

Toma de Decisiones a partir de Conocimiento Extraído tras el Análisis de Comportamientos Previos. Aplicación Práctica para la Dirección de Proyectos de Desarrollo de Software

Decision making from knowledge obtained after previous behavior analysis. Practical implementation to project management of software development

A. Peralta¹, F. P. Romero¹

Recibido: 9 de octubre de 2015

Aceptado: 10 de noviembre de 2015

Resumen

En la disciplina de desarrollo del software y en otras disciplinas existe un problema inherente en la organización de tareas para realizar un proyecto tanto en su definición como durante su elaboración. La incertidumbre en la definición de la duración de las tareas y el rendimiento de las personas que las realizan conforman un entorno en el que las técnicas de Soft-Computing pueden aportar nuevos enfoques, con el fin de ayudar a los participantes de un proyecto a administrar su tiempo, priorizar sus actividades y reajustar el trabajo para alcanzar los objetivos marcados. Para ello, a partir de los registros de tiempo de cada uno de los participantes en el proyecto, se extraerá el conocimiento necesario para apoyar la toma de decisiones en la distribución de tareas del trabajo diario y la elaboración de estimaciones para futuros proyectos.

Palabras Clave: Planificación de proyectos, Estimación de tareas, Personal Software Process, Reconocimiento de patrones, Clustering

¹ PhD Investigador Departamento de Tecnologías y Sistemas de la Información, Universidad de Castilla - La Mancha, Paseo de la Universidad, 4, 13071, Ciudad Real, España. aperalta@campus.eae.es, FranciscoP.Romero@uclm.es

Abstract

Within the discipline of software development as well as in other disciplines there is an inherent problem in organizing tasks for a project both in its definition as during processing. The uncertainty in the definition of the duration of the tasks and performance of the people who make them make up an environment in which the soft-computing techniques can provide new approaches in order to help project participants to manage your time correctly, prioritize their activities and readjust work successfully to achieve the objectives. To do this, from the time records of each of the participants in the project, the necessary knowledge with which support both the decision for the distribution of tasks in their daily work and the preparation of estimates to be extracted future projects.

Keywords: *Project planning, estimation tasks, Personal Software Process , Pattern Recognition, Clustering.*

1. Introducción

En la actualidad, uno de los principales problemas al que se enfrentan las empresas, especialmente las dedicadas al desarrollo de *software*, es la estimación de la duración de sus proyectos. A pesar de contar con gente de reconocida experiencia para el desarrollo de las tareas, es muy habitual que en el momento de hacer la estimación el conjunto de tareas a realizar y su definición no estén totalmente acotadas para definir las de forma completa. Además, durante el transcurso del proyecto suelen aparecer nuevas tareas o proyectos que afectan al equipo del proyecto y se dificulta el cumplimiento de su planificación.

Al problema de definición de tareas se le une el “factor humano” de la construcción del *software*, es decir, las personas que realizan el trabajo de desarrollo. El escenario existente en la consultoría tecnológica de multitud de países está caracterizado por altos niveles de superposición, es decir, que muchas personas realizan la misma tarea a la vez o una persona realiza varias tareas al mismo tiempo. A su vez los desarrolladores suelen estar multia-signados ^[1] es decir, realizan trabajos de diferentes proyectos en intervalos temporales cortos. A esto hay que añadir la flexibilidad de horarios, las interrupciones durante el trabajo, realización de horas extras e impuntualidad, así como la definición difusa de los objetivos. Es por ello que manejar correctamente esta incertidumbre es una constante en la gestión del proyecto.

Por estos motivos, aunque el estimador se esfuerce, los resultados de su trabajo no logran ser infalibles y precisos. La multitud de factores que influyen en el transcurso de un proyecto pueden impedir la estimación realizada,

lo cual al final redunda negativamente tanto en los beneficios de la empresa como en la situación del equipo, pudiendo aparecer factores como el estrés y la insatisfacción que darán lugar a un trabajo de menor calidad.

Dado que el desarrollo de productos de *software* implica no sólo escribir instrucciones de programación y ejecutarlas en un ordenador sino que requiere cumplir requisitos del cliente a un coste y planificación acordada, se han definido metodologías que ayuden a los profesionales a realizar su trabajo dentro de una serie de buenas prácticas en Ingeniería del Software. Una de las técnicas más destacadas es el *Personal Software Process* (PSP) ^[1]. Dentro del PSP se definen métodos que permiten realizar una mejor planificación y seguimiento del trabajo, establecer metas mesurables que posibilitan la realización de un rastreo constante de la consecución de las metas, a su vez, ofrece la posibilidad de utilizar un proceso bien definido y medido de trabajo.

Tanto PSP en el ámbito individual como su extensión a equipos TSP (*Team Software Process*) requieren la recopilación y análisis de métricas con un elevado nivel de detalle. Una vez que los datos necesarios han sido recopilados al nivel preciso, el análisis estadístico de los mismos, permite efectuar una planificación realista, rastreo, predicción, y control de los proyectos y productos de *software*. Para llevar a cabo la recopilación de estos datos es necesario que el desarrollador utilice una determinada metodología como puede ser la propuesta en los llamados “cuadernos de ingeniería” o “cuadernos de registro de tiempos” en los que se lleva un control de los datos mostrados en la tabla 1, relativos a la distribución temporal de las labores realizadas por el trabajador.

TABLA 1. DATOS REGISTRADOS EN EL CUADERNO DE INGENIERÍA

Fecha	Día en el que se realiza el trabajo.
Hora Inicio	Hora en la que se inició el trabajo en la tarea.
Hora Fin	Hora en la finalizó el trabajo en la tarea.
T. Interrupción	Tiempo de interrupción.
T. Efectivo	Tiempo de trabajo, sin interrupciones.
Proyecto	Proyecto al que corresponde la tarea.
Actividad	Tarea realizada durante la sesión.
Comentarios	Comentarios sobre el desarrollo de la tarea.
Otros datos	Lugar, porcentaje de avance en la tarea, etc.

Habitualmente, la estimación de duración en proyectos *software* o de modo general, la planificación de tareas, supone una línea de investigación de gran interés, contando con numerosos estudios centrados en el uso de diversas técnicas como las propuestas por autores como Boehm ^[2] y Abdel-Haid^[3]. Sin embargo, en la mayoría de los casos, el conocido “factor humano”, la incertidumbre, o las distintas eventualidades hacen que las estimaciones planificadas en una primera aproximación, dejen de ser tan precisas o útiles como se pretendía.

Ante la problemática expuesta previamente, en este trabajo se propone la utilización de diversas técnicas enmarcadas dentro del *Soft-Computing* como la lógica borrosa, o algoritmos de clustering con el fin de facilitar la planificación y desarrollo de las tareas de un Ingeniero de *Software*. Como fuente de datos para realizar este trabajo de extracción de conocimiento se establecen las técnicas y herramientas definidas *Personal Software Process* (PSP).

Concretamente se persiguen los siguientes objetivos:

1. Detectar patrones y extraer conclusiones para poder construir “estimaciones basadas en conocimiento”. Estas estimaciones se confeccionarán mediante técnicas de optimización heurística que tengan en cuenta los factores clásicos y humanos de la construcción de aplicaciones *software*.
2. Proporcionar las herramientas necesarias para realizar una programación adaptativa de las tareas que un componente del equipo de proyecto debe realizar durante un ciclo temporal, permitiendo que tanto el objetivo del proyecto como la cohesión del equipo de desarrollo se vea lo menos comprometido posible por factores externos.

Este trabajo se organiza de la siguiente forma: en la sección número 2 se realiza una breve semblanza de los trabajos relacionados tanto con la planificación de proyectos, la utilización de simuladores y la literatura existente. En la sección 3 se desarrolla el proceso de extracción de conocimiento que va desde la selección y preproceso de las fuentes de datos hasta la caracterización de los elementos y la definición de patrones de comportamiento. En la sección 4 se muestran los resultados obtenidos en un caso de estudio para finalizar con las conclusiones y el trabajo futuro en la sección 5.

2. Trabajos relacionados.

La estimación de la duración de tareas, o en general la planificación de proyectos supone una problemática especialmente costosa y de gran importancia en el desarrollo de *software*. Por este motivo en los últimos años se han desarrollado numerosos trabajos que tratan, mediante el uso de técnicas basadas en Ingeniería del *Software*, de estudiar posibles soluciones a este problema. Sin embargo, la incertidumbre motivada en muchos casos por el anteriormente citado “factor humano” hace necesario investigar soluciones que conjuguen técnicas propias del campo de la Ingeniería del *Software*, como el PSP, con otras basadas en *Soft-Computing* como la lógica borrosa o los algoritmos de clustering. Actualmente el uso conjunto de ambas técnicas continúa sin ser suficientemente analizado.

No obstante, un campo especialmente estudiado en los últimos años está siendo el referido al PSP. Concretamente, muchos de los trabajos están siendo orientados al desarrollo de herramientas de apoyo al PSP en la recopilación de los datos necesarios para su análisis. Ejemplos de ello pueden ser herramientas automatizadas como *Leap* ^[4], *PSP Studio* o *PSP Dashboard*. Mediante este tipo de aplicaciones el desarrollador puede, de una manera amigable y sencilla ingresar mediante cajas de diálogo los datos relativos al esfuerzo supuesto, el tamaño del proyecto desarrollado y defectos detectados. Además también permiten mostrar análisis de los datos si el usuario lo desea. Por último, proyectos de última generación como *Hackystat* ^[5] permiten la recolección automática de métricas mediante sensores adjuntos a las herramientas de desarrollo, trasladando los datos a un servidor y dependiendo de su análisis, enviando mensajes de alerta, por ejemplo, a los desarrolladores.

Por otro lado, uno de los mayores avances en la estimación y gestión de proyectos ha venido dado por la creación de modelos sobre el mismo proceso de desarrollo. Estos modelos permiten comprobar el efecto de ciertas actuaciones al éxito del proyecto con el fin de apoyar en la toma de decisiones a los gestores de los proyectos ^[6]. Se distinguen dos tendencias principales:

- Modelos de estimación estáticos:
Son empíricos y univariantes. Se caracterizan por la utilización de fórmulas obtenidas empíricamente de proyectos reales, para predecir los datos requeridos. Entre los más utilizados está COCOMO (CONstructive COst MOdel) en sus versiones I y II ^[2].

- Modelos de estimación dinámicos:

Aplican dinámica de sistemas al desarrollo de proyectos de *software* ^[3]. Son de relevancia debido a su naturaleza dinámica y a su capacidad de adaptación a la evolución temporal del proyecto.

Sin embargo, son escasos los trabajos que combinen estas técnicas con otras basadas en *Soft-Computing* para la gestión de tareas con incertidumbre, variables en el tiempo, o la caracterización de los participantes en un proyecto *software*. Para comprobar este punto fue realizada una revisión sistemática sobre las principales fuentes de artículos relacionados con el desarrollo de *software*. Para ello fue seguida la metodología propuesta por Kitchemham ^[7] y la plantilla de protocolo definida por Biolchini et al., ^[8] como guía para las etapas de planificación y ejecución de la revisión sistemática.

Entre los trabajos encontrados y vinculados a este ámbito destaca el elaborado por Ramos ^[9] en el que obtienen aspectos fundamentales del proyecto a partir de un simulador cuyo núcleo está constituido por un modelo dinámico definido.

Por último Padberg ^[10] propone un modelo probabilístico de simulación basándose en una pequeña base de conocimiento que establece para cada desarrollador su probabilidad de terminar en tiempo una tarea así como grados de dependencias entre las tareas calculadas por el acoplamiento y la cohesión del diseño.

3. Extracción de patrones de comportamiento en ingenieros *software*

El proceso de extracción de patrones es una instanciación del proceso de Descubrimiento de Conocimiento en Bases de Datos ^[11] con la finalidad de conseguir detectar comportamientos y patrones útiles para su posterior utilización en la estimación, planificación y distribución del trabajo de los proyectos *software*.

3.1. Fuentes de datos

Sin lugar a dudas, el éxito de cualquier proceso de extracción de conocimiento está condicionado al uso de una fiable fuente de datos. Sin embargo, disponer de un conjunto de datos de calidad, suficientemente amplios y re-

presentativos, puede suponer un problema. Esto es debido a que en pocas ocasiones los desarrolladores tienen la costumbre de anotar sus tiempos y completar con sus datos cuadernos de ingeniería, haciendo que sea difícil contar con datos reales suficientemente extensos. Es por ello que habitualmente es necesario utilizar otras fuentes de datos como los generados de experimentos controlados, o simuladores de proyectos *software*.

Actualmente se han realizado varios experimentos basados en el desarrollo de proyectos por parte de estudiantes, dentro de alguna asignatura de ingeniería del *software* o afín. Existen incluso datos facilitados por experimentadores que han realizado trabajos en Alemania ^[12] y Finlandia ^[13].

Antes de comenzar el procesamiento de los datos se deben obtener éstos a partir de las fuentes de datos existentes y según el formato en el que éstas ofrezcan la información procesarla convenientemente mediante componentes *software* al efecto.

3.2. Preproceso

Para llevar a cabo el análisis de datos es ineludible previamente realizar inicialmente una limpieza del material con el que se va a trabajar, con el objetivo de detectar aquellos datos que pudieran ser erróneos. De modo similar, cuando se trabaja con textos es necesario un preproceso de los mismos.

En este estudio, la información procesada se brinda por los cuadernos de ingeniería, siendo necesario realizar una limpieza de los datos y los descriptores de las tareas y proyectos. Los diferentes procesos que deben realizarse para la limpieza de la información que será procesada son los siguientes:

- Detección de sesiones de trabajo incorrectas: En este paso son detectadas las sesiones cuyos datos son incongruentes. Por ejemplo, sesiones con fecha de finalización anterior a la de inicio, interrupciones con valores negativos, etc.
- Corrección ortográfica: Mediante el uso de diccionarios se realizan correcciones ortográficas en los nombres y descripciones de las tareas y proyectos definidos.
- Agrupación de tareas o proyectos según su nombre: en muchas ocasiones, varias tareas o proyectos con nombres ligeramente distintos, se tratan realmente del mismo. Por ejemplo, las tareas “Analizar-Revisar

Software” y “Analizar/Revisar Software” muy probablemente sean la misma. En este punto son agrupadas las tareas o proyectos con suficiente similitud en sus nombres.

Como resultado de este proceso se obtiene una serie de sesiones de trabajo caracterizadas por los siguientes datos.

3.3. Métricas y definición de dimensiones de estudio.

Mediante el análisis de los datos proporcionados y el uso de ciertas métricas, se pretende como primer objetivo determinar el comportamiento de un profesional dentro de una dimensión temporal y no de tareas. Para ello es necesario realizar un análisis preliminar con respecto a su tiempo dedicado al trabajo en su distribución diaria y semanal.

Posteriormente, el conocimiento extraído de este análisis es utilizado para programar las labores de un profesional durante una o varias semanas teniendo en cuenta todos los factores que determinan su distribución habitual del tiempo (tiempo trabajado según el día, interrupciones, división de una tarea entre días consecutivos, etc.). Además, mediante el estudio de los datos se pretende determinar ciertos atributos que puedan llegar a caracterizar a cada profesional de acuerdo a su capacidad de trabajo para cada tipo de tareas, su eficiencia o fiabilidad.

- **Caracterización de los días.**

Para llevar a cabo la caracterización de los días es necesario un análisis de los datos proporcionados por el libro de ingeniería de cada profesional. Una posible relación de datos a extraer para la definición del comportamiento en un día seña la mostrada en la tabla 2.

TABLA 2. CARACTERÍSTICAS A EXTRAER DEL TRABAJO DE UN DÍA.

Tiempo total trabajado.	07:20:00
Tiempo total de interrupción.	00:27:00
Tiempo medio de sesión.	01:28:00
Número de sesiones.	5
Número de interrupciones.	2
Número de tareas terminadas.	3

Otras métricas que pueden ser importantes para su posterior utilización son el porcentaje de tiempo dedicado a tareas imprevistas, si han existido lugares diferentes en los que se ha trabajado, el número de proyectos distintos en los que se han trabajado, el tiempo total en el lugar de trabajo, etc.

A partir de estas métricas es posible definir una medida de similitud entre días y una medida de rendimiento del día, así como establecer medias de trabajo entre diferentes días identificando el día de la semana que es, la media del tiempo trabajado, el volumen de interrupciones, y como estas y los tiempos de las sesiones afectan al número de tareas acabadas.

- **Caracterización de las semanas de trabajo.**

Una semana es un conjunto ordenado de días. Por tanto, una semana se caracterizará por la secuencia ordenada de las métricas calculadas para cada uno de sus días, teniendo en cuenta la naturaleza del día dentro de la semana (día laborable, festivo, víspera, número de semana, etc.). Además se cuenta con los valores agregados de estas métricas. De este modo es posible modelar una semana como un vector de siete valores, correspondiente a sus días.

- **Caracterización de los profesionales.**

La caracterización de cada profesional, el conocimiento de sus habilidades, sus debilidades o su capacidad de trabajo, supone un aspecto clave para la planificación de futuros proyectos. Con este objetivo son calculadas una serie de métricas capaces de caracterizar cada empleado mediante una serie de atributos. Para ello es necesario el análisis de la siguiente información a partir del estudio de los datos procedentes de sus libros de ingeniería:

1. Número total de sesiones.
2. Tiempo total de trabajado efectivo, sin interrupciones.
3. Tiempo total de las interrupciones.
4. Tiempo total dedicado a tareas de análisis/desarrollo
5. Número de reparaciones, es decir, el número de veces que se indicó una tarea como finalizada y sin embargo en una sesión posterior es continuada ya que realmente no había sido concluida correctamente.

A partir de esta información, es posible realizar el cálculo de las métricas usadas para caracterizar cada empleado. Esta caracterización se realiza mediante técnicas lingüísticas procedentes de la lógica borrosa mediante una

serie de variables y etiquetas lingüísticas predefinidas (muy baja, baja, media, alta, muy alta). Un ejemplo de características son las mostradas en la tabla 3.

TABLA 3. CARACTERÍSTICAS DEFINIDAS MEDIANTE ETIQUETAS LINGÜÍSTICAS

Característica	Descripción
Experiencia en análisis	% tiempo en tareas de análisis.
Experiencia en desarrollo	% de tiempo en tareas de desarrollo
Fiabilidad	% de reaparición de tareas.
Rendimiento	% consecución completa de tareas

- **Estudio de las descripciones de las tareas.**

Tal y como se mencionó anteriormente, uno de los primeros pasos en el preproceso de los datos es la agrupación de tareas con nombres muy similares según una función de distancia. Tras esta primera agrupación de tareas es posible detectar tareas que se repiten periódicamente, por ejemplo, reuniones de seguimiento, completar el parte semanal de horas, etc, e identificar tareas similares, analizando si son periódicas, ya que pueden formar pequeños grupos de tareas tratados como grupos en vez de tareas individuales.

3.4. Clustering y representación de conocimiento

Es habitual realizar la planificación del desarrollo de un proyecto con la base de un calendario laboral normal. En él los días laborables tienen un número de horas de trabajo con una cierta distribución temporal (4 días de 9 horas y uno de 7, o 5 días de 8 horas, etc.). Como variaciones de este calendario existen ciertas variaciones previsibles como pueden ser los días festivos, las vacaciones del trabajador, los días dedicados a formación, etc. Esta planificación normalmente es insuficiente, ya que por un lado no tiene en cuenta posibles imprevistos (viajes inesperados, enfermedad del trabajador, incidencias en otros proyectos, etc.), y Además porque la distribución del tiempo no suele ser una distribución realista de cómo se distribuye el tiempo semanalmente, es decir, los trabajadores no trabajan de forma efectiva todos los días las mismas horas ya que esta efectividad está influida por más factores, previstos o imprevistos.

Los algoritmos de *clustering* nos permiten realizar agrupaciones de elementos similares con el fin de descubrir patrones regulares en nuestros datos de estudio. Para poder utilizar este tipo de algoritmos se describe cada unidad temporal (día o semana) mediante un vector ordenado con los valores de las métricas que la definen. De esta manera, se obtiene una agrupación de semanas con la cual se pueda encontrar distintos tipos de comportamiento. Cada uno de estos comportamientos son definidos a partir de las características presentes en ellos y ausentes en los demás (por ejemplo, semana con día festivo, o con entrega) obtenidas mediante un proceso de post-clasificación. De esta forma, se identifica en qué semanas se trabaja mejor y cuales peor, incluso es posible estudiar la evolución de los tiempos de trabajo (y de otros factores) durante la propia semana. Ejemplo del conocimiento que se puede extraer de este proceso son reglas como “los días anteriores o posteriores al final de semana o a un festivo el rendimiento es menor”, o la regla “los días que se efectúa una entrega al cliente las horas superan el horario laboral”.

Con el fin de representar el comportamiento en cuanto a distribución del tiempo de una persona se puede recurrir a la representación del prototipo como la media de todos los comportamientos posibles. Esta representación simplifica en exceso la representación del este tipo de comportamientos, por lo que se ha optado por la utilización de una representación basada en Categorías Prototípicas Deformables Borrosas ^[14]. En esta representación un concepto engloba un conjunto de prototipos, los cuales representan la buena, baja o media compatibilidad de los ejemplares con el concepto. De esta forma, el comportamiento de un empleado en cuanto a su distribución semanal del tiempo se puede representar mediante diferentes clases que se encuentran en el dominio.

Este conocimiento será la base para realizar una planificación adaptativa de las tareas que deben realizar en un equipo de durante una semana, así se presentan los efectos que un posible viaje inesperado o una enfermedad puedan tener sobre la distribución de tareas entre los componentes del equipo en cuestión.

4. Caso de estudio

El escenario para este caso de estudio es una consultora con un equipo de 5 personas responsables de los desarrollos relativos a un mismo proyecto.

La principal tarea de los empleados es el mantenimiento de un proyecto, que además genera nuevos desarrollos que se deben estimar, planificar y distribuir correctamente en el tiempo del que disponen los componentes del equipo. Las entregas de las versiones de mantenimiento se concentran los jueves por la tarde, lo cual hace que la distribución del tiempo se vea muy afectada por estos hitos.

Para estudiar el comportamiento de este equipo de trabajo se ha partido del estudio de todos los datos expuestos en cada uno de los cuadernos de ingeniería de cada empleado. Estos datos han sido recolectados de forma privada y confidencial.

En primer lugar se ha comenzado por una limpieza de los datos recibidos, continuando por una fase de procesamiento y cálculo de métricas, y analizando con un análisis de las mismas.

En la tabla 4 se muestra el número de sesiones totales realizadas e indicadas por cada empleado en su cuaderno de ingeniería, con datos erróneos. Se muestra a su vez, el número de tareas distintas realizadas y el número real tras realizar el análisis de sus descripciones.

TABLA 4. RESULTADOS PRELIMINARES (E: EMPLEADO)

	Sesiones	% Errores	Tareas	Tareas agrupadas
E 1	687	2%	245	230
E 2	495	1.8%	180	143
E 3	325	3.5%	215	190
E 4	780	5.7%	536	486
E 5	415	1.5%	286	242

Tras realizar la limpieza de datos, se llevó a cabo un análisis de los mismos, con el objetivo de calcular las métricas capaces de caracterizar cada empleado mediante una serie de atributos. Para ello fueron calculadas las ya menciona-

das métricas usadas para la caracterización de los miembros obteniéndose los resultados en cuanto a etiquetas lingüísticas que se muestran en la tabla 5.

TABLA 5. RESULTADOS PRELIMINARES (E: EMPLEADO)

	Experiencia en análisis	Experiencia en desarrollo	Fiabilidad	Rendimiento
E 1	alta	baja	media/alta	medio
E 2	media	baja	baja	bajo
E 3	baja	media/alta	media	medio/bajo
E 4	alta	alta	alta	alto
E 5	media	baja	media/alta	media

Por último se ha realizado la agrupación de semanas a partir de los datos del tiempo trabajado en cada una de ellas mediante la aplicación de un algoritmo de *clustering*, detectándose tres tipos distintos de semanas (figura 1), las cuales se han podido caracterizar como:

- Semanas con entrega (64%), en la que se trabaja más, existiendo un pico de trabajo los días de la entrega.
- Semanas sin entrega (20%), en la que los tiempos de trabajo son más constantes y bajos.
- Semanas de escaso trabajo, incompletas o resultado de una semana anterior de estrés debido a festividades, especialmente en lunes y viernes (14%).

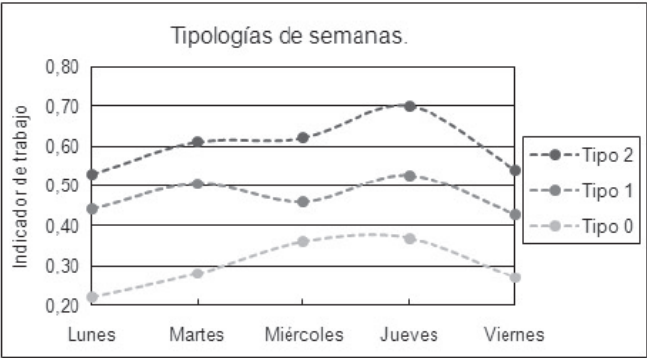


FIGURA 1. TIPOS DE SEMANAS SEGÚN NIVEL DE TRABAJO

Con este conocimiento se puede inferir que las horas trabajadas al día siguiente de la entrega son inversamente proporcionales a las realizadas el día que se efectúa la entrega, independientemente de las tareas pendientes que puedan existir.

5. Conclusiones y trabajo futuro

En este trabajo se ha presentado una serie de técnicas enmarcadas dentro del descubrimiento de conocimiento que permiten el modelado del trabajo de los desarrolladores *software* en lo que se refiere a la distribución de su tiempo de trabajo.

Con ello se han conseguido definir comportamientos en cuanto a tipología de días o semanas de trabajo, o características propias del profesional mediante el uso de técnicas como los algoritmos de clustering, la lógica borrosa o las Categorías Prototípicas Deformables Borrosas.

Como consecuencia lógica del conocimiento extraído el próximo paso es la aplicación real de este conocimiento en la planificación y estimación de nuevos proyectos, sobre todo con el objetivo de que la planificación sea flexible y permita responder a crisis e imprevistos de forma eficiente. Además se pretende extraer conocimiento de datos procedentes de otros estudios, de simuladores y de experimentos controlados con el fin de poder definir de forma más completa los posibles comportamientos de los componentes de los equipos de desarrollo.

6. Referencias

- [1] W. Humphrey. "El Proceso de Software Personal". 16th edn. Volume 1. Prentice Hall, College Station, Texas, 2001.
- [2] B. W. Boehm. "Software Cost Estimation with COCOMO II". Volume 1. Prentice Hall, 2000.
- [3] T. Abdel-Haid, S. Madnick. "Software project dynamics". An integrated approach. Volume 1. Prentice Hall Software Series, 1991.
- [4] C. Moore. "Project leap: Personal process improvement for the differently disciplined". In: The 21st International Conference on Software Engineering, Springer, 1998.

- [5] P. Johnson, H. Koe, J. Agustin, C. Chan. "Beyond the personal software process, metrics collection and analysis for the differently disciplined". In: 25th International Conference on Software Engineering, Springer, 2003.
- [6] M. Ruiz, I. Ramos. "A simplified model of software project dynamics". The Journal of Systems and Software 59. 209-309, 2001.
- [7] B. Kitchenham. "Procedures for performing systematic reviews". Technical report, Software Engineering Group, Department of Computer Science, Keele University and Empirical Software Engineering National ICT Australia, 2004.
- [8] J. Biolchini, P. Gomes, A. Cruz. Travassos, G. "Systematic review in software engineering". Technical report, Systems Engineering and Computer Science Department, UFRJ: Rio de Janeiro, Brazil, 2005.
- [9] I. Ramos. "Modelado dinámico y aprendizaje automático aplicado a la gestión de proyectos software". Revista de Procesos y Métricas de las Tecnologías de la Información. 21-28, 2004.
- [10] F. Padberg. "A software process scheduling simulator". In: 25th International Conference on Software Engineering, Springer. 816-817, 2003.
- [11] U. Fayyad, G. Piatetsky-Shapiro, P. Smith. "The KDD process for extracting useful knowledge from volumes of data". Communications of the ACM 39, 1996.
- [12] L. Prechelt, B. Unger. "An experiment measuring the effects of personal software process (psp) training". IEEE Transactions on Software Engineering, 20-33, 2000.
- [13] P. Abrahamsson, K. Kautz. "Personal software process: Classroom experiences from Finland". In: European Conference on Software Quality, Springer. 175-185, 2002.
- [14] J. Olivas. "Contribución al estudio experimental de la predicción basada en Categorías deformables borrosas". PhD thesis, Universidad de Castilla La Mancha, Ciudad Real, 2000.