PREDICCIÓN Y CLASIFICACIÓN DE RIESGOS EN PROYECTOS DE SISTEMAS DE INFORMACIÓN

Alba, C.; Rodríguez, V.; Ortega, F.; Villanueva, J.

Abstract

The development of a software project is subjected to many risks and treatments of diverse nature. The omission of risk management causes out of date product deliveries, with higher cost than planned and/or presenting failures concerning the specifications and requirements established by the client.

It is considered of great utility for the Project Manager to find a model that, taking basic information of the project, would be able to predict its difficulty and to classify it depending on its risk, so that it would be able to propose coherent as well as proportional actions up to the level of problems expected during the development.

The main objective of this work is to study the availability of the application of data mining techniques that allows to characterize the risk of software projects and to predict the approximate number and type of failures (extreme, major, minor) before the development phase. It will also identify the characteristic needed to develop the estimation, as project characteristics, developing language, use of CASE tools, team size, number of systems to integrate, using real data gathered by ISBG. In order to achieve this, the current systems are analyzed, selecting the more available characteristics in ISBG data base.

Keywords: Software projects, Risk management, Data mining, Project Management

Resumen

El desarrollo de un proyecto software está sometido a multitud de riesgos y amenazas de diversa naturaleza. La omisión de la gestión de estos riesgos provoca entregas de productos finales fuera de plazo, con mayor coste respecto a lo planificado y/o presentando incumplimiento de especificaciones y requerimientos establecidos por el cliente.

Se considera de gran utilidad para los Directores de Proyecto encontrar un modelo que, partiendo de información básica del proyecto sea capaz de predecir su dificultad y clasificarlo en función de su riesgo, de modo que se puedan proponer medidas coherentes y proporcionadas al nivel de problemas esperado durante su desarrollo.

El principal objetivo de este trabajo es estudiar la viabilidad de la aplicación de técnicas de minería de datos que permitan caracterizar el riesgo de un proyecto software y predecir el número aproximado de fallos y tipo de los mismos (extremos, graves, leves) antes de comenzar el desarrollo. Así como identificar los atributos necesarios para la estimación como las características del proyecto, lenguaje de desarrollo, herramientas CASE, tamaño del equipo, sistemas a integrar, usando un conjunto datos recopilados por *ISBSG*. Para ello se analizan otros sistemas, seleccionando atributos viables y disponibles en la base de datos ISBSG.

Palabras clave: Proyectos Software, Riesgo, Data mining, Dirección de proyectos.

1. Descripción del problema

Dentro de la ingeniería de los sistemas de información muchos estudios concluyen que los productos que se obtienen tienen un gran número de deficiencias, con desviaciones de retraso a la entrega, tanto en costes como en tiempo.

Muchos autores intentaron analizar las causas de esta peculiaridad de la ingeniería de sistemas de información, como es el caso de Alfred Spector, presidente de Transarc Corporation, que en 1986 publicó un artículo comparando la construcción de puentes con el desarrollo software [1]. La premisa consistía en que los puentes normalmente se construyen en el tiempo establecido, con el presupuesto asignado y no se caen. Por el contrario, el desarrollo software nunca se completa en los plazos asignados, ni de acuerdo con el presupuesto asignado, y por lo tanto fracasa. La razón fundamental de esta diferencia está en el diseño extremadamente detallado. El diseño de un puente permanece inalterable y no admite cambios y por lo tanto el contratista tiene pocas posibilidades de cambiar las especificaciones. Otra diferencia es que cuando un puente se cae, se investigan las causas y se acumulan para otras futuras construcciones, mientras que en el desarrollo software los fracasos se ocultan y no se obtiene el beneficio producido por las lecciones aprendidas.

Tras esta primera aproximación, se comenzó de forma más rigurosa a estudiar el problema y sus posibles soluciones. En los últimos años surgen distintas instituciones que realizan informes y análisis estadísticos, como pueden ser: GAO (Government Account Office) que analiza proyectos de desarrollo de software para el Gobierno Americano o ESPITI (European Software Process Improvement Training Initiative) que realiza estudios sobre los principales problemas en el desarrollo de software a nivel europeo, y cuyos resultados son muy similares a los obtenidos en otro de los informes más aceptados, el CHAOS (Standish Group), indicando que los mayores problemas están relacionados con la especificación, la gestión y la documentación de los proyectos.

Los resultados de las investigaciones CHAOS son los más contrastados a nivel mundial en la industria de las Tecnologías de la Información (TI), y representan una década de datos que incluyen más de 50.000 proyectos y que indican los niveles de éxito o fracaso de los proyectos informáticos. El objetivo de estas investigaciones es proporcionar una comprensión de las razones por las que fracasan los proyectos, así como de los principales factores de riesgo, analizando las claves que pueden reducir los fracasos. El objeto de investigación del grupo Standish Group se centra en identificar el alcance de los fracasos del software, los factores principales que causan el fracaso de los proyectos software y los ingredientes clave que pueden reducir el fracaso de los proyectos.

Los resultados aportados por CHAOS hasta 2004 muestran una mejora en la gestión de los proyectos de TI, con un crecimiento en el número de proyectos con éxito y una caída en proyectos fracasados, mientras que los proyectos que sufren muchos cambios tendían a estabilizarse, aunque parece alcanzarse una barrera que impide el crecimiento de los proyectos con éxito y la tendencia es el mantenimiento de los niveles conseguidos en los últimos años.

Dada la ralentización de la mejora, se comienza la búsqueda de las causas con más exhaustividad, detectándose como factores críticos de los proyectos con problemas los siguientes:

- Falta de información por parte de los usuarios
- Especificaciones y requisitos incompletos o cambiantes
- Falta de apoyo de los directivos

- Incompetencia tecnológica
- Falta de recursos
- Expectativas no realistas
- Objetivos poco claros
- Plazos temporales no realistas
- Uso de tecnología novedosa

Como se observa en estos puntos, muchas de las causas están relacionadas con factores humanos, motivo por el cual, a mediados de los años 90 empezaron a aparecer iniciativas de aplicación metodologías de mejora de procesos, utilizando los modelos CMM (Capability Maturity Model, 1987), SPICE (Software Process Improvement and Capability dEtermination, ISO/IEC 15504), BOOTSTRAP, etc.

De los estudios realizados sobre ESPITI se observa que, para todos los sectores, las unidades de producción se ven afectadas por deficiencias en las especificaciones. También concluyen con unos resultados significativos en cuanto al uso de métodos de calidad, como por ejemplo que el 65% de las compañías europeas no utilizan procesos de mejora del software, que el 86% de las compañías europeas no emplean métodos de valoración del software y que el 80% no siguen ISO 9000. Sin embargo la disposición general de los equipos técnicos y de gestión en el uso de herramientas y métodos de calidad de desarrollo de proyectos e ISO 9000 es positiva.

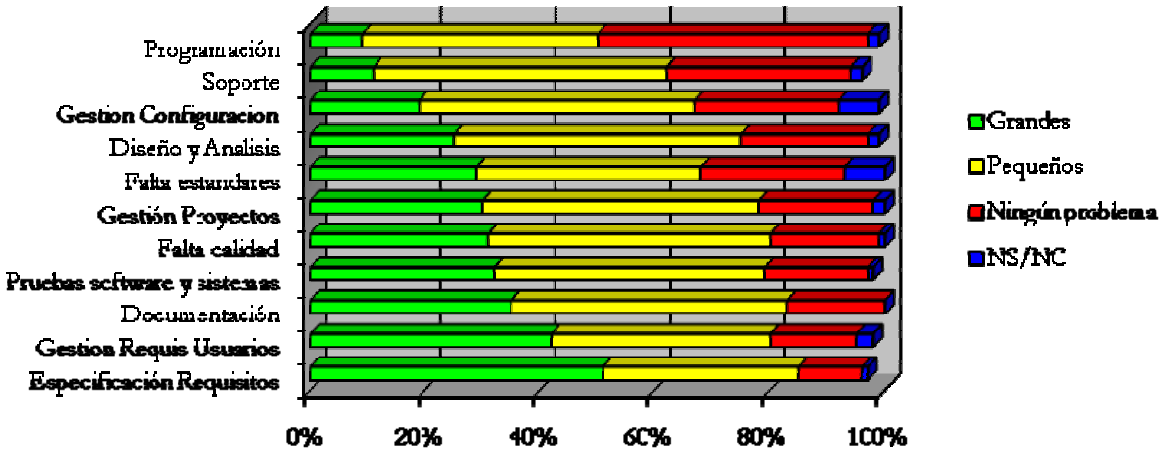

Figura 1. Influencia de los factores de fracaso en los proyectos. Fuente: ESPITI.

La filosofía o tendencia a ocultar las desviaciones provocadas por los defectos o fallos de desarrollo generan un espiral perjudicando progresivamente la solución del mismo, dado que esta manera de actuar no favorece el disponer de información histórica que permita adquirir conocimiento o lecciones aprendidas para futuros desarrollos, que es la base de cualquier ciclo de mejora continua para el desarrollo y mejora de productos.

Otra de las causas a tener en cuenta es que los equipos de desarrollo software siguen centrando sus esfuerzos en el propio desarrollo, en la integración de sistemas, en las comunicaciones, y en definitiva, en los aspectos puramente técnicos de los proyectos de sistemas de información, dejando totalmente al margen los aspectos de dirección de proyectos diferentes de la calidad o la gestión de costes y plazos. Si bien estos factores son fundamentales, la dirección de proyectos requiere la consideración de otros condicionantes. Para ello existen propuestas como IPMA (International Project Management Assotiation) y

PMI (Project Management Institute). Como ejemplo, esta última considera nueve factores a dirigir: costes, plazo, alcance, integración, recursos humanos, aprovisionamiento, calidad y riesgo. Es precisamente este punto, el riesgo, el que se detecta como una carencia generalizada en los proyectos software.

Una vez descrito el entorno en el que se encuentra el problema, se plantea el objetivo del trabajo expuesto en este artículo que es encontrar o definir una metodología que permita generar un modelo que facilite las labores de los directores de proyectos.

Definir un protocolo que permita a un director de proyecto conocer, prevenir e intentar mitigar cualquier riesgo al que pueda estar sometido un proyecto de sistemas de información es una labor casi inabordable para una persona, debido principalmente a la enorme cantidad y diversidad de amenazas presentes desde antes del comienzo del proyecto.

Por esta razón, se plantea la analizar la realización de un modelo de minería de datos que permita a los directores de proyectos software conocer los principales riesgos a los que puede estar sometido el proyecto, así como la gravedad de los mismos, desde antes del comienzo del proyecto en función de cierta información básica y en la mayor parte de los casos, disponible, lo cual permitiría emplear los pocos recursos normalmente asignados a la gestión de riesgos en los problemas potenciales con mayor probabilidad de ocurrencia, teniendo en consideración a su vez la gravedad de los mismos.

2. Evolución de la gestión de riesgos

La gestión de los riesgos en el software es un tema cuya evolución va pareja a la evolución de las técnicas y métodos del software.

Las primeras formas de identificar los factores de riesgo en los proyectos de sistemas de información proceden de estudios de riesgos producidos y sus posteriores agrupaciones, otras proceden de encuestas a directores de proyectos o entrevistas a expertos. Autores como McFarlan [2], Boehm [3] o Barki [4] desarrollaron su estudios para definir los grupos de factores de riesgo con estas técnicas.

Otra perspectiva a la hora de analizar la importancia de los riesgos en los proyectos de software es que son expresados normalmente como una combinación de probabilidad de ocurrencia e impacto en el rendimiento del proyecto. En esta línea existen tres métodos ampliamente conocidos de gestión de riesgos software en la literatura: SRE (Software Risk Evaluation), SERIM (Software Engineering Risk Management) y la DoD Guide (Department of Defense).

También se ha de tener en cuenta que cuando se habla del término riesgo se asocia normalmente con el concepto de éxito del proyecto. Éxito es un término muy difícil de precisar. Es algo muy subjetivo. Por supuesto, la apreciación de éxito dentro de un proyecto variará en función de la percepción de cada persona. Normalmente, las personas afectadas o relacionadas con el proyecto desde fuera de la organización utilizarán criterios basados en el coste o en los plazos, mientras que las personas involucradas en el proyecto de forma interna coinciden en que el grado de éxito del proyecto puede medirse como el logro del alcance de desarrollo. Uno de los trabajos más interesantes a este respecto se debe a Agarwal [5]. En el estudio se sostiene que para evaluar el éxito hay que fijarse en dos aspectos diferentes de los proyectos: factores internos y externos. El interno, aborda el éxito desde el punto de vista de la ejecución, seguimiento y control del proyecto a corto plazo, mientras que el segundo criterio lo aborda desde el punto de vista del valor final de los entregables que se les dará a los usuarios, teniendo un mayor impacto a más largo plazo.

La utilización de técnicas inteligentes para dar solución a este problema también está presente en trabajos que emplean redes neuronales difusas (fuzzy neural networks) para

evaluar el rendimiento de un proyecto, redes bayesianas para los factores que afectan al proceso de desarrollo de software, etc. Estos problemas pueden derivar en factores de riesgos pero su objetivo no era ese en concreto, por tanto se ha de llegar a un acuerdo para definir como se han de medir o cuantificar los factores de riesgo.

3. Metodología planteada

Como se comentaba en apartados anteriores, dado que se partirá de un conjunto de datos históricos, se plantea utilizar una metodología orientada a la minería de datos, ya que se pretende conseguir mediante técnicas y herramientas extraer un conocimiento implícito que actualmente no conocemos y que se encuentra almacenado en el conjunto de datos. Utilizar esta metodología tiene como objetivo predecir de forma automatizada tendencias y comportamientos o construir un modelo desconocido.

La metodología CRISP-DM [6] estructura el ciclo de vida de un proyecto de minería de datos en seis fases, que interactúan entre ellas de forma iterativa durante el desarrollo del proyecto con se puede ver en la siguiente figura.

La primera fase, “Análisis del Problema”, incluye la comprensión de los objetivos y requerimientos del proyecto desde una perspectiva empresarial, con el fin de convertirlos en objetivos técnicos y en una planificación.

La segunda fase de “Análisis de Datos” comprende la recolección inicial de datos, en orden a que sea posible establecer un primer contacto con el problema, identificando la calidad de los datos y estableciendo las relaciones más evidentes que permitan establecer las primeras hipótesis. En la siguiente fase, la metodología establece que se proceda a la preparación de los datos, de tal forma que puedan ser tratados por las técnicas de modelado. La preparación de datos incluye las tareas generales de selección de datos a los que se va a aplicar la técnica de modelado (variables y muestras), limpieza de los datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato.

Las siguientes fases de la metodología no abordadas en este trabajo serán “Preparación de los Datos” se encuentra muy relacionada con la fase de modelado, puesto que en función de la técnica de modelado que vaya a ser utilizada los datos necesitan ser procesados en diferentes formas. Por lo tanto las fases de preparación y modelado interactúan de forma sistemática. En la fase de “Modelado” se seleccionan las técnicas de modelado más apropiadas para el proyecto de minería de datos específico. Antes de proceder al modelado de los datos se debe de establecer un diseño del método de evaluación de los modelos, que permita establecer el grado de bondad de los mismos. Una vez realizadas estas tareas genéricas se procede a la generación y evaluación del modelo. Los parámetros utilizados en la generación del modelo dependen de las características de los datos.

En la fase de “Evaluación” se evalúa el modelo, no desde el punto de vista de los datos, sino del cumplimiento de los criterios de éxito del problema. Se debe revisar el proceso seguido, teniendo en cuenta los resultados obtenidos, para poder repetir algún paso en el que, a la vista del desarrollo posterior del proceso, se hayan podido cometer errores. Si el modelo generado es válido en función de los criterios de éxito establecidos en la primera fase, se procede a la explotación del modelo.

4. Descripción del conjunto de datos de estudio.

Se ha optado por dar un enfoque al trabajo orientándolo a un problema de minería de datos por tanto el primer punto a solucionar es el conjunto de datos. Se plantea la posibilidad de realizar una captura de datos procedente de proyectos actuales. Esta solución tiene el problema de ser lenta puesto que la vida media de un proyecto puede ser de meses o años

con lo que se tardará en conseguir un conjunto de datos suficientemente representativo de todas las condiciones de desarrollo y de distintas organizaciones.

Por tanto se ha optado por buscar la información en una base de datos de prestigio que contiene datos procedentes del cierre de proyecto, como es el caso de la del (ISBSG) *International Software Benchmarking Standards Group,* organización sin ánimo de lucro que se fundó en el año 1.997. La información contenida en su base de datos es el resultado de varios años de cooperación previa de varias asociaciones nacionales de métricas de software, las cuales intentaban desarrollar y promocionar el uso de medidas para mejorar los procesos y productos software. Los datos almacenados en esta base de datos han sido recogidos y valorados por expertos en análisis y desarrollo de software, por lo que se garantiza la calidad y fiabilidad de los mismos.

Los datos de los que se parte para realizar el estudio proceden de 3.024 proyectos, cada uno de ellos con 100 variables o atributos. Algunas de estas tienen información meramente contextual del proyecto, es decir, aportan datos sobre la ejecución o tipo de proyecto pero no se pueden utilizar en un proceso de minería de datos. En este grupo se encuentran variables como alcance del proyecto, técnicas de desarrollo o tipo de organización. Además de estas variables con información textual se disponen de variables con valores numéricos y categóricos.

5. Análisis del problema y del conjunto de datos.

El primer punto a definir son los objetivos que se han de conseguir. Este objetivo general debe de transformarse en objetivos específicos que sean medibles y alcanzables mediante la aplicación de técnicas de minería de datos. Por lo tanto, del objetivo general se derivan los siguientes objetivos específicos:

- Identificar los atributos que permitan ser utilizados en un modelos predictivos basados en datos capaces de estimar el número de defectos potenciales que contendrá el sistema de información a desarrollar, categorizándolos en leves, medios y graves. También se puede plantear que el modelo nos prediga la posibilidad de ocurrencia de cada uno de ellos.
- Los atributos también deberían posibilitar desarrollar un sistema que permita extraer el conocimiento implícito contenido en una base con datos procedentes de cierre de proyectos.

Teniendo en cuenta que el conjunto de datos está formado por proyectos distintos, se están mezclando cosas no comparables. Hay que estar seguro que sólo se consideran proyectos en los que se midan las distintas variables (esfuerzo, riesgos, tamaño, etc.) de la misma forma que en el proyecto a comparar, estimar o evaluar. También se ha de considerar proyectos en los que los atributos disponibles son similares a los disponibles en el proyecto a tratar.

Un ejemplo de esta situación es el caso del tamaño, esfuerzo o calidad de muestreo:

- Si el tamaño del proyecto es relevante para el caso de estudio, sólo puede compararse con proyectos en los que se haya usado el mismo método de medición del tamaño.
- El esfuerzo tiene en cuenta distintos niveles de detalle. En función de los valores tomados por el atributo *Resource Level* se mide el esfuerzo del equipo de desarrollo, o el del equipo de soporte, o las operaciones de computación y también el esfuerzo realizado por el cliente o usuario final.
- En cuanto a la calidad de los datos debe considerarse el campo Data Quality Rating. Sólo deben tenerse en cuenta proyectos con valor A o B en este campo.

El ISBSG sugiere que los criterios más importantes para seleccionar proyectos similares que son:

- Size: Si el tamaño del proyecto a evaluar o comparar es muy grande, no aporta mucho valor estudiar o seleccionar proyectos pequeños y viceversa.
- Development Type: Hay tres valores posibles, nuevo desarrollo, mejora o desarrollo sobre un proyecto hecho.
- Primary Programming Language, o Language Type (ej. 3GL, 4GL)
- Development Platform (ej. Mainframe, midrange o PC)

Otros criterios a tener en cuenta son: Organisation Type, Business Area Type, Application Type, User base and Development Techniques.

Hay que tener en cuenta que a medida que se añaden criterios de selección, el número de proyectos seleccionados se reduce.

En la fase de análisis de la calidad de los datos se ha de garantizar el cumplimiento de todas las condiciones de calidad de datos mencionadas anteriormente.

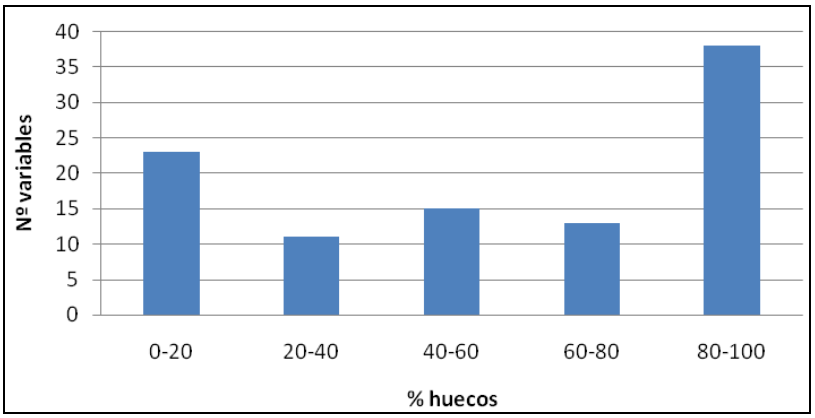

Figura 2. Distribución de las variables según % de valores perdidos.

En relación a la calidad de los datos, se ha observado en un estudio preliminar que será uno de los puntos sobre los que habrá que centrar los esfuerzos, dado que existen:

- Outliers, valores de proyectos que siendo reales se salen del funcionamiento ordinario del resto de proyectos con los que se deberá decidir su eliminación de la muestra.
- Variables categóricas, que plantean la dificultad de que muchos de los métodos predictivos necesitan convertir esas categorías en variables numéricas. Esto complica la selección de la técnica de modelado y por ello será necesaria la transformación de estas variables.
- Valores vacios, puesto que en algunos atributos no se dispone de información de algunos proyectos, con lo que se tendrán que utilizar técnicas inteligentes o combinaciones de varias para estimar el valor que podría tomar ese atributo en ese proyecto o como último caso eliminar el proyecto de la muestra.

La Figura 2 muestra el número de variables que presentan distintos intervalos de porcentaje de huecos o valores perdidos. Se detecta una alta presencia de valores perdidos o falta de información dentro de las variables de la base de datos. Únicamente hay 5 variables que contienen toda la información respecto a todos los proyectos recogidos, mientras que 77 variables tienen pérdidas de información superiores a un 20% de los casos.

Después de aplicar los filtros comentados el conjunto de datos se verá claramente reducido tanto en número de atributos como en proyectos. Los atributos, después de las transformaciones de categóricas a numéricas y del rechazo de las que aportan información contextual, se estima que estarán en torno a un 30% de los iníciales.

Como resultado de este análisis los atributos que podrán ser utilizados para modelar contendrán información sobre:

- Métricas de tamaño, como valores relacionados con los puntos de función, y sus valores de ajuste o líneas de código.
- Esfuerzos realizados para el desarrollo del proyecto, tanto de información final como de información desglosada por cada una de las fases del proyecto (planificación, especificación, diseño, construcción, pruebas, implantación).
- Plazos de ejecución, es decir, duración, tiempos o porcentaje de inactividad.
- Defectos detectados, tanto leves, medios o graves, así como el total de defectos.
- Plataforma, lenguaje, herramientas CASE y metodología utilizada para el desarrollo del producto.
- Equipo de trabajo, tamaño del equipo, información sobre la ubicación y usuarios concurrentes.

6. Conclusiones y desarrollos futuros.

Los defectos pueden considerarse un indicador del riesgo, si bien se ha de analizar si a partir de este conjunto de datos se puede discriminar entre orígenes de los riesgos o sólo definir un indicador de la gravedad.

Vista la viabilidad de utilizar técnicas de minería de datos sobre el conjunto de datos proporcionado por ISBSG y que se dispone de variables que se pueden definir como objetivo, ya sea para estimar el número o la posibilidad de riesgos generados por el producto software, y que además estos resultados pueden ser alcanzables mediante técnicas supervisadas de predicción, se plantea como línea futura de trabajo, conseguir un modelo capaz de clasificar los errores en tres niveles (leve, medio o grave) y estime la presencia de cada uno de ellos.

La utilización de un modelo basado en datos plantea la posibilidad de que se pueda implantar el modelo en cada organización, ajustándose a las características de los desarrollos propios. Los modelo así generados son generalistas, por lo tanto se podrían aplicar a otras organizaciones. Pero además se plantea la posibilidad de recoger datos para ajustarse a las necesidades o peculiaridades de esa nueva organización, con lo que se podría re-entrenar el modelo para ajustar sus parámetros a la propia organización.

Dada la viabilidad de fases iníciales de la metodología CRISPDM se plantea como futuros trabajos la aplicación de técnicas de minería de datos como los mapas auto-organizados (SOM) [7], que se pueden utilizar para realizar proyecciones y agrupaciones del conjunto de datos permitiendo encontrar elementos anómalos, así como reducir la dimensión del conjunto inicial de datos. Esta técnica también se puede utilizar para identificar comportamientos similares entre proyectos independientemente de que sobre un atributo no se disponga de información dado que es robusta a la presencia de valores vacios.

En cuanto a la predicción se plantea la utilización de una de las técnicas más prometedoras en la modelización de problemas complejos no lineales, el algoritmo MARS [8] (Multivariate Adaptive Regression Splines), que se adapta perfectamente al volumen y la tipología de

datos. Deberá realizarse también una comparación de los resultados con los mecanismos tradicionales de estimación del riesgo.

Referencias

[1] Spector A. et al “A computer science perspective of bridge design”, *Communications of the ACM*, Vol 29, 267 - 283 (April 1986)

[2] McFarlan, F. “Portfolio approach to information systems”. *Harvard Business* Review 59, pp 142–150. 1981

[3] Boehm, B. “Software risk management: principles and practices”. *IEEE Software 8*, pp. 32–41. 1991

[4] Barki, H., Rivard, S., Talbot, J. “Toward an assessment of software development risk”. *Journal of Management Information Systems* 10, pp. 203–225. 1993

[5] Agarwal, N., Rathod, U. “Defining ‘success’ for software projects: An exploratory revelation”. *International Journal of Project Management*. Elsevier. 2006.

[6] *CRISP-DM 1996, CRoss-Industry Standard Process for Data Mining.*

URL: http://www.crisp-dm.org/.

[7] Kohonen, T., 2006. Self-organizing neural projections. Neural Networks. 19 (6-7), 723-733.

[8] Friedman, J. H. (1991a). Multivariate Adaptive Regression Splines (with discussion), *Annals of Statistics*, 19, 1-141.

Correspondencia (Para más información contacte con):

Universidad de Oviedo
Área de Proyectos de Ingeniería
c/ Independencia, 13
33004 Oviedo, Asturias (España).
Phone: +34 985 10 42 72
Fax: + 34 985 10 42 56
E-mail: secre@api.uniovi.es
URL: http://www.api.uniovi.es