



## GESTIÓN DEL CONOCIMIENTO PARA LA ELABORACIÓN DE UN MODELO FORMAL DE ASIGNACIÓN DE PERSONAL A EQUIPOS DE PROYECTOS DE SOFTWARE

### Resumen / Abstract

La formación de los equipos de proyectos de software se realiza, por lo general, de forma empírica. Sin embargo, en este proceso se deben considerar múltiples factores. En la literatura resultan escasos los trabajos donde se modele este proceso. La mayor parte de ellos constituyen propuestas que no consideran ningún aspecto vinculado a la formación del equipo como un todo. En este trabajo se describen los principales resultados del proceso de gestión de conocimientos realizado para obtener los elementos considerados en un modelo formal para la asignación de personal a equipos en proyectos de software. Estos elementos fueron identificados mediante el uso del método Delphi y la aplicación de tests.

*The formation of software project teams is carried out, in general, in an empiric manner. However, in this process should consider multiple factors. In literature the works where this process is modeled are scarce, and most of them do not consider any aspect linked to the formation of the team as a whole. In this work, the main results of the process of knowledge management carried out to obtain the elements considered in a formal model for the assignment people to teams in software projects is described. These elements were identified through the use of the Delphi method, and of the application of psychological tests.*

### Palabras clave / Key words

Gestión de conocimientos, método Delphi, tests psicológicos, modelo formal, asignación de personal, equipo de proyecto de software.

*Knowledge management, Delphi method, psychological tests, formal model, assignment people, software project team.*

**Margarita André Ampuero,**  
Departamento de Ingeniería de  
Software, Instituto Superior  
Politécnico "José Antonio  
Echeverría", CUJAE, Ciudad de La  
Habana, Cuba.  
e-mail: mayi@ceis.cujae.edu.cu

**María Gulnara Baldoquín de la Peña,**  
Departamento de Matemática, Instituto  
Superior Politécnico "José Antonio  
Echeverría", CUJAE, Ciudad de La  
Habana, Cuba.  
e-mail: gulnara@ind.cujae.edu.cu

**Silvia Teresita Acuña Castillo,**  
Universidad Autónoma de Madrid,  
Madrid, España.  
e-mail: silvia.acunna@uam.es

## INTRODUCCIÓN

Muchas investigaciones reconocen que los recursos humanos juegan un papel crítico en el éxito o fracaso de un proyecto de software [1; 2; 3; 4]. Sin embargo, el personal continúa siendo el factor menos formalizado en los modelos de procesos, los cuales se centran más en aspectos técnicos [5; 6]. La asignación al proyecto de personas sin las competencias necesarias y los problemas entre los miembros del equipo de proyecto, se detectan como dos de las principales dificultades asociadas con factores humanos que afectan el éxito de los proyectos de software [7; 8]. En general, la asignación de personal a los roles de un proyecto y la conformación del equipo se realiza de forma empírica, con su correspondiente impacto en la calidad de los proyectos. Esta situación se torna más compleja en organizaciones medianas y grandes, donde la cantidad de combinaciones de asignaciones posibles, en

dimensiones relativamente significativas de empleados disponibles y roles a cubrir, puede convertirse en un problema con un espacio de soluciones demasiado grande. Esto hace que esta etapa sea prácticamente imposible de abordar de manera eficiente, sin la ayuda de sistemas automatizados de soporte a la decisión que se basen en algoritmos de solución de modelos matemáticos que representen el problema a resolver lo más objetivamente posible.

Lo anterior pone de manifiesto la importancia de contar con un modelo formal que, insertado en un sistema de soporte a la decisión, apoye el proceso de asignación haciéndolo más objetivo. Sin embargo, si bien el problema de asignación ha sido objeto de estudio desde hace varias décadas, resultan escasos los trabajos donde se modele la asignación de personal en el ámbito del software y desde la perspectiva del equipo.

En [2; 9; 10; 11] se describen modelos que abordan la asignación de personal a proyectos de software. Estas propuestas tienen en común que no consideran ningún factor asociado a la formación del equipo como un todo, centrándose en la asignación individual de personas a tareas o roles del proyecto.

En el trabajo se exponen los principales resultados del proceso de gestión del conocimiento realizado para obtener los elementos esenciales que posibilitan la elaboración de un modelo formal, haciendo uso del método Delphi, como método de consulta a expertos, y de la aplicación de tests psicológicos y herramientas de minería de datos para identificar patrones útiles en la formación de los equipos de proyectos software. El artículo está estructurado como sigue: en la Sección 2 se discuten trabajos relacionados con la temática de asignación de personal a proyectos de software, la Sección 3 presenta el proceso de gestión del conocimiento empleado en el modelado de la asignación de personal a equipos en proyectos de software y finalmente, en la Sección 4 se plantean las conclusiones del trabajo.

## TRABAJOS RELACIONADOS

La asignación de personal a proyectos de software ha sido un tema poco abordado en la literatura. De hecho, de manera general, los modelos de procesos y las metodologías de desarrollo de software se centran más en aspectos técnicos que en cuestiones humanas u organizacionales. En este sentido, reconocidos modelos de procesos como: *People-CMM* (P-CMM) [12], Proceso de Software Personal (PSP) [13], Proceso de Software en Equipo (TSP) [14] y el Proceso Unificado de *Rational* (RUP) [15], aunque incorporan el factor humano, no modelan el proceso de asignación de personal al proyecto, ni formalizan las competencias necesarias para el desempeño de los roles. A continuación, se analizan brevemente trabajos donde se proponen modelos de procesos para la asignación de personal a proyectos de software.

En [2] se propone un proceso de gestión de recursos humanos en proyectos de desarrollo de software basado en el reuso del conocimiento organizacional de las competencias y las asignaciones previas del personal. Acorde al proceso, los jefes de proyecto asignan personas a cada tarea del proyecto, tomando en cuenta el perfil definido lo que incluye tanto competencias, como experiencias y formación académica en base a la norma NBR ISO10006:2000. Esta norma también recomienda considerar en la asignación los intereses y las

relaciones interpersonales entre los miembros del equipo de proyecto, factores que no se toman en cuenta en el proceso. El proceso propuesto no está soportado por un modelo formal.

En [11] se propone un método de planificación de *releases* (versión del producto que se obtiene en cada iteración de construcción, según RUP) para desarrollar software de manera incremental. El método asigna funcionalidades a los *releases*, tomando en consideración los recursos técnicos, los riesgos y las restricciones de presupuesto, y busca maximizar el valor ganado (contribución de la funcionalidad al ser asignada al *release*). En la asignación se toma en cuenta la productividad de los recursos humanos para ejecutar los diferentes tipos de tareas. El método sólo resulta aplicable en organizaciones de software maduras que cuenten con procesos bien definidos y medidos.

En [10] se aborda la asignación de personal a proyectos de software teniendo en cuenta: las competencias de los empleados para desempeñar tareas, y la caracterización de los proyectos en cuanto a las tareas en las que se descompone y las competencias que son necesarias para llevarlas a cabo. La propuesta también toma en consideración la planificación de las tareas, así como la disponibilidad y el costo de los empleados. La asignación de personal se realiza a través de una herramienta flexible que soporta el modelo formal propuesto.

Estas propuestas caracterizan los proyectos, o etapas de éstos, no en términos de los roles necesarios para llevarlos a cabo, sino en términos de las tareas necesarias para completarlos. La capacidad de desempeño de los empleados también queda caracterizada en base a la capacidad para ejecutar las tareas objeto de asignación. Sin embargo, este enfoque, en general, es más complejo y presupone contar con una planificación de las tareas en etapas muy tempranas del proyecto, situación difícil de lograr en organizaciones no maduras. Adicionalmente, establecer cronogramas previo a la asignación, puede provocar cambios en la planificación y en el equipo en función de los niveles reales de competencia del personal asignado [16], e inhabilita al equipo de participar en la elaboración del plan del proyecto. Sin embargo, como se plantea en [3], resulta muy importante para el éxito del proyecto contar desde un inicio con el equipo completo y con las personas adecuadas. Además, la participación temprana del equipo en la planificación permite fortalecer el compromiso con el proyecto y elaborar planes más realistas [16].

Este trabajo propone caracterizar los proyectos y la capacidad de los empleados en términos de los roles necesarios para llevar a cabo el proyecto. Al definir cada rol se le asocian un conjunto de tareas.

El Modelo de Proceso de Software Orientado a las Capacidades [9] incluye, como elemento original, las características de la conducta profesional o capacidades de comportamiento (competencias genéricas), y propone dos procedimientos: uno para determinar las capacidades de los miembros del equipo, y otro para asignar personas a los roles en dependencia de sus capacidades y las requeridas por los roles. Los resultados obtenidos en su aplicación demuestran la efectividad de considerar las competencias genéricas en el proceso de asignación. Sin embargo, no se consideran las competencias técnicas y el procedimiento de asignación de los empleados a proyectos se hace basado en criterios

## GESTIÓN DEL CONOCIMIENTO PARA LA ELABORACIÓN DE UN MODELO FORMAL DE ASIGNACIÓN DE PERSONAL A EQUIPOS DE PROYECTOS DE SOFTWARE

estadístico empíricos y no a través de la modelación matemática como se plantea en esta investigación.

Las propuestas evaluadas consideran diferentes factores en la asignación. Sin embargo, resulta común el hecho de no considerar en la asignación ningún factor que contribuya a la formación del equipo como un todo, logrando un adecuado balance, y tomando en consideración las relaciones interpersonales entre sus miembros, como se sugiere en [16; 17]. Lo anterior pone de manifiesto la conveniencia de elaborar un modelo formal que tome en consideración la mayor cantidad de factores posibles y que contribuya, tanto a la asignación individual a los roles del proyecto, como a la formación del equipo como un todo. A continuación, se describen los principales resultados del proceso de gestión de conocimientos llevado a cabo para la deducción de un modelo formal que cumpla con dichos requisitos.

### PROCESO DE GESTIÓN DE CONOCIMIENTOS

En esta sección se enuncian los principales resultados del proceso de gestión de conocimientos desarrollado con el objetivo de obtener los principales elementos a tomar en cuenta en el modelo de asignación, los cuales fueron identificados mediante el uso del método Delphi, y de la aplicación de *tests* psicológicos.

El método Delphi se aplicó en cuatro etapas con diferentes objetivos. En la primera etapa, dado que el éxito de la utilización de los métodos de expertos en un proceso de Ingeniería de Conocimiento depende, entre otros factores, de la adecuada selección de los expertos a encuestar, se utilizó con el objetivo de obtener los criterios para seleccionar a los expertos a consultar en las etapas restantes. En la segunda etapa se empleó el método Delphi, con el propósito de obtener los principales factores a considerar en el modelo de asignación y en la tercera etapa, se utilizó con el objetivo de elaborar una propuesta de roles invariantes y competencias que sirva de base en la implementación del modelo de asignación.

#### Principales resultados de la aplicación del método Delphi en las tres primeras etapas

En la primera etapa, participaron como expertos doce profesionales de siete instituciones cubanas con al menos veinte años de experiencia en el desarrollo y la dirección de proyectos de software. Al concluir las dos rondas desarrolladas, se obtuvieron como criterios para seleccionar a los expertos los que se enuncian a continuación:

- Contar con diez o más años de experiencia participando activa y exitosamente en el desarrollo de software.
- Poseer conocimientos de ingeniería de software y de técnicas de dirección de proyectos.
- Haber dirigido exitosamente como mínimo a tres personas en al menos dos proyectos de media o alta complejidad.

En la segunda etapa, se obtuvieron los factores más importantes a tener en cuenta en el modelo de asignación. Éstos fueron: las competencias (técnicas, genéricas, experiencia en proyectos similares y experiencia en el rol) para desempeñar los diferentes roles del proyecto de software, la sinergia del equipo, la disponibilidad del personal (en función de su carga), y el costo por lejanía (para el caso de desarrollos a distancia). La votación de los expertos mostró una correspondencia total entre necesidad

de considerar los factores y la prioridad que asignaron a dichos factores. Así, los factores que obtuvieron la mayor prioridad, en ese orden, fueron:

- Maximizar competencia de los empleados
- Maximizar sinergia del equipo
- Balancear la carga de trabajo
- Minimizar el costo por lejanía

El concepto de sinergia de equipo utilizado en este trabajo fue dado por la Real Academia de la Lengua Española que plantea: "Sinergia es acción y creación colectivas; es unión, cooperación y concurso de causas para lograr resultados y beneficios conjuntos; es concertación en pos de objetivos comunes".

A partir de los resultados obtenidos en la tercera etapa de la aplicación del método Delphi, se elaboró una propuesta que incluyó: roles invariantes para enfrentar proyectos de desarrollo e implantación de software, competencias requeridas para su desempeño (tanto técnicas como genéricas), así como el nivel de importancia de cada competencia para el desempeño de cada rol.

Como resultado de los criterios expresados por los expertos en la segunda etapa, se identificó la sinergia del equipo como un factor de suma importancia a tomar en cuenta durante el proceso de asignación de personal a proyectos de software. Sin embargo, existen pocos estudios, en el ámbito del desarrollo de software, acerca del impacto que tienen los factores de grupo como: la cohesión, el conflicto, la estructura del equipo y la coordinación [1]. En [18] se propone determinar el índice de cohesión o conflicto de un equipo en función de las relaciones recíprocamente positivas o negativas entre sus miembros. Tomando en cuenta que esta propuesta resulta factible de aplicar y que no se cuenta con un método para determinar la sinergia del equipo (por lo complejo que resulta evaluar este factor), se propone que, en el modelo de asignación, se incorpore como objetivo, minimizar las incompatibilidades entre los miembros del equipo en lugar de maximizar la sinergia.

Con el propósito de identificar e incorporar en el modelo de asignación patrones que contribuyan a la formación de equipos de proyectos de software se desarrolló un experimento que consistió en la aplicación de un conjunto de *tests* psicológicos y la aplicación del método Delphi en una cuarta etapa. Los patrones a identificar debían favorecer la formación del equipo en tanto contribuyeran: a balancear el equipo, a lograr una mejor asignación a los roles establecidos, y a evitar incompatibilidades entre los miembros.

#### Patrones para la formación de equipos en proyectos software

Como parte del experimento se aplicaron *tests* psicológicos a 336 personas, todas vinculadas al desarrollo de software, pertenecientes a 25 organizaciones. De ellas, 146 (44%) profesionales que laboran en organizaciones de software y 190 (56%), personas que pertenecen a la academia (79 profesores y 111 estudiantes del último año de la carrera de Informática de dos universidades: el Instituto Superior Politécnico "José Antonio Echeverría" (CUJAE) y la Universidad de Ciencias Informáticas (UCI)). En el trabajo participaron psicólogos del Grupo de Diagnóstico del Centro de Formación y Desarrollo del Capital Humano (FORDES)

perteneciente al Ministerio de la Informática y las Comunicaciones.

La batería de *tests* aplicada estuvo formada por seis cuestionarios: Myers-Briggs [19], 16PF [20], Belbin [21], los cuales constituyen tres de los *tests* más utilizados en las investigaciones que abordan la formación de equipos de proyectos de software [22; 23]. Los cuestionarios LPC0.2 y LPC0.3, los cuales fueron diseñados por los psicólogos con el objetivo de identificar patrones de rechazo basado en los factores del 16PF y en las dimensiones del Myers-Briggs, respectivamente. El último cuestionario se diseñó para caracterizar a los participantes y recoge: datos generales (sexo, edad, carrera, estudiante/profesional, organización), experiencia (tanto en el desarrollo como en la dirección de proyectos), así como preferencias y rechazo en función del desempeño de roles funcionales.

**Breve caracterización de los tests seleccionados**

El Indicador de Tipo de Myers-Briggs (MBTI) es un *test* de autopercepción que mide cuatro dimensiones diferentes de las preferencias humanas: Extroversión (E)-Introversión (I), Intuición (N)-Sentidos (S), Emoción (F)-Pensamiento (T), y Juicio (J)-Percepción (P). A partir de los valores de cada dimensión se identifica el tipo psicológico de la persona (de los 16 tipos posibles) [19]. En investigaciones de software el *test* ha sido utilizado, fundamentalmente, para identificar: el tipo o el patrón que predomina en el personal del software [24; 25], los tipos más adecuados para desempeñar roles [4] en el proceso de desarrollo, para evaluar la factibilidad o no de la diversidad de tipos en el equipo [26], el impacto de las distintas dimensiones en los resultados del proyecto y en la cohesión del equipo [27; 28].

El *test* 16 Factores de la Personalidad (16PF) es un instrumento de valoración objetiva, elaborado mediante investigación psicológica, con el fin de ofrecer, en el menor tiempo posible, una visión muy completa de la personalidad. Se basa en la medición de 16 dimensiones funcionalmente independientes y psicológicamente significativas. Los 16 factores se identifican con letras que van de A a Q4 (A-Afabilidad, B-Razonamiento, C-Estabilidad Emocional, E-Dominancia, F-Animación, G-Atención a normas, H-Atrevimiento, I-Sensibilidad, L-Vigilancia, M-Abstracción, N-Privacidad, O-Aprensi3n, Q1-Apertura al cambio, Q2-Autosuficiencia, Q3-Perfeccionismo, Q4-Tensi3n). Adem3s, cada uno de los factores tiene un polo negativo (-) y uno positivo (+) en funci3n de la puntuaci3n [20]. Por ejemplo, una persona con estabilidad - (C-) es considerada reactiva e insegura, mientras que C+ describe una persona estable, adaptada y madura. Su aplicaci3n para la evaluaci3n de rasgos de personalidad se realiza, principalmente, en tareas de selecci3n y clasificaci3n del personal. En [5] se propone como v3a para evaluar las competencias gen3ricas del personal a ser asignado a los roles de proyectos de software. Belbin es un *test* de autopercepci3n que permite identificar los roles de equipo preferidos y evitados de cada persona. Meredith Belbin, su creador, define rol de equipo como "nuestra particular tendencia a comportarnos, contribuir y relacionarnos socialmente" e identifica nueve roles clasificados en tres categor3as [21]:

- Roles Mentales (RM): Cerebro-CE, Monitor-Evaluador-ME y Especialista-ES.
- Roles de Acci3n (RA): Impulsor-IS, Implementador-ID y Finalizador-FI.

- Roles Sociales (RS): Coordinador-CO, Cohesionador-CH e Investigador de Recursos-IR.

Las categor3as representan las dimensiones del grado de orientaci3n de las personas hacia el desempe1o de tareas (roles de acci3n), hacia el mundo de las ideas (roles mentales) o hacia las relaciones con las personas (roles sociales). El *test* ha sido utilizado en investigaciones vinculadas al software para evaluar el impacto de diferentes roles de Belbin en un equipo de proyecto [29; 30; 31].

**Principales resultados del experimento**

El *test* de Belbin se proces3o utilizando dos procedimientos: el cl3sico (que identifica roles primarios y secundarios en funci3n de la jerarqu3a), y una variante que toma en cuenta ambos par3metros: jerarqu3a y puntuaci3n. Con esta variante se limita el hecho de establecer la preferencia por el desempe1o de un rol a una puntuaci3n superior a 9. Esto permite lograr mayor precisi3n en la preferencia. Acorde a esta variante se identifican como preferidos: roles primarios, secundarios y terciarios, acorde a las siguientes reglas:

- Rol primario: jerarqu3a = 1 y puntuaci3n ≥ 18.
- Rol secundario: jerarqu3a = 1 6 2 y 15 ≤ puntuaci3n < 18
- Rol terciario: jerarqu3a = 1, 2 6 3 y 10 ≤ puntuaci3n < 15

Al culminar el procesamiento de los *tests*, se dispon3a de un gran volumen de datos, por lo que la b3squeda de patrones se tornaba una tarea muy compleja. Como consecuencia, se decidi3o aplicar herramientas de miner3a de datos. *Weka* result3o la herramienta seleccionada, por ser una aplicaci3n de c3digo abierto, de libre distribuci3n, y por no comprometerse con una metodolog3a en particular. El primer paso fue la etapa de preparaci3n de los datos e incluy3o: limpieza (con el prop3sito de eliminar el mayor n3mero posible de datos err3neos o inconsistentes) y transformaci3n (con el prop3sito de presentar los datos de la manera m3s apropiada para la miner3a). Para lograrlo, se solicit3o a cada persona que completara o corrigiera sus respuestas en caso de errores u omisiones. Las transformaciones se realizaron con la ayuda de la herramienta *Kettle*, y sus principales prop3sitos fueron: crear nuevos atributos y cambiar su tipo de modo que resultaran m3s f3cil de comprender o de procesar por la herramienta de miner3a de datos utilizada. Adem3s, como parte del experimento se aplic3o el m3todo Delphi, en una cuarta etapa, con vistas a identificar la contribuci3n de los roles de Belbin a los roles funcionales invariantes propuestos en la tercera etapa (ver Tabla 1).

**TABLA 1**  
**Contribuci3n de los roles de Belbin a los roles funcionales**

Roles Funcionales	Roles de Belbin								
	Mentales			Acci3n			Sociales		
	C E	E S	M E	I S	I D	F I	C O	C H	I R
Jefe de Proyecto				X			X		
Dise1ador Gr3fico		X				X			
Gestor de Cambios			X			X		X	
Gestor de Configuraci3n						X		X	
Arquitecto	X		X	X					X
Analista	X	X							
Dise1ador	X	X							
Dise1ador de Base de Datos		X				X			
Programador	X	X			X	X			
Probador					X	X			
Especialista de Seguridad		X							X
Especialista de Calidad				X		X		X	
Documentador					X	X		X	

## GESTIÓN DEL CONOCIMIENTO PARA LA ELABORACIÓN DE UN MODELO FORMAL DE ASIGNACIÓN DE PERSONAL A EQUIPOS DE PROYECTOS DE SOFTWARE

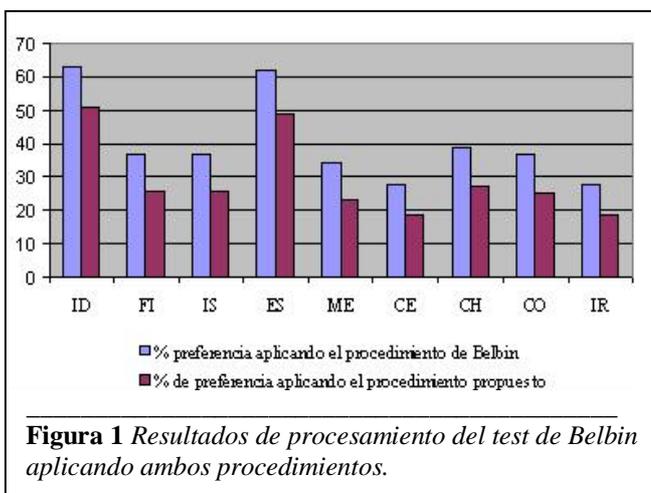
Adicionalmente, se realizó un análisis más detallado de los resultados alcanzados por los miembros de seis equipos de proyectos catalogados como exitosos y de alta complejidad (por su tamaño y el uso de nuevas tecnologías). La Tabla 2 muestra un resumen del análisis realizado.

A partir de los resultados obtenidos por el procesamiento de los *tests* (utilizando la herramienta *Weka*) y evaluando los criterios expresados por los expertos como resultado de la cuarta etapa de la aplicación del Delphi, se identificaron los siguientes patrones. Estos patrones como se observa en la Tabla 2, se cumplen en los seis equipos evaluados.

**Patrón 1:** Se requiere la presencia de todas las categorías de roles de Belbin, con predominio en la preferencia por los roles de acción sobre los roles mentales, y de los mentales sobre los sociales.

Este comportamiento se identifica como patrón a partir de evaluar el resultado de procesar el *test* de Belbin aplicando ambos procedimientos (ver Figura 1). Adicionalmente, el patrón se cumple en cada uno de los proyectos analizados en la Tabla 2 y está presente en la opinión emitida por los expertos que se resume en la Tabla 1.

<b>TABLA 2</b>						
<b>Análisis de equipos de proyectos</b>						
Criterios	Equipo de proyecto					
	1	2	3	4	5	6
Total de miembros	17	10	11	7	9	9
<b>Resultados del MBTI</b>						
Cantidad por tipos (MBTI)	7	6	8	5	6	6
Preferencia por dimensión	E (88%) S (71%) T (59%) J (71%)	E (70%) S (80%) T (90%) J (80%)	E (64%) S (73%) T (73%) J (55%)	E (57%) S (100%) T (86%) J (57%)	E (56%) S (67%) T (100%) J (67%)	E (88%) S (71%) T (59%) J (71%)
Patrones Significativos Patrón (%)	ST(70%) SJ(70%) TJ(70%)	ST(70%) SJ(70%)	ST(55%)	ST(86%)	ST(67%) TJ(67%)	SJ(67%)
Jefe de Proyecto (MBTI)	ESTJ	ENTJ	ESTJ	ESTJ	ESTJ	ENTJ
<b>Resultados de Belbin</b>						
Roles por categorías	<b>Preferencia por rol</b>					
ID - FI - IS	13 - 6 - 2	7 - 4 - 2	9 - 3 - 2	7 - 4 - 2	13 - 6 - 1	7 - 7 - 1
CE - ES - ME	8 - 4 - 6	2 - 2 - 5	2 - 2 - 7	2 - 2 - 3	3 - 2 - 6	3 - 2 - 6
CO - CH - IR	2 - 6 - 3	1 - 2 - 1	1 - 5 - 1	1 - 2 - 1	1 - 3 - 1	1 - 4 - 1
Categorías de roles (RA - RM - RS)	<b>Balance entre categorías de roles</b>					
Jefe de Proyecto (roles preferidos)	21>18>11	13>9>4	14>11>7	13>8>4	14>11>5	15>11>6
	<b>Roles de liderazgo preferidos</b>					
	CO	IS	IS	IS	IS	CO



La poca diferencia entre el predominio de los roles de acción y los mentales se explica por la propia naturaleza del trabajo. En el desarrollo de software es preciso que prevalezca la acción (IS, ID, FI) para que el proyecto termine en tiempo y con calidad. Sin embargo, la ingeniería de software es una disciplina que exige creatividad. La opinión de los expertos

(Tabla 1) corrobora este planteamiento, ya que los roles funcionales que los expertos asociaron con tareas esenciales como el análisis, el diseño y la programación, están asociados tanto a roles de acción como a mentales.

Resulta importante señalar que si bien Belbin apuntó la necesidad de prevalencia de los roles de acción, sólo alertó que el resto de las categorías (sociales y mentales) no debían estar sobre representadas. Por tanto, resulta nueva la precisión acerca del balance entre las categorías de roles.

**Patrón 2:** Se requiere la presencia de al menos una persona con preferencia por el rol Cerebro (CE) en el equipo.

Este experimento permite concluir que la presencia del rol Cerebro resulta imprescindible para un equipo de proyecto de software. Como se observa en la Tabla 1, hubo concordancia en el criterio de los expertos sobre la contribución del rol CE en el desempeño de los principales roles funcionales de un proyecto: arquitecto, analista, diseñador y programador. Además, como se observa en la Tabla 2 en todos los equipos analizados existen al menos dos personas que tienen preferencia por el rol CE. Al realizar un análisis detallado de los roles funcionales que desempeñan

las personas que en los equipos evaluados mostraron preferencia por el rol CE, se comprueba que coinciden con roles claves como el arquitecto, el analista, el diseñador y el programador. El equipo 1, encargado del proyecto más complejo de todos los evaluados, incluso cuenta con ocho personas con preferencia por este rol.

**Patrón 3:** Al asignar el Jefe de proyecto se debe verificar la preferencia por uno de los siguientes roles: Impulsor y Coordinador. Además, comprobar de su tipo psicológico, según Myers-Briggs, cumple el patrón ExxJ (Extroversión (E)-Juicio (J)).

Los expertos concuerdan que los roles de equipo Coordinador (CO) e Impulsor (IS) contribuyen al desempeño del rol funcional Jefe de Proyecto (ver Tabla 1). Esta relación confirma que lo propuesto por Belbin también se cumple en el ámbito del software. Como se muestra en la Tabla 2, en todos los casos las personas que desempeñaron el rol de Jefe de Proyecto tenían preferencia por los roles CO y/o IS. Un estudio más detallado acerca del desempeño de doce de los Jefes de Proyecto involucrados en el experimento, permitió detectar que los que muestran preferencia por el rol IS se proyectan como impulsores desde posiciones de liderazgo técnico, a diferencia de los que prefieren CO que se muestran más como coordinadores generales del proyecto. Esta diferencia en cuanto a estilo de liderazgo ya había sido señalada en [30; 32]. Un análisis de las 42 personas que participaron en el experimento con al menos cinco años de experiencia en la dirección de proyectos, permitió comprobar que 36 de ellas (86%) tenían entre sus roles preferidos, CO y/o IS. Además, se observa una marcada preferencia por el rol IS sobre el CO. Este comportamiento sugiere que el estilo de liderazgo técnico resulta predominante en el ámbito del software.

En los proyectos analizados, se pudo constatar que todos los Jefes de proyectos cumplen el patrón ExTJ. En [4] se sugiere utilizar el patrón ENFJ para asignar un Jefe de proyecto. Sin embargo, al analizar los 42 participantes con al menos cinco años de experiencia en la dirección de proyectos, se constató que el 90% de ellos cumplen sólo el patrón ExxJ; aunque existe un predominio de los tipos ESTJ, ENTJ y ENFJ, en ese orden. Tomando en cuenta estos comportamientos, se propone establecer como patrón, útil en la asignación del Jefe de proyecto, tener el subtipo EJ, partiendo del hecho que resulta muy importante que los Jefes de proyecto sean extrovertidos y planificados.

El procesamiento de los cuestionarios 16PF y LPC0.2 arrojó información que contribuye a caracterizar el comportamiento de los profesionales del software en general. Sin embargo, no fue posible identificar patrones de incompatibilidad a partir de estos resultados. En 15 de los 16 factores del 16PF, más del 50% de los participantes tenían puntuaciones medias. Sólo en el caso del factor N, el 51% tuvo puntuación en el polo positivo. Este resultado indica que el 51% de los participantes son personas analíticas con un enfoque intelectual, lo que de manera general, caracteriza a los profesionales del software; por lo que no aporta elementos significativos a los efectos del trabajo.

Al evaluar los resultados del cuestionario LPC0.2, se detecta que más del 50% de los participantes rechaza un

conjunto de comportamientos: A-(55%), C-(51%), E-(67%), G-(72%), Q2+ (69) y Q3-(66%). Estos resultados indican que, de manera general, se rechaza a las personas dependientes, poco prácticas, no esmeradas, que no consideran la opinión del grupo ni prestan atención a sus compromisos, y que prefieren trabajar solas y no comprometerse. Estas manifestaciones de rechazo, como en el caso anterior, caracterizan a los profesionales del software y confirman la importancia de un grupo de competencias que los expertos calificaban de muy necesarias como es el caso de: el trabajo en equipo, la tenacidad, el compromiso con la organización y la independencia.

Al finalizar el experimento se pudo concluir que el uso de *tests* psicológicos puede contribuir a la formación de equipos de proyectos de software. Sin embargo, los *tests* por sí solos no identifican a los buenos ingenieros, ni sus resultados pueden predecir, de manera absoluta, la calidad de la ejecución de un trabajo. Es preciso lograr un balance entre las características psicológicas, las competencias y la experiencia del personal [27; 28; 32].

## CONCLUSIONES

El proceso de gestión de conocimientos realizado permitió identificar un conjunto de elementos significativos para la elaboración de un modelo formal de asignación de personal a equipos de proyectos de software:

- La aplicación del método Delphi, como método de consultas a expertos, permitió identificar los principales factores a considerar en la elaboración de un modelo formal de asignación de personal a equipos de proyectos de software, elaborar una propuesta de roles invariantes y competencias necesarias para enfrentar proyectos de software, y obtener criterios acerca de la contribución de los roles de Belbin a los roles funcionales del proyecto, útil en la identificación de patrones.
- La aplicación de los *tests* psicológicos y la herramienta de minería de datos permitió identificar patrones que contribuyen a la formación de equipos de proyectos de software como: el adecuado balance entre las categorías de roles de equipo, la presencia del rol Cerebro y las exigencias para el desempeño del rol Jefe de proyecto (preferencias por los roles Impulsor y/o Coordinador y la pertenencia al subtipo psicológico EJ que denota preferencia por la Extroversión y la Planificación). 🏠

## REFERENCIAS

1. ACUÑA, S. T., GÓMEZ, M. y JURISTO, N. "Towards understanding the relationship between team climate and software quality-a quasi-experimental study". *Empirical Software Engineering*. 13(4): 401-434, 2008.
2. DE CARVALHO, L. R. "Planejamento da alocação de recursos humanos em Ambientes de desenvolvimento de software orientados à Organização". Tese para a obtenção do grau de mestre em ciências em engenharia de sistemas e computação. Rio de Janeiro: COPPE, Universidade Federal do Rio de Janeiro, 2003.

## GESTIÓN DEL CONOCIMIENTO PARA LA ELABORACIÓN DE UN MODELO FORMAL DE ASIGNACIÓN DE PERSONAL A EQUIPOS DE PROYECTOS DE SOFTWARE

3. DE MARCO, T. y LISTER, T. *Peopeware: Productives Projects and Teams*. [s.l.]: Dorset House, 1999.
4. GORLA, N. y WAH, Y. W. "Who should work with whom?: building effective software project teams". *Communications of the ACM*. 47(6): 79-82, 2004.
5. ACUÑA, S.T, JURISTO, N., y MORENO, A. M. "Emphasizing Human Capabilities in Software Development". *IEEE Software*. 23(2): 94-101,
6. ANDRÉ, M., BALDOQUÍN, M. G., ACUÑA, S. T. y ROSETE, A. "A formalized model for the assignment of human resources to software projects". En: *XIV Congreso Latino Ibero Americano de Investigación de Operaciones (CLAIO 2008)* (Cartagena de Indias, Colombia: J. Amador, C. Paternita, J. Velázquez, 2008)
7. CHARETTE, R. N. "Why software fails". *IEEE Spectrum*. 42(9): 42-49, 2005.
8. RYAN, R. "IT Project Management: Infamous Failures, Classics Mistakes, and Best Practices". *Mis Quarterly Executive*. 6(2): 67-78, 2007.
9. ACUÑA, S.T. y JURISTO, N. "Chapter 5: Software Process Modeling: Socio-Technical Perspectives". En: *Software Process Modeling*. [s.l.]: Springer, 2005. 111-139
10. BARRETO, A.S. "Apoio à Decisão Gerencial na Alocação de Recursos Humanos em Projetos de Software". Tese para a obtenção do grau de mestre em ciências em engenharia de sistemas e computação. Rio de Janeiro: COPPE, Universidade Federal do Rio de Janeiro, 2003.
11. NGO-THE, A. y RUHE, G. "A Systematic Approach for Solving the Wicked Problem of Software Release Planning" *Soft Computing*. 12(1): 95-108, 2008.
12. CURTIS, B., HEFLEY, W. E. y MILLER, S. A. Capability Maturity Model (P-CMM) [Technical Report CMU/SEI-2001-MM-01]. 2.0. Software Engineering Institute, 2001.
13. HUMPHREY, W. S. *A Discipline for Software Engineering*. [s.l.]: Addison-Wesley Longman, Inc, 1995.
14. HUMPHREY, W. S. *Managing Technical People: Innovation, Teamwork and the Software Process*. [s.l.]: Addison-Wesley, 1998.
15. JACOBSON, I., BOSCH, G. y RUMBAUGH, J. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley Iberoamericana, 2000.
16. PMI. *Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. Third Edition. [s.l.]: Project Management Institute, 2004.
17. ISO. "ISO 10006:2003 Quality management systems-Guidelines for quality management in projects". [en línea]. 2003, Disponible en: <http://www.iso.org>
18. YANG, H. L. y TANG, J. H. "Team Structure and Team Performance in IS Development: A Social Network Perspective". *Information & Management*. 41: 335-349, 2004.
19. BRIGGS, I., KIRBY, L.K. y MYERS, K.D. *Introducción al Type (MBTI). Una guía para entender los resultados de su evaluación Myers-Briggs Type Indicador*. Sixth Edition. California: Consulting Psychologists Press, 2004.
20. CATTELL, R. B., CATTELL, A. K. y CATTELL, H. E. P. *Sixteen Personality Factor Questionnaire*. [s.l.]: Institute for Personality and Ability Testing, 1993.
21. BELBIN, R. M. *Management Teams: Why they Succeed or Fail*. Second Edition. [s.l.]: Butterworth Heinemann, 2004.
22. MCDONALD, S. y EDWARDS, H. M. "Who Should Test Whom? Examining the use and abuse of personality tests in software engineering". *Communications of the ACM*. 50(1): 67-71, 2007.
23. THOMSON, C. y HOLCOMBE, M. "20 years of teaching and 7 years of research: research when you teach". En: *Proceedings of 3rd South East European Formal Methods Workshop* ([s.l.]: [s.n.], 2007) 141-153.
24. CAPRETZ, L.F. "Personality types in software engineering". *International Journal of Human Computer Studies*. 58(2): 207-214, 2003.
25. VARVEL, T. et al. "Team Effectiveness and Individual Myers-Briggs Personality Dimensions". *Journal of Management in Engineering*. 20(4): 141-146, 2004.
26. RUTHERFOORD, R. H. "Using personality inventories to help form teams for software engineering class projects". *SIGCSE Bull*. 33(3): 76-76, 2001.
27. ACUÑA, S. T., GÓMEZ, M. y JURISTO, N. *Empirical study of how personality, team processes and task characteristics relate to satisfaction and software quality?* [s.l.]: ESEM 2008, 2008. 291-293.
28. KARN, J.S. y COWLING, A.J. *An Initial Study of the Effect of Personality on Group Cohesion in Software Engineering Projects*. [s.l.]: Department of Computer Science, University of Sheffield, 2004. Research Report CS-04-01.
29. HENRY, S. M. y STEVENS, K. T. "Using Belbin's Leadership Role to Improve Team Effectiveness: An Empirical Investigation". *The Journal of System and Software*. 44: 241-250, 1999.
30. KLAUS, P. "Belbin's Company Worker, The Self-Perception Inventory, and Their Application to Software Engineering Teams". Master of Science degree in Computer Science. Virginia: Virginia Polytechnic Institute, 2001.
31. STEVENS, K. T., HENRY, S. M. "Analyzing Software Team using Belbin's Innovative Plant Role". [en línea]. 2002, Disponible en: [www.radford.edu/~kstevens2/](http://www.radford.edu/~kstevens2/)
32. TURLEY, R.T. y BIEMAN, J.M. "Competencies of exceptional and nonexceptional software engineers". *Journal of Systems and Software*. 28(1): 19-38, 1995.