Tenth LACCEI Latin American and Caribbean Conference (LACCEI'2012), Megaprojects: Building Infrastructure by fostering engineering collaboration, efficient and effective integration and innovative planning, Panama City, Panama.

Modelo de Gestión de Amenazas en Proyectos de Software

Diana Leonor Tinjacá Rodríguez
Universidad Distrital "Francisco José de Caldas, Bogotá, Colombia, dltinjacar@udistrital.edu.co

Víctor Hugo Medina García
Universidad Distrital "Francisco José de Caldas, Bogotá, Colombia, vmedina@udistrital.edu.co

Germán Andrés Méndez Giraldo
Universidad Distrital "Francisco José de Caldas, Bogotá, Colombia, gmendez@udistrital.edu.co

RESUMEN

Este articulo expone un modelo para seleccionar y priorizar en ambientes de incertidumbre un conjunto de amenazas criticas en un proyecto de desarrollo de software y así minimizar los riesgos y aumentar el nivel de cumplimiento de los requerimientos del usuario. El modelo propuesto está basado en los postulados de la técnica de inteligencia artificial *"Rough Set*" y permite obtener, desplegar y utilizar el conocimiento adquirido por el equipo de trabajo a lo largo del proyecto caracterizando, clasificando y reduciendo de manera iterativa e incremental las posibles amenazas al cual puede estar expuesto. El modelo fue validado con proyectos de entidades estatales del distrito evidenciando una mejora progresiva del 16,48% en el cumplimiento de los requerimientos del usuario.

Palabras claves: Proyecto, Riesgo, Software, Incertidumbre, Discernibilidad.

ABSTRACT

This article presents a model for selecting and prioritizing in environments of uncertainty, a set of critical threats in a software development project and so, to minimize risks and to increase the level of performance from the user's requirements. The proposed model is based on the principles of artificial intelligence technique "*Rough Set*" and allows to obtain, to spread and to use the acquired knowledge by the team of work throughout the project characterizing, classifying and reducing in a recurrent and incremental way the possible threats to which the project may be exposed. The model was validated with state agencies projects in the city showing a progressive improvement of 16.48% in the performance from the user's requirements.

Keywords: Project, Risk, Software, Uncertainty, Discernment.

1. INTRODUCCIÓN

Hoy en día las Organizaciones se encuentran inmersas en un torbellino de escenarios inciertos por efecto de la globalización y la constante reestructuración económica, social, política y organizacional que exigen su entorno local o internacional (Calvo, 2009), (Beck, 1998). Al igual que las organizaciones, los proyectos de desarrollo de software que estas lideran, son sistemas complejos dinámicos y cambiantes (Holland, 2004), (Denhardt, 2008), (Serlin). La complejidad se denota porque en el desarrollo del proyecto interactúan un número considerable de variables divergentes como el recurso humano, tiempo, presupuesto, alcance y método de desarrollo, los cuales deben coordinarse para lograr suplir las necesidades y requerimientos que los usuarios o interesados demandan (PMI, 2008), (Schwalbe, 2006).

Los proyectos como sistemas complejos se van modificando en forma dinámica de acuerdo a las circunstancias o sucesos temporales que pueda ocurrir en el desarrollo del mismo, y se ven expuestos a posibles eventos no deseados o amenazas que pueden afectar el desarrollo del mismo, creando así incertidumbre y caos por el desconocimiento de lo que pueda suceder en el futuro (Harvard Business Review. 1999). En un proyecto de

.

desarrollo de software es casi imposible encontrar eventos determinísticos, ya que el gestionarlo implica la ejecución de una serie de acciones impredecibles, aleatorias e inciertas que escapan muchas veces de las intenciones iniciales.

Existen innumerables estudios que exponen las amenazas más comunes y con mayor probabilidad de ocurrencia en proyectos de desarrollo de software, sin embargo uno de los principales inconvenientes para los gestores o líderes de este tipo de proyectos es identificar y seleccionar el conjunto mínimo de amenazas en ambientes de incertidumbre que se deben tratar en un momento dado para mitigar los efectos de estos eventos en caso de que se materialicen.

La Gestión del Conocimiento se presenta como una estrategia que permite tratar los sistemas complejos dinámicos bajo escenarios de incertidumbre ya que es disciplina que permite obtener, desplegar y utilizar el conocimiento adquirido por un grupo de personas frente a un contexto dado, recurriendo a la experiencia y enfrentarse a un entorno cambiante y turbulento (Peña, 2006). Un sistema de conocimientos es un sistema capaz de simular el conocimiento humano y su razonamiento, aproximando la capacidad humana para resolver problemas (Barr et al., 1981).

Dentro de los sistemas basado en conocimiento se distinguen un subconjunto de sistemas de inteligencia artificial los cuales incorporan algún tipo de "inteligencia" mediante el uso de técnicas derivadas de búsqueda inteligente, la representación del conocimiento, descubrimiento de conocimiento, entre otros (Aznar, 2005).

Las clases de problemas a los que da respuesta la inteligencia artificial incluyen la inferencia basada en el conocimiento, el razonamiento de información incierta o incompleta, las diversas formas de percepción y aprendizaje, y las aplicaciones a problemas tales como el control, predicción, clasificación y optimización (Toshinori, 2008).

Este articulo expone un modelo apoyado en la técnica de inteligencia artificial "*Rough set*" para seleccionar el conjunto de amenazas criticas en las cuales se deben enfocar los gestores de proyectos de desarrollo de software, para mitigar los riesgos y dar cumplimiento a los requerimientos y necesidades demandados por el usuario en este tipo de proyectos.

2. FUNDAMENTOS DE LA TEORÍA ROUGH SET

La técnica *"Rough Set"* se basa en la suposición de que con todo objeto X del universo U que está considerando se puede asociar alguna información (datos, conocimiento), expresado por medio de atributos que describen el objeto (Pawlak, 1982a), (Pawlak, 1982b), (Mara).

"Rough" se traduce como "vago, impreciso", es decir que rough set es un conjunto de objetos que, en general, no pueden ser caracterizados de manera precisa en términos de la información disponible, consiste entonces en un conjunto de objetos descrito por otro conjunto, en este caso, de atributos.

2.1. SISTEMA DE INFORMACIÓN Y SISTEMA DE DECISIÓN

La teoría del *"Rough Set"* asume la representación del conocimiento de los objetos en forma de una tabla de información, que es un caso especial de un sistema de información. En las filas de la tabla se indican los objetos (acciones, alternativas, eventos, candidatos, pacientes, empresas, etc.), mientras que las columnas corresponden a los atributos (característica, variables, condiciones, indicadores, etc.). Las entradas en la tabla son los valores del atributo (Pawlak, 1982a).

Definición 1: Sistema de Información y Sistema de decisión. Sea un conjunto de atributos $A=a_1,a_2,\ldots a_n$ y un conjunto U no vacío llamado universo (Objetos, entidades, situaciones o estados) descritos usando los atributos a_i, al (U,A) se le denomina Sistema de Información. Si cada elemento de U se le agrega un nuevo atributo d llamado decisión, indicando la decisión tomada en ese estado o situación, entonces se obtiene un Sistema de decisión $(U,Au(d))$ donde d no pertenece A.

.

El "valor de la decisión" puede representar un valor cualitativo o cuantitativo con el cual se clasifica el objeto. Con el atributo de decisión d es posible realizar una clasificación del universo de objetos U. Sea el conjunto $V_d = (1, ..m)$ un conjunto de enteros, entonces $(X_1, .. X_n)$ es una colección de clases de equivalencia, llamadas clases de decisión, en donde dos objetos pertenecen a la misma clase si tiene el mismo valor para el atributo de decisión.

2.2. RELACIÓN DE INSEPARABILIDAD O DISCERNABILIDAD

La inseparabilidad significa que dos elementos son separables si el valor de un atributo dado para ellos es diferente, en caso contrario los elementos son inseparables (Mara).

Definición 2: A cada subconjunto de atributos B de A , $B \subset A$ está asociada una relación binaria de inseparabilidad denotada por R, la cual es el conjunto de pares de objetos que son inseparables uno de otros por esta relación.

Es posible construir una matriz donde se indique la separabilidad en un conjunto utilizando los datos de la tabla de información. En la matriz de discernibilidad se tiene un índice por cada combinación o de dos objetos del sistema de información. Este índice contiene la lista de atributos por cada par de objetos que tienen diferentes valores. Cada atributo en esta lista puede discernir entre estos dos objetos (Pawlak, 1982a).

Una relación de inseparabilidad que sea definida a partir de formar subconjuntos de elementos de U que tienen igual valor para un subconjunto de atributos de atributos B de A (B), es una relación de equivalencia que es reflexiva, simétrica y transitiva.

2.3. CONJUNTOS APROXIMADOS SUPERIOR E INFERIOR

Sea el sistema de información (U , A), (U = Universo, A = Atributos) y los conjuntos (B) y (X), donde B es un subconjunto de atributos A y X es un subconjunto de elementos de U. Se puede aproximar X usando solamente la información contenida en B construyendo dos conjuntos llamados Aproximación Inferior y Aproximación Superior (Pawlak, 1982a), (Mara).

La aproximación inferior $B(X)_{inf}$ de un conjunto se define como la colección de casos cuyas clases de equivalencia están contenidas completamente en el conjunto X, mientras que la aproximación superior $B(X)_{supe}$ se define como la colección de casos cuyas clases de equivalencia están al menos parcialmente contenidas en el conjunto X, es decir, hay intersección diferente de vacío. La diferencia de estas dos aproximaciones es la región duda o frontera.

2.4. REGLAS DE DECISIÓN

La Teoría *Rough Set* analiza las decisiones pasadas (experiencia histórica) de un determinado decisor de manera cuantitativa, para basándose en esas decisiones, explicitar reglas. Estas reglas constituyen la esencia de las decisiones pasadas y contribuyen a objetivizarlas. Las reglas obtenidas pueden ayudar a tomar decisiones futuras y se estructuran en forma de sentencias lógicas (si <condiciones> entonces <clase de decisión>) (Pawlak, 1982a).

3. MODELO PROPUESTO GESTIÓN DE AMENAZAS PARA PROYECTOS DE DESARROLLO DE SOFTWARE - "APOLÍNEO"

El nombre "APOLÍNEO" viene del latín Apollineus expresan orden, armonía y equilibrio atribuidos a Apolo, dios Griego. Es un concepto que se contrapone al caos, el desorden y la irracionalidad. El modelo "APOLINEO" busca

.

reducir la incertidumbre y la irracionalidad en el proceso de selección y priorización de amenazas (eventos no deseados) en proyectos de desarrollo de software y de manera iterativa e incremental lograr un equilibrio para mitigar los riesgos asociados al proyecto y dar cumplimiento a los requerimientos del usuario.

Tomando en cuenta que las actividades fundamentales que se deben tomar en cuenta en la gestión de riesgos en proyectos de desarrollo de software son la identificación, evaluación, tratamiento y control de las amenazas o eventos no deseados (PMI, 2008), (Airley, 2009), (AS/NZS, 2004), (Magerit, 2006), (Mcmsnus, 2004), (Pandian, 2007)], el modelo Apolíneo propone los siguientes objetivos:

1) Gestionar la incertidumbre en la identificación y selección de amenazas (eventos no deseados) que afecta un proyecto de desarrollo de software, mediante la aplicación de los postulados de la teoría de conjuntos aproximados.
2) Analizar, evaluar, priorizar y controlar los riesgos asociados a los proyectos de desarrollo de software con base en el reducto de amenazas (eventos no deseados) obtenidos en la gestión de la incertidumbre.
3) Identificar y validar reglas o patrones de comportamiento de las amenazas (eventos no deseados) mediante la evaluación iterativa del cumplimiento de los requerimientos del proyecto.

Figura 1: Modelo Apolíneo. Fuente: Autores

Este trabajo se centra en describir el proceso requerido para gestionar la incertidumbre en la selección de las amenazas del proyecto basado en los postulados de la técnica "*Rough Set*". El proceso contempla los siguientes pasos:

3.1. ESTRUCTURAR EL SISTEMA DE INFORMACIÓN Y DECISIÓN

Las filas de la tabla de información están conformadas por el grupo total de amenazas o eventos no deseados a los cuales puede estar expuesto el proyecto de desarrollo de software, las columnas corresponden los atributos o las características que describen las amenazas. Las entradas en la tabla son la valoración numérica que el líder o gestor del proyecto le proporcione a cada uno de los atributos y en la columna final se encuentra el atributo de decisión o clase (inferior, superior o duda) al cual pertenece la amenaza según el conocimiento del gestor del proyecto.

La clase superior es el conjunto de amenazas que con seguridad no deben considerarse en la gestión del riesgo del proyecto, la clase inferior, es el conjunto de eventos seleccionados y con seguridad deben considerarse en el proceso de gestión del riesgo del proyecto y la clase duda será el conjunto de eventos que no pueden clasificarse con certeza en ninguna de las dos categorías descritas anteriormente.

.

En la Tabla 1 se presenta un ejemplo de un Sistema de Información para seleccionar las amenazas de un proyecto de desarrollo de software.

Tabla 1. Sistema de Información de un proyecto de desarrollo de software

U		Atributos				Decisión
Amenazas		Intesidad	Visibilidad	Velocidad	Conocimiento	Clase
1	Resistencia al cambio por parte del usuario	3	3	3	2	Duda
2	Conflicto entre los usuarios	2	3	2	2	Superior
3	Inexperiencia del usuario	1	1	2	2	Inferior
4	Usuarios no comprometidos con el proyecto	1	1	1	1	Inferior
5	Falta de participación de los usuarios	3	1	3	1	Inferior
6	Cambios continuos de requerimientos	2	2	2	2	Inferior
7	Definición inadecuada de requerimientos	3	3	3	3	Inferior
8	Cambios de última hora en los requerimientos	3	3	3	3	Inferior
9	Planificación deficiente del proyecto	1	1	1	1	Superior
10	Control insuficiente al avance del proyecto	2	2	2	2	Superior

Los elementos que constituyen el Sistema de Información de la tabla 2, son:

a) El conjunto de objetos o amenazas: $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Las amenazas relacionadas en la tabla 1 son ejemplo de numerosos estudios hasta la fecha (Boemh, 1992), McConell, 1998), (Wallace et al. 2004).

b) Conjunto de atributos

$$C = \{Intensidad, Invisibilidad, Velocidad, Conocimiento\}$$

El atributo intensidad describe la determinación de la amenaza por perseverar su objetivo, es decir el grado de fuerza que tiene frente a otros eventos para que se materialice. El atributo de invisibilidad describe la capacidad del evento no deseado para mantener en secreto la consecución de su objetivo es decir no es tan evidente su manifestación si no hasta que ya se está materializando. El atributo velocidad cuantifica el período de tiempo en que un evento puede ser capaz de materializarse y el atributo conocimiento de evento no deseado se refiere al nivel de experiencia y conocimiento que tiene el equipo de trabajo con respecto al evento.

Cada uno de estos atributos puede tener un valor de bajo (1), medio (2) o alto (3) según las condiciones del proyecto.

c) Atributo de decisión $D = \{Superior, Duda, Inferior\}$

3.2. REDUCCIÓN DE ATRIBUTOS DE ACUERDO A LA RELACIÓN DE INSEPARABILIDAD

Una vez se cuente con el Sistema de Información, es preciso evaluar y obtener la función de inseparabilidad de los atributos a partir de la matriz de discernabilidad.

Para el ejemplo de la tabla 1, la matriz de discernibilidad es una matriz simétrica nxn que muestran las relaciones entre atributos con base en las posibles combinaciones de las amenazas que hacen parte del Sistema de Información, expresando 0 cuando la relación entre atributos es diferente y 1 cuando la relación es igual, fruto de este ejercicio se obtiene la función de discernibilidad o expresión boleaba que expresa las diferentes relaciones entre los atributos del sistema de información.

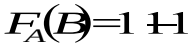
.

Tabla 2. Matriz de discernibilidad

AMENAZAS		*Intesidad*	*Visibilidad*	*Velocidad*	*Conocimiento*
1	2	0	1	0	1
1	3	0	0	0	1
1	4	0	0	0	0
1	5	1	0	1	0
1	6	0	0	0	1
1	7	1	1	1	0
1	8	1	1	1	0
1	9	0	0	0	0
1	10	0	0	0	1
1	11	0	0	0	0
2	1	0	0	1	1
2	3	0	0	0	0
2	4	0	0	0	0
2	5	1	0	1	1
2	6	0	1	0	0

La matriz de la tabla 2 se puede representar como una función de discernibilidad booleana así:

Simplificado esta expresión utilizado la técnica de simplificación booleana Quine-McCluskey se obtiene la siguiente expresión minimizada:



Esta función simplificada muestra que se reduce el atributo velocidad, esto quiere decir que es suficiente trabajar con los atributos intensidad, visibilidad y conocimiento y obviar el atributo velocidad para evaluar las amenazas del proyecto.

Así el reducto obtenido para el sistema de información de la tabla 1 es:

3.3. OBTENER CONJUNTO DE APROXIMACIÓN INFERIOR PARA SELECCIONAR AMENAZAS DEL PROYECTO

Con el reducto de atributos $IND_A(B)$ obtenido en el paso anterior se procede a seleccionar del conjunto de amenazas que deben ser consideradas por los gestores de los proyectos para ser evaluadas y tratadas. Los eventos no deseados o amenazas seleccionadas serán aquellas que pertenezcan al subconjunto de aproximación inferior y las amenazas que se descartaran serán aquellas que pertenezcan al subconjunto de aproximación superior.

Así para el caso del Sistema de Información que se presenta en la tabla 1, se obtiene los subconjuntos del conjunto de amenazas U con base en el reducto de atributos 

$X=1,6,7,8,10$, que es el conjunto formado por las amenazas clasificadas en la clase inferior, se obtienen los siguientes subconjuntos

Aproximación Inferior: $B(x)_{inf}=(1)$

Aproximación Superior: 

Región Duda: 

En este caso la amenaza que se debe concentrar el equipo de trabajo del proyecto será la amenaza Nº1 "Resistencia al cambio por parte del usuario".

.

3.4. REPETIR EL PROCESO DE MANERA ITERATIVA E INCREMENTAL

Con base en las amenazas seleccionadas y obtenidas en los pasos anteriormente descritos, el gestor del proyecto inicia la evaluación, tratamiento y control de las amenazas priorizadas y resultado de estas actividades quedaran un reducto de amenazas para seleccionar nuevamente, por lo que el proceso descrito se repita de manera iterativa e incremental a medida que se vaya desarrollando el proyecto y hasta tanto se logre reducir los riesgos críticos y alcance un alto nivel de cumplimiento en los requerimientos del usuario.

4. APLICACIÓN, VALIDACIÓN Y RESULTADOS DEL MODELO "APOLÍNEO"

Para la validación del modelo "Apolíneo" se aplico un experimento durante un periodo de 5 meses con dos proyectos (Tabla 3) que desarrollan software de manera iterativa dentro de una entidad estatal (Universidad Distrital Francisco José de Caldas). El experimento se dividió en dos fases una antes y después de la aplicación del modelo, las cuales se compararon con el fin de evaluar la relación causal entre la variable dependiente (nivel de cumplimiento de los requerimientos de los usuarios) y la independiente (conjunto de amenazas seleccionados y priorizados para la gestión de los riesgos del proyecto).

Tabla 3. Proyectos que hicieron parte de la validación del modelo "Apolíneo"

Código	Nombre del Proyecto	Nombre de la Organización	Fecha Inicial	Fecha Final
1	Sistema Bodega de Datos- Fase II	Universidad Distrital	15/07/2011	14/07/2013
2	Sistema de Gestión Académica- Fase II	Universidad Distrital	15/07/2011	14/07/2013

El experimento comparó los resultados obtenidos de 12 iteraciones (Tabla 4) definidas en los proyectos a lo largo de los 5 meses de evaluación, las primeras 6 iteraciones contemplan el tiempo en el cual se lleva a cabo una gestión de riesgos tradicional y las iteraciones restantes se aplica el modelo "Apolíneo" contando con el apoyo de los líderes de los proyectos.

Tabla 4. Relación de iteraciones definidas para los proyectos

Código del Proyecto	Nº iteración	Días Hábiles	Fecha Inicial	Fecha Final
1	1	15	15/07/2011	04/08/2011
1	2	16	05/08/2011	26/08/2011
1	3	20	29/08/2011	23/09/2011
1	4	19	26/09/2011	21/10/2011
1	5	18	24/10/2011	09/11/2011
1	6	16	10/11/2011	30/11/2011
2	1	15	15/07/2011	04/08/2011
2	2	16	05/08/2011	26/08/2011
2	3	20	29/08/2011	23/09/2011
2	4	19	26/09/2011	21/10/2011
2	5	18	24/10/2011	09/11/2011
2	6	16	10/11/2011	30/11/2011

Para este experimento se contemplaron 30 amenazas (Tabla 5) categorizadas así: Gestión de Usuarios, Gestión de Requerimientos, Planeación y Control del Proyecto, Equipo de Trabajo y Ambiente Organizacional.

.

Tabla 5. Relación de Amenazas base para el experimento

Categoría	Amenazas
Gestión de Usuarios	Resistencia al cambio por parte del usuario
	Conflicto entre los usuarios
	Inexperiencia del usuario
	Usuarios no comprometidos con el proyecto
	Falta de participación de los usuarios
Gestión de Requerimientos	Cambios continuos de requerimientos
	Definición inadecuada de requerimientos
	Cambios de última hora en los requerimientos
Planeación y Control del Proyecto	Planificación deficiente del proyecto
	Control insuficiente al avance del proyecto
	Inadecuada estimación de los recursos
	Comunicación ineficiente
	Recursos asignados insuficientes
	Gestión inadecuada o nula de los cambios
	Poca o nula claridad en el alcance del proyecto
	Cambios en el alcance del proyecto
	Conflicto de recursos con otros proyectos
	Fondos inadecuados o interrumpidos
Equipo de Trabajo	Inexperiencia de los miembros del equipo de trabajo
	Falta de habilidades y competencias de los miembros del equipo
	Déficit de personal
	Volatilidad o cambios de personal
	Falta de compromiso y motivación del equipo de trabajo
	Roles y responsabilidades no definidas completamente
Ambiente Organizacional	Cambios de dirección durante el desarrollo del proyecto
	Cambios del ambiente legal o regulatorio institucional
	Falta de compromiso y apoyo de la alta dirección en el proyecto
	Conflictos políticos
	Falta de personal de planta con conocimientos en TI

Con base en reuniones de trabajo con los líderes de los proyectos que hicieron parte de la validación de este modelo, se definieron 6 tipos de atributos: Intensidad, Visibilidad, Velocidad, Conocimiento, Control y Frecuencia.

Al final de este experimento se obtuvieron los siguientes resultados:

a. Reducto de Atributos y Amenazas: Una vez estructurado los Sistemas de Información de los proyectos y valorado la discernabilidad de los atributos (6) y las amenazas (30) que sirvieron de base para este experimento, se logro reducir los atributos y amenazas de los proyectos como se observa en la tabla 6 y 7.

Tabla 6. Reducto de atributos obtenidos en los proyectos

REDUCTO DE ATRIBUTOS							
Nº Proyecto	Nº Iteración	Reducto de Atributos					
		Intesidad	Visibilidad	Velocidad	Conocimiento	Control	Frecuencia
1	4	SI	SI	NO	SI	SI	NO
2	4	SI	SI	SI	SI	SI	SI
1	5	SI	NO	NO	SI	SI	NO
2	5	NO	SI	SI	SI	SI	NO
1	6	SI	NO	NO	SI	SI	NO
2	6	NO	SI	SI	NO	NO	NO

.

Tabla 7. Reducto de amenazas obtenidas en los proyectos

Nº Proyecto	Nº iteración	Evento Seleccionados
1	4	Resistencia al cambio por parte del usuario
1	4	Cambios continuos de requerimientos
1	4	Control insuficiente al avance del proyecto
1	4	Comunicación ineficiente
1	4	Gestión inadecuada o nula de los cambios
2	4	Conflicto entre los usuarios
2	4	Inexperiencia del usuario
2	4	Usuarios no comprometidos con el proyecto
2	4	Falta de participación de los usuarios
2	4	Cambios continuos de requerimientos
2	4	Comunicación ineficiente
2	4	Poca o nula claridad en el alcance del proyecto
2	4	Cambios en el alcance del proyecto
2	4	Inexperiencia de los miembros del equipo de trabajo

b. Nivel de Cumplimiento de Requerimientos de los Proyectos

Una vez aplicados los controles correspondientes en las diferentes iteraciones a el conjunto de amenazas seleccionadas, se evidencio una mejora iterativa e incremental del 16,48% en el cumplimiento de los 73 requerimientos de los proyectos que fueron objeto de estudio como se puede apreciar en la tabla 8. La escala de evaluación de los requerimientos oscila entre 1 y 5, siendo el 5 un requerimiento cumplido y 1 un requerimiento no considerado en la iteración.

Tabla 8. Nivel de cumplimiento de Requerimiento de los proyectos

	Fase 1 – Sin Modelo			Fase 2- Con Modelo		
	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5	Iteración 6
Proyecto 1	3.38	3.25	2.71	3.32	3.75	4.00
Proyecto 2	3.00	3.25	2.33	3.80	3.63	3.00
Promedio	2.99			3.58		
Diferencia	16.48					

c. Base de Conocimientos

En el proceso de la aplicación del modelo "Apolíneo" se pudieron obtener aproximadamente 20 reglas de conocimiento que sirven de referencia para cualquier proyecto de desarrollo de software, algunas de estas se encuentra relacionadas en la tabla y se leen de la siguiente manera, **SI** la intensidad, la visibilidad, la velocidad y la frecuencia es alta (3) y el conocimiento y control es medio (2) ENTONCES el proyecto es propenso a la amenaza "Resistencia al cambio por parte del Usuario"

Tabla 9. Reglas de conocimiento obtenidas en el proceso de validación

Nº iteración	Nº proyecto		Atributos Descriptores							Amenazas
			Intesidad	Visibilidad	Velocidad	Conocimiento	Control	Frecuencia		
4	1	**IF**	3	3	3	2	2	3	**THEN**	Resistencia al cambio por parte del usuario
4	1	**IF**	2	2	2	2	2	2	**THEN**	Cambios continuos de requerimientos
4	2	**IF**	2	2	3	1	2	1	**THEN**	Usuarios no comprometidos con el proyecto
4	2	**IF**	2	2	2	3	2	2	**THEN**	Falta de participación de los usuarios
4	2	**IF**	2	2	2	2	2	2	**THEN**	Cambios continuos de requerimientos
4	2	**IF**	2	2	2	2	2	2	**THEN**	Inexperiencia de los miembros del equipo de trabajo

.

4. CONCLUSIONES

Los proyectos de desarrollo de software se caracterizan por estar inmersos en ambientes de incertidumbre y estar sujetos a un conjunto de amenazas impredecibles que pueden afectar su exitoso desarrollo.

La técnica de inteligencia artificial *"Rough Set"* base del modelo "Apolíneo" permite reducir la incertidumbre de seleccionar y priorizar un conjunto de amenazas que los gestores de los proyectos de desarrollo de software deben tratar para mitigar los riesgos asociados al proyecto.

El modelo "Apolíneo" permite gestionar el conocimiento adquirido por los gestores y lideres de proyectos de desarrollo de software, ya que permite clasificar, representar, organizar, almacenar y compartir la información que generan las amenazas asociadas a lo lardo del proyecto.

La validación del modelo "Apolíneo" arrojo una mejora progresiva e incremental en el nivel de cumplimiento de los requerimientos o necesidades del usuario en los proyectos que hicieron parte del experimento.

Si los gestores o líderes de los proyectos de desarrollo de software enfocan su esfuerzo en un conjunto reducido de amenazas, podrán aplicar controles efectivos para minimizar los efectos de los riesgos asociados al proyecto y dar cumplimiento a las expectativas de los usuarios o interesados.

REFERENCIAS

Airley, Richard. (2009). Managing and Leading Software Projects .
Aznar, Fidel. (2005). Fundamentos de Inteligencia Artificial.
AS/NZS 4360:2004.. (2004). Risk Management Standard.
Barr & Feigenbaum. (1981). "The Handbook of Artificial Intelligence", Vol. I.
Beck,Ulrich. (1998). La sociedad del riesgo global: Hacia una nueva modernidad.
Boemh, B. (1992). "Software Risk Management: Principles and Practices".
Calvo, Juan. (2009). Globalización.
Denhardt, R. (2008).Teoria de la Administración pública: El estado de la disciplina.
Harvard Business Review. (1999).La Gestión en la Incertidumbre.
Holland, Jhon. (2004). El orden oculto de como la adaptación crea la complejidad.
Mara, C. "Rough Sets- Técnica de reducción de Atributos y generación de reglas para clasificación de datos".
McConell, Steve. (1998). Software Project Guide.
Magerit. (2006). Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información - Método.
Mcmsnus, John. (2004). Risk Management in Software Development Projects.
Pawlak, A. (1982). Rough sets.
Pawlak, A. (1994). Rough membership function.
Peña, Alejandro. (2006). Sistemas Basados en Conocimiento: Una base para su concepción y Desarrollo.
PMI. (2008). Project Management Institute. Guía de los Fundamentos para la Dirección de Proyectos (PMBOK).
Pandian C. Ravindranath. (2007). Applied software risk management: a guide for software project managers.
Schwalbe,k. (2006).Information Tecnology Project Management.
Serlin, José. Conocimiento de la Gestión de las Organizaciones: Sistemas Complejos dinámicos adaptativos.
Toshinori, M. (2008). Fundamentals of the New Artificial Intelligence.
Wallace and M. Keil. (2004). "Software Project Risk and their Effect on Outcomes".

Autorización y Renuncia

Los autores autorizan a LACCEI para publicar el escrito en las memorias de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito