente de ventaja competitiva de una organización (Nonaka et al., 2006; Nonaka & Takeuchi, 1995), presentándose la necesidad de administrar y almacenar este conocimiento (Alavi & Leidner, 2001; Liao, 2003), por lo cual surge la gestión del conocimiento, encargada de buscar la manera de hacer aprovechable el conocimiento de las personas pertenecientes a la organización y hacer que este conocimiento sea de fácil acceso en cualquier momento con el fin de crear nuevo conocimiento que permita a la organización ser creadora de nuevas estrategias para hacerla competitiva (Nonaka & Takeuchi, 1995).

La gestión del conocimiento, según Alavi y Leidner (2001), es el proceso de identificación, captura y utilización del conocimiento en una organización con el fin de incrementar la competitividad organizativa. Con base en esto, es relevante conocer la definición de conocimiento, ya que en ocasiones no se diferencia entre los términos conocimiento, información y dato, utilizándose indistintamente. La información está conformada por datos que fueron capturados de observaciones del mundo o capturados por máquinas y al realizar análisis o interpretaciones tienen pertinencia y propósito; el conocimiento es una combinación fluida de experiencias e información, reflexión, valores; alguien ha aplicado su propia sabiduría, considerado sus implicaciones más amplias (Davenport & Prusak, 1999). En este sentido el conocimiento se encuentra mucho más relacionado con la acción, ya que le permite al individuo tomar decisiones y actuar, Por otra parte, varios autores han desarrollado taxonomías para ampliar la descripción del tipo de conocimiento; sin embargo, la más destacada en la literatura es la propuesta por Nonaka y Takeuchi (1995); estos autores dividen el conocimiento en explícito y tácito. El explícito es aquel conocimiento que puede ser transmitido utilizando lenguaje formal, mientras que el tácito es un conocimiento personal, adquirido por la experiencia el cual resulta difícil de formalizar y comunicar. La tabla 1 presenta una clasificación dada por Alavi y Leidner (2001) sobre la taxonomía del conocimiento.

Tabla 1. Taxonomías de conocimiento y ejemplos

Tipos de Conocimiento	Definiciones	Ejemplos
Tácito	El conocimiento está en las acciones, experiencias y está envuelto en un contexto específico	Formas de relacionarse con un cliente específico.
Tácito Cognitivo	Modelos mentales	Creencias individuales sobre relaciones causa-efecto.
Tácito Técnico	"Know-how" aplicable a un trabajo específico.	Habilidades en cirugía.
Explícito	Articulado, conocimiento generalizado.	Conocimiento sobre los principales clientes de una zona.
Individual	Creado por e inherente al Individuo.	Percepciones conseguidas a través de un proyecto concluido
Social	Creado por e inherente a las acciones colectivas de un grupo.	Normas de comunicación entre grupos.
Declarativo	<i>Know-about</i>	Qué medicamento es apropiado para una enfermedad.
Causal	<i>Know-why</i>	Comprender por qué los medicamentos son eficaces.
Condicional	<i>Know-when</i>	Comprender cuándo prescribe un medicamento.
Relacional	<i>Know-with</i>	Comprender cómo interactúa un medicamento con otros grupos de medicamentos.
Pragmático	Utilidad de un conocimiento para una organización	Mejores prácticas, estructura de negocio, experiencias en proyectos, dibujos de ingeniería, informes de mercado.

Fuente: Alavi y Leidner, (2001) Traducción libre

1.2 Modelos de gestión de conocimiento

Según Riesco (2006) clasifica los modelos de gestión de conocimiento en dos categorías: Modelos de clasificación y medición del capital intelectual y Modelos de gestión de conocimiento en donde se incluyen también los modelos de gestión de la información. Para Riesco (2006), la diferencia entre estas dos categorías radica en que los primeros tienen un tratamiento más estático y acumulativo, en cambio los segundos hacen hincapié en la característica dinámica del conocimiento.

Algunos modelos de clasificación del capital intelectual son: Modelo Balanced Scorecard (Kaplan y Norton, 1996); Modelo Navigator de Skandia (Edvinson, 1996; Edvinsson y Malone, 1997); Modelo Technology Broker (Brooking, 1996), Modelo Universidad de West Ontario (Bontis, 1996); Modelo Canadian Imperial Bank (Saint-Onge, 1996) y Modelo Capital Intelectual (Drogonetti y Roos, 1997); Modelo Intellectual Assets Monitor(Sveiby, 1997); Modelo de Dirección Estratégica por competencias (Bueno, 1998); Modelo de Intelect (Euroforum, 1998); el Modelo Nova (Camisón, Palacios y Devece, 2000). La siguiente tabla presenta una síntesis de cada uno de los modelos presentados anteriormente respecto a la clasificación del capital intelectual propuesta en cada modelo.

Tabla 2. Clasificación de los modelos del capital intelectual.

Modelos	Clasificación del CI
Modelo Balanced Scorecard (Kaplan y Norton, 1996)	• Perspectiva Financiera • Perspectiva del cliente • Perspectiva de procesos internos de negocio • Perspectiva del aprendizaje y de crecimiento
Modelo Navigator de Skandia (Edvinson & Malone, 1999)	• Enfoque Financiero • Enfoque de clientes. • Enfoque humano. • Enfoque de procesos • Enfoque de renovación y desarrollo
Modelo Technology Broker (Brooking, 1999)	• Activos de mercado • Activos de propiedad intelectual • Activos centrados en el

	individuo • Activos de infraestructura.
Modelo Universidad de West Ontario (Bontis, 1996);	• Capital Humano • Capital Estructural • Capital Clientes.
Modelo Canadian Imperial Bank (Saint-Onge, 1996)	• Capital Financiero • Capital Humano • Capital Estructural • Capital Clientes.
Modelo Capital Intelectual (Ross, Edvinsson y Drogonetti, 1997)	• Capital Humano • Capital Estructural • Capital Clientes.
Modelo Intellectual Assets Monitor(Sveiby, 2001)	• Competencias de los colaboradores • Componente interno • Componente externo.
Modelo de Dirección Estratégica por competencias (Bueno, 1998);	• Capital Humano • Capital organizativo • Capital Tecnológico. • Capital Relacional.
Modelo de Intelect (Euforum., 1998)	• Capital Humano • Capital Estructural • Capital Relacional.
Modelo Nova (Camson, Palacios, & Devece, 2000)	• Capital Humano • Capital organizativo • Capital social • Capital de innovación y aprendizaje

Fuente: Tomado de (Riesco, 2006).

Respecto a los modelos de gestión de la información, tres modelos se destacan en la literatura, los cuales abordan aspectos de la gestión del conocimiento, estos son: El modelo Itami, el modelo Rowley y el modelo sintetizador de (Cornella, 2000). El modelo itami propone tres niveles para realizar actividades de gestión de la información, estos niveles son: Información sobre el entorno, Información Interna e información proyectada. En el modelo de Rowley se hace una diferenciación entre dos disciplinas, la microinformática que estudia la relación entre las personas concretas y la información y la macroinformática que estudia relaciones informáticas entre la organización concreta y el entorno social. (Conella, 2000) propone un modelo sintetizados para la gestión de la información que abarca cuatro etapas: 1) Obtención de la información, 2) Estructuración de la

información, 3) Distribución de la información, 4) Uso de la Información. Otros modelos de gestión de conocimiento son: El modelo de gestión de conocimiento de KPMG Consulting (Tejedor y Aguirre, 1998), el modelo de Arthur Andersen (Arthur, Andersen, 1999), Knowledge Management Assessment Tool KMAT, entre otros.

Respecto a los modelos de gestión de conocimiento, el conocimiento es adquirido en un proceso a través del aprendizaje, las experiencias, creencias, comprensión, valores, reflexión lo cual genera la capacidad de interpretar información y toma de decisiones (Davenport & Prusak, 1999; Nonaka & Takeuchi, 1995; Vargas, 2003). Así como variadas son las definiciones de conocimiento, también puede encontrarse diferentes enfoques y aproximaciones al concepto de gestión de conocimiento.

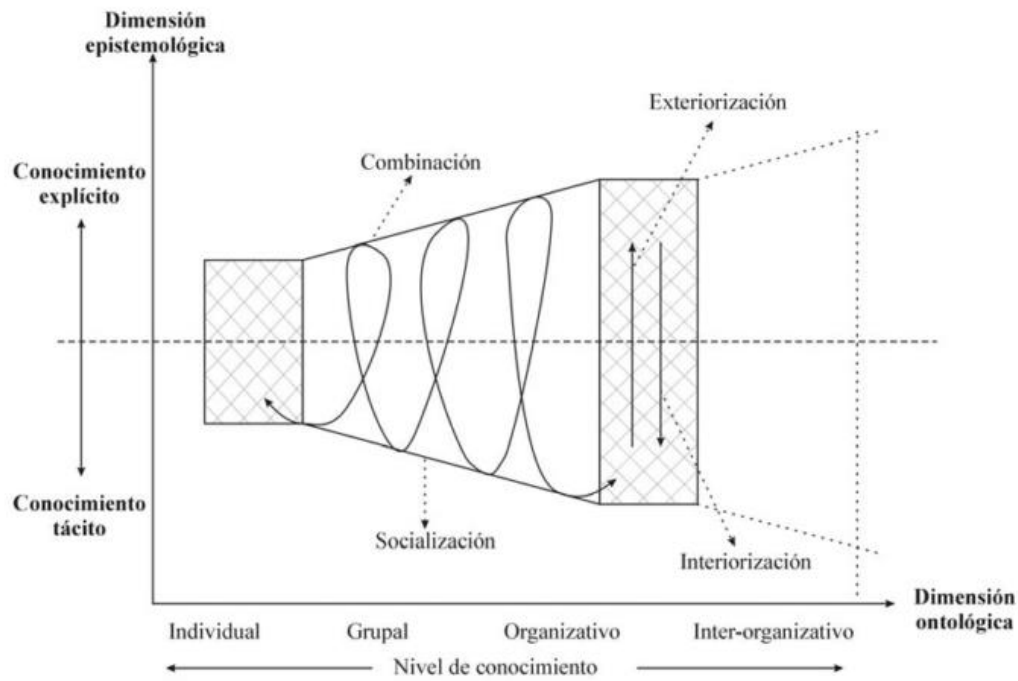
- La gestión de conocimiento es la capacidad de la empresa para crear conocimiento nuevo, diseminarlo en la organización e incorporarlo en productos, servicios y sistemas (Nonaka & Takeuchi, 1995).
- La gestión de conocimiento es cualquier proceso para crear, adquirir, capturar, compartir y usar conocimiento, cualquiera sea el lugar donde resida, para mejorar el aprendizaje y el desempeño en las organizaciones (Quintas, Lefrere, & Jones, 1997).
- La gestión de conocimiento es la construcción y aplicación sistemática, explícita y deliberada de conocimiento para maximizar la efectividad organizacional con respecto al conocimiento, por lo que usa sus activos de conocimiento (Wiig, 1997).
- Proceso de identificación, captura y utilización del conocimiento en una organización con el fin de incrementar la competitividad organizativa (Alavi & Leidner, 2001).

De las anteriores definiciones se desprende que la mayoría definen la gestión del conocimiento como un proceso donde se crea, se captura, se comparte y se utiliza el conocimiento en la organización para mejorar el desempeño de la organización.

Uno de los modelos más relevantes en la literatura para la gestión de conocimiento es el propuesto por (Nonaka & Takeuchi, 1995) conocido como la espiral de creación de conocimiento o modelo SECI por sus procesos (Socialización, Externalización, Combinación e Interiorización), es este modelo los autores (Nonaka & Takeuchi, 1995) conciben la creación de conocimiento

organizativo como un proceso en espiral que se inicia con la generación de un conocimiento tácito en el nivel individual, que posteriormente se moviliza y amplifica a través de un proceso de transformación entre el conocimiento tácito y explícito a través de las cuatro formas de conversión de conocimiento y cristalizado en niveles ontológicos más altos. A esto lo llaman espiral de conocimiento (figura 1).

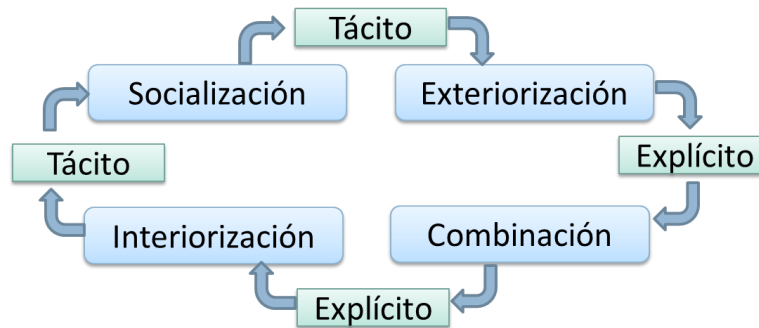
Figura 1. Espiral de creación de conocimiento organizativo



Fuente: (Nonaka & Takeuchi, 1995)

según Nonaka y Takeuchi (1995) el conocimiento tácito y el explícito no son entidades separadas, sino complementarias. Se encuentra una interacción e intercambio entre ellos en las actividades creativas de los seres humanos. Es lo que se denomina conversión de conocimiento. Por lo tanto, Nonaka y Takeuchi (1995) proponen cuatro formas de conversión del conocimiento: socialización, exteriorización, interiorización y combinación como se aprecia en la figura 2.

Figura 2. El ciclo de Gestión conocimiento



Fuente: (Nonaka & Takeuchi, 1995).

El proceso de socialización (de tácito a tácito) permite compartir experiencias y crear conocimiento tácito. La forma de adquirir conocimiento tácito sin utilizar el lenguaje es mediante la observación, la imitación y la práctica.

El proceso de exteriorización es la conversión de conocimiento tácito en conceptos explícitos. Esta conversión hace posible que el conocimiento tácito sea expresado en metáforas, analogías, conceptos, hipótesis o modelos.

El proceso de combinación (de explícito a explícito) implica la combinación de distintos cuerpos de conocimiento explícito. Se combina e intercambia conocimiento a través de distintos medios, tales como documentos, juntas conversaciones telefónicas o redes de comunicación. La combinación se origina mediante la vinculación del conocimiento recién creado con el conocimiento existente en distintas partes de la organización, por ejemplo, se puede reflejar en un nuevo producto o servicio.

Por último el proceso de interiorización la conversión de conocimiento explícito en conocimiento tácito. Para que el conocimiento explícito se vuelva tácito, es de gran utilidad que el conocimiento se verbalice o diagrame en manuales o historias orales. La documentación ayuda a interiorizar lo que han experimentado, enriqueciendo, por tanto su conocimiento tácito.

Así mismo cada una de los procesos del modelo SECI, puede ser soportado mediante el uso de tecnologías de información tal como lo señala (Tiwana, 2000; Riesco, 2006) en la siguiente tabla:

Tabla 3. Herramientas en los procesos de creación de conocimiento

Socialización	Externalización
• Comunicaciones cara a cara • Videoconferencias • Webcams • Herramientas de realidad virtual.	• Herramientas para captar procesos • Habilidad de rastreo • Redes de iguales • Sistemas expertos • Plataformas de discusión
Interiorización	Combinación
• Redes de conocimiento colectivo • Bases de datos/memoria Corporativa • Reconocimiento de patrones • Redes de trabajo	• Herramientas de conocimiento sistémico • Herramientas colaborativas • Intranets, groupware • Listas de discusión • Foros • Bases de datos de buenas prácticas.

Fuente: Tomado de (Tiwana, 2000; Riesco, 2006)

Por otro lado (Gómez, 2006), clasifica los modelos de gestión de conocimiento en tres categorías :

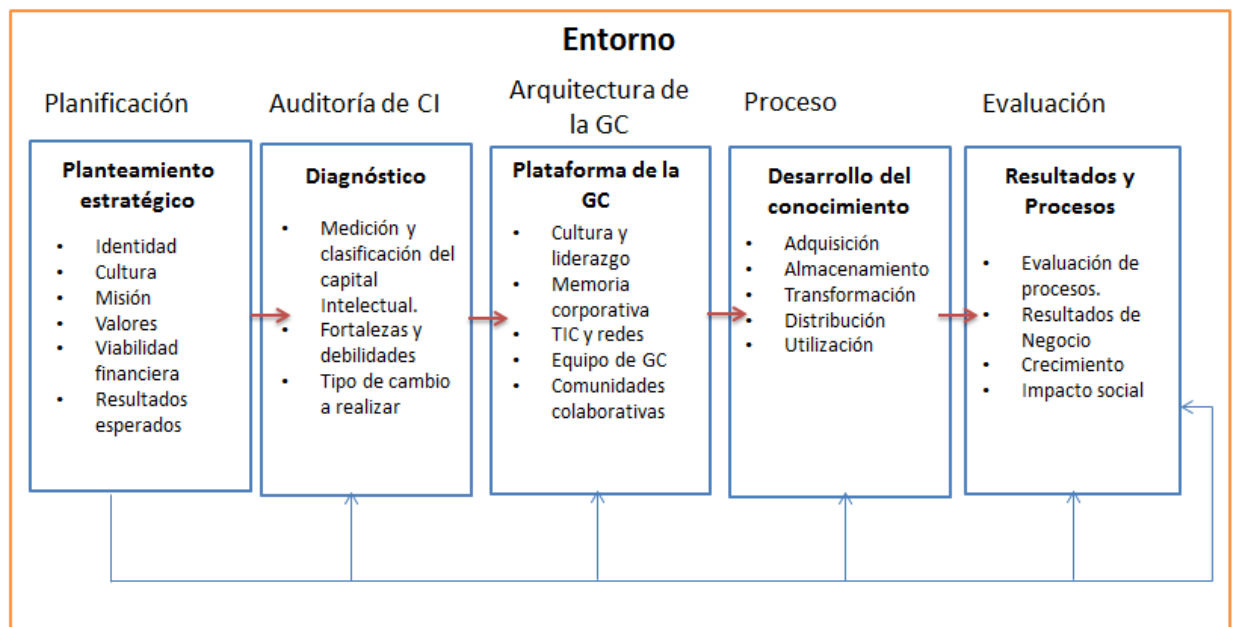
1. Almacenamiento, acceso y transferencia: se centran en el desarrollo de metodologías, estrategias y técnicas para almacenar el «conocimiento» disponible en la organización en depósitos de fácil acceso para propiciar su posterior transferencia entre los miembros de la organización.
2. Sociocultural: centrados en el desarrollo de una cultura organizativa adecuada para el desarrollo de procesos de gestión del conocimiento.
3. Tecnológicos: modelos en los que destaca el desarrollo y la utilización de sistemas (por y herramientas tecnológicas para la gestión del conocimiento).

En la revisión de la literatura se pudo identificar que la mayoría de los modelos son de carácter teórico y se centran en la definición de aspectos a tener en cuenta para la gestión de conocimiento o en la definición de categorías y niveles para clasificar y gestionar el conocimiento, sin embargo estos modelos dan pocos detalles sobre el cómo se podrían aplicar en un determinado contexto. Para la presente investigación se va a tomar como base el modelo propuesto por (Nonaka y Takeuchi, 1995) para definir un modelo propio que contribuya a la incorporación de la gestión de conocimiento en la planificación de proyectos software en grupos

de investigación; mediante la aplicación de herramientas y prácticas de referencia identificadas en una revisión sistemática de la literatura.

Otro modelo de gestión de conocimiento es el modelo propuesto por (Riesco,2006), este modelo se diferencia de los otros en el sentido que aborda la gestión de conocimiento desde una dimensión integrada en donde sus principales características son: premisa, la finalidad, asunciones básicas, perspectiva integrada y particular, perspectiva social y tecnologica, definicion de principios, estructura funcional y desarrollo sistémico. Este modelo se denomina Modelo Integrado Situacional de Gestión del conocimiento (MIS), el modelo se esquematiza en la figura 3.

Figura 3. Modelo Integrado Situacional de Gestión de conocimiento



Fuente: Tomado de Riesco (2006).

Según (Riesco, 2006) algunas cuestiones clave al inicio de un proyecto de gestión de conocimiento son:

- ¿Quiénes somos?
- ¿Por qué emprendemos un proyecto de gestión de conocimiento?
- ¿Dónde nos gustaría estar?
- ¿Con que infraestructura contamos?
- ¿Cómo integrar la gestión de conocimiento en la estrategia del negocio?

- ¿Viabilidad Financiera del proyecto?
- ¿Qué resultados esperamos?
- ¿Cuál es nuestro Capital Intelectual?
- ¿Cuál es el grado de desarrollo del conocimiento en sus distintas fases?
- ¿Qué relación existen entre estabilidad y dinamicidad del conocimiento en nuestra institución?
- ¿Cuáles son las fortalezas y debilidades?
- ¿Qué criterios se utilizan en la valoración?
- ¿Qué tipos de cambios se necesitan realizar para implantar con éxito el proyecto de Gestión de conocimiento?
- ¿Qué modelo y estrategias son los más adecuados en función del trabajo y tipo de cambio a realizar?
- ¿Qué conocimiento valioso se necesita y de cuál se dispone?
- ¿Dónde almacenar el conocimiento y cómo organizarlo?
- ¿Qué actividades existen para actualizar el conocimiento? ¿Quién y cada cuánto se hace?

1.3 Planificación de proyectos software

La disciplina de la computación encargada de ofrecer métodos y técnicas para el desarrollo y mantenimiento del software de calidad es la Ingeniería del Software (ACM, AIS, y IEEE-CS., 2005; Pressman, 2010) Así mismo, la IEEE (Institute of Electrical and Electronics Engineers) Computer Society define la ingeniería de software como: *“La aplicación de un enfoque sistemático, disciplinado, hacia el desarrollo, operación y mantenimiento del software; es decir la aplicación de ingeniería del software.”* (IEEE Std 610.12, 1990, p.67)

Ahora bien, teniendo clara la definición de Ingeniería del software, para entender el proceso de desarrollo software es importante conocer ¿qué se entiende por el concepto de “Software”?, ¿Cuáles son las características diferenciadoras del software de otro tipo de productos que se pueden construir, como es el hardware? según la IEEE por el termino software se entiende: “Los programas de ordenador, procedimientos y documentación posiblemente asociados y los datos relativos a la operación de un sistema informático” (IEEE Std 610.12, 1990, p. 66). En contraste con el hardware que es definido por la IEEE como “Equipos físicos utilizados para procesar, almacenar, o transmitir los programas de ordenador o de datos” (IEEE Std 610.12, 1990 p.36).

En este sentido, según (Pressman, 2010) las características del software son:

- *El software es un elemento del sistema lógico, en lugar de físico.*
- *El software se desarrolla o modifica con intelecto, no se fabrica en un sentido clásico*, esto significa que los proyectos de software no se pueden gestionar como si fueran proyectos de fabricación.
- *El software no se estropea, o se desgasta*: Cuando un componente de hardware se estropea se sustituye por una pieza de repuesto, pero para el software no hay piezas de repuesto. Cada fallo en el software indica un error en el diseño o en el proceso mediante el que se tradujo el diseño a código máquina ejecutable. Por tanto, el mantenimiento del software tiene una complejidad considerablemente mayor que la del mantenimiento del hardware.
- Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a la medida.

Del mismo modo Kruchten, Ahlqvist, y Stefan (2001) menciona algunos aspectos que diferencian la ingeniería del software de otras ingenierías como la mecánica y la eléctrica, este autor sugiere que las diferencias se deben a la naturaleza blanda del software y las características que lo diferencian son:

- Ausencia de teorías fundamentales o al menos aplicada a la práctica, lo cual dificulta razonar del software, sin haberlo construido.
- Entorno altamente cambiante, pero difícil de predecir el impacto del cambio.
- La rápida evolución de la tecnología no permite una evaluación adecuada, y lo hace difícil de mantener y evolucionar los sistemas heredados.
- Costos de fabricación muy bajos.

Respecto a las áreas de conocimiento que comprende la ingeniería del software, se destaca en la literatura un documento denominado SWEBOK (Software Engineering Body of Knowledge) creado por la Software Engineering Coordinating Committee, promovido por la IEEE Computer Society cuyos objetivos son: caracterizar los contenidos de la Ingeniería del Software, proveer acceso a través de las temáticas al conjunto de conocimientos de la Ingeniería del Software, promover una visión consistente de la Ingeniería del Software en todo el mundo, clarificar la posición de la Ingeniería del Software respecto a otras disciplinas, como las ciencias de la computación o las Matemáticas, proveer una base para su desarrollo curricular y la creación de materiales de certificación. Según el SWEBOK (2004) en su tercera versión, las áreas de conocimiento que comprende la ingeniería del software son:

1. Requisitos de Software.
2. Diseño de Software
3. Construcción de Software
4. Pruebas de Software
5. Mantenimiento de Software
6. Gestión de la configuración
7. Gestión de la Ingeniería de Software
8. Proceso de Ingeniería de Software
9. Herramientas y métodos de la Ingeniería de Software
10. Calidad del Software

Las cinco primeras áreas de conocimiento hacen referencia a etapas del ciclo de vida del software (IEEE Std 610.12, 1990; SWEBOK, 2004), definido en la norma ISO 12207 como: “un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando desde la definición hasta la finalización de su uso” y las cinco últimas están relacionadas con una perspectiva de ingeniería.(SWEBOK, 2004; Vizcaíno, Soto, García, Ruiz, & Piattini, 2006).

Por otra parte, el proceso de desarrollo software es definido por el IEEE estándar 610.12 (1990) como un proceso que consiste en traducir las necesidades de los usuarios en los requisitos de software, la transformación de los requisitos de software en el diseño, la codificación, pruebas y mantenimiento del software. A su vez estas actividades pueden superponerse o realizarse iterativamente, dependiendo del modelo de ciclo de vida del software seleccionado dependiendo del alcance del proyecto, de la situación problema a resolver y de la necesidad de realimentación e iteración de las etapas de desarrollo (Palma, Paniagua, Mart, & Mar, 2000), decisión que en ocasiones es un factor determinante del éxito o fracaso del proyecto software.

Así mismo, Pressman (2002) destaca tres fases en el desarrollo software, estas son: Fase de definición (equivalente a la etapa de requerimientos), fase de desarrollo (diseño y construcción) y fase de Mantenimiento (Pruebas y mantenimiento), la Tabla 4 presenta los principales resultados que se buscan obtener en cada una de las fases.

Tabla 4. Fases genéricas de desarrollo de software

Fase de definición <i>Se centra en el qué</i>	Fase de desarrollo <i>Se centra en el cómo</i>	Fase de mantenimiento <i>Se centra en el cambio</i>
- Qué información deberá procesarse - Función y rendimiento que se desea - Qué comportamiento del sistema - Interfaces que deberán ser establecidas - Restricciones de diseño existentes - Criterios de validación en el sistema	- Cómo se diseñan las estructuras de datos - Detalles procedimentales - Cómo ha de traducirse en un lenguaje de programación - Desarrollo de interfaces - La función dentro de una arquitectura software	- Corrección de errores - Adaptaciones requeridas a medida que evoluciona el entorno del software - Cambios debidos a mejoras producidas por los requisitos cambiantes del cliente

Fuente: Elaboración propia adaptado de Pressman (2002)

La necesidad de la ingeniería del software y con está la definición de modelos y/o metodologías para desarrollar software, data de la década de 1970, en la denominada “Crisis del software” (Pressman, 2002). Cuando los programas software fueron creciendo en complejidad y la técnica de code & fix (codificar y corregir) quedó obsoleta, surgió la necesidad de generar modelos al proceso de desarrollo software con el fin de lograr realizar los desarrollos de una manera estructurada y organizada. Algunos modelos existentes para el desarrollo software son: Modelo en cascada (Buchanan et al., 1983), Modelo del ciclo de vida en espiral (Boehm, 1988), desarrollo por prototipos (Kahn, 1994), entre otros.

Otro concepto que ha sido de interés desde la década de 1990 en cuanto a la mejora de la calidad y productividad de la ingeniería del software, es la mejora de procesos software, (Software Process Improvement - SPI, por sus siglas en inglés) (Bjornson & Dingsøyr, 2008; Capote et al., 2008; Pino, Piattini, & Oktaba, 2006), según señala Capote et al. (2008,p 138) “*un programa de SPI tiene como propósito el mejoramiento continuo a través de un ciclo de vida iterativo. Dicho programa contribuye al aumento gradual de satisfacción del cliente, a la obtención de un software de excelente calidad y la generación de lecciones aprendidas*”. El campo de SPI tiene un enfoque de gestión en lugar de tratar directamente con las técnicas utilizadas para escribir el código, y se centra principalmente en la gestión

de las empresas de software para mejorar su práctica (Bjornson & Dingsøyr, 2008).

La realización de los desarrollos software, se gestionan mediante proyectos; Kerzner (2001) menciona que un proyecto considera una serie de actividades y tareas, las cuales tienen un comienzo y un final definidos, límites consolidados y consumen recursos. Por su parte, el PMI (Project Management Institute) quien publica el PMBOK (Project Management Body Knowledge) define un proyecto como un esfuerzo temporal, llevado a cabo para lograr un producto, servicio o resultado único. En resumen un proyecto es un esfuerzo temporal que emplea unos recursos para alcanzar un fin.

En cuanto a la gestión en ingeniería del software, esta puede definirse como la aplicación de actividades de gestión en: planificación, coordinación, medición, monitoreo, control e informes que asegure un desarrollo y mantenimiento del software sistemático, disciplinado y cuantificado (IEEE Std 610.12, 1990). Las actividades de gestión tienen lugar en tres niveles, (1) organizacional y gestión de la infraestructura, (2) gestión de proyectos, (3) medición y control del programa planificado (SWEBOK, 2004). En este sentido, la gestión organizacional es de gran importancia ya que plantea políticas a seguir, por ejemplo: políticas organizativas y estándares que proporcionan el marco en el que se desenvuelve la ingeniería del software. Otro aspecto importante es el personal, teniendo en cuenta que en ingeniería del software hay un esfuerzo humano aplicando diversos conocimientos, se debe contar con personal altamente preparado y motivado para la realización del proyecto (Lindner & Wald, 2010). Además para una gestión eficaz se debe centrar en las cuatro P's: Personal, producto, proceso y proyecto (Pressman, 2002; Sommerville, 2005). Sin embargo existen aspectos específicos de los productos de software y los procesos del ciclo de vida del software que complican la gestión eficaz, algunos de los cuales según (Hazzan & Hadar, 2008) son:

- La ingeniería de software necesariamente incorpora aspectos de la creatividad y la disciplina, pero el mantenimiento de un equilibrio adecuado entre los dos es difícil.
- El grado de novedad y complejidad del software es a menudo extremadamente alto.
- En la percepción de los clientes a menudo existe una falta de aprecio por la complejidad inherente a la ingeniería del software.
- Hay una rápida tasa de cambio en la tecnología subyacente.

Por otra parte, la gestión de proyectos se encuentra organizada principalmente en cinco grupos de procesos: iniciación (I), planeación (P), ejecución (E), monitoreo – control (M) y cierre (C) (PMBOK, 2008). A continuación se describe brevemente cada uno de ellos en el contexto de proyectos software.

Iniciación: durante esta etapa se establece la visión del proyecto (el qué), la misión por cumplir, los objetivos propuestos, la descripción de los argumentos que permiten elaborar su justificación, así como la identificación y descripción de las restricciones y supuestos (Morales, Alexander, Luzardo, & Lourdes, 2007).

Planeación: La selección de un modelo adecuado de ciclo de vida, la adaptación y el despliegue de ciclos de vida del software, se emprenden a la luz del alcance particular y de los requisitos del proyecto. También se seleccionan métodos y herramientas pertinentes (Dorfman & Thayer, 2002; Pfleeger, 2001; Pressman, 2002; Sommerville, 2005). Se especifica el producto o productos en los entregables, se evalúan oportunidades de utilizar componentes de software anteriores, se estima recursos requeridos para acometer el esfuerzo de desarrollo del software, especificando su descripción, disponibilidad, la fecha cronológica en la que es requerido, y el tiempo durante el cual será aplicado (Morales et al., 2007); cabe destacar que la estimación del costo es una actividad en la cual se requiere la consideración de muchas variables: humanas, técnicas, de entorno, políticas; variables que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo (Morales et al., 2007; Sommerville, 2005).

Ejecución: En esta etapa se emprenden las actividades del proyecto según el cronograma planificado. En este proceso, se utilizan recursos y se producen entregables (Pfleeger, 2001; Sommerville, 2005). Se promulga el proceso de medición junto con el proyecto del software, asegurándose de que se recogen datos relevantes y útiles (Fenton & Neil, 1999; Pfleeger, 2001; Sommerville, 2005). Se evalúa continuamente a intervalos predeterminados la adhesión a los diferentes planes. Se analizan los resultados y las condiciones de acabado de cada tarea y se evalúan los entregables en términos de las características que ellos requieren (Dorfman & Thayer, 2002; Sommerville, 2005).

Monitoreo y control: Se evalúan los distintos métodos, herramientas y técnicas empleadas para ver su eficacia y adecuación, y se valora sistemática y periódicamente el propio proceso para conocer su relevancia, utilidad y eficacia en el contexto del proyecto. Cuando se juzga necesario, se llevan a cabo y se gestionan los cambios (Dorfman & Thayer, 2002; Pressman, 2002).

Cierre: Se han completado las tareas tal y como se especificaron en los planes, y se confirman los criterios para lograr un acabado satisfactorio. Todos los

productos planificados han sido entregados con características aceptables (Sommerville, 2005).

Adicionalmente, los aspectos claves que debe contemplar un proyecto, especialmente en los procesos de planificación, monitoreo y control, han sido definidos en las 9 áreas de conocimiento planteadas en el PMBOK. La tabla 5 presenta una breve descripción de cada una de ellas.

Tabla 5. Áreas del PMBOK

Áreas de conocimiento	Definición	Grupos de procesos
Gestión de la Integración del Proyecto	Incluye los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades de la dirección de proyectos dentro de los grupos de procesos de dirección de proyectos.	• Desarrollar el Acta de constitución del Proyecto (I) • Desarrollar el Plan para la dirección del Proyecto (P) • Dirigir y Gestionar la Ejecución del Proyecto (E) • Monitorear y Controlar el Trabajo del Proyecto (M) • Realizar Control Integrado de Cambios (M) • Cerrar el Proyecto o la Fase (C)
Gestión del Alcance del Proyecto	Los procesos necesarios para garantizar que incluya todo el trabajo para completar el proyecto exitosamente.	• Recopilar los Requisitos (P) • Definir el Alcance (P) • Crear la Estructura de Desglose del Trabajo (EDT) (P) • Verificar el Alcance (M) • Controlar el Alcance (M)
Gestión del Tiempo del Proyecto	Gestiona el proyecto para finalizar el proyecto a tiempo	• Definir las Actividades (P) • Secuenciar las Actividades (P) • Estimar los Recursos para las actividades (P) • Estimar la Duración de las Actividades (P) • Desarrollar el Cronograma (P) • Controlar el Cronograma (M)

Gestión de los Costos del Proyecto	Procesos involucrados en estimar, presupuestar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado.	• Estimar los Costos (P) • Determinar el Presupuesto (P) • Controlar los Costos (M)
Gestión de la Calidad del Proyecto	Gestiona el proyecto para que cumpla con los objetivos con los que fue planteado el proyecto.	• Planificar la Calidad (P) • Realizar el Aseguramiento de Calidad (E) • Realizar el Control de Calidad (M)
Gestión de los Recursos Humanos del Proyecto	Incluye los procesos que organizan, gestionan y conducen el equipo del proyecto.	• Desarrollar el Plan de Recursos humanos (P) • Adquirir el Equipo del Proyecto (E) • Desarrollar el Equipo del proyecto (E) • Dirigir el Equipo del Proyecto (E)
Gestión de las Comunicaciones del Proyecto	Incluye los procesos requeridos para garantizar que la generación, la recopilación, la distribución, el almacenamiento, la recuperación y la disposición final de la información del proyecto sean adecuados y oportunos.	• Identificar a los Interesados (I) • Planificar las Comunicaciones (P) • Distribuir la Información (E) • Gestionar las Expectativas de los Interesados (E) • Informar el Desempeño (M)
Gestión de los Riesgos del Proyecto	Incluye los procesos relacionados con llevar a cabo la planificación de la gestión, la identificación, el análisis, la planificación de	• Planificar la Gestión de Riesgos (P) • Identificar los Riesgos (P) • Realizar Análisis Cualitativo de riesgos (P) • Realizar Análisis Cuantitativo de riesgos (P) • Planificar la Respuesta a

	respuesta a los riesgos, así como su monitoreo y control en un proyecto.	los riesgos (P) • Dar seguimiento y Controlar los riesgos (M)
Gestión de las Adquisiciones del Proyecto	Incluye los procesos de compra o adquisición de los productos, servicios o resultados que es necesario obtener fuera del equipo del proyecto.	• Planificar las Adquisiciones (P) • Efectuar las Adquisiciones (E) • Administrar las Adquisiciones (M) • Cerrar las Adquisiciones (C)

Fuente: PMBOK versión 4

Según (Oktaba, Piattini, Pino, Orozco, & Alquicira, 2009) un 80% de los proyectos iniciados acaban con algún tipo de problema, el 31.1% son cancelados antes del tiempo y el 52.7% cuestan el 189% más que sus estimaciones originales, dentro de los problemas los más significativos son: dificultades para terminarlos en el tiempo exigido, costos superiores, mala gestión de los proyectos y pocos seguimientos, lo cual hace necesario de una adecuada administración de los proyectos e implementación de buenas prácticas de dirección de proyectos. En este sentido una de las etapas primordiales de la gestión de proyectos es la elaboración del plan del proyecto, el cual contiene los siguientes aspectos: Objetivos, alcance, recursos, actividades, adquisiciones, formación, gestión del riesgo, estimaciones de costo y esfuerzo y plan para realizar el seguimiento y control al proyecto (Oktaba et al., 2009).

Por otra parte en la literatura y en la comunidad de desarrollo software se han realizado diversos esfuerzos con el fin de establecer modelos y estándares enfocados a la mejora del proceso del software (software process improvement-SPI), muchos de estos esfuerzos se han enfocado en pequeñas y medianas empresas; por ejemplo en la Secretaria de Economía de México creo el programa Prosoft para definir el modelo de referencia de procesos para la industria del software Moprosoft. Así mismo en Brasil se crea el proyecto “MPS.BR” para definir e implementar modelos para la mejora de procesos del software orientado a PYMES, el proyecto propone 3 modelos: un modelo de referencia de procesos (MR-MPS), un método de evaluación de procesos (MA-MPS) y un modelo de negocio (MN-MPS).

Otro modelo de mejora de procesos del software es el modelo COMPETISOFT basado en el modelo Moprosoft e integra prácticas y elementos de varios modelos y estándares conocidos, tales como (ISO 9000, 2000; SW-CMM, 1995; CMMI, 2006, ISO/IEC 15504-2, 2003; ISO/IEC 12207, 2008 Y PMBOOK, 1995). Este modelo de procesos aborda 3 categorías: Alta dirección, gerencia y operación. La categoría de alta dirección establece la razón de ser de la organización, lo que desea alcanzar y las estrategias para lograrlo. La categoría de Gerencia aborda los aspectos de gestión de procesos, cartera de proyectos, gestión de recursos y gestión de conocimiento, finalmente en la categoría de operación se encuentra lo relacionado con la administración del proyecto, desarrollo del software y el mantenimiento del software.(Oktaba et al., 2009)

1.4 Gestión de conocimiento en el desarrollo de software

Las organizaciones en el área de la ingeniería de software requiere un uso intensivo de conocimientos en el negocio, actividades y técnicas en las cuales participan muchas personas que trabajan en las diferentes fases y actividades del proceso de desarrollo; estos conocimientos no son estáticos en la organización, sino que se encuentra en constante crecimiento a medida que se desarrollan diversos proyectos, razón por la cual, a menudo, las organizaciones tienen problemas para identificar el contenido, la ubicación y uso del conocimiento. La invención de software requiere un esfuerzo amplio en el levantamiento de requerimientos, análisis, validación y diseño; estos muchas veces se ven afectados por no tener en cuenta los errores cometidos en el pasado, haciendo más costosa la implementación de estos proyectos (Corbin et al., 2007). Por otra parte, hay una alta rotación del personal en organizaciones que desarrollan software, llevándose con ellos el conocimiento adquirido durante el ejercicio de su trabajo, lo cual muestra la necesidad de un mejor aprovechamiento de los conocimientos en la organización (Dingsøyr & Conradi, 2002; Ioana Rus & Lindvall, 2002).

Algunos autores como (Aurum et al., 2008; Bjørnson, 2007; Corbin et al., 2007; Desouza & Awazu, 2006; Dingsøyr & Conradi, 2002; Iversen et al., 2004; Ioana Rus & Lindvall, 2002) destacan la importancia de incorporar la gestión de conocimiento en la ingeniería del software, principalmente por su la variedad y gran proporción de conocimientos. Corbin et al. (2007, p.3) destaca “la gestión de conocimiento permite en los proyectos de desarrollo de software, descubrir nuevo conocimiento, capturarlo, almacenarlo, recuperarlo, compartirlo y entenderlo, y eso sucede en el desarrollo de proyectos de ingeniería de software”.

Incorporar la gestión de conocimiento en el proceso de desarrollo software contribuye a mejorar este proceso tal como menciona (Aurum et al., 2008; Ward & Aurum, 2004): la gestión de conocimiento contribuye a reducir los costos, cumplir con los tiempos planificados y mejorar la calidad del proyecto. A su vez, las organizaciones desarrolladoras de software pueden ver la gestión del conocimiento como el aprovechamiento del conocimiento de los individuos en beneficio de la organización (Pinto, Sierra, & Gomez, 2012) . La gestión de conocimiento une las actividades diarias de producción, iniciativas de mejora y objetivos planteados, apoyando así el establecimiento de una organización de aprendizaje (Ioana Rus & Lindvall, 2002); además se pueden prevenir riesgos y proponer estrategias de mitigación, (Bjornson & Dingsøyr, 2008; Dingsøyr, Bjornson, & Shull, 2009; Ioana Rus & Lindvall, 2002)

De otra parte el desarrollo de software implica el conocimiento de varios tipos de técnicas, dominio de la aplicación, tecnologías, productos y conocimiento del proyecto, también es probable que algunos conocimientos se encuentren en la organización, implementado en bases de datos o en papel, sin saber quién lo tiene y como potencializar este conocimiento (Dingsøyr & Conradi, 2002).

Para Mathiassen y Pourkomeylian (2003) el conocimiento tácito juega un papel muy importante en la mejora del proceso software porque facilita una apreciación profunda de las prácticas para evaluar las capacidades actuales, para diseñar e implementar nuevos procesos y proyectos (Mathiassen y Pourkomeylian, 2003). Estos autores plantean la idea de crear y compartir el conocimiento a través de los diferentes proyectos e individuos. Falbo, Borges, y Rosa (2004) manifiesta que la gestión del conocimiento contribuye fundamentalmente a la mejora continua del proceso y en consecuencia de los productos resultantes el contexto específico del desarrollo software.

Del mismo modo, Alhawari, et al. (2012) sostienen que la Gestión de conocimiento contribuye a reducir los riesgos en proyectos de tecnologías de información. Así mismo, Borjesson y Mathiassen (2004) afirman que para cambiar, de manera exitosa las prácticas de software, se requiere aprendizaje y al gestionar el conocimiento permite corregir errores y modificar los procesos en base a la experiencia práctica.

Por otra parte con el uso y desarrollo de nuevas tecnologías de información y comunicación facilitan el acceso a las fuentes de información de la organización (García, 2013), con diferentes herramientas tecnológicas se logra contribuir con procesos de gestión de conocimiento.

Diferentes investigaciones han realizado diversos trabajos con el fin de gestionar conocimientos por ejemplo: compartir conocimiento en equipos de proyectos software mediante memoria transitiva (Chen, Li, Clark, & Dietrich, 2013), otro trabajo patrones como objetos para gestionar conocimiento en organizaciones de desarrollo software, la creación de objetos de conocimiento para facilitar el uso de conocimientos por parte de los ingenieros (Martín, Guzmán, Urbano, & Llorens, 2012), el siguiente trabajo es una herramienta para gestionar el conocimiento en el desarrollo de nuevos productos en la industria de combustibles en esta investigación generó un aplicativo para ser usado por los ingenieros que desarrollan productos en este sector petroquímico y lograr gestionar conocimientos en la realización de los productos (Zhen, Wang, & Li, 2013), un estudio sobre la forma de adquirir y diseminar el conocimiento tácito en la ingeniería de Software (Heredia, Garcia, Amescua, & Sanchez, 2013), el desarrollo de una herramienta para gestionar decisiones en proyectos de software distribuidos (Garrido, Vizcaino, Andrada, Monasor, & Piattini, 2012).

1.5 Revisión Sistemática de literatura

Una revisión sistemática de la literatura es un medio de identificación, evaluación e interpretación de toda la investigación disponible relevante para una pregunta de investigación particular o área temática o fenómeno de interés. Los estudios individuales que contribuyen a una revisión sistemática se llaman estudios primarios, una revisión sistemática es una forma de estudio de secundaria (Kitchenham & Charters, 2007),

Según Kitchenham y Charters (2007) hay muchas razones para llevar a cabo una revisión sistemática de la literatura. Las razones más comunes son: (a) Para resumir la evidencia existente sobre el tratamiento o la tecnología por ejemplo, para resumir la evidencia empírica de los beneficios y limitaciones de un método ágil específico (b) Identificar las lagunas en la investigación actual para sugerir áreas para una mayor investigación. (c) Proporcionar un marco o fundamentación con el fin de posicionar adecuadamente las nuevas actividades de investigación.

Para realizar la revisión sistemática de literatura se agrupa en tres principales fases: Planificación de la revisión, ejecución de la revisión y reporte de resultados.

Las actividades previas a la revisión más importantes son la definición de las preguntas de investigación que la revisión sistemática se abordarán y la producción de un protocolo de revisión (es decir, plan) la definición de los procedimientos de revisión básicos.

Un protocolo de revisión especifica los métodos que se utilizarán para llevar a cabo una revisión sistemática específica. Un protocolo predefinido es necesario para reducir la posibilidad de sesgo del investigador. Por ejemplo, y sin un protocolo, es posible que la selección de los estudios individuales o el análisis puede ser impulsado por las expectativas investigador. En medicina, los protocolos de revisión generalmente se someten a revisión por pares (Kitchenham & Charters, 2007).

Planificando la revisión: Se revisa documentación sobre el área que se desea conocer, se identifican personas que conozcan sobre el área de investigación, siendo esto una práctica bastante útil facilitando conformar un panel de expertos, este panel podría ayudar y aconsejar sobre la inclusión o exclusión de estudios. El estado inicial del de la revisión sistemática es un proceso iterativo de definición, clasificación y refinamiento de información (Tranfield, Denyer, & Smart, 2003). Posteriormente realizar las preguntas de investigación que se desean resolver y con esto formalizar el protocolo de revisión, visto como un plan que ayudara en el paso a paso, este contiene las preguntas, los criterios de inclusión y exclusión, bases de datos que se van a consultar la base de datos (Kitchenham & Charters, 2007; Tranfield et al., 2003).

Conduciendo la revisión: Una búsqueda sistemática comienza con la identificación de palabras clave y términos de búsqueda, que se construyen a partir del estudio de alcance, la literatura y las discusiones o consultas con personas expertas, después, el investigador debe decidir sobre las palabras claves que son más adecuados para la búsqueda sobre la temática a investigar (Tranfield et al., 2003). El resultado será una lista completa de artículos y ponencias en los que se basará la revisión. Posteriormente extraer los datos de los documentos encontrados, llevar un archivo con todos los documentos con la principal información como título, autores, palabras claves, año de publicación, resumen. La extracción de información del contenido de los documentos estará relacionada con el aporte a las preguntas de investigación planteadas.

Reporte de resultados: Se sintetiza la información extraída y se pueden generar dos reportes en un primer reporte un análisis descriptivo, por ejemplo cuales son los autores, contribuciones de diferentes países, años de publicación, disciplinas en las se enfoca el tema, etc. El otro reporte es el análisis temático en el que se analizan las contribuciones de los trabajos, se justifican y se fundamentan las conclusiones, además las revisiones sistemáticas proveen al investigador una evidencia sobre la investigación realizada y decisiones tomadas (Tranfield et al., 2003).

1.6 Metodología de Investigación-acción

La metodología de investigación-acción I-A es un método particular de investigación que combina la teoría y la práctica (investigadores y profesionales) a través del cambio y reflejo en una situación problemática inmediata dentro de un marco ético mutuamente aceptable. La investigación-acción es un proceso iterativo en el cual investigadores y profesionales actúan juntos en un determinado ciclo de actividades, incluyendo el diagnóstico de problemas, intervención de la acción y el aprendizaje reflexivo (Avison, Lau, Myers, & Nielsen, 1999). El objetivo de la I-A es resolver problemas prácticos actuales mientras se expande el conocimiento científico. A diferencia de otros métodos de investigación, donde el investigador trata de estudiar los fenómenos de organización pero no el cambio de ellos, en investigación – acción se refiere a crear un cambio en la organización y al mismo tiempo estudiar el proceso. Es fuertemente orientado hacia la colaboración y el cambio que implica tanto a los investigadores y los sujetos (Baskerville & Myers, 2004).

Según Baskerville y Myers (2004), la Investigación- Acción se presentó por primera vez en la obra de Lewin (1947), quien desarrolló el método en el Centro de Investigación para la dinámica de Grupo (Universidad de Michigan) para el estudio de la psicología social en el marco de la teoría del campo, cuando Los científicos trataron de comprender las complejas causas de la gran variante "Enfermedades sociales" y la idea de la acción social se levantó. Los científicos intervinieron en cada uno experimental caso de cambiar algún aspecto de la del paciente ser o alrededores. Desde científico y terapeuta fueron encerrados juntos en esta búsqueda, el los científicos participantes en sus propias investigaciones.

La investigación-acción supone que los sistemas sociales complejos no pueden ser reducidos para el estudio significativo (Baskerville, 1999). Las organizaciones humanas, como un contexto en el que interactúan con tecnologías de la información, solo se pueden entender como entidades enteras y el análisis de estos no debe limitarse a un análisis por variables o componentes, el cual no dará un conocimiento útil sobre toda la organización (Baskerville, 1999). Según Checkland y Scholes (1990) la realidad social, la cual se registra como un "hecho" respecto del mundo social está continuamente siendo construida y reconstruida en el dialogo y el discurso entre seres humanos, y la acción que ellos toman, el investigar la realidad social se convierte entonces en un descubrimiento organizado de cómo los agentes humanos dotan de sentido sus mundos

percibidos, y cómo esas percepciones cambian en el tiempo y difieren de una persona o grupo a otra.

El trabajo de Lewin buscaba una teoría general de cómo tales el cambio social podría facilitarse, su original modelo de investigación-acción incluyen la iteración de seis etapas por etapas: (1) análisis, (2) investigación de los hechos, (3) conceptualización, (4) planificación, (5) la aplicación de acción, (6) evaluación.

Las principales características de la Investigación-acción son:

- Complejidad de los sistemas sociales
- Centra en problemas no estructurados, es decir aquellos en donde los objetivos están claramente definidos y por lo general hay una única realidad percibida.
- Combina la teoría y la practica
- Los investigadores toman sentido de sus experiencias
- Se propone una acción para mejorar una situación problema y hay una reflexión de la intervención
- No puede ser totalmente planeada y dirigida por un camino particular

Según Sampieri et al. (2010), hay dos tipos de investigación acción, práctica y participativa, la primera se caracteriza por estudiar prácticas locales del grupo o comunidad, involucrar indagación individua o en grupo, implementar un plan de acción para resolver el problema o introducir una mejora y el liderazgo es ejercido por el investigador y uno o varios miembros de la comunidad; para el caso de la metodología de investigación acción participativa, esta estudia temas sociales que constriñen las vidas de personas o una comunidad, resalta la colaboración equitativa de un grupo o comunidad y se enfoca en el cambio para mejorar el nivel de vida y desarrollo humano de los individuos.

1.6.1 El proceso de investigación-acción

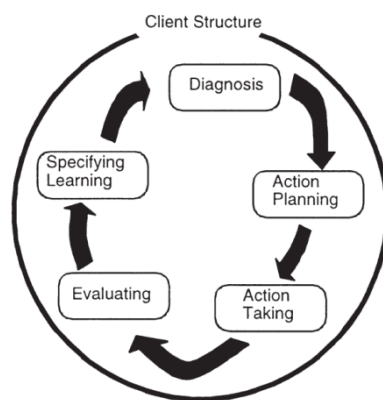
Inicialmente Blum (1955) explica la esencia de la investigación-acción como un simple proceso de dos etapas.

- En primer lugar, la etapa de diagnóstico consiste en un análisis conjunto de la situación social por el investigador y los sujetos de la investigación. Las hipótesis se formulan sobre la naturaleza del ámbito de la investigación.

- En segundo lugar, la fase terapéutica consiste en experimentos colaborativos cambio. En esta etapa son los cambios introducidos y estudiados.

Por otra parte, en (Baskerville, et al ,1999) se menciona que la descripción prevaleciente del ciclo de investigación-acción es un proceso cíclico que incluye cinco etapas. El primer método requiere el establecimiento de una infraestructura de cliente del sistema o entorno de la investigación. Luego, cinco fases identificables son iteradas: (1) diagnóstico, (2) la planificación de acciones, (3) la adopción de medidas, (4) evaluación y (5) especificando el aprendizaje. La siguiente figura presenta el esquema de del ciclo de la investigación-acción.

Figura 4. El ciclo de la investigación-acción



Fuente: Tomado de (Baskerville, et al ,1999)

Acerca del proceso de investigación –acción, Checkland (1998) plantea que Inicialmente el investigador encontrará una situación del mundo real que parece relevante para temas de investigación que él o ella consideran significativo. Entonces es importante acordar cuidadosamente los roles del investigador(es) y las personas en la situación del problema. Luego, es esencial declarar tanto el marco de ideas como la metodología. Ahora si puede empezar el trabajo más grande, Esto consistirá en que el investigador se involucre en la acción de la situación. Mientras se hace esto, el investigador intenta interpretar la experiencia acumulada, haciendo uso del F declarado y M. Esto puede de forma adecuada causar un re-pensar de las fases más tempranas - y de nuevo es la declaración explícita del marco intelectual la que hace esto posible. Finalmente, el investigador acuerda una salida de la situación (qué puede ser un acto arbitrario puesto que la propia situación continuará evolucionando a través de tiempo) y reflexiona sobre la experiencia para extraer las diversas lecciones aprendidas.

2. OBJETIVOS

2.1 Objetivo General

Proponer un modelo para la gestión de conocimiento en grupos de investigación que desarrollan software, que contribuya a la planificación de sus proyectos, tomando como caso de estudio el Grupo de Investigación en Sistemas y Tecnologías de Información (STI), adscrito a la Escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander.

2.2 Objetivos Específicos

1. Identificar, mediante una revisión de la literatura, las principales contribuciones científicas y herramientas aplicables para la gestión de conocimiento en la gestión de proyectos software.
2. Caracterizar el proceso de desarrollo software en grupos de investigación pertenecientes a la Escuela de Ingeniería de Sistemas de la UIS y grupos aliados al grupo STI, que sirva de base para la identificación de los conocimientos y procesos relevantes a gestionar.
3. Identificar conocimientos y procesos relevantes en la gestión de proyectos software en grupos de investigación, que contribuyan a mejorar su planificación, a partir del caso de estudio del Grupo de Investigación STI.
4. Definir el modelo que incorpore la gestión de conocimiento, teniendo en cuenta las herramientas, conocimientos y procesos relevantes identificados, el cual contribuya a la planificación de proyectos software.

3. METODOLOGÍA

La presente investigación se llevó a cabo mediante un enfoque cualitativo, específicamente se siguió la metodología de investigación-acción, la cual combina la teoría y la práctica, donde investigadores y profesionales actúan juntos en un determinado ciclo de actividades, incluyendo el diagnóstico de problemas, intervención de la acción y el aprendizaje reflexivo (Avison et al., 1999) con el propósito de resolver problemas cotidianos e inmediatos y mejorar prácticas concretas (Sampieri et al., 2010).

En esta investigación, el grupo de investigación Sistemas y Tecnologías de la información (STI) adscrito a la escuela de Ingeniería de Sistemas constituye el entorno de investigación, en donde se presentan los espacios para realizar diversas actividades con los miembros del grupo, entre los cuales se encuentran entrevistas, talleres, encuesta y revisión de documentos de proyectos anteriores desarrollados por el grupo de investigación.

En cuanto al proceso de investigación-acción, Baskerville (1999) menciona que la descripción prevaleciente del ciclo de investigación-acción es un proceso cíclico que incluye cinco etapas: (1) diagnóstico, (2) la planificación de acciones, (3) tomando acción, (4) evaluación y (5) especificando el aprendizaje. Cabe aclarar que el alcance de este proyecto llega hasta proponer un modelo para la gestión de conocimiento en grupos de investigación que desarrollan software, que contribuya a la planificación de sus proyectos y no contempla la aplicación del modelo en el grupo de investigación para evaluar su impacto en el proceso de planificación de proyectos software; sin embargo el modelo fue presentado a algunos integrantes del grupo de investigación STI para evaluar su aplicabilidad en el grupo y se presentó en una ponencia internacional en el evento LACCEI 2013 realizada en Cancún, México durante el mes de agosto de 2013.

A continuación se presentan las actividades que se desarrollaran en este proyecto para cada una de las etapas de la Investigación-acción.

Etapas	Actividades Desarrolladas	Objetivo al que contribuye
Diagnóstico o se introduce en la situación	•Conocimiento inicial de las relaciones sociales de la organización y estructura organizativa, mediante observación en el entorno de	1. Identificar, mediante una revisión de la literatura, las principales contribuciones científicas y herramientas aplicables

problema del mundo real	investigación (Grupo STI) • Se identificaron problemáticas relacionadas con la planificación de proyectos software. • Revisión de literatura. • Para obtener información de grupos de investigación se elaboraron instrumentos de recolección de información como encuestas.	para la gestión de conocimiento en la gestión de proyectos software.
Planificand o la acción	• Diseño de preguntas para entrevistas, para el entorno de investigación. • Análisis de documentos del grupo de investigación • Ejecución y obtención de información de las encuestas, mediante el análisis y tabulación de la información recogida. • Se caracterizó el proceso de desarrollo de software.	2. Caracterizar el proceso de desarrollo software en grupos de investigación pertenecientes a la Escuela de Ingeniería de Sistemas de la UIS y grupos aliados al grupo STI, que sirva de base para la identificación de los conocimientos y procesos relevantes a gestionar.
Tomando acción y evaluando	• Se realizan entrevistas a los integrantes del grupo de investigación STI. • Lluvia de ideas y talleres con los integrantes del grupo. • Se realizaron observaciones, revisión de documentos, registros y material generado en la investigación. • Al evaluar los resultados se identificaron conocimientos y procesos relevantes llevados a cabo en los desarrollos de proyectos Software y finalmente se sintetiza los resultados en la propuesta de un modelo de referencia en la planificación de proyectos de desarrollo Software en grupos de investigación.	3. Identificar conocimientos y procesos relevantes en la gestión de proyectos software en grupos de investigación, que contribuyan a mejorar su planificación, a partir del caso de estudio del Grupo de Investigación STI. 4. Definir el modelo que incorpore la gestión de conocimiento, teniendo en cuenta las herramientas, conocimientos y procesos relevantes identificados, el cual contribuya a la planificación de proyectos software.
Especifica	Se realiza la documentación de	4. Definir el modelo que

ndo lo aprendido	los hallazgos obtenido en la investigación	incorpore la gestión de conocimiento, teniendo en cuenta las herramientas, conocimientos y procesos relevantes identificados, el cual contribuya a la planificación de proyectos software.
-------------------------	--	--

Fuente: Elaboración propia.

Una de las actividades más importantes durante la investigación fue la revisión sistemática de la literatura realizada para conocer documentos que traten sobre la gestión de conocimiento en proyectos software con el fin de profundizar sobre los siguientes aspectos:

1. Herramientas de gestión de conocimiento en la gestión de proyectos software.
2. Prácticas de gestión de conocimiento en la gestión de proyectos software.
3. Beneficios que genera la gestión de conocimiento en la gestión de proyectos software.
4. Factores de éxito de la Gestión de conocimiento en la gestión de proyectos software.

La revisión sistemática se basó en lineamientos metodológicos presentados en (Kitchenham & Charters, 2007; Tranfield et al., 2003) donde se describe el proceso para realizar este tipo de revisiones, se plantean varios aspectos como: preguntas de investigación, identificación de investigaciones relevantes, proceso de selección de artículos, evaluación y síntesis de resultados. Los cuales se agruparan en tres principales fases: Planificación de la revisión, ejecución de la revisión y reporte de resultados.

En la fase de planificación, se definió el protocolo de la revisión sistemática el cual contiene preguntas sobre los que se desea investigar, las estrategias para la identificación de estudios relevantes y los criterios de inclusión y exclusión en la revisión. Se realizaron lecturas previas que hablaban sobre la temática que se desea investigar y se determinaron palabras claves para realizar la búsqueda de la presente revisión las cuales se muestran en la Tabla 6. Se contó con la asesoría de expertos en el campo en diferentes etapas de la revisión.

Tabla 6. Palabras claves

Gestión conocimiento	Proyectos software
• Knowledge management • Tacit knowledge • Explicit knowledge • Knowledge Identification • Knowledge Acquisition • Knowledge Creation • Knowledge Codification • Knowledge Application • Knowledge sharing • Knowledge Transfer • knowledge Protection • knowledge Assessment	• Software Project management • Software Project planning • Software Project executing • Software Project monitoring • Software Project controlling • Software Project closing • Software Project integration management • Software Project scope management • Software Project time management • Software Project cost management • Software Project Quality Management • Software Project Human Resource Management • Software Project Communications Management • Software Project Risk Management • Software Project Procurement Management • Software Project Stakeholders Management

Fuente. Elaboración propia

Se realizaron posibles combinaciones de términos de gestión del conocimiento y proyectos software para encontrar los documentos para esta investigación. Las bases de datos científicas más relevantes usadas para realizar la búsqueda fueron: Scopus y la ISI web of Knowledge ya que estas dos bases de datos cubren todas las áreas del conocimiento y son de gran relevancia y calidad en la producción científica.

Adicionalmente en la etapa de planificando la acción se realizó el diseño de las preguntas para la encuesta la cual se aplicó a los grupos de investigación de la escuela de ingeniería de sistemas, por otra parte se realizó el análisis de los

proyectos entregados por parte de los estudiantes. En la realización de dicho instrumento se procedió a consultar caracterizaciones en la industria y grupos de investigación en el desarrollo de software, dando como resultado pocos estudios sobre esta temática.

En la siguiente etapa se realizó entrevistas a los integrantes de los grupos de investigación, se aplicó la encuesta vía web y se tabuló las respuestas, se realizaron talleres con los integrantes del grupo de investigación, cuyos integrantes estuvieran inmersos en proyectos de desarrollo Software. Dado lo anterior se pudo establecer las características del proceso llevado a cabo en los grupos de investigación.

Al evaluar los resultados se identificaron conocimientos y procesos relevantes llevados a cabo en los desarrollos de proyectos Software y finalmente se sintetiza los resultados en la propuesta de un modelo de referencia en la planificación de proyectos de desarrollo Software en grupos de investigación.

4. RESULTADOS

4.1 Caracterización del proceso de desarrollo Software en grupos de investigación

Según Colciencias: “Se define grupo de investigación científica o tecnológica como el conjunto de personas que se reúnen para realizar investigación en una temática dada, formulan uno o varios problemas de su interés, trazan un plan estratégico de largo o mediano plazo para trabajar en él y producir unos resultados de conocimiento sobre el tema cuestión. Un grupo existe siempre y cuando demuestre producción de resultados tangibles y verificables, fruto de proyectos y otras actividades de investigación convenientemente expresadas en un plan de acción (proyectos) debidamente formalizado”.

De esta definición se destaca que la producción de un grupo de investigación es importante para verificar su existencia y clasificación, los grupos pueden producir diferentes productos, como se está tratando en la presente investigación acerca del desarrollo software se indicara los requerimientos de existencia, calidad y puntaje asignado por Colciencias en el nuevo modelo de medición propuesto en el 2013, la cual se aprecia en la tabla 7.

Tabla 7. Producto Software como resultado de investigación, requerimientos de existencia, categorías y puntajes relativos

Nombre Producto	Requerimientos de existencia	Categoría	Requerimientos de calidad	Puntajes
Software	Software: Título del producto o proceso (*), Número del registro aprobado por la Dirección Nacional de Derechos de Autor, año de obtención. Identificación del proyecto: Nombre, código, contrato, número del radicado del informe técnico de avance o final donde reporta el desarrollo del	SF_A	Certificación de la entidad financiadora en la cual se especifique que este software es el resultado de una investigación y se haga claridad sobre el nivel de innovación. Realizar la	10

	software.		descripción del Análisis, Diseño, Implementación y Validación	
	Certificación de la entidad financiadora en la cual se especifique que este software es el resultado de una investigación y se haga claridad sobre el nivel de innovación.	SF_B	Software producto de un proyecto de investigación financiado por el SNCyT. (COLCIENCIAS)	6

Fuente: tomado de (Colciencias, 2013)

En la anterior tabla se aprecia la forma y cantidad de puntos que se dan para un producto como lo son los productos Software, en el anterior modelo de medición dado por Colciencias para grupos de investigación era suficiente el registro de autor de un producto software para que el grupo obtuviera los puntos por productividad.

Un grupo de investigación está conformado por las personas que desempeñan alguna tarea relacionada con la actividad del grupo. Los integrantes se clasifican en tres categorías generales: investigadores, estudiantes y auxiliares de investigación (Colciencias, 2008). Dentro de la categoría de investigadores estaría un líder del grupo, quien es el encargado de orientar el grupo hacia el cumplimiento de sus objetivos, otros investigadores, cuya función es dirigir trabajos específicos de una investigación y pueden llegar a desempeñar algunas actividades administrativas, por lo general este rol es desempeñado por profesores o estudiantes de posgrado, y por último auxiliares de investigación que se encargan de tareas muy puntuales dentro de la investigación, por lo general son estudiantes de pregrado y las tareas realizadas en el grupo de investigación están enmarcadas en su trabajo de grado (Gaviria et al., 2007; González, 2009; León et al., 2011).

Adicionalmente una clasificación de los integrantes, recientemente propuesta por Colciencias en el modelo de medición de grupos de investigación se presenta a continuación en la siguiente tabla.

Tabla 8. Tipos, subtipos de integrantes y los requisitos respectivos.

Tipo	Subtipo	Requisitos
INVESTIGADORES	Investigador Sénior	Se deben cumplir las siguientes tres condiciones simultáneamente. - Nivel de Formación: Doctorado finalizado o equivalente en producción (15 productos de nuevo conocimiento con calidad A). - Producción mínima: Diez (10) productos de calidad A en los últimos diez (10) años. - Productos de Formación: Director de tesis de doctorado o cuatro (4) trabajos de maestría finalizadas en los últimos diez (10) años.
	Investigador Asociado	Se deben cumplir las siguientes tres condiciones simultáneamente. - Nivel de Formación: Doctorado o Maestría finalizados o equivalentes en producción (7 productos de nuevo conocimiento). - Producción mínima: Dos (2) productos de calidad A y cuatro (4) productos de nuevo conocimiento en los últimos cinco (5) años. - Productos de Formación: Haber dirigido trabajo de pregrado, de maestría o tesis de doctorado durante los últimos cinco años.
	Investigador Junior	Graduado con formación de doctorado finalizada, integrante de un grupo de investigación, que hace parte de un proyecto de investigación del grupo; o maestría finalizada con autoría de tres productos de nuevo conocimiento por año
INVESTIGADORES EN FORMACIÓN	Estudiante de Doctorado	En formación de doctorado
	Estudiante de Maestría	En formación de maestría
	Jóvenes Investigadores	Con formación de pregrado, integrante de un grupo de investigación, hace parte de un proyecto y está inscrito en el programa de jóvenes investigadores
ESTUDIANTES DE PREGRADO	Estudiante de Pregrado	En formación de pregrado
	Integrante vinculado con	Vinculado a un grupo de investigación y no cumple con ninguna de las anteriores

INTEGRANTE VINCULADO	Doctorado	definiciones
	Integrante vinculado con Maestría	Vinculado a un grupo de investigación y no cumple con ninguna de las anteriores definiciones
	Integrante vinculado con Pregrado	Vinculado a un grupo de investigación y no cumple con ninguna de las anteriores definiciones
	Integrante vinculado	Vinculado a un grupo de investigación y no cumple con ninguna de las anteriores definiciones

Fuente: Elaboración basada en (Colciencias, 2012)

En los grupos de investigación se desarrollan diversos tipos de proyectos en la cual los integrantes aportan sus habilidades y capacidades específicas en la realización de uno o varios proyectos. Entre los resultados parciales y totales que se producen en grupos de investigación se encuentran: documentos como informes de investigación, tesis de grado, libros y artículos, diseños, construcción de máquinas, ponencias en eventos, software, entre otros (Gaviria et al., 2007; González, 2009; León et al., 2011). Así mismo, cabe resaltar el valor del software en comparación con los otros tipos de productos fruto de una investigación.

En este sentido la actividad de desarrollo software en contextos de investigación, según León et al., (2011) se da para cumplir alguno de los siguientes fines: agilizar la realización de cálculos, gestionar el almacenamiento de grandes volúmenes de información, simular el desarrollo de procesos y fenómenos de toda índole, gestionar y documentar procesos en general, servir como mediador en procesos pedagógicos y didácticos enmarcados dentro de la educación en línea, ilustrar nuevas técnicas de programación, acelerar el análisis de datos, apoyar la realización de diagnósticos y gestionar la consulta de información.

Algunas de las dificultades en grupos de investigación, en los cuales realizan actividades de desarrollo software radica en que los conocimientos no son gestionados adecuadamente, los conocimientos y experiencia que los miembros de los equipos de proyecto crean y adquieren durante los proyectos software constituyen un valioso activo para el grupo de investigación y al no realizarse una adecuada gestión de estos conocimientos, se pierden parte de los conocimientos, los cuales son de gran utilidad para otros proyectos que se desarrollen en el interior del grupo (Kebede, 2010; Mejía, 2007).

En lo concerniente con proyectos de investigación en donde se contemple el desarrollo de un producto software, la situación es aún más preocupante, dado que también existe poca planificación del desarrollo software, debido a que los requerimientos del software van surgiendo a medida que el proyecto de investigación avanza, implicando menor tiempo para el desarrollo del software que el asignado para realizar la investigación, sin embargo por lo general el software debe entregarse como resultado del proyecto de investigación en el que se enmarca (Candida & Natali, 2002; León et al., 2011; Rose, Pedersen, Hosbond, & Krammergaard, 2007).

Adicionalmente del conocimiento referente al proceso de investigación se debe tener un amplio conocimiento sobre la metodología de desarrollo software que se adapte a la situación problema actual y con base en esta cómo debería formularse el proyecto software, situación que empeora cuando no hay proyectos en los que se cuente con personal con experiencia en áreas de ingeniería de sistemas o en ingeniería del software (León et al., 2011; Rose et al., 2007; Vizcaíno et al., 2006).

Esta investigación realizó un diagnóstico acerca del entorno y dificultades que enfrentan los integrantes de los grupos de investigación en el proceso de planificación de proyectos software enmarcado en actividades de investigación, este diagnóstico se llevó a cabo mediante la aplicación de una encuesta aplicada a los estudiantes de pregrado vinculados a los grupos de investigación que desarrollan software pertenecientes al Escuela de ingeniería de sistemas de la UIS (GIIB, SIMON, STI) con el fin de identificar las principales características de los grupos de investigación en actividades de desarrollo software, los resultados detallados de la encuesta se presentan en el ANEXO 1, además se envió la encuesta a dos grupos de investigación con los cuales el grupo STI tiene alianza, así mismo se realizaron algunas entrevistas a algunos profesores vinculados a los grupos de investigación de la escuela de ingeniería de sistemas de la UIS, con el fin de identificar dificultades de los estudiantes en la planificación y ejecución de sus proyectos de grado en donde uno de los productos finales es el desarrollo de un software. La encuesta tomo como grupo objetivo a los estudiantes de pregrado adscritos a algún grupo de investigación, quienes estén realizando desarrollo de software como proyecto de grado o lo hayan realizado recientemente en un periodo inferior a dos años.

Esta encuesta fue desarrollada vía web se remitió a los líderes de los grupos de investigación para que se los hiciera llegar a cada uno de los estudiantes del grupo de investigación que tuvieran las características ya mencionadas, además enviaron correos a los estudiantes y en algunos casos acompañamiento

personalizado para recibir su colaboración al contestar la encuesta, este instrumento de medición se construyó basándose en una revisión de la literatura especializada lo cual dio como resultado un diseño inicial, que se aplicó a una pequeña parte de la muestra.

Posteriormente, una vez tenido los resultados de la encuesta se tomó como caso de estudio el grupo de investigación STI para realizar diversas actividades con los miembros del grupo entre las cuales se encuentran charlas, entrevistas, taller sobre gestión de conocimiento en el proceso de desarrollo software, entre otras con el fin de profundizar en los hallazgos encontrados en la encuesta e identificar oportunidades de mejora en la planificación de proyectos software, mediante la implementación de prácticas y herramientas de gestión de conocimiento identificadas en la literatura.

En primera instancia y haciendo énfasis en el desarrollo de software realizado por estudiantes como proyecto de grado de ingeniería de sistemas, se elaboró, con base en los resultados obtenidos en la encuesta y en la revisión de la literatura, una descripción de las actividades seguidas en cada una de las etapas del proceso de desarrollo software llevado a cabo en grupos de investigación, la cual se describe en la tabla 9.

Tabla 9. Características del proceso de desarrollo Software en grupos de investigación

Etapas del proyecto de grado	Actividades seguidas en el proceso de desarrollo software.
Elaboración del tema	• En la mayoría de los casos directores o codirector del proyecto propone un tema, en menor proporción los estudiantes proponen un tema y si llama la atención de los profesores se da el aval al tema. • El director del grupo de investigación le da aval al tema dependiendo si la temática se ajusta a la temática del grupo de investigación • Se construye el título del proyecto y el objetivo general
Presentación del tema ante el comité de proyectos	• Recibir las correcciones del comité, si las hay • Esperar la aprobación del tema.
Elaboración del plan de proyecto	• Se plantean los objetivos del proyecto, tantos generales como específicos con la ayuda del director o codirector.

	• Levantamiento de requisitos del software (en algunos casos estos requisitos son entregados a los estudiantes por el director, codirector o beneficiarios del proyecto, pero de igual forma es necesario realizar actividades con los futuros usuarios para conocer de primera mano los requisitos del software) las actividades que usan los estudiantes para el establecimiento de requisitos son: Entrevistas a los interesados, grupos de opinión, talleres, cuestionarios, observaciones y lluvias de ideas. • Se selecciona la tecnología que se va usar • Entorno (Escritorio, Web, Dispositivos móviles) • Lenguajes de programación • Análisis de software o herramientas tecnológicas a usar • Seleccionar la metodología que más se adapte a las necesidades del proyecto software • Verificar los recursos físicos necesarios para el óptimo desarrollo del software • Estimar las tareas o actividades a realizar, con las estimaciones respectivas de tiempo (cronograma), los estudiantes utilizan diversas formas tales como, hablar con compañeros, la orientación del director o codirector, el grado de dificultad de las tareas, basados en datos históricos de otros proyectos ejecutados en la escuela o consultando expertos. • Realizar la estimación de costo y horas de trabajo (presupuesto), mediante la orientación del director o codirector del proyecto basado en experiencias anteriores en el desarrollo de software tanto propias como de compañeros. • Revisión de literatura, sobre la temática del proyecto • Elaborar el plan de proyecto, con la información recogida de los anteriores pasos.
Actividades complementarias paralelas a la elaboración del plan	• En el proyecto se puede llegar a definir realizar un artículo científico, una ponencias en un evento académico, un documento para el registro del software o algún documento complementario según se acuerde con el director del proyecto
Presentar el plan de proyecto	• Presentar el plan de proyecto ante el calificador previamente seleccionado por el director de proyecto. • Esperar la aprobación del plan de proyecto
Aprobación del plan de proyecto y Proyecto de	• Una vez aprobado el plan de proyecto el estudiante ha aprobado la materia trabajo de grado I

grado I	• Después de aprobarse el plan por parte del comité, este asigna un evaluador. • Cuando el plan es aprobado por el evaluador, este pasa al comité de proyectos quienes dan aval al plan para continuar con la siguiente etapa.
Desarrollo del plan de proyecto	• Realizar el análisis del sistema a través de diagramas UML que representan la interacción del sistema • Diseño de las bases de datos • Diseño de las interfaces de usuario • Según la metodología seleccionada se realiza el análisis, diseño y desarrollo • Desarrollo de las interfaces (se escribe el código) • Pruebas del software; se usa el software en diferentes formas con el fin de encontrar los errores del sistema • Implementación o puesta en marcha, se pone en marcha el software con un grupo de prueba o con los todos los posibles usuarios dependiendo del alcance del proyecto. • Elaboración del manual de usuario y manual técnico. • Elaboración de documentos complementarios al desarrollo de software (ej.: artículo científico, ponencia) • Recopilación de información para ensamblar la tesis de grado
Sustentación del proyecto	• El director del proyecto solicita los calificadores. • El comité asigna los calificadores • Se programa la sustentación • Los estudiantes sustentan su proyecto en la fecha y lugar programados. • Los evaluadores califican el proyecto, si este tiene una nota sobre 3,5 los estudiantes han aprobado su proyecto de grado
Entrega de documentos a la universidad	• Una vez sustentado el proyecto los estudiantes deben realizar los cambios que indiquen los calificadores a sus tesis y entregarla a la biblioteca de la universidad para que entre a formar parte de la base de datos que sirve de consulta para los estudiantes. • Hacer el registro del software.

Fuente: Elaboración propia

Adicionalmente estos proyectos se pueden realizar en forma individual o en grupo los cuales pueden estar integrados por dos, tres y excepcionalmente cuatro

estudiantes para realizar el proyecto, según la encuesta realizada en un 14% se realiza el proyecto en forma individual, en 81% grupos de 2 personas y un 5% realizan sus proyectos en equipos de tres o cuatro personas. En cuanto a remuneración económica para los estudiantes el 81% de los encuestados no cuenta con algún tipo de apoyo económico, frente a un 19% que si cuenta con apoyo económico. Por otra parte los directores de estos proyectos tienen formación de Maestría y Doctoral.

Por otra parte en estos grupos también se encuentran proyectos de extensión que en ocasiones tienen contratados ingenieros para desarrollar los proyectos y no van relacionados con tesis de pregrado. Adicionalmente, según León, Gómez, & Pimentel (2011) definir el alcance del proyecto en grupos de investigación, puede ser una tarea bastante compleja ya que los estudiantes se están enfrentando en la mayoría de los casos con sus primeros desarrollos y el manejo inadecuado de los cambios que ocurren en el proyecto puede modificar radicalmente la finalización del proyecto.

4.2 Conocimientos y procesos relevantes para la planificación de proyectos software en grupos de investigación universitarios.

Con base en las actividades desarrolladas con integrantes de los grupos de investigación y la revisión de la literatura realizada, se identificaron las principales dificultades en el proceso de planificación de proyectos software y los conocimientos y procesos relevantes que podrían ser gestionados por medio de las herramientas y técnicas identificadas en la literatura y comprendidas en el modelo propuesto de esta investigación, el cual se presenta en la siguiente sección. En el Anexo 1 se presenta el informe de resultados completos de la encuesta realizada a estudiantes en grupos de investigación. A continuación se presenta una síntesis de los principales hallazgos identificados en la encuesta, entrevistas y talleres realizados con estudiantes de grupos de investigación desarrolladores de software.

Los estudiantes entrevistados y/o encuestados manifiestan algunas dificultades en el proceso de planificación de proyectos software; una de ellas se presenta en la etapa inicial del proceso, con la elaboración del plan de proyecto donde surgen dudas acerca de la forma de elaborar los planes, por lo cual principalmente se recibe una asesoría del director y/o codirector de proyecto, además se acude a consultar otras fuentes de información, siendo la más consultada el plan de algún compañero que tenga un proyecto relacionado con el proyecto a realizarse, así

mismo se consulta los planes y/o libros de proyectos de grado realizados por estudiantes del grupo de investigación que estén publicados en la página de la biblioteca de la universidad o de la escuela, sin embargo no hay un sitio establecido por el grupo para acceder a esta documentación a través de un buscador que clasifique los documentos según temáticas y/o dominios de conocimiento. Otra dificultad que se presenta es que no siempre los informes de proyectos presentan los resultados en forma detallada del proyecto ni las dificultades y cambios que se presentaron en el proceso, muchos de estos presentan solamente los aspectos positivos con el fin de obtener una buena calificación y cumplir con el requisito de grado.

Por otra parte varios estudiantes mencionan dificultades en la escogencia de la metodología de desarrollo software a seguir en el proyecto, ya que en los cursos de formación se aprende varias metodologías de forma teórica, pero llegar a definir la metodología adecuada y cómo se puede implementar y seguir en el proyecto es una de las actividades más complejas para los estudiantes, por lo cual este es un conocimiento que es importante para adquirir y aprender con base en las experiencias y resultados obtenidos en otros proyectos, teniendo en cuenta las razones por las cuales se seleccionó la metodología y los resultados obtenidos durante su implementación en el proyecto.

Otro conocimiento relevante está relacionado con el lenguaje de programación con el cual se va a desarrollar el software dado que a pesar de que el estudiante pueda tener conocimiento previo del lenguaje de programación, este conocimiento por lo general es básico teniendo en cuenta que la mayoría de ellos se enfrentan a sus primeros desarrollos de software, por lo tanto es importante crear estrategias en las cuales se pueda adquirir conocimientos de otros estudiantes que hayan desarrollado software en el mismo lenguaje de programación y de esta forma poder reducir la curva de aprendizaje, ya que en los desarrollos y avances tecnológicos aparecen constantemente conocimientos que no están en los manuales y estos pueden contribuir a otros proyectos similares.

En cuanto a la estimación de esfuerzo y tiempo dedicado, es una gran dificultad que se presenta en la planificación de los proyectos, ya que en muchos casos están en frente a los primeros proyectos de desarrollo, por lo cual un conocimiento relevante es lograr presupuestar los tiempos y esfuerzos que se deben plantear en un proyecto.

Así mismo, conocer el estimado de duración de tareas o entregables de otros proyectos es conocimiento útil para tener en cuenta cuando el estudiante realice

su planificación del proyecto. En la encuesta realizada se obtuvo que la mayoría de los estudiantes realizan el cronograma en Excel y Word, pero no usan una herramienta más especializada para la elaboración de los cronogramas como Microsoft Project o Primavera, la cual sería una buena herramienta para que realicen el cronograma de sus proyectos y asignar los recursos necesarios para cada actividad. Así mismo, en cuanto a lo referente a la finalización oportuna de los proyectos, el 60% de los estudiantes encuestados no han finalizado el proyecto y de los que ya finalizaron el restante 40% excedió el tiempo estimado para la realización del proyecto.

Por otra parte, también es pertinente que el personal inmerso en los proyectos de desarrollo posea conocimientos sobre el análisis de riesgos ya que en la encuesta aplicada en más del 90% no realiza o desconoce cómo realizar este proceso, lo cual conlleva a no contemplar situaciones adversas que al materializarse pueden ocasionar retrasos o fracasos en el proyecto.

Como se puede observar la mayoría de los problemas identificados hacen referencia a los procesos de gestión del alcance y gestión del tiempo propuestos en el (PMBOK, 2012), tal vez a causa del desconocimiento de los estudiantes sobre la aplicación de buenas prácticas en la gestión de proyectos, por lo tanto el modelo propuesto presenta una breve descripción de las actividades relacionadas con estos procesos y como estas pueden aplicarse en los grupos de investigación desarrolladores de software.

4.3 Modelo de Gestión de conocimiento en la planificación de proyectos software

Con base en la revisión de la literatura y la caracterización del proceso de desarrollo software en grupos de investigación universitarios, la presente investigación presenta un modelo de gestión de conocimiento en la planificación de proyectos software diseñado para guiar la incorporación de la gestión de conocimiento en los grupos de investigación que desarrollan software, el modelo está conformado por los siguientes tres componentes: en primer lugar el modelo contempla la clasificación y descripción de prácticas y herramientas de gestión de conocimiento que fueron identificadas en la literatura como aplicables en organizaciones desarrolladoras de software, presentando un ejemplo de su posible aplicación en los grupos de investigación universitarios con el fin de mejorar el proceso de planificación de proyectos software; el segundo y tercer componente hacen referencia a las principales dificultades presentadas en la planificación de proyectos software, las cuales fueron identificadas en la encuesta, entrevistas y en talleres realizados con integrantes de grupos de investigación desarrolladores de

software, estos componentes hacen referencia a dos procesos principales de la gestión de proyectos según el PMBOK en su quinta versión se denominan la gestión del alcance y la gestión de tiempos, Así mismo en la incorporación de la gestión de conocimiento intervienen algunos factores de éxito y facilitadores de la gestión de conocimiento, los cuales según (Lee & Choi, 2002) son: la cultura, las personas, la estructura organizativa y las tecnologías de información. La figura 5 esquematiza el modelo propuesto.

Figura 5. Modelo de gestión de conocimiento en la planificación de proyectos software



Fuente: Elaboración propia.

A continuación se describen cada uno de los tres componentes junto con su aplicabilidad en la planificación de proyectos software en grupos de investigación.

4.4 Prácticas y Herramientas para la Gestión de conocimiento aplicables en grupos de investigación que desarrollan software

Teniendo en cuenta que uno de los objetivos de esta investigación es Identificar prácticas y herramientas para la gestión de conocimiento aplicables en grupos de investigación que desarrollan software, mediante una revisión sistemática de la literatura a continuación se presenta una clasificación de dichas prácticas y herramientas. Las prácticas y herramientas fueron clasificadas en 3 categorías de acuerdo a su enfoque central, Prácticas centradas en la Gestión del conocimiento explícito, Prácticas centradas en la Gestión del conocimiento tácito y Prácticas de apoyo a la Gestión de conocimiento. Una práctica puede definirse como un

conjunto de métodos, normas y procedimientos, que establecen pautas regulares y predecibles de comportamiento para la coordinación de ciertos recursos individuales, con el fin de desarrollar actividades o procesos concretos dentro de las fronteras organizativas (Cohen & Bacdayan, 1994); la Tabla 10 presenta dicha clasificación.

Tabla 10. Clasificación de Herramientas y prácticas de gestión del conocimiento

Prácticas centradas en la Gestión del conocimiento Explícito	Prácticas centradas en la Gestión del conocimiento Tácito	Prácticas de Apoyo a la Gestión de conocimiento	
		Cultura Organizativa	Seguimiento y control
• Publicaciones • Gestión Documental • Capacitaciones	• Grupos temáticos • Reuniones • Lecciones Aprendidas	• Motivación • Principios • Incentivos • Aprendizaje	• Análisis post-mortem • Control de procesos • Gestión de riesgos • Sistemas de gestión de portafolios
Herramientas			
• Repositorios de conocimiento • Plantillas y formatos • Directorios de expertos • Aula virtual • Sistemas de gestión documental	• Sistemas colaborativos • Sistema de razonamiento basado en casos	• Análisis de debilidades, fortalezas, Amenazas y oportunidades (matriz DOFA) • Software de Gestión de proyectos como Microsoft Project y Primavera.	

Fuente: Elaboración propia.

La primera categoría está centrada en la gestión del conocimiento explícito formadas por prácticas y herramientas enfocadas en almacenar y transferir conocimiento estructurados, codificados y/o formalizados en la organización con el fin de mejorar la memoria organizacional de las organizaciones que para esta situación de interés son los grupos de investigación; prácticas como creación de informes y publicaciones de los resultados parciales y finales de los proyectos,

gestión documental, definición de plantillas , formatos y repositorios de conocimiento son prácticas enfocadas en el proceso de combinación de conocimientos lo cual implica la combinación de distintos cuerpos de conocimiento explícitos, mientras que prácticas y herramientas como capacitaciones, directorio de expertos y aula virtual están centradas en los procesos de exteriorización e interiorización de conocimientos. La tabla 11 presenta una descripción de cada una de las prácticas y herramientas que conforman esta categoría, junto con una posible forma de aplicación en los grupos de investigación que desarrollan software.

Tabla 11. Prácticas de gestión de conocimiento centradas en la gestión del conocimiento explícito

Prácticas y herramientas	Descripción	Aplicabilidad en los grupos de investigación
Creación de informes y Publicaciones	Esta práctica hace referencia a documentar los avances del proyecto en informes, presentaciones, videos , folletos, imágenes y demás formas de documentos estructurados, Las publicaciones internas o externas son un método de distribución del conocimiento, ya que las publicaciones son depósitos de conocimiento que mediante la gestión documental se convertirán en un activo estratégico (Manzanares & Gómez, 2010).	Sintetizar las investigaciones e incentivar a los estudiantes para que las publiquen en revistas científicas, eventos académicos, informes, reportes y/o casos y se pongan a disposición de los integrantes del grupo de investigación.
Gestión documental	Practica que permite administrar la documentación existente de diferentes autores, organizarla o inclusive fragmentarla con el	Almacenar información relevante de cada uno de los proyectos que sirva de guía para futuros desarrollos, acompañado del uso de formatos o guías que faciliten

	fin de que sea más útil y de fácil acceso para los interesados (Aurum et al., 2008).	el acceso y consulta a los integrantes del grupo.
Capacitación de los estudiantes.	Ser capaces de identificar las fuentes de conocimiento dentro de la organización, valorar a los empleados por lo que saben y tener procedimientos eficaces para la transferencia de este conocimiento son prácticas que permitirán una adecuada repartición de conocimiento útil dentro de la organización (Zack, McKeen, & Singh, 2009)	Identificar falencias en los conocimientos que tengan los integrantes del grupo de investigación y que puedan ser útiles para mejorar el proceso de desarrollo software, organizar cursos o generar espacios para que compartan ese conocimiento, aprovechando la infraestructura existente para capacitar a los estudiantes.
Repositorios de conocimientos	Hacen referencia a la memoria organizacional. Son sistemas para almacenar y difundir el capital intelectual de la organización (Petter, Mathiassen, & Vaishnavi, 2007; Portillo, Vizcaíno, Piattini, & Beecham, 2012).	Identificar y almacenar conocimientos útiles para la planificación de proyectos de desarrollo software.
Definición de Plantillas y formatos	Las plantillas son formatos establecidos para planes de proyectos, presupuestos, o documentación de proyectos anteriores, etc., que son continuamente reutilizados para mantener la consistencia entre los proyectos. Una parte importante del tiempo se dedica a documentar un proyecto a través de planes	Definir plantillas y formatos en donde se almacene información relevante a la toma de decisiones en el proyecto, tales como plantillas para documentación de requisitos, seguimiento a metodología y cronograma; así mismo manejo de plantillas exigidas por la universidad y entidades para las cuales se está realizando el

	de desarrollo, de calidad, y prueba, así como requisito y las especificaciones de diseño. Las plantillas compartidas entre colegas pueden ayudar a los gerentes de proyectos de software aprovechar el saber cómo (know-how), del conocimiento de proyectos anteriores (Petter et al., 2007).	proyecto del software.
Páginas amarillas de expertos	Una lista de personas de la organización ordenada sistemáticamente y alfabéticamente, figurando filiación, dirección, funciones, sus habilidades, áreas de conocimiento en la cual son expertos y otros datos de interés para cada una de ellas.(Alhawari et al., 2012; Evangelista, Esposito, Lauro, & Raffa, 2010).	En los grupos de investigación, se puede almacenar los datos de personas que han trabajado en proyectos del grupo, expertos en un área específica de conocimiento y/o personas que han dado algún tipo de asesoría o calificación a los proyectos. También pueden ser contactos que tenga el grupo de investigación con otros grupos o instituciones.
Aula virtual	Plataformas virtuales para dar capacitaciones al personal de una organización.	Implementar moodle para el seguimiento a cursos y capacitaciones ofrecidas por el grupo de investigación a sus integrantes.

Fuente: Elaboración propia

La segunda categoría de prácticas y herramientas para la gestión de conocimiento están enfocadas en la gestión del conocimiento tácito como se aprecia en la tabla 12, en donde los miembros de la organización intercambian sus conocimientos implícitos en ellos y adquiridos por medio de la experiencia. Esta categoría está relacionada con los procesos de socialización y exteriorización del conocimiento propuesto por Nonaka y Takeuchi en 1995. Estrategias como la formación de

grupos temáticos, comunidades de práctica y redes de trabajo colaborativo, están direccionadas a la interacción entre los integrantes de la organización para intercambio de conocimientos y la creación de una cultura de trabajo en equipo, junto con el establecimiento de reuniones periódicas y la implementación de sistemas de razonamiento basado en casos permite sacar provecho a las lecciones aprendidas en la organización permitiendo identificar oportunidades de mejora, disminución de errores y reproceso, establecer ideas innovadoras y mantener un aprendizaje continuo al interior de la organización.

Tabla 12. Prácticas de gestión de conocimiento centradas en la gestión del conocimiento tácito

Prácticas y herramientas	Descripción	Aplicabilidad en los grupos de investigación
Grupos temáticos.	Los grupos o comunidades temáticas están formados por un grupo de personas que comparte un interés común en un tópico específico, quienes deben tener unos objetivos o metas definidos en un tiempo con productos tangibles que se puedan socializar con los integrantes (Gradillas Leverté, 2001).	Organizar los estudiantes en grupos con intereses comunes (p. ej. Objetivos del proyecto, el mismo lenguaje de programación) con el fin de que socialicen sus experiencias en la temática que los agrupa y así generar una apropiación de este conocimiento.
Reuniones	Las reuniones son una práctica bastante usada en la gestión y transferencia del conocimiento, ya que este modelo requiere un alto nivel de socialización para tener éxito. Razón por la cual las reuniones cara a cara se convierten en un importante canal de comunicación al permitir que se compartan normas, actitudes, comportamientos,	Organizar periódicamente reuniones con todos los estudiantes que estén involucrados con actividades de desarrollo software, y realizar presentaciones sobre los avances de cada uno de los proyectos, además socializar las dificultades u obstáculos que hayan tenido en cada una de las etapas del proceso de ejecución del proyecto y la forma en cómo las resolvieron, especificando el

	experiencias y conocimientos (Basri & Rory, 2011; Richardson, Casey, McCaffery, Burton, & Beecham, 2012).	aprendizaje obtenido en el proceso.
Sistemas colaborativos	Los sistemas colaborativos permiten a las personas conectarse virtualmente a reuniones, discusiones, lluvia de ideas y compartir conocimiento. Usando sistemas de colaboración los participantes crean nuevo conocimiento vía exploración y desarrollo, lo cual permite una comprensión del saber porque (know-why) de los proyectos (Petter et al., 2007). Este tipo de conocimiento es tácito al usar sistemas colaborativos y se centra en la creación de soluciones que se ajustan al contexto presente del proyecto. Repositorios de contenido almacenan el conocimiento documentado en una base de datos centralizada para crear un almacén de conocimientos dentro de la organización y ayudan a explotar los conocimientos existentes mediante la reutilización en futuras situaciones. La atención se centra en el almacenamiento de las mejores prácticas y métodos para documentar los	En el grupo de investigación se pueden implementar herramientas de trabajo colaborativo que permitan hacer seguimiento a los proyectos desarrollados y manejo de versiones por el equipo de trabajo, en este sistema el docente o líder del proyecto podrá emitir sugerencias y veritos sobre la calidad del sistema a desarrollar, así mismo se pueden establecer discusiones respecto al diseño, alcance y desarrollo del sistema. Esta plataforma debe incorporar herramientas de comunicación como chats, foros, noticias, y mensajes internos, entre otras funcionalidades. Otra alternativa es la formación de comunidades de práctica en torno a temas de interés del grupo de investigación.

	conocimientos obtenidos en proyectos anteriores, con ello la captura de pericia organizativa.	
Lecciones aprendidas	Lecciones aprendidas son conocimientos adquiridos a través de experiencias y observaciones realizadas durante la ejecución de tareas y la resolución de problemas, en las cuales dejan enseñanzas conclusiones y recomendaciones para futuras situaciones (Basri & Rory, 2011; Kebede, 2010; Ioana Rus & Lindvall, 2002)	Almacenar todos los incidentes y situaciones problemas que se han presentado en los proyectos de desarrollo software junto con la solución implementada y los resultados obtenidos. También puede ser útil para la gestión del conocimiento específico del dominio de la aplicación y la identificación de oportunidades de investigación y desarrollos software que se originaron de proyectos anteriores pero no pudieron ser implementados por cuestión de alcance, tiempo y/o disponibilidad de recursos.
Sistemas de razonamiento basado en casos	Un sistema de razonamiento basado en casos (SRBC) es un modelo de razonamiento que nos permite resolver problemas, entender situaciones y aprender de ellas, integrado con los mecanismos asociados a la memoria. Este modelo ofrece un nuevo paradigma para la construcción de sistemas inteligentes que se basa en la utilización de la experiencia previa (Dingsøyr & Conradi, 2002; Garrido et al., 2012; Gestión et al., 2006; Irona Rus & Maryland, 2002)	Diseño e implementación de un software que facilite la gestión del tiempo y la selección de la metodología de desarrollo software en el proyecto con base en resultados obtenidos en proyectos anteriores, parametrizando los casos para que el sistema identifique el proyecto más similar y de alternativas de apoyo a la toma de decisiones.

La tercera categoría de prácticas y herramientas para la gestión de conocimiento están orientadas a crear un ambiente de trabajo propicio para la gestión de conocimiento abarcado el principal factor de éxito en los proyectos de gestión de conocimiento, la cultura (Lee & Choi, 2002), así mismo en esta categoría se contemplan prácticas y herramientas destinadas a hacer seguimiento y control tanto a las acciones planteadas para gestionar conocimiento como a los procesos relacionados con la gestión de proyectos en la organización. La Tabla 13 presenta una descripción de cada práctica y/o herramienta identificada en la literatura junto con su posible forma de implementación en los grupos de investigación que desarrollan software con el objetivo de mejorar la planificación en sus proyectos.

Tabla 13. Prácticas y herramientas de apoyo a la gestión de conocimiento

Prácticas y herramientas	Descripción	Aplicabilidad en los grupos de investigación
Análisis post-mortem	El análisis post-mortem consta de un análisis exhaustivo al proceso de desarrollo software una vez se haya finalizado, este análisis involucra un grupo de expertos, quienes se reúnen con los desarrolladores y realizan un análisis de causa y efecto de cada una de las dificultades o fallas del procesos ayudados del diagrama de espina de pescado, finalmente se documenta este análisis para uso del resto de la organización (Birk, Dingdoyr, & Stalhane, 2002). El conocimiento obtenido mediante vía post-mortem puede ayudar en la reducción de la repetición de los errores del pasado, mediante la obtención de un	Cada vez que se finalice un proyecto en el grupo de investigación realizar un análisis post-mortem, ayudados de un grupo de expertos pertenecientes al grupo, generando la respectiva documentación donde se incluya la descripción del proyecto, el producto resultante, el método usado y los tiempos y esfuerzo realizados.

	recuento histórico de lo que salió mal, la mejora de los procesos y el desempeño en proyectos futuros (Dingsøyr et al., 2009; Dingsøyr & Conradi, 2002); lo cual apunta a explicaciones de saber por qué del conocimiento, a través del análisis exploratorio y a menudo documentado por el administrador del proyecto o de los miembros del equipo del proyecto para facilitar el intercambio de toda la organización (Petter et al., 2007).	
Creación de principios de cultura organizacional centrada en la gestión de conocimiento	La práctica de implementación de una cultura apoyada en los principios organizacionales, sirve como sistema de apoyo a la gestión del conocimiento que se desea aplicar a una compañía, ya que las prácticas que se implementaran en la gestión del conocimiento se convertirán en la base genérica de obtención de ventajas competitivas (Manzanares & Gómez, 2010).	Establecer los principios del grupo centrados en la calidad de los productos, el cumplimiento de los tiempos de entrega, la adquisición de nuevos conocimientos para aplicarlos en futuros desarrollos de software, teniendo como eje central la gestión de conocimiento, concientizando a los integrantes sobre la importancia del aporte de cada uno de ellos en los actuales y futuros desarrollos de software en el grupo de investigación.
Motivar a los estudiantes a compartir su	La práctica de motivar a los trabajadores a compartir su conocimiento, es muy útil en las organizaciones y es un sistema de apoyo efectivo para la	Motivar las iniciativas de transmitir conocimiento, a través de premios, reconocimientos públicos, brindándoles entornos que les permitan un óptimo

conocimiento	gestión del conocimiento, ya que se aplica a la efectiva transferencia, recuperación y aplicación del conocimiento e incluye aspectos de tipo humano (Manzanares & Gómez, 2010).	desarrollo de habilidades, buscando aumentar en los estudiantes los espacios participativos para que tengan la oportunidad de aportar sus conocimientos en el mejoramiento del proceso.
Control del proceso	El control de proceso es un sistema de apoyo de la gestión del conocimiento en las organizaciones, porque facilita la identificación de conocimientos claves para ser compartidos a la vez que permite un mejor almacenamiento o administración de este conocimiento (Manzanares & Gómez, 2010)	A través de la identificación de las características del proceso de desarrollo software en grupos de investigación, determinar las etapas con sus respectivos requerimientos, con el fin de identificar los conocimientos que deben adquirir los estudiantes para llevar a cabo cada etapa, y así facilitar el seguimiento y evaluación del proyecto.
Gestión de riesgos	“La Gestión de los Riesgos del Proyecto incluye los procesos para llevar a cabo la planificación de la gestión de riesgos, así como la identificación, análisis, planificación de respuesta y control de los riesgos de un proyecto. Los objetivos de la gestión de los riesgos del proyecto consisten en aumentar la probabilidad y el impacto de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos negativos en el proyecto.” (PMI, 2012)	Identificar y establecer estrategias para la gestión del riesgo en los proyectos de desarrollo software realizado por el grupo de investigación, con el fin de reducir la probabilidad y/o impacto de materializarse un riesgo sobre el proyecto en desarrollo.

DOFA	Esta clásica técnica de administración puede identificar características claves en los proyectos Software en la cual se pueden identificar fortalezas, amenazas, debilidades y oportunidades (Petter et al., 2007)	Identificar las debilidades, oportunidades, fortalezas y amenazas que tiene el grupo de investigación respecto a los proyectos que puede ejecutar.
Software de gestión de proyectos	Por otra parte un aspecto práctico que ofrecen diferentes herramientas online es Realizar disposición de horarios, planes y programaciones con identificación clara de tareas, responsables y recursos disponibles (Rose et al., 2007; Vizcaíno et al., 2006).	Por otra parte con respecto a las herramientas cuyo fabricante es Microsoft, hay un programa cuyo nombre es el convenio MSDN entre universidades y Microsoft para el uso de software licenciado a través de los estudiantes, por lo cual se puede usar cualquier software de Microsoft. Entre estos productos encontramos SharePoint, SharePoint server, Project, sistemas operativos etc. Las cuales estarían disponibles para uso de los estudiantes que desarrollan software en los grupos de investigación. Microsoft Project es una herramienta ampliamente usada para la gestión de proyectos, también se encontró una aplicación de nombre Ganttter la cual está en la web para la gestión de proyectos basada en el software de Microsoft Project, al ingresar y registrarse en el sitio la interfaz gráfica y el registro en esta herramienta es similar a lo que se hace en

		Microsoft Project, además incluye compatibilidad con google drive por lo cual se tendrá acceso a los documentos relacionados con el proyecto, también se puede compartir el cronograma del proyecto con todo el equipo y tener de tal forma que el equipo de desarrollo puede ver el cronograma en cualquier lugar y todo queda almacenado en la nube.
Sistemas de gestión de portafolios	Las inversiones y sistemas de gestión de portafolios se realizan para identificar, planificar y obtener un seguimiento de las inversiones en los proyectos. Esta herramienta es útil para seleccionar y planificar futuros proyectos sobre la base de las necesidades de la organización. Los líderes del proyecto pueden explotar el conocimiento de los proyectos anteriores, generados por la organización y el entorno, con el fin de desarrollar y documentar diferentes aspectos que lleven a la organización a tomar decisiones para contratar o mantener sus inversiones en un proyecto (Petter et al., 2007). Por otra parte, los mapas de	Implementar un sistema de gestión de portafolios para la gestión de proyectos en grupos de investigación.

	conocimiento ayudan a quienes buscan algún conocimiento lo puedan encontrar en las personas que tengan la experiencia necesaria.(Broadbent, Weill, & Neo, 1999; Petter et al., 2007)	
--	---	--

Fuente: elaboración propia

De igual modo (Zack et al., 2009) identifican 12 prácticas de gestión de conocimiento y analizan su influencia en el desempeño de la organización, estas doce prácticas también pueden ser clasificadas en esta tercera categoría dado que representan estrategias que pueden ser incorporadas por la organización para generar una cultura propicia para la gestión de conocimiento, las prácticas propuesta por (Zack et al., 2009) son:

- KP1 Reconocer explícitamente el conocimiento como un elemento clave en el ejercicio de planificación estratégica.
- KP2 Realizar benchmarking del conocimiento estratégico frente a la de los competidores.
- KP3 Lograr desarrollar un conocimiento estratégico con los mapas de conocimiento para la creación de valor.
- KP4 Ser capaces de identificar las fuentes de conocimiento dentro de la organización.
- KP5 Los empleados son valorados por lo que saben.
- KP6 Buscar oportunidades para experimentar y aprender más acerca de los clientes.
- KP7 Buscar oportunidades para experimentar y aprender más sobre los productos y servicios.
- KP8 Buscar oportunidades para experimentar y aprender más acerca de las tecnologías y las operaciones internas.
- KP9 La organización alienta y premia el intercambio de conocimientos.

- KP10 Tener procedimientos internos eficaces para la transferencia de mejores prácticas en toda la organización.
- KP11 Aprovechar las fuentes externas de conocimiento de manera efectiva, incluyendo el conocimiento del cliente.
- KP12 El grupo de gestión del conocimiento es una fuente reconocida de creación de valor dentro de la organización.

Por otra parte se tienen herramientas tecnológicas clasificadas como herramientas que contribuyen con la gestión del conocimiento que son utilizadas transversalmente a las anteriores clasificaciones y categorías descritas. La Tabla 14 presenta el nombre de estas herramientas identificadas en la literatura.

Tabla 14. Herramientas de gestión conocimiento

Herramientas de gestión conocimiento
4Everedit
Addss
Bugzilla
Caws
CollabDev
DOCTOR
Galaxy wiki
Google docs
Google groups
IBistro
KNOWFACT
KnowledgeTree
LiveNet
Lotus Quickr
Microsoft Sharepoint
MoinMoin
MULTIMIND
PAKME
Saperion ECM
TWiki
Xerox Docushare

Fuente: tomado de (Portillo et al., 2012)

4.5 Gestión del alcance de los proyectos de desarrollo software en grupos de investigación universitarios

La gestión del alcance según el PMBOK en su versión 5 “incluye los procesos necesarios para garantizar que el proyecto incluya todo el trabajo requerido y únicamente el trabajo para completar el proyecto con éxito, se enfoca primordialmente en definir y controlar qué se incluye y que no se incluye en el proyecto”(PMI, 2012). La gestión del alcance aborda los siguientes 6 procesos: 1. Planificar la gestión del alcance, 2. Recopilar requisitos, 3. Definir el alcance, 4. Crear la estructura de desglose de trabajo, 5. Validar el alcance y 6. Controlar el alcance del proyecto. Los primeros 4 procesos hacen parte de la planificación de los proyectos mientras que los procesos 5 y 6 están relacionados con el seguimiento y control a los proyectos. Dado que el objetivo de este proyecto es centrarnos en el proceso de planificación de proyectos, solo se abordaran los 4 primeros procesos relacionados con la planificación.

4.5.1 Planificar la gestión del Alcance: este proceso consiste en definir la forma en cómo se va a validar y controlar el alcance del proyecto y la forma en cómo se analizarán, documentaran y gestionaran los requisitos. Este proceso es de gran importancia en la planificación de proyectos software dado que permite definir los interesados del proyecto y las formas en las cuales se van a tomar los requisitos del software, para llevar a cabo este proceso se debe tener en cuenta los siguientes factores: la cultura del grupo de investigación, las políticas establecidas en el grupo y los reglamentos institucionales que rigen los desarrollos ya sea por proyectos de grado o reglas establecidas en contratos con entidades públicas o privadas.

4.5.2 Recopilar Requisitos: Proceso mediante el cual se definen los requisitos del software y se elabora el acta de requisitos, la cual debe ser aprobada por los interesados del proyecto; a partir de esta recopilación de los requisitos y con base en las normas que rigen el proyecto se define el alcance del proyecto. Es importante que en este proceso participen todos los interesados del proyecto dado que esto garantizara que el resultado satisfaga las necesidades de los clientes y/o usuarios del software. Así mismo es de gran utilidad realizar la matriz de trazabilidad de requisitos con el fin de vincular los requisitos del producto desde su origen hasta los entregables del proyecto y de esta forma contribuir a que cada requisito esté vinculado con los objetivos de la organización y del proyecto. Algunas de las técnicas que se pueden utilizar en este proceso son: Entrevistas, formación de grupos temáticos, técnicas

grupales de creatividad y de toma de decisiones, cuestionarios, encuestas y analogías con otros proyectos, etc.

Por otra parte, algunas de las herramientas utilizadas para la definición de requisitos se presentan a continuación en la tabla 15: Herramientas de requerimiento: en este caso, la característica más frecuentemente provista es el sistema de seguimiento de incidencias, que hace que los usuarios sean más conscientes de los sucesos y cambios. Esta característica esta generalmente complementada con técnicas de percepción visual tales como destacar aspectos importantes con colores diferentes y la posibilidad de insertar comentarios.

Tabla 15. Herramientas de Requerimientos.

Herramientas de requerimientos (SRT)
Arena
DOORS
EGRET
ERequirements
GatherSpace
Rational Requisite Pro
Rational Requirement Composer

Fuente: tomado de (Portillo et al., 2012)

4.5.3 Definición del alcance: Describe los límites del proyecto mediante la definición de cuales requisitos serán abordados en el desarrollo del proyecto y cuáles serán excluidos del proyecto. Como principales resultados se obtiene la descripción detallada de los requisitos y el enunciado del alcance, la definición de los criterios de aceptación, entregables, exclusiones, supuestos y restricciones. Para la definición del alcance se debe tomar en consideración los procedimientos planteados en la organización, la definición del problema que se va a abordar y la experiencia sobre el área para el cual va a ser desarrollado el software.

4.5.4 Definición de la EDT: En este proceso se hace la subdivisión de los entregables de proceso en subentregables del proyecto y se definen las actividades necesarias para la obtención de los entregables.

4.6 Gestión del tiempo en proyectos de desarrollo software en grupos de investigación universitarios

La gestión del tiempo contempla la definición del cronograma de trabajo, detallando cada una de las actividades a realizar para dar cumplimiento a los objetivos del proyecto. Los procesos que conforman la gestión del tiempo son: 1. La planificación de la gestión del cronograma, 2. Definición de actividades, 3. Secuenciación de actividades, 4. Estimación de recursos por actividad, 5. Estimación de la duración de las actividades, 6. Desarrollo del cronograma y 7. Controlar el cronograma. Los 6 primeros procesos están relacionados con la planificación de proyectos y el proceso 7 hace referencia al control del proceso.

4.6.1 Definir la gestión del cronograma: se encarga de establecer las políticas y procedimientos para administrar el cronograma, este proceso debe tener en cuenta los siguientes factores ambientales de los grupos de investigación: Cultura, Recursos físicos, las habilidades de estudiantes, profesores e investigadores relacionadas tanto con el proceso de desarrollo software como con el dominio de campo de aplicación del software a desarrollarse. Así mismo entre los activos de conocimiento que posee el grupo de investigación y pueden ser de utilidad son: las plantillas establecidas, documentación de proyectos anteriores, software existente, normas y políticas establecidas por la organización, experiencia de los miembros del grupo sobre el dominio de la aplicación.

Entre las técnicas para la definición de la gestión del cronograma se encuentran: Juicios de Expertos, el cual puede ser apoyado de un directorio de expertos dentro del grupo de investigación destacando sus principales áreas de conocimiento, técnicas analíticas como planificación gradual y análisis de alternativas.

4.6.2 Definir actividades: para la definición de las actividades específicas que se deben abordar en el proyecto se debe considerar los siguientes factores: cultura, políticas de la universidad respecto a la gestión de proyectos, reglamentos de grado, contratos, entre otros. Así mismo debe hacer uso de las lecciones aprendidas en el grupo de investigación de otros proyectos similares, plantillas, formatos y guías existentes. Como técnicas se pueden utilizar la descomposición de actividades en subactividades, la planificación gradual, el juicio de expertos y analogías con proyectos similares.

4.6.3 Secuenciar actividades: algunas técnicas utilizadas en la ejecución de este proceso son la diagramación por precedencia y la determinación de dependencias entre las actividades, puede hacerse con base en proyectos

anteriores y en la experiencia del equipo de desarrollo y gestión del proyecto; así mismo debe considerar los siguientes aspectos: normas institucionales e internas del grupo de investigación, sistemas de autorización de proyecto, restricciones de recursos como tiempo, personas, material, etc.

4.6.4 Estimar recursos: Definición de todos los recursos necesarios para la realización de los proyectos, entre los cuales están información, personal, expertos, licencias de software, equipos de cómputo, infraestructura física, etc y asignar los recursos a cada una de las actividades realizadas para la consecución de los entregables. Esto permitirá identificar limitaciones para el acceso a los recursos y posibles atrasos en el proyecto.

4.6.5 Estimar duración de actividades: entre los factores del entorno de la organización se destacan: la disponibilidad de recursos, estimaciones previas y ubicación de los recurso para el caso que se encuentren fuera de las instalaciones del grupo de investigación y/o universidad. Como activos de conocimiento de la organización se encuentra la experiencia en proyectos similares, calendarios y metodologías previamente definidos. Entre las técnicas útiles para este proceso se encuentran: juicio de expertos, estimación por analogía, estimación parametrizada, enfoque pesimista, optimista y probabilista, técnicas grupales, etc.

4.6.6 Desarrollar el cronograma: para la definición del cronograma del proyecto se puede usar herramientas como el método de la ruta crítica y cadena crítica y analogías con otros proyectos; se debe considerar los siguientes factores del entorno: el tiempo estimado por la universidad para el desarrollo del proyecto(principalmente cuando es un proyecto de grado o que se está recibiendo financiación para realizar el proyecto), los canales de comunicación existente, la tecnología a utilizar y las herramientas de planificación. Como activos de conocimiento se encuentran la experiencia, la metodología de desarrollo, planes aprobados y reglamentos.

5. CONCLUSIONES

En cuanto al desarrollo software enmarcado en actividades de investigación, en el interior de grupos de investigación, es realizado por equipos de desarrollo poco maduros, los cuales están conformados por un alto porcentaje por estudiantes que en ocasiones se enfrentan a sus primeros desarrollos presentándose problemas en la definición del alcance y en la gestión de tiempos del proyecto, así mismo al terminar el proyecto se van del grupo llevándose consigo parte del conocimiento adquirido en la elaboración del proyecto, el cual no queda explícito en los productos resultantes del proyecto, por tanto este conocimiento no puede ser aprovechado para el beneficio del grupo y sus futuros integrantes, lo anterior genera que el conocimiento obtenido no permanezca en la organización, y los nuevos integrantes deben planificar sus proyectos de desarrollo sin tener en cuenta el conocimiento generado por otros y repitiendo errores del pasado. Adicionalmente se observó que la mayoría de los problemas identificados hacen referencia a los procesos de gestión del alcance y gestión del tiempo propuestos en el PMBOK, 2012, tal vez a causa del desconocimiento de los estudiantes sobre la aplicación de buenas prácticas en la gestión de proyectos, por lo tanto el modelo propuesto presenta una breve descripción de las actividades relacionadas con estos procesos y como estas pueden aplicarse en los grupos de investigación desarrolladores de software.

La incorporación de la gestión de conocimiento en el proceso de desarrollo Software contribuye con el aprovechamiento de los conocimientos generados por los integrantes del equipo de desarrollo, que para el caso de los grupos de investigación vinculados a universidades, son en su mayoría estudiantes, contribuyendo así al aprendizaje de los nuevos integrantes del grupo, dándoles más elementos para realizar exitosamente sus proyectos. Asimismo las diferentes prácticas y herramientas presentadas pueden ser tomadas como referencia para los grupos de investigación e implementarlas con el fin de obtener un aprendizaje continuo con base en proyectos anteriores y generar espacios en donde los integrantes del grupo puedan compartir experiencias logrando tener un mejor rendimiento y productividad en el desarrollo de los proyectos software.

Prácticas y herramientas de gestión de conocimiento como: Publicaciones, Gestión documental, participación en eventos, capacitaciones, repositorios de contenidos, sistemas de gestión de proyectos y páginas de directorios de expertos contribuyen a gestionar el conocimiento explícito de la organización, generando una adecuada apropiación del conocimiento de proyectos anteriores sirviendo de

base para futuros proyectos. Actualmente las anteriores herramientas se encuentran en mayor proporción en plataformas web, con características más robustas, las cuales permiten una mayor difusión y almacenamiento de conocimientos útiles que contribuyan al aprendizaje de los estudiantes y a reducir tiempos, errores y reprocesos en los proyectos de desarrollo Software.

Esta investigación se limitó al desarrollo software producido en grupos de investigación vinculados a instituciones universitarias, centrándose en el proceso de desarrollo software producido por estudiantes, sin embargo los resultados de esta investigación pueden ser una guía para la incorporación de la gestión de conocimiento en el desarrollo software a otros tipos de organizaciones de investigación, así mismo futuras investigaciones pueden orientarse en la definición de modelos y metodologías que incorporen las prácticas y herramientas mencionadas o extender el trabajo a organizaciones cuyas actividades de desarrollo software no están inmersas en un entorno de investigación.

BIBLIOGRAFÍA

- ACM. (2005). Computing Curricula. *Overview Report. Draft*. Retrieved from http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf
- Alavi, M., & Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly*, 25(1), 107–136. doi:10.2307/3250961
- Alhawari, S., Karadsheh, L., Nehari Talet, A., & Mansour, E. (2012). Knowledge-Based Risk Management framework for Information Technology project. *International Journal of Information Management*, 32(1), 50–65.
- Amit, R., & Shoemaker, P. (1993). Strategic assets and organizational rent. *Strategic Management Journal*, 14(1), 33–46. doi:10.1002/smj.4250140105
- Aurum, A., Daneshgar, F., & Ward, J. (2008). Investigating Knowledge Management practices in software development organisations – An Australian experience. *Information and Software Technology*, 50(6), 511–533. doi:10.1016/j.infsof.2007.05.005
- Avison, D., Lau, F., Myers, M., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, 42(1), 94–97. doi:10.1145/291469.291479
- Baskerville, R. L. (1999). Investigating Information Systems With Action Tutorial Investigating Information Systems with Actio. *Communications of the Association for Information Systems*, 2(19), 1–32.
- Baskerville, R., & Myers, M. (2004). SPECIAL ISSUE ON ACTION RESEARCH IN INFORMATION SYSTEMS: MAKING IS RESEARCH RELEVANT TO PRACTICE — FOREWORD. *MIS Quarterly*, 28(3), 329–335.
- Basri, S., & Rory, V. O. (2011). A Study of Knowledge Management Process Practices in Very Small Software Companies, 3(4), 636–644.
- Birk, A., Dingdoyr, T., & Stalhane, T. (2002). Postmortem : Never Leave a project without it. *IEEE Software*, 19(3), 43–45. doi:10.1109/MS.2002.1003452
- Bjørnson, F. O. (2007). *Knowledge Management in Software Process Improvement*. Norwegian University of Science and Technology.

- Bjornson, F. O., & Dingsøyr, T. (2008). Knowledge management in software engineering : A systematic review of studied concepts , findings and research methods used. *Knowledge Management*. doi:10.1016/j.infsof.2008.03.006
- Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 61–72.
- Borjesson, A., & Mathiassen, L. (2004). Successful Process Implementation. *IEEE Software*, 21(4), 36–44.
- Broadbent, M., Weill, P., & Neo, B. S. (1999). Strategic context and patterns of IT infrastructure capability. *Information Systems*, 8(December 1996), 157–187.
- Brooking, A. (1999). *Corporate Memory*. London; International Thomson Business Press.
- Buchanan, B. G., Barstow, R., Bechtal, R., Bennet, J., Clan- cey, W., Kulikowsky, C., ... Waterman, D. A. (1983). Constructing an expert sys- tem. *Building Expert Systems*, 127–167.
- Camson, Palacios, & Devece. (2000). *Un nuevo modelo para la medición del capital intelectual en la empresa*.
- Candida, A., & Natali, C. (2002). Knowledge Management in Software Engineering Environments. *Measurement*, 238–253.
- Capote, J., Jesús, A. De, Ramírez, G., Alberto, C., Software, P., Las, E. N., & Empresas, P. Y. M. (2008). Gestión del conocimiento como apoyo para la mejora de procesos software en las micro , pequeñas y medianas empresas. *revista ingeniería e investigación*, 28(1), 137–145.
- Checkland, P., & Scholes, J. (1990). *Soft systems methodology in action*. John Wiley & Sons Ltd.
- Chen, X., Li, X., Clark, J. G., & Dietrich, G. B. (2013). Knowledge sharing in open source software project teams: A transactive memory system perspective. *International Journal of Information Management*, 33(3), 553–563. doi:10.1016/j.ijinfomgt.2013.01.008
- Cohen, M., & Bacdayan, P. (1994). Organizational routines are stored as procedural memory: Evidence from a laboratory study. *Organization science*, 4(4), 554–568. Retrieved from <http://www.jstor.org/stable/2635182?seq=1>

- Colciencias. (2008). *Modelo de medición de Grupos de Investigación, Tecnológica o de Innovación* (pp. 1–40).
- Colciencias. (2012). *Modelo de Medición de Grupos de Investigación Científica, Desarrollo Tecnológico e Innovación Dirección de Fomento a la Investigación, documento en construcción.* (pp. 1–100).
- Colciencias. (2013). *Modelo de medición de grupos de investigación, desarrollo tecnológico e innovación.*
- Collis, D. J., & Montgomery, C. A. (1995). Competing on resources: strategy in the 1990s. *Harvard Business Review*, 73(4), 118–128.
- Corbin, R. D., Dunbar, C. B., & Zhu, Q. (2007). A three-tier knowledge management scheme for software engineering support and innovation. *Journal of Systems and Software*, 80(9), 1494–1505. doi:10.1016/j.jss.2007.01.013
- Davenport, T., & Prusak, L. (1999). *Ecología de la información.* Oxford University Press.
- Desouza, K. ., & Awazu, Y. (2006). Knowledge Management at SMEs: five peculiarities. *Journal of Knowledge Management*, 10(1), 32–43.
- Dingsøyr, T., Bjørnson, F. O., & Shull, F. (2009). What Do We Know about Knowledge Management Practical Implications for Software Engineering. *Software, IEEE*, 26(3), 100–103. doi:10.1109/MS.2009.82
- Dingsøyr, T., & Conradi, R. (2002). A Survey of Case Studies of the use of Knowledge Management in Software Engineering. *International Journal of Software Engineering and Knowledge Engineering*, 12(4), 391–414.
- Dorfman, M., & Thayer, R. (2002). *Software Engineering, IEEE Computer Society Press.*
- Edvinson, L., & Malone, M. (1999). *El capital Intelectual. Barcelon: Gestión 2000.*
- Euforum. (1998). *Proyecto intellect. Medición del capital intelectual.*
- Evangelista, P., Esposito, E., Lauro, V., & Raffa, M. (2010). The Adoption of Knowledge Management Systems in Small Firms. *Electronic Journal of Knowledge Management*, 8(1), 33–42.

- Falbo, R., Borges, L., & Rosa, F. M. (2004). Using knowledge management to improve software process performance in a CMM level 3 organization. In *Proceedings of the Fourth International Conference on Quality Software* (pp. 162–169).
- Fenton, N. E., & Neil, M. (1999). Quantitative analysis of faults and failures in a complex software system. *IEEE Transactions on Software Engineering*, 25(5), 675–689.
- Garcia, M. (2013). El rol de las tecnologías de la información y comunicación en la gestión del conocimiento : un desafío estratégico en el nuevo contexto empresarial Challenge in the New Business Environment. *Ciencias Sociales*, XIX(2), 322–333.
- Garrido, P. J., Vizcaino, A., Andrada, J., Monasor, M. J., & Piattini, M. (2012). DPMTool: A Tool for Decisions Management in Distributed Software Projects. *2012 IEEE Seventh International Conference on Global Software Engineering Workshops*, 22–27. doi:10.1109/ICGSEW.2012.7
- Gaviria, M., Mejía, A., & Henao, D. (2007). Gestión del conocimiento en los grupos de investigación de excelencia de la Universidad de Antioquia. *Revista Interamericana de Bibliotecología*, 30(2), 137–163.
- Gestión, A., Conocimiento, D. E. L., El, E. N., Vizcaíno, A., Soto, J. P., García, F., ... Piattini, M. (2006). Aplicando Gestión del Conocimiento en el Proceso de Mantenimiento del Software. *Inteligencia Artificial*, 91–98.
- Gómez, D. R. (2006). Modelos para la creación y gestión del conocimiento : una aproximación teórica, 25–39.
- González, J. (2009). Modelo para el desarrollo de la gestión del conocimiento en los centros de investigación de las universidades públicas colombianas. Caso aplicativo universidad pedagógica y tecnológica de colombia (UPTC). *Gestión y Estrategia*, 3(5), 47–62.
- Gradillas Leverté, M. (2001). Propuesta para la formulación de una estrategia de gestión del conocimiento. Retrieved from <http://www.gestiondelconocimiento.com/web/gestion-del-conocimiento/documentos;jsessionid=D4C2DCBA16C49A9E07B81067644420BB>
- Grant, R. M. (1991). The resource-based theory of competitive advantages: implications for strategy formulation. *California Management Review*, 33(3), 114–135. doi:10.2307/41166664

- Hazzan, O., & Hadar, I. (2008). Why and how can human-related measures support software development processes? *Journal of Systems and Software*, 81(7), 1248–1252. doi:10.1016/j.jss.2008.01.037
- Heredia, A., Garcia, J., Amescua, A., & Sanchez, M. (2013). Interactive Knowledge Asset Management : Acquiring and Disseminating Tacit Knowledge. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*, 147, 133–147.
- IEEE Std 610.12. (1990). IEEE Standard Glossary of Software Engineering Terminology. *IEEE standar*, 1–184. doi:10.1109/IEEESTD.1990.101064
- Iversen, J. H., Mathiassen, L., & Nielsen, P. A. (2004). Risk in Software Process An Action Improvement : *Management Information Systems*, 28(3), 395–433.
- Kahn, U. (1994). Managing knowledge-based systems development using standard life-cycle techniques. In *Managing Expert Systems* (pp. 120–160).
- Kebede, G. (2010). Knowledge management: An information science perspective. *International Journal of Information Management*, 30(5), 416–424. doi:10.1016/j.ijinfomgt.2010.02.004
- Kerzner, H. (2001). Strategic Planning for Project Management Using a Project Management Maturity Model. In *John Wiley & Sons* (p. 272).
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering.
- Kruchten, P., Ahlqvist, S., & Stefan, B. (2001). User interface design in the rational unified process. In *Object modeling and user interface design* (pp. 161–196).
- Lee, H., & Choi, B. (2002). Knowledge Management Enablers, Processes, and Organizational Performance: An Integration and Empirical Examination. *Journal of Management Information Systems*, 20(1), 179–228.
- León, N. E., Gómez, L. C., & Pimentel, J. I. (2011). Computational Tool for Management and Assessment of Software Projects Under Research Activities. *Scientia*, (47), 141–146.
- Lewin, K. (1947). Normative Action Research. *Organization Studies*, 13, 19–34. doi:10.1177/001872674700100103

- Liao, S. (2003). Knowledge management technologies and applications—literature review from 1995 to 2002. *Expert Systems with Applications*, 25, 155–164. doi:10.1016/S0957-4174(03)00043-5
- Lindner, F., & Wald, A. (2010). Success factors of knowledge management in temporary organizations. *International Journal of Project Management*, 1–12. doi:10.1016/j.ijproman.2010.09.003
- Manzanares, D., & Gómez, G. (2010). ESTRATEGIA DE GESTION DEL CONOCIMIENTO Y ACTITUD INNOVADORA EN EMPRESAS DE CASTILLA-LA MANCHA. UN ESTUDIO EXPLORATORIO. *Investigaciones Europeas de Dirección y Economía de la Empresa*, 16(1), 31–54. doi:10.1016/S1135-2523(12)60002-1
- Martín, D., Guzmán, J. G., Urbano, J., & Llorens, J. (2012). Patterns as objects to manage knowledge in software development organizations. *Knowledge Management Research & Practice*, 10(3), 252–274. doi:10.1057/kmrp.2012.15
- Mathiassen, L., Pourkomeylián, P. (2003). Managing knowledge in a software organization. *Journal of Knowledge Management*, 7, 63–80.
- Mejía, M. (2007). Estructura organizativa de los grupos de investigación de la Universidad de Antioquia como fuente de creación de conocimiento * Resumen. *Revista Interamericana de Bibliotecología*, 30(2), 89–112.
- Morales, L., Alexander, J., Luzardo, G., & Lourdes, M. (2007). Administración de proyectos en ingeniería del software. *Telos*, 9(1), 26–41.
- Nonaka, I., Krogh, georg von, & Voelpel, S. (2006). Organizational Knowledge Creation Theory: Evolutionary Paths and Future Advances. *Organization Studies*, 27(8), 1179–1208. doi:10.1177/0170840606066312
- Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company. How Japanese Companies Create the Dynamics of Innovations*. (Nueva York, Ed.). Oxford University Press.
- Oktaba, H., Piattini, M., Pino, F. J., Orozco, M., & Alquicira, C. (2009). *Competisoft. Mejora de procesos software para pequeñas y medianas empresas y proyectos* (p. 248).
- Palma, J. T., Paniagua, E., Mart, F., & Mar, R. (2000). Ingeniería del conocimiento. de la extracción al modelado de conocimimientto. *Revista Iberoamericana de Inteligencia Artificial*, 46–72.

- Petter, S., Mathiassen, L., & Vaishnavi, V. (2007). Five Keys to Project. *IT Professional*, 9(3), 42–46. doi:10.1109/MITP.2007.44
- Pfleeger. (2001). *Software Engineering Theory And Practice* (p. 652).
- Pino, F. J., Piattini, M., & Oktaba, H. (2006). Revisión Sistemática de Mejora de Procesos Software en Pequeñas y Medianas Empresas de Software.
- Pinto, L., Sierra, L., & Gomez, L. (2012). Importancia de la gestión del conocimiento en la ingeniería de software Luis Fernando Sierra Joya Consejero de la Facultad ; In *Latin American and Caribbean Conference* (pp. 1–8).
- PMBOK. (2008). *Project Management Body Knowledge". Project Management Institute, fourth edition.*
- PMI. (2012). *Project Management Body Knowledge". Project Management Institute, fourth edition.*
- Portillo, J., Vizcaíno, A., Piattini, M., & Beecham, S. (2012). Tools used in Global Software Engineering: A systematic mapping review. *Information and Software Technology*, 54(7), 663–685. doi:10.1016/j.infsof.2012.02.006
- Pressman, R. (2002). *Ingeniería de Software. Un enfoque práctico. Quinta edición.* Mc Graw Hill.
- Pressman, R. (2010). *Ingeniería de Software. Un enfoque práctico. Séptima edición.* Mc Graw Hill.
- Quintas, P., Lefrere, H., & Jones, G. (1997). Knowledge management: A strategic agenda. *Long Range Planning*, 30(3), 385–391.
- Richardson, I., Casey, V., McCaffery, F., Burton, J., & Beecham, S. (2012). A Process Framework for Global Software Engineering Teams. *Information and Software Technology*, 54(11), 1175–1191. doi:10.1016/j.infsof.2012.05.002
- Riesco. (2006). *El negocio es el conocimiento* (p. 280).
- Rose, J., Pedersen, K., Hosbond, J., & Krammergaard, P. (2007). Management competences, not tools and techniques: A grounded examination of software project management at WM-data. *Information and Software Technology*, 49(6), 605–624. doi:10.1016/j.infsof.2007.02.005

- Rus, Ioana, & Lindvall, M. (2002). Knowledge Management in Software Engineering. *IEEE Software*, 19(3), 26–38.
- Rus, Irona, & Maryland. (2002). Knowledge Management in Software Engineering. *Knowledge Management*.
- Saint-Onge, H. (1996). “*Tacit Knowledge: The Key to Strategic Alignment of Intellectual Capital*.”
- Sampieri, H., Fernández, C., & Baptista, M. (2010). *Metodología de la investigación* (Quinta edi.). McGRAW_HILL.
- Sommerville, I. (2005). *Ingeniería del Software. Séptima edición* (Septima., p. 687). Pearson Educación.
- Sveiby, K. (2001). (2001). “*The invisible Balanced Sheet*.”
- SWEBOK. (2004). *Guide to the Software Engineering Body of Knowledge. A project of the IEEE Computer Society Professional Practices Committee, IEEE Computer Society*.
- Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management*, 14(3), 207–222. doi:10.1111/1467-8551.00375
- Vargas, G. (2003). *Tratado de epistemología* (pp. 1–303). San pablo.
- Vizcaíno, A., Soto, J. P., García, F., Ruiz, F., & Piattini, M. (2006). Aplicando Gestión del Conocimiento en el Proceso de Mantenimiento del Software. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 31(31), 91–98.
- Ward, J., & Aurum, A. (2004). Knowledge Management in Software Engineering – Describing the Process. In *Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04)* (pp. 137 – 146). Australian. doi:10.1109/ASWEC.2004.1290466
- Wiig, K. (1997). Integrating intellectual capital and knowledge management. *Long Rang Planning*, 30(3), 339–405. doi:http://dx.doi.org/10.1016/S0024-6301(97)90256-9

Zack, M., McKeen, J., & Singh, S. (2009). Knowledge management and organizational performance: an exploratory analysis. *Journal of Knowledge Management*, 13(6), 392–409. doi:10.1108/13673270910997088

Zhen, L., Wang, L., & Li, J.-G. (2013). A design of knowledge management tool for supporting product development. *Information Processing & Management*, 49(4), 884–894. doi:10.1016/j.ipm.2013.01.008