

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 4/07/2012 | Aceptado : 10/09/2012

Sistema inteligente de mitigación de riesgos

Intelligent system for risk mitigation

Yoanny Torres Rubio^{1*}, Dasiel Cordero Morales², Yadira Ruiz Constanten²

¹ Centro de Telemática. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370

² Centro de Informatización de la Seguridad Ciudadana. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370

*Autor para la correspondencia: ytrubio@uci.cu

Resumen: la acumulación de experiencias en la concepción de proyectos, el surgimiento de organizaciones y estándares dedicados a la planificación y administración de los mismos, han permitido que su desarrollo se haya convertido en una actividad planificada y controlada. La gestión de riesgos durante el ciclo de vida de desarrollo de software es un proceso complejo estrechamente vinculado al dominio que tenga el equipo de desarrollo sobre el tema. De su correcta gestión dependerán gran parte de los resultados, teniendo en cuenta las ventajas que tanto tecnológicas como económicas reportará para el equipo de desarrollo la mitigación de los mismos. Los riesgos han de analizarse para propiciar el aprovechamiento de las diversas oportunidades que puedan ofrecer y de la misma forma, evitar que sean muy severos e irreparables, los daños que puedan provocar. Los sistemas basados en casos ayudan y agilizan la toma de decisiones simulando las cadenas de razonamiento que realiza un experto para resolver un problema de su dominio. Su vinculación con distintos elementos del proceso de desarrollo de software posibilita la obtención de resultados más acertados a partir del conocimiento que se les introduce basado en la experiencia acumulada. Actualmente el centro de Informatización para la Seguridad Ciudadana ISEC, ha enfrentado dificultades para una rápida y acertada gestión de riesgos. La utilización de las facilidades que brindan los sistemas basados en casos para la gestión de riesgos en un proyecto de desarrollo de software, resultaría ventajoso al brindar las herramientas necesarias para que los líderes de proyecto, basados en experiencias anteriores, realicen una planificación más acertada, teniendo en cuenta los diversos contratiempos que pudieran surgir.

Palabras clave: gestión, mitigación, riesgos, sistema basado en el conocimiento, sistema basado en casos.

Abstract: *the accumulation of experiences in project design, and the emergence of standards organizations and engaged in planning and administration of the same, have allowed their development has become an activity planned and controlled. Risk management during the life cycle of software development is a complex process closely linked to the domain with the development team on the subject. In its correct management will depend largely on the results, taking into account both technological advantages for the team will bring economic development, mitigating them. To have a tool that automates this, streamline the decision-making process, is essential. In Artificial Intelligence there are different knowledge-based techniques that allow solution of problems in different ways. One of the most common techniques is knowledge Based Systems, which are computer systems that use domain knowledge to solve problems pertaining to this. This solution is essentially the same as that obtained by a person experienced in the problem*

domain. It is of great importance to use the advantages provided by this type of systems for developing applications to streamline the process of managing risks. To this will be discussed in this article, showing that facilitates the use of minimal resources and higher quality.

Keywords: *management, mitigation, risk, knowledge-based system, cases-based system.*

1. Introducción

El Instituto de Gestión de Proyectos (del inglés, *Project Management Institute* PMI®) es una organización internacional sin fines de lucro que asocia a profesionales para la gestión de proyectos (PMBOK, 2008). Actualmente, es la más grande del mundo en su rubro; dado que se encuentra integrada por más de 260.000 miembros alrededor de 171 países (CIBERTEC, 2010). El mismo, basa sus ideas fundamentales en que: un proyecto es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único (García, 2009).

El desarrollo de un proyecto debe ser una actividad planificada y controlada, puesto que requiere participación humana en un tiempo dado para su cumplimiento. La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y costo definidos (Soft Expert, 2007). Teniendo en cuenta que todo reposa sobre la base de obtener un producto con calidad.

Los proyectos de desarrollo de software se diferencian de los otros proyectos de ingeniería tradicional en la naturaleza lógica del producto. El software se desarrolla, no se fabrica en un sentido clásico. La gestión del proyecto de software es el primer nivel del proceso de ingeniería de software, porque cubre todo el proceso de desarrollo. Para lograr un proyecto de software fructífero se debe comprender el ámbito del trabajo a realizar, los riesgos en los que se puede incurrir, los recursos requeridos, las tareas a llevar a cabo, el esfuerzo a consumir y el plan a seguir.

Como cualquier actividad humana, el desarrollo de un proyecto incluye la ocurrencia de riesgos, los cuales son eventos o condiciones inciertas, que si se producen, afectan de manera positiva o negativa al menos un objetivo del proyecto (Pressman, 2007). Una acertada gestión de riesgos es la mejor manera de mitigar un riesgo o un conjunto de estos, ya que permite identificar, analizar y responder a los riesgos a lo largo de la vida de un proyecto, con el propósito de aumentar la probabilidad y el impacto de los eventos positivos y disminuir la de los eventos adversos para el proyecto. Constituye, dentro del desarrollo de un proyecto de software un proceso generalmente complejo, que requiere seguimiento constante, dependiendo en gran medida de las experiencias acumuladas, el conocimiento adquirido y los estándares definidos.

La automatización de los procesos relacionados con la gestión de riesgos garantizaría la obtención de resultados que apoyen la toma de decisiones en aras de enfrentar, mitigar o aprovechar la ocurrencia de uno o varios riesgos. Posibilitando además, que el proyecto en sí se desarrolle con la mayor calidad posible, cumpliendo con los alcances definidos en el menor período de tiempo y con la utilización del mínimo de recursos.

Entre las técnicas más distintivas utilizadas actualmente para este fin se encuentran las que permiten predecir e identificar un resultado determinado o un evento que puede ocurrir simulando el razonamiento humano. A la rama de las Ciencias de la Computación dedicada al desarrollo o uso de los ordenadores, con los que se intenta reproducir los procesos de la inteligencia humana se le denomina Inteligencia Artificial (IA) (Gálvez, 1998).

La identificación y mitigación de riesgos a lo largo del ciclo de vida del software en el Centro Informatización de la Seguridad Ciudadana (ISEC), de la Facultad 2 en la Universidad de las Ciencias Informáticas (UCI), es realizado por parte de la dirección de cada proyecto, analizando y calculando según diferentes criterios de estimación y con el uso de metodologías de Gestión de Riesgos (GR). Este es un proceso largo y complejo, que se realiza de forma manual y que incluye el seguimiento y control de muchos de los integrantes del equipo de desarrollo.

Actualmente el centro no cuenta con una herramienta que permita, ayude o agilice el proceso de gestión de riesgos en el ciclo de vida de los proyectos de desarrollo de software. Este trabajo tiene como objetivo proveérsela mediante un sistema inteligente, y de esta forma automatizar este proceso de vital importancia. Todo ello con el fin de identificar los riesgos más comunes a ocurrir, según las características del proyecto en cuestión, así como posibles estrategias para su tratamiento.

La aplicación informática desarrollada, “Sistema Inteligente de Mitigación de Riesgos”, relaciona los sistemas basados en casos con los procesos incluidos en la gestión de riesgos. El objetivo de dicho sistema es aprovechar las ventajas que brindan las técnicas de IA, para facilitar el manejo de algunos de los elementos distintivos en los proyectos de desarrollo de software, que permiten identificar la ocurrencia de un determinado evento, para su aprovechamiento o mitigación.

El sistema brinda la posibilidad de que dadas características relevantes del proyecto, se logre obtener de forma automática los riesgos potenciales a incidir a lo largo del ciclo de desarrollo, así como una posible mitigación o aprovechamiento de los mismos. Lo anterior, permite tener una visión adelantada, agilizando el proceso de toma de decisiones.

2. Materiales y métodos

Visual Paradigm es una herramienta visual de ingeniería de software para el modelado, que tiene un entorno de trabajo que muestra colección de menús, barra de herramientas y ventanas que permite realizar diagramas. Puede ser utilizada por una gran variedad de usuarios como analistas de sistemas, analistas de negocios e ingenieros de software. Provee soporte para la generación de código y la ingeniería inversa. Se integra con algunas herramientas como: Eclipse, Netbeans, Jbuilder, Oracle, entre otras. Se utilizará *Visual Paradigm Suite* versión 3.4.

La plataforma Java es un entorno capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a bytecode. Provee una máquina virtual que se encarga de la ejecución de aplicaciones, un conjunto de bibliotecas estándares que ofrecen funcionalidades comunes así como una interfaz para la programación de aplicaciones haciendo uso del lenguaje Java. Esto permite que los usuarios interactúen con la máquina virtual de Java y exploten el conjunto de funcionalidades brindadas por la misma, así como sus bibliotecas. Además las aplicaciones sobre esta plataforma pueden también ser desarrolladas en la web.

Groovy es un lenguaje orientado a objetos, nacido con la misión de acoplarse de forma efectiva con Java, trabajar en conjunto con la Máquina Virtual de Java JVM y soportar los tipos de datos estándar, pero añadiendo características dinámicas y sintácticas presentes en otros lenguajes como Python, Smalltalk o Ruby (Zulueta y Despaigne, 2009). El código fuente Groovy se compila a bytecodes, y es posible instanciar objetos Java desde Groovy y viceversa. Lo que Groovy aporta es una sintaxis que aumenta enormemente la productividad. Todo lo anterior, unido a que la mayor parte de código escrito en Java es totalmente válido en Groovy hace que este lenguaje sea de muy fácil adopción.

NetBeans es un entorno de desarrollo hecho principalmente para el lenguaje de programación Java, pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y sin restricciones de uso. Incluye soporte para Groovy y Grails desde la versión 6.5, incluyendo un tipo de proyecto y posibilidad de ejecutar comandos Grails desde el propio IDE. El soporte para Groovy incluye resaltado de sintaxis y autocompletado de nombres de variables y métodos, y sigue evolucionando a muy buen ritmo. Se utilizará la versión 6.9.

Grails es un framework para aplicaciones web libre desarrollado sobre el lenguaje de programación Groovy. Grails pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como convención sobre configuración, proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de

configuración al programador.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional de código abierto. Ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características de estabilidad y rendimiento. Cuenta con interfaces para distintos lenguajes, como son: C/C++, Java, Delphi, Python, Perl, PHP, entre otros. Se utilizará PostgreSQL 8.2.

FDD es una metodología ágil para el desarrollo de sistemas a corto plazo. Se basa en un proceso de iteraciones cortas de aproximadamente dos semanas que producen un software funcional que el cliente y la dirección de la empresa, centro o proyecto pueden ver y monitorear. Estas iteraciones se deciden en base a las funcionalidades que son pequeñas partes del software con significado para el cliente. Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado y con su utilización se obtienen resultados periódicos y tangibles.

RUP es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos de software. Las características principales están dadas en que es un proceso iterativo e incremental, centrado en la arquitectura y dirigido por casos de usos. Además su desarrollo se basa en componentes, tiene un único lenguaje para modelar constituyendo un proceso integrado (BMP, 2008).

En el desarrollo de este trabajo se utiliza la metodología RUP como apoyo a la metodología FDD, seleccionada para guiar el diseño y la implementación. Se generarán algunos artefactos correspondientes al modelamiento del negocio, requerimientos, implementación y pruebas para garantizar que quede documentada la mayor cantidad de detalles posibles, para el posterior desarrollo del sistema.

Inductivo-Deductivo: Se utilizó para el planteamiento del objetivo y la extracción de las ideas fundamentales para la elaboración y fundamentación del trabajo de diploma.

Histórico-Lógico: El método histórico permitió estudiar la trayectoria real de la GR, además de herramientas inteligentes que permitan su tratamiento y acontecimientos fundamentales en el decursar de la historia de ambos. El método lógico permitió investigar las leyes generales del tema y las peculiaridades de los marcos estudiados. La utilización de este método permitió que el estudio no se limitara a una simple descripción de los hechos sino que facilitó el descubrimiento de la lógica objetiva del desarrollo histórico de la GR y su integración con herramientas inteligentes para su tratamiento.

3. Resultados y discusión

El Sistema Inteligente de Mitigación de Riesgos se basa en la información que introduce el usuario referente a la descripción de un proyecto. Utiliza técnicas de Inteligencia Artificial, específicamente de un sistema basado casos, para distinguir por rasgos la descripción dada. Luego los compara con los almacenados en la base de casos con que cuenta el sistema experto y devuelve un grupo de casos semejantes. A partir de ellos y con el uso de un razonador que forma parte también de la arquitectura de un sistema basado en casos) devuelve una respuesta semejante a la adquirida según la similitud del caso entrado con los que se tienen en la base de casos, y a partir de esta información, obtener las posibles acciones a realizar o estrategias a tomar. Y por último, adaptar la solución permitiendo la retroalimentación y aprendizaje del sistema.

Es una aplicación web, basada en modelo Cliente-Servidor desarrollada sobre la plataforma Java. Posee un diseño sencillo con una interfaz amigable. Incluye también una ayuda para facilitar el uso y comprensión del sistema por parte de los usuarios.

La solución desarrollada combina principalmente un sistema de gestión de información con las técnicas de razonamiento basado en casos. Lo primero permite el almacenamiento, inserción, actualización y eliminación de toda la información referente a los proyectos, los usuarios, además de los riesgos y su posible mitigación. La combinación con un sistema de RBC permite que, a partir de la información insertada, y las características de los proyectos, los riesgos y las clasificaciones de éstos con sus respectivas mitigaciones, se conviertan en el conocimiento almacenado para su utilización en el proceso seguido por el razonador basado en casos para la identificación de riesgos. Además facilita mostrar estos resultados de forma organizada y comprensible para el usuario facilitando su posterior análisis.

Base de casos definida y el proceso de identificación de riesgos

La base de casos cuenta con 31 rasgos, definidos según los elementos fundamentales del modelo de gestión de riesgos MOGERI. Dichos elementos constituyen rasgos de los proyectos que permiten predecir, a partir de las experiencias acumuladas, la ocurrencia de situaciones tanto adversas como favorables para un determinado proyecto de desarrollo de software. Además, se adaptaron los elementos obtenidos, a partir del estudio de otras metodologías y modelos de gestión de riesgos, formando un híbrido que se ajustara de forma correcta, con las condiciones existentes en el centro ISEC.

Las siguientes tablas definen los rasgos de la base de casos con su valor de semejanza para la comparación y el dominio sobre el que están definidos.

Tabla 1. Rasgos con 5 criterios de comparación y dominios.

	Muy Alta	Alta	Media	Baja	Ninguna
Muy Alta	1	0,7	0,4	0,1	0
Alta	0,7	1	0,6	0,2	0
Media	0,4	0,6	1	0,5	0
Baja	0,1	0,2	0,5	1	0,3
Ninguna	0	0	0	0,3	1

Los valores de comparación y los dominios anteriores, están definidos para los rasgos: modelo de proyecto, dominio del negocio, experiencia en la plataforma, nivel de conocimiento del lenguaje, las herramientas y los desarrolladores (Tabla 1).

La siguiente tabla (Tabla 2) muestra los valores y dominios correspondientes a los rasgos experiencia del equipo, la motivación, la experiencia del mismo, el nivel de dificultad del lenguaje, la cohesión del equipo, la experiencia en las herramientas y la estabilidad de los requisitos.

Tabla 2. Rasgos con 4 criterios de comparación y dominios.

Estabilidad de Requerimientos Experiencia/motivación/Dificultad lenguaje/Cohesión del equipo/ Experiencia con las herramientas	Muy Estables	Estables	Poco Estables	Inestables
	Mucha	Media	Baja	Ninguna
Mucha/Muy Estables	1	0,6	0,2	0
Media/Estables	0,6	1	0,5	0
Baja/Poco Estables	0,2	0,5	1	0,1
Ninguna/Inestables	0	0	0,1	1

La tabla que aparece a continuación (Tabla 3) expone los valores de dominio y de comparación de los rasgos complejidad de la aplicación, el tiempo promedio dedicado al proyecto, la información que se maneja, el estado del cronograma y la relación con el cliente.

Tabla 3. Rasgos con 3 criterios de comparación y dominios.

Cronograma	Muy Variable	Variable	Poco Variable
Relación con el cliente	Buena	Regular	Mala
Información que se maneja	Confidencial	Pública	Privada
Complejidad de la aplicación	Alta	Media	Baja
Alta/Confidencial/Buena	1	0,5	0
Media/Pública/Regular	0,5	1	0,2
Baja/Privada/Mala	0	0,2	1

La última tabla (Tabla 4) exhibe los datos para la comparación y el dominio del rasgo relación con el cliente.

Tabla 4. Rasgo con 1 criterio de comparación y dominio.

Relación con el cliente	Accesible	Poco Accesible
Accesible	1	0,5
Poco Accesible	0,5	1

Los rasgos que aparecen a continuación fueron definidos con un dominio continuo:

- | | |
|---|---------------------------------|
| Cantidad de profesores. | Cantidad de estudiantes de 2do. |
| Cantidad de profesores en adiestramiento. | Cantidad de estudiantes de 3ro. |
| Cantidad de profesores prestando servicios. | Cantidad de estudiantes de 4to. |
| Cantidad de funcionalidades simples. | Cantidad de estudiantes de 5to. |
| Cantidad de funcionalidades medias. | Cantidad total de PCs. |
| Cantidad de funcionalidades complejas. | Cantidad disponible de PCs. |
| | Tiempo por PC. |

Estos valores, son comparados siguiendo una estrategia de pertenencia a intervalos. Si los valores son iguales se retorna 1. De no ser iguales pero, si el módulo de la resta de ambos es igual a 2, se retorna 0.7 (lo que constituye su valor de semejanza), y si este cálculo retorna 3 o 4 entonces se devolverá 0.3. De lo contrario se asume que los valores son muy diferentes y se retorna 0.

$$F(S) \begin{cases} V1 = V2 & \delta i = 1 \\ |V1 - V2| \leq 2 & \delta i = 0.7 \\ 2 < |V1 - V2| \leq 4 & \delta i = 0.3 \\ & 0 \end{cases}$$

La identificación de riesgos es la funcionalidad de mayor relevancia en el sistema. A través de los datos del proyecto de un usuario, este último ejecuta la acción Identificar Riesgos. De esta forma, se comparan los datos entrados con los casos almacenados en la base de conocimientos a partir de las funciones de semejanza correspondientes para luego evaluar el caso. Cada dato, según su valor y dominio tendrá una función de semejanza, la cual compara dicho valor con su rasgo correspondiente en cada Caso de la Base de Conocimientos. Al terminar de analizar y comparar cada

rasgo predictor (columna de la Base de Casos o campo independiente), se procede a evaluarlo (rasgo objetivo o fila de la Base de casos). La función de evaluación correspondiente a los casos es la siguiente:

$$(E) = \frac{\sum_{i=0}^n W_i \delta_i(R1, R2)}{\sum_{i=0}^n W_i}$$

Donde W representa el peso o nivel de importancia de cada rasgo predictor y δ representa la función de semejanza definida para la comparación del caso existente (R2) con el caso nuevo (R1).

La comparación es realizada analizando cada dato insertado referente a un proyecto, con cada rasgo predictor de la base de casos retornando su valor de semejanza. Si son iguales se retornará 1 (valor que constituye la semejanza). De lo contrario, se procede a devolver el valor correspondiente a cada situación mostrada en la tabla, comparando el valor de dominio de la columna correspondiente a la variable retornando el valor de semejanza perteneciente a tal comparación. Esto será multiplicado por el peso o nivel de importancia definida para cada rasgo predictor. Evaluando el resultado con la utilización de la fórmula matemática mostrada anteriormente.

De esta forma, cada caso nuevo es comparado con los existentes y posteriormente evaluado. Este proceso se realiza de forma iterativa para todos los rasgos de la Base de casos obteniendo así el caso que exprese mayor semejanza a la descripción entrada. Luego, se obtienen los riesgos asociados a este caso y se le muestran al usuario en una página con algunos datos del proyecto y los datos de los posibles riesgos a incidir.

La necesidad de automatizar el proceso de identificación de riesgos, planteó la tentativa de utilizar la aplicación para la obtención de resultados inmediatos en el centro. De 8 grandes proyectos existentes en el mismo, la aplicación se utilizó en 5 de ellos, de los cuales en 4 ofreció buenos resultados a corto y mediano plazo, mientras que en solo uno ofreció resultados a largo plazo, dependiendo del análisis de los especialistas del proyecto.

Lo anterior facilitó aminorar el tiempo en las actividades referentes a la gestión de riesgos y de esta forma disminuir la utilización de un grupo de recursos, facilitando la culminación de diferentes fases de los proyectos según el alcance definido en el tiempo requerido. Dichos resultados hacen resaltar la importancia derivada de la utilización de esta herramienta en las etapas tempranas del ciclo de vida de desarrollo de software.

4. Conclusiones

La solución desarrollada con el nombre “Sistema Inteligente de Mitigación de Riesgos”, constituye una herramienta capaz de ayudar a la toma de decisiones. Para el logro de la misma, se definieron los procesos fundamentales relacionados con la gestión de riesgos de software. Tras el análisis detallado de los métodos que abordan el tema, se seleccionó MOGERI como la metodología adecuada para ello, de acuerdo con las características de dicho centro de desarrollo de software.

La formalización de un modelo computacional donde se aplican técnicas de inteligencia artificial, a consideración de la mitigación de riesgos como entidad principal dentro de los procesos de desarrollo de software, permitió poner en práctica esta solución en proyectos reales del centro. Fueron enunciados los algoritmos propuestos para considerar la semejanza entre cada par de proyectos. Además se describen las características generales de una herramienta automatizada que implementa el RBC de forma que se pueda acumular la experiencia en la detección de defectos en los proyectos de software y emplearla en la ejecución de nuevas revisiones.

Del total de proyectos analizados, el 80% arrojó buenos resultados, mostrando un ahorro de recursos y tiempo, y cumpliéndose con los objetivos propuestos en el alcance definido.

Referencias

- BMP. Best Manufacturing Practices. [en línea] 2008. [Consultado el: 10 de agosto de 2011]. Disponible en: [<http://www.bmpcoe.org/>].
- CIBERTEC. Gestión de Proyectos de Software bajo el enfoque PMI. [en línea] 2008 [Consultado el: 23 de agosto de 2010]. Disponible en: [http://www.cibertec.edu.pe/2/modulos/JER/JER_Interna.aspx?ARE=2&PFL=2&JER=3338].
- Expert, Soft. Soft Expert. [en línea] 2007. [Consultado el: 10 de diciembre de 2010]. Disponible [www.softexpert.es/norma-pmbok.php].
- GÁLVEZ, DANIEL. *Curso de Sistemas Basados en el Conocimiento*. 1998.
- GARCÍA, JOSÉ LUIS PEREZ. Expertos TIC.Red Social de profesionales y empresas. *Gestión de Proyectos*. [en línea] 2009. [Consultado el: 25 de agosto de 2011]. Disponible en: [<http://www.expertostic.com/articulos/gestion-proyectos.html>].
- PMBOK. *Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK)*. Four Campus Boulevard. Newton Square, Pennsylvania 19073-3299: Project Management Institute, Inc., 2008.
- PRESSMAN. *Ingeniería del Software. Un enfoque Práctico*. México, McGraw-Hill: Version 1.1. CMMI. C. M. S. E. Institute. Pittsburgh, Carnegie Mellon Software Engineering Institute., 2002.
- ZULUETA, YELENY y DESPAIGNE, EDER. *Revista Española de Innovación, Calidad e Ingeniería de Software*. Barcelona: s.n., 2009. Vol. 5.