



PROPUESTA DE ROLES INVARIANTES Y COMPETENCIAS PARA ENFRENTAR PROYECTOS DE SOFTWARE

Resumen / Abstract

La gestión de recursos humanos basado en el enfoque por competencias, aunque muy necesaria para las organizaciones de *software*, no resulta una práctica aún generalizada. Este trabajo tiene como propósito contribuir con la implementación de esta disciplina, para ello se describe el procedimiento utilizado y los principales resultados de la aplicación del método de consulta a expertos, Delphi, en la elaboración de una propuesta de roles invariantes y de competencias para enfrentar proyectos de *software*.

The human resources management based on the focus by competences, although very necessary for the software organizations, it is not an even widespread practice. This work has as purpose to contribute with the implementation of this discipline, for it is described it the used procedure and the main results of the application from the consultation method to experts, Delphi, in the elaboration of a proposal of invariant roles and of competences to face software projects.

Palabras clave / Key words

Gestión de recursos humanos, competencias, roles, proyectos de software, método Delphi.

Human resources management, competences, roles, software's projects, Delphi method.

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

**Jorge Soler McCook y Juan Luis
Rodríguez Dáguales,** Empresa
DESOF

e-mail: soler@esc.cu,
juanl@idi.desoft.cu

Recibido: 12/05/2009

Aprobado: 15/03/2010

INTRODUCCIÓN

La industria de software ha experimentado un desarrollo vertiginoso y ha tenido un impacto creciente en prácticamente todas las ramas del desarrollo de la sociedad. Sin embargo, aún resulta significativo el número de proyectos de software que no culminan con éxito [1; 2; 3]. Entre las principales causas que conllevan a estos resultados, se identifican las asociadas con factores humanos, situación que denota una débil gestión del personal en el marco de la gestión de proyectos.

Estos resultados evidencian que la industria de software constituye un buen ejemplo de cómo el uso de la tecnología, en vez de reemplazar a las personas, confirma que el talento constituye realmente el activo más importante en cualquier organización. La gestión de ese talento, se convierte en una necesidad creciente, en especial en aquellas organizaciones que se conocen como intensivas en conocimiento, como las de software.

Sin embargo, el personal constituye el factor menos formalizado en los modelos de proceso y las metodologías de desarrollo de software. La falta de marcos de referencia (para establecer roles y competencias), unido a la falta de métodos y herramientas de software que ayuden a evaluar el desempeño laboral, dificulta que se

realice una adecuada gestión de los recursos humanos basado en el enfoque por competencias. En este trabajo se realiza una propuesta de roles invariantes y competencias para abordar proyectos de software. El trabajo se estructura como sigue: en la Sección 2 se caracteriza la gestión de recursos humanos en el ámbito del software, en la Sección 3 se describe y fundamenta la propuesta de roles invariantes y competencias que sirve de marco de referencia para la implementación en las organizaciones de software de la gestión de recursos humanos basado en el enfoque por competencias y finalmente, la Sección 4 contiene las conclusiones del trabajo.

LA GESTIÓN DE RECURSOS HUMANOS EN LAS ORGANIZACIONES DE SOFTWARE

En esta sección se realiza un análisis de los principales aspectos que caracterizan la gestión de recursos en las organizaciones de software, así como sus principales insuficiencias. Para comenzar, se enuncian los principales problemas que afectan los resultados de los proyectos con el propósito de identificar la incidencia de factores vinculados con aspectos humanos.

Principales problemas que afectan el desarrollo de los proyectos de software

El Reporte CHAOS, una de las investigaciones más citadas al caracterizar la situación de la industria de software [3], revela que sólo el 35% de los proyectos pueden ser catalogados como exitosos, considerando que terminan en tiempo, en presupuesto, con calidad y que cumplen con los requerimientos del usuario. En este sentido varios trabajos [4; 5; 6] que analizan, a partir de estudios realizados, las principales causas que conllevan a estos resultados, concluyen que los problemas humanos constituyen una de ellas.

La industria de software nacional no está exenta de estos problemas y también experimenta proyectos fuera de cronograma, presupuesto y calidad [7]. Una encuesta realizada a 45 profesionales pertenecientes a doce organizaciones de software cubanas, permitió confirmar que además de los problemas de planificación, los problemas asociados con factores humanos, constituyen uno de los de mayor incidencia en el resultado de los proyectos.

En la Tabla 1 se resumen los principales problemas que afectan el desarrollo de los proyectos de software, tanto a nivel nacional como internacional, clasificados de acuerdo a su origen en: personal, proceso, producto y tecnología. A partir de este análisis es posible concluir que los problemas de liderazgo, la asignación de personal no adecuado y los problemas entre los miembros del equipo de proyecto se detectan como tres de las dificultades comunes asociadas con factores humanos que afectan el resultado de los proyectos de software. Estas dificultades reflejan una débil gestión de los recursos humanos.

El factor humano en los proyectos de software

La gestión eficaz de un proyecto de software se centra en las llamadas cuatro P: personal, producto, proceso y proyecto, donde el orden no resulta arbitrario [2]. En este sentido, son varias las investigaciones donde se reconoce que los recursos humanos juegan un papel crítico en el éxito o fracaso de un proyecto de software [2; 8; 9; 10]. Sin embargo, el personal

continúa siendo el factor menos formalizado en los modelos de procesos y las metodologías de desarrollo de software, las cuales se centran más en aspectos técnicos que en los aspectos humanos [8; 9; 10; 11]. Lo mismo sucede en la academia, donde la mayor parte de los cursos que se imparten en carreras afines, se centran en cuestiones técnicas y en el desempeño individual [11]. La disciplina personal y de trabajo en equipo, la formación de roles, la comunicación y el liderazgo, son temas que se abordan, en muchas ocasiones, de manera teórica o no logran implementarse a plenitud en la ejecución de proyectos reales [11; 12].

TABLA 1
Principales problemas que afectan el resultado de los proyectos de software

| Categoría | Problemas |
|--------------------|--|
| Personal | Asignación de personas sin las competencias individuales necesarias |
| | Problemas con las relaciones de trabajo entre los miembros del equipo |
| | Problemas de formación y superación continua del personal |
| | Problemas de liderazgo / Insuficientes líderes |
| | Motivación |
| | Incorporación tardía de personas al proyecto |
| | Relaciones inadecuadas con los clientes |
| | Usuario no involucrado |
| | Inadecuada comunicación y soporte de la alta dirección |
| | Expectativas no realistas |
| | Inadecuadas condiciones de trabajo |
| Proceso | Problemas de planificación (pobre estimación) |
| | Abandonar el plan bajo presión |
| | Inadecuado control de los cambios |
| | Inadecuado control de configuración |
| | Insuficiente manejo de riesgos |
| | Pobre determinación de los requisitos |
| | Insuficientes recursos |
| | Pobre documentación |
| | No uso o mal uso de metodologías de trabajo |
| | Insuficientes prácticas de calidad (uso de métricas) |
| Metas no realistas | |
| Producto | Agregar tamaño y características innecesarias al producto en correspondencia con la nueva tecnología |
| | Cambios en los requerimientos del producto |
| Tecnologías | Uso de tecnología inmadura |
| | Adoptar nuevas tecnologías no adecuadas para el proyecto |
| | Introducir nuevas tecnologías en el medio del proyecto |

La falta de formalización del factor humano en los modelos de procesos y las insuficiencias en la formación académica, entran en contradicción con la necesidad que tiene la industria de prestarle una atención especial a la gestión de recursos humanos (GRH), atendiendo a un conjunto de peculiaridades que posee, como son:

- En el profesional del software confluyen condiciones propicias para que experimente gran movilidad: es especializado (por sus conocimientos y habilidades), pero a su vez generalista (por la gran cantidad y variedad de dominios de aplicación de su trabajo) y muy

PROPUESTA DE ROLES INVARIANTES Y COMPETENCIAS PARA ENFRENTAR PROYECTOS DE SOFTWARE

demandado (debido al impacto de la informatización en prácticamente todas las ramas de desarrollo de un país).

- El cambio continuo en el estado del arte en esta rama, debido al desarrollo vertiginoso de las Tecnologías de la Información y las Comunicaciones, exige un personal técnicamente competente en continua superación.
- El desarrollo de software, debido al aumento significativo en el volumen y complejidad de los productos, dejó de ser artesanal para convertirse en un trabajo esencialmente en equipo. Por lo tanto, fue preciso establecer varios roles, en su mayoría, con alta dependencia funcional. Estas exigencias, unidas a la necesidad de establecer una adecuada interacción con los usuarios, permiten caracterizar la ingeniería de software como un proceso social complejo. Como consecuencia, en el profesional del software se tornan imprescindibles habilidades como: la negociación, la comunicación, la toma de decisiones y la capacidad de trabajar en equipo.

Gestión por competencias en las organizaciones de software

Todo lo anterior muestra que la industria de software constituye un buen ejemplo de cómo el uso de la tecnología, en vez de reemplazar a las personas (como algunos sospechaban), confirma que el talento constituye realmente el activo más importante en cualquier organización. La gestión de ese talento, que los profesionales de Recursos Humanos identifican como capital humano, se convierte en una necesidad creciente, en especial en aquellas organizaciones que se conocen como intensivas en conocimiento como las de software [13]. En este sentido, se reconoce que la GRH es mucho más eficiente si se basa en las competencias individuales de los trabajadores, teniendo en cuenta que la gestión por competencia apoya los procesos de selección, contratación, capacitación, evaluación y compensación del personal [14]. De ahí, que varias investigaciones resalten lo necesario que resulta incorporar en los modelos de procesos de software la parte humana, y las interacciones en las que participan; en especial, las competencias para ejecutar los roles del proyecto, y la disciplina y organización del trabajo interpersonal [8; 15].

Sin embargo, de manera general sucede que cada gestor de equipo determina las necesidades de competencias de una manera no muy sistemática; o sea, no se basan en reglas preestablecidas, sino que utiliza sus percepciones y experiencia. Esta práctica, si bien resulta útil, debe aplicarse de conjunto con las técnicas sistemáticas, las cuales posibilitan mejorar las decisiones de los directivos [8], y contribuyen a gestionar el conocimiento de la organización.

Investigaciones para obtener marcos de referencias indican que la gestión por competencias está muy poco desarrollada y que los resultados obtenidos para evaluar el rendimiento de los profesionales son, habitualmente, muy ambiguos [16].

En las organizaciones de software cubanas se ha avanzado en la comprensión de la disciplina de gestión por competencias. Se han realizado trabajos, encaminados a la identificación de perfiles de competencias, en organizaciones como: CUBACEL, GEOCUBA, Correos de Cuba, DESOFT, CITMATEL y el Ministerio de la Informática y las Comunicaciones (MIC). Sin embargo, su implementación no

constituye una práctica generalizada debido, fundamentalmente, a la ausencia de marcos de referencia (donde aparezcan claramente definidos los roles, sus responsabilidades y las competencias requeridas para su desempeño), métodos y herramientas que soporten su aplicación. En este sentido, también incide que la introducción de esta disciplina en la formación académica aún resulta incipiente; a pesar que como se plantea por Cuesta, A. (2007) [17], la competencia laboral se constituye en elemento central y decisivo, pero que debe ir de conjunto con el proceso de formación y gestión en sistema.

PROPUESTA DE ROLES INVARIANTES Y COMPETENCIAS PARA ENFRENTAR PROYECTOS DE SOFTWARE

La elaboración de los marcos de referencia no resulta una tarea fácil, ya que las diferentes organizaciones y modelos asignan diversos nombres a los mismos roles, e incluso algunos roles se generalizan o se superponen con otros. Por otra parte los distintos tipos de proyectos exigen roles específicos. Con las competencias ocurre algo similar. Si bien se cuenta con diccionarios [18], no existe una nomenclatura y definición única [19; 20].

La falta de marcos de referencia (para establecer roles y competencias), unido a que en muchas ocasiones no se cuenta con un registro histórico actualizado de los equipos que permita re-usar el conocimiento de la organización, dificulta que se realice una adecuada propuesta de la estructura del proyecto. Esta situación afecta, sin dudas, el resto de los procesos de gestión de recursos humanos: asignación de personal, capacitación, evaluación y remuneración.

A continuación se realiza una propuesta de roles invariantes para proyectos de desarrollo e implantación de software y de las competencias requeridas para su adecuado desempeño. Esta propuesta puede servir de punto de partida y soporte en la implementación de la gestión de recursos humanos por competencias en las organizaciones de software cubanas. Para elaborar la propuesta se analizaron los roles definidos en reconocidas metodologías y modelos de proceso de desarrollo de software.

Propuestas de roles invariantes para proyectos de desarrollo e implantación de software

En este trabajo se utiliza el concepto de rol que se emplea en el Proceso Unificado de Desarrollo de Rational (RUP). Este considera que un rol es un puesto que puede ser asignado a una persona o conjunto de personas que trabajan juntas en un equipo, y que requiere habilidades y responsabilidades como: realizar determinadas actividades y desarrollar determinados artefactos. Los miembros de un equipo de proyecto generalmente cubren varios roles [21]. En la Tabla 2 se resumen los roles establecidos en cada una de propuestas metodológicas evaluadas.

A partir del estudio de las propuestas mencionadas anteriormente y de lo establecido por López, Y. y André, M. (2006) [22], se puede concluir que es posible identificar un conjunto de roles comunes o invariantes que participan en proyectos de desarrollo de software. Éstos son:

- Roles técnicos, responsables del desarrollo del sistema: Analista, Diseñador de Interfaz-Usuario,

| TABLA 2 Roles establecidos en propuestas metodológicas evaluadas |
|--|
| Proceso Unificado de Desarrollo de Rational (RUP) [21; 23] |
| Los roles de RUP se clasifican en cinco grandes grupos: <ul style="list-style-type: none"> Analistas: Analista del Proceso de Negocios, Diseñador de Negocios, Revisor del Modelo de Negocios, Analista de Sistema, Especificador de Requisitos, Revisor de Requisitos y Diseñador de la Interfaz Usuario. Desarrolladores: Arquitecto, Revisor de la Arquitectura, Diseñador, Diseñador de Cápsula, Diseñador de Base de Datos, Revisor del Diseño, Programador, Revisor del Código, Integrador. Probadores: Diseñador de Prueba, Probador. Directivos: Director de Control de Cambio, Director de Configuración, Director de Implantación, Ingeniero de Proceso, Director del Proyecto, Revisor del Proyecto. Otros: Stakeholder, Desarrollador de Cursos, Artista Gráfico, Administrador de Sistema, Documentador Técnico, Especialista en Herramientas. |
| Extreme Programming (XP) [24] |
| Programador, Cliente, Encargado de pruebas, Encargado de seguimiento, Entrenador, Consultor y Gestor. |
| Crystal [25] |
| Patrocinador Ejecutivo, Jefe de Proyecto, Experto en el Dominio, Experto de uso, Programador-Diseñador, Diseñador de Interfaz de Usuario, Probador, Programador Técnico. |
| Proceso de Software en Equipo (TSP) [15] |
| Gerente de Interfaz-Usuario, Gerente de Diseño, Gerente de Implementación, Gerente de Planeación, Gerente de Procesos, Gerente de Calidad, Gerente de Soporte Técnico y Gerente de Pruebas. |

Arquitecto (Diseñador de alto nivel), Diseñador (de bajo nivel), Diseñador de Base de Datos (puede estar incluido en el rol anterior), Programador y Probador.

- Roles de gestión, responsables de la planificación y la ejecución del proyecto: Jefe de Proyecto (incluye funciones asociadas a la Planificación y Control del proceso), Especialista de Calidad, Gestor de Configuración, Gestor de Cambios, Documentador y Especialista en Seguridad (este rol se incorpora por la importancia que tiene tomar en cuenta los requisitos no funcionales asociados a la protección y seguridad del sistema).

Sin embargo, resulta importante señalar que no necesariamente son éstos los roles a establecer en cada proyecto. Es posible desglosar o integrar estos roles como ocurre en algunas de las propuestas estudiadas e incluso se pueden definir nuevos roles acorde al tipo de proyecto en cuestión. Por otra parte, existe un conjunto de empresas cuya misión no solo consiste en realizar proyectos de desarrollo, sino que ejecutan proyectos de implantación. Para el caso de estos proyectos también resulta posible identificar tres roles invariantes: Jefe de Proyecto de Implantación, Especialista Operacional (con experiencia en el dominio del sistema) y Especialista Técnico (con experiencia en la actividad informática).

Propuesta de competencias técnicas y genéricas requeridas para el desempeño de los roles

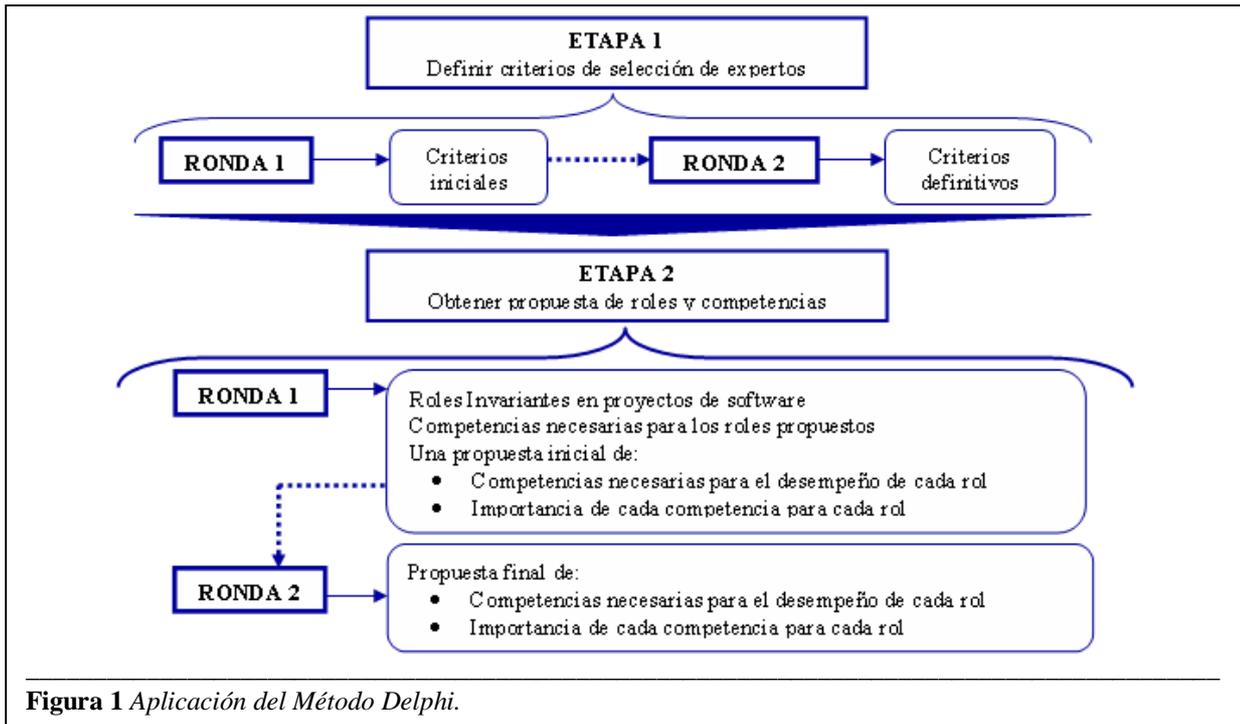
En este trabajo se utiliza el concepto de competencia laboral expuesto por Cuesta, A. (2007) [17] que plantea: “las competencias laborales son *características subyacentes* en las personas, asociadas a la experiencia, que como tendencia están *causalmente relacionadas* con *actuaciones exitosas en un puesto de trabajo* contextualizado en una determinada cultura organizacional”. Además, se distinguen dos categorías de competencias: las genéricas (que definen características referidas al comportamiento general del empleado, independientes de los conocimientos técnicos concretos. Ejemplos: negociación, planificación y organización, liderazgo, capacidad de análisis, etc.) y las técnicas o específicas (asociadas a conocimientos y habilidades técnicas específicas de cada puesto de trabajo).

Al evaluar la propuesta de roles de cada uno de los procesos y metodologías de desarrollo de software analizadas, se constató que en ningún caso se formalizan las competencias requeridas para su adecuado desempeño. En este sentido, se destaca el modelo de proceso de software propuesto por Acuña, S.T., Juristo, N. y Moreno, A.M. (2006) [8], que incluye como elemento original, las capacidades de comportamiento o características de la conducta profesional. Acuña, S. T. (2006) propone asignar las personas a roles del proyectos en función de las competencias genéricas requeridas para cada rol. Para ello las clasifica en cuatro grupos: habilidades intrapersonales, habilidades organizativas, habilidades interpersonales y habilidades directivas [8].

Tomando como base la propuesta anterior, encuestas a profesionales de la industria cubana de software, las propuestas de competencias tanto de la empresa DESOFT como del Proyecto Europeo Tuning [19], y el análisis de diccionarios de competencias [18], se proponen 26 competencias: 19 genéricas y 7 técnicas. Una vez elaborada la propuesta de roles invariantes para enfrentar proyectos de desarrollo e implantación de software, y de las competencias requeridas para el desempeño de dichos roles, se decidió someterla a la consideración de un comité de expertos para lo cual se aplicó el método Delphi en dos etapas, como se muestra en la Figura 1. El objetivo fundamental era obtener el criterio de los expertos sobre: la propuesta de roles y competencias, así como la importancia de cada competencia en el desempeño de los diferentes roles (Etapa 2). Sin embargo, como paso previo a la conformación de dicho comité, los autores del trabajo decidieron aplicar el método para obtener los criterios de selección de los expertos (Etapa 1).

En la primera etapa, después de dos rondas, se obtuvo como resultado que los expertos debían ser profesionales con conocimientos de ingeniería de software, que tuvieran 10 o más años de experiencia en el desarrollo y la dirección de proyectos de software, y que hubiesen desarrollado al menos dos proyectos de mediana o gran complejidad donde hubiesen dirigido al menos a tres personas. Posteriormente, se circuló entre los candidatos una plantilla para obtener su síntesis curricular y aplicando los criterios de selección, se creó el comité el cual está formado por 35 expertos provenientes de la industria y de la academia. A cada experto se le suministró una matriz en cuyas filas aparecían los 16

PROPUESTA DE ROLES INVARIANTES Y COMPETENCIAS PARA ENFRENTAR PROYECTOS DE SOFTWARE



roles propuestos y cuyas columnas contenían las 26 competencias. Así, los expertos debían completar 416 celdas ($16 \times 26 = 416$) con un valor entre 1 y 5: 5-imprescindible para desempeñar el rol, 4-muy útil para desempeñar el rol, 3-útil para desempeñar el rol, 2-quizás sea útil para desempeñar el rol, 1-no es importante para desempeñar el rol. Para facilitar el trabajo de los expertos, se asociaron comentarios donde se describían las principales responsabilidades de los roles y una breve definición de cada competencia. Adicionalmente, los expertos tenían la posibilidad de adicionar y/o eliminar tanto roles como competencias.

Una vez recogidas las 35 matrices se obtuvieron 14560 datos (416×35 expertos) los cuales fueron procesados con el paquete estadístico MINITAB, utilizando gráficos de *boxplots*. Estos gráficos muestran la mediana y el rango del intercuartil ya que los datos son ordinales. Por un problema de espacio no resulta posible reflejar el procesamiento realizado a los 16 roles. A continuación, se explican los pasos del procedimiento aplicado, utilizando como ejemplo el procesamiento del rol Jefe de Proyecto:

1- Procesar, para cada rol, los datos emitidos por los expertos en las 26 competencias. La Figura 2 muestra el procesamiento de los datos para el rol Jefe de Proyecto.

2- Analizar los valores *outliers* (representados por *). Tomando en cuenta la naturaleza de los datos procesados, la escala utilizada y el concepto de *outlier* que ofrece el MINITAB, los autores del trabajo definieron como *outlier* aquellos valores de la variable que diferían en al menos dos unidades de los extremos de los *boxplots*. En el caso en que el rango del intercuartil tomaba valor 1 y existían bigotes (ya fuera superior, inferior o ambos) se consideraba *outlier* el valor que difiriera en una unidad del extremo del bigote.

3- Eliminar los *outliers*. Una vez que se eliminaron los 22 *outliers* identificados para el rol Jefe de Proyecto, se procesaron nuevamente los datos y se obtuvo el gráfico que se muestra en la Figura 3.

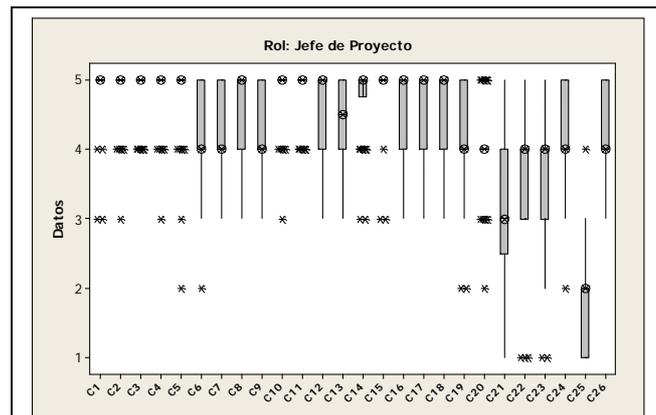


Figura 2 Análisis del rol Jefe de Proyecto.

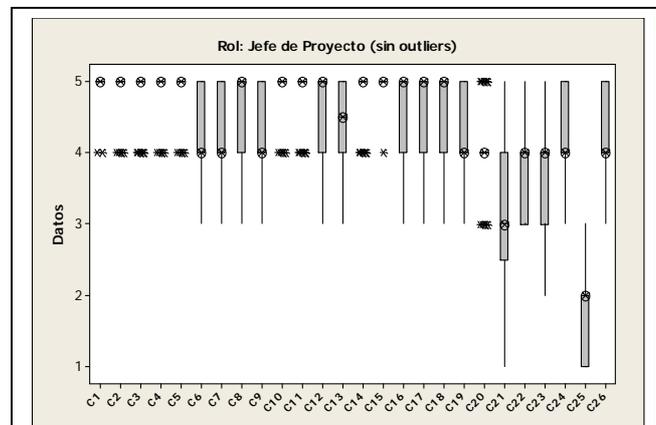


Figura 3 Análisis del rol Jefe de Proyecto (sin outsiders).

4- Identificar las competencias requeridas para desempeñar cada rol. A partir del análisis del gráfico de la Figura 3, se pudo concluir que las competencias 1 hasta la 20, la 22, la 24 y la 26, eran necesarias para desempeñar el rol de Jefe de Proyecto, ya que como se observa, todos los expertos concuerdan que cada una de estas competencias al menos resulta útil (valor ≥ 3) para el rol. Para llegar a conclusiones acerca de las competencias 21, 23 y 25, fue preciso analizar los valores de la mediana y la moda. Esto permitió concluir que tanto la competencia 21 como la 23, resultaban necesarias para el desempeño del rol y que en el caso de la competencia 25, debido a la gran dispersión en el criterio emitido por los expertos, se sugiere someterla a una segunda ronda de evaluación.

5- Identificar el nivel de importancia de cada competencia para el desempeño de cada rol. Al culminar el paso anterior, para todos los roles se identificaron: las competencias requeridas, las no requeridas y las que debían ser sometidas a una segunda ronda. Sin embargo, entre las competencias requeridas no resultaba muy claro precisar, en una parte de los casos, el grado de importancia de cada una, debido a la dispersión de los datos. Este hecho puede tener su explicación en la poca formalización que existe, en la mayor

parte de las organizaciones, de los roles y las competencias necesarias para enfrentar los proyectos de software. Al evaluar esta situación, los autores del trabajo decidieron transformar los datos obtenidos con la escala de valores de 1 a 5, tomando en consideración una escala más simple, capaz de reflejar la información esencial obtenida bajo la escala anterior y que a su vez resultara más fácil de utilizar por los expertos al aplicarse la segunda ronda. Este cambio de escala constituye una modificación en la aplicación del método Delphi que no modifica la esencia del mismo. La nueva escala contempla tres valores:

2- la competencia es requerida en un alto por ciento.

1- la competencia es necesaria en alguna medida.

0- la competencia no es necesaria.

Al procesar todos los roles aplicando la nueva escala se obtuvo la matriz que se muestra en la Figura 4. Las celdas en blanco indican que esa competencia para ese rol debía ser sometida a la segunda ronda. Así, para el caso del rol Jefe de proyecto las competencias 22 y 25 debían someterse a la segunda ronda, la competencia 21 se evalúa como necesaria en alguna medida y el resto de las competencias se evalúan como muy importantes para el desempeño del rol.

| Roles Invariantes | Competencias genéricas | | | | | | | | | | | | | | | | Competencias Técnicas | | | | | | | | | |
|---------------------------------|---------------------------------|-----------------------|-------------------------------------|------------------------|--------------------------|--|---|---|------------------------|-------------------|--------------------|---------------------------|---------------|------------------------|--------------------------------|------------------------|-----------------------|----------------------|----------------------|--------------------|--------------------------------------|---|------------------------------------|---|-------------------------------|----------------------|
| | Trabajo en equipo y cooperación | Capacidad de análisis | Capacidad de planificar y organizar | Capacidad de Controlar | Capacidad de Negociación | Capacidad de Adaptación (Flexibilidad) | Capacidad de Investigar (exploración, búsqueda, indagación) | Capacidad de Estrategia (Busca, analiza y define estrategias) | Pensamiento conceptual | Manejo de riesgos | Toma de decisiones | Tenacidad (Perseverancia) | Independencia | Orientación al cliente | Compromiso con la organización | Proactivo (Iniciativa) | Comunicación oral | Comunicación escrita | Aprendizaje continuo | Dominio del idioma | Dominio del lenguaje de programación | Dominio de la herramienta de modelación | Dominio de gestor de base de datos | Dominio de la metodología de desarrollo | Habilidades de Diseño Gráfico | Dominio del Producto |
| Jefe de proyecto | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | | 2 | 2 | | 2 |
| Diseñador Gráfico | 2 | 2 | | 0 | | 2 | 2 | | | | 1 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | | 0 | 0 | | | 2 |
| Gestor de Cambios | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | | 1 | 1 | 1 | | 2 |
| Gestor de Configuración | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | | 2 | | 2 | | | 2 | | 1 | 1 | 2 | 1 | | | 1 | 1 | 0 | 2 |
| Arquitecto | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 |
| Especialista de Seguridad | | 2 | | 2 | | 1 | 2 | | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | | | | 2 | 1 | 0 | 2 |
| Analista | 2 | 2 | 2 | | 2 | 2 | 2 | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | | 2 |
| Diseñador | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 |
| Diseñador de Base de Datos | 2 | 2 | 1 | 1 | | 2 | 2 | 1 | | 2 | 2 | | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 | 2 | 0 | 2 |
| Programador | 2 | 2 | | | | 2 | 2 | | 2 | | | 2 | 2 | | 2 | 2 | 1 | | 2 | 1 | 2 | 2 | 2 | 2 | | 2 |
| Documentador | 2 | 1 | 1 | | 0 | 1 | | | 1 | | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | | 0 | | 0 | 2 | 2 | 2 | 2 |
| Probador | 2 | | 1 | | 0 | | 1 | | | | | 2 | 1 | | 2 | 1 | | | 2 | 1 | | | 0 | | 0 | 2 |
| Especialista de Calidad | 2 | 2 | 2 | | | | 2 | 1 | | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | | | | 0 | 2 |
| Jefe de Proyecto (Implantación) | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | | 0 | 2 |
| Especialista Operacional | 2 | 2 | | 1 | 2 | | 2 | 1 | | | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | | | 0 | 0 | 2 |
| Especialista Técnico | 2 | 2 | 1 | | | 2 | 2 | | 2 | | | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 1 | 2 | | | 2 |

Figura 4 Matriz que contiene los resultados de la primera ronda.

PROPUESTA DE ROLES INVARIANTES Y COMPETENCIAS PARA ENFRENTAR PROYECTOS DE SOFTWARE

En la segunda ronda, participaron 34 expertos. De igual forma se empleó el MINITAB para procesar las 34 matrices, utilizando gráficos de *boxplots*. En el procesamiento de los datos se utilizó el procedimiento aplicado en la ronda anterior, sólo que en el segundo paso, dada la escala utilizada, se consideraron *outliers* todos los valores considerados como tal por el MINITAB, puesto que estos expresan una opinión conceptualmente diferente de la mayoría de los otros expertos y estadísticamente significativa.

Al culminar el procesamiento de la segunda ronda se obtuvo la matriz que se muestra en la Figura 5. Esta matriz recoge

una propuesta que incluye: roles de invariantes para enfrentar proyectos de desarrollo e implantación de software, competencias requeridas para desempeñar los roles e importancia que tiene cada competencia para desempeñar cada rol. Las celdas donde aparecen números fraccionarios, indican falta de concordancia en el criterio emitido por los expertos:

1,5 - indica que la competencia es necesaria pero no hay concordancia en la medida.

0.5 - indica que no hay concordancia en si la competencia resulta necesaria para el desempeño del rol.

| Roles Invariantes | Competencias genéricas | | | | | | | | | | | | | | | Competencias Técnicas | | | | | | | | | | |
|---------------------------------|---------------------------------|-----------------------|-------------------------------------|------------------------|--------------------------|--|---|--|------------------------|-------------------|--------------------|---------------------------|---------------|------------------------|--------------------------------|------------------------|-------------------|----------------------|----------------------|--------------------|--------------------------------------|---|------------------------------------|---|-------------------------------|----------------------|
| | Trabajo en equipo y cooperación | Capacidad de análisis | Capacidad de planificar y organizar | Capacidad de Controlar | Capacidad de Negociación | Capacidad de Adaptación (Flexibilidad) | Capacidad de Investigar (exploración, búsqueda, indagación) | Capacidad de Estratega (Busca, analiza y define estrategias) | Pensamiento conceptual | Manejo de riesgos | Toma de decisiones | Tenacidad (Perseverancia) | Independencia | Orientación al cliente | Compromiso con la organización | Proactivo (Iniciativa) | Comunicación oral | Comunicación escrita | Aprendizaje continuo | Dominio del idioma | Dominio del lenguaje de programación | Dominio de la herramienta de modelación | Dominio de gestor de base de datos | Dominio de la metodología de desarrollo | Habilidades de Diseño Gráfico | Dominio del Producto |
| Jefe de proyecto | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 1 | 2 |
| Diseñador Gráfico | 2 | 2 | 1 | 0 | 1 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 0.5 | 0 | 0 | 0 | 2 | 2 |
| Gestor de Cambios | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 2 |
| Gestor de Configuración | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1.5 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 2 |
| Arquitecto | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 |
| Especialista de Seguridad | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 1.5 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 0 | 2 |
| Analista | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 1 | 2 |
| Diseñador | 2 | 2 | 2 | 2 | 1.5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 |
| Diseñador de Base de Datos | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 2 |
| Programador | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 1 | 2 |
| Documentador | 2 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | 2 | 0 | 1.5 | 0 | 2 | 2 | 2 |
| Probador | 2 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1.5 | 0.5 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 2 |
| Especialista de Calidad | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 1 | 2 | 0 | 2 |
| Jefe de Proyecto (Implantación) | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 2 |
| Especialista Operacional | 2 | 2 | 1.5 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 1 | 0 | 0 | 2 |
| Especialista Técnico | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 1.5 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1.5 | 2 | 1 | 2 | 1 | 0 | 2 |

Figura 5 Propuesta de roles y competencias (resultados de la segunda ronda).

Aunque la definición de roles depende en gran medida del tipo de proyecto y de la organización, esta propuesta puede servir de punto de partida y soporte en la implementación de la gestión de recursos humanos por competencias en las organizaciones de software cubanas. No obstante, se sugiere que al comenzar cada proyecto se evalúen cuáles son los roles más convenientes a utilizar y se valore la importancia real de cada competencia para el desempeño de cada rol.

A partir de los resultados obtenidos al concluir la segunda ronda y dado que es posible que un empleado desempeñe varios roles en el proyecto, se realizó un análisis con el

objetivo de identificar cuáles eran las competencias que los expertos consideraban como muy necesarias con independencia del rol a ejecutar. Este análisis reflejó la relevancia general de las competencias genéricas con respecto a las técnicas, las cuales resultan extremadamente necesarias, pero en la ejecución de roles específicos. Se destacan las competencias de trabajo en equipo, compromiso con la organización y aprendizaje continuo.

Un análisis similar se realizó para el caso de los roles. El arquitecto, los jefes de proyecto (de ambos tipos), el analista y el diseñador resultan los roles que, según los expertos, requieren una calificación alta en prácticamente

todas las competencias para que sean ejecutados de manera satisfactoria.

CONCLUSIONES

Gestionar los recursos humanos basado en el enfoque por competencias resulta muy necesario para las organizaciones de software, ya que permite preservar y desarrollar su personal en una industria donde la tecnología cambia de manera vertiginosa. Los recursos humanos constituyen el factor menos formalizado en los modelos y metodologías de desarrollo de software, las cuales no ofrecen un marco de referencia, que contenga claramente definidas las competencias genéricas y técnicas requeridas para desempeñar cada rol establecido acorde a las características del proyecto. La propuesta de roles invariantes para enfrentar proyectos de desarrollo e implantación de software, y de competencias tanto técnicas como genéricas que se obtuvo como resultado de la aplicación del método Delphi, puede servir de punto de partida y soporte en la implementación de la gestión de recursos humanos por competencias en las organizaciones de software cubanas. 🏠

REFERENCIAS

1. DECARVALHO, 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.
2. PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Science, 2004.
3. RUBINSTEIN, D. *Standish Group Report: There's less development chaos today*, *SD Times*. 2007. 169. 19-20.
4. CHARETTE, R. N. "Why software fails". *IEEE Spectrum*. 42(9): 42-49, 2005.
5. NEIL, D. "Proyectos Informáticos: Fracasos y Lecciones Aprendidas". *Revista de Derecho y Tecnologías de la Información*. 4: 2006.
6. RYAN, R. "IT Project Management: Infamous Failures, Classics Mistakes, and Best Practices". *Mis Quarterly Executive*. 6(2): 67-78, 2007.
7. DELGADO, M. D. "Un modelo para la gestión de Revisiones en proyectos de software utilizando Razonamiento Basado en Casos". Tesis Doctoral. Ciudad de La Habana, 2006.
8. ACUÑA, S.T., JURISTO, N. y MORENO, A.M. "Emphasizing Human Capabilities in Software Development". *IEEE Software*. 23(2): 94-101, 2006.
9. 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 Latin Ibero-American Congress on Operations Research CLAIO 2008* (Cartagenas de Indias, Colombia: 2008).
10. DE MARCO T, T. LISTER. *Peopeware: Productives Projects and Teams*. Dorset House, 1999.
11. ANDRÉ, M. y LÓPEZ, Y. "Creando un profesional con disciplina en el proceso de desarrollo de software". *Ingeniería Industrial*. XXVII(1): 27-30, 2006.
12. VLIET, H. V. "Reflections on Software Engineering Education". *IEEE Software*. 23(3): 55-61, 2006.
13. URQUIZA, A. "Aplicación de modelos de competencias a la gestión de sistemas de información". *Revista de Innovación, Calidad e Ingeniería del Software*. 3(1): 23-37, 2007.
14. CUESTA, A. *Tecnología de Gestión de Recursos Humanos*. La Habana: La Habana: Academia, 2005.
15. HUMPHREY, W. S. *Managing Technical People: Innovation, Teamwork and the Software Process*. [s.l.]: Addison-Wesley, 1998.
16. GRZEDA, M. M. "In Competence we trust: addressing conceptual ambiguity". *Journal of Management Development*. 24(6): 530-545, 2005.
17. CUESTA, A. "Sistema y tecnologías de gestión de recursos humanos". Tesis Doctoral. Ciudad de La Habana, 2007.
18. ALLES, M. *Dirección estratégica de recursos humanos. Gestión por competencias: el diccionario*. Argentina: Granica, 2002.
19. "Tuning Educational Structures in Europe". 2004.
20. DOWNEY, J. AND N. POWER. "An Artifact-centric Framework for Software Development Skills". En: *In ACM SIGMIS CPR Conference* (Missouri: 2007) 186-195.
21. JACOBSON, I., G. BOSCH, AND J. RUMBAUGH. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley Iberoamericana, 2000.
22. LÓPEZ, Y. y ANDRÉ, M. "Roles en el Proceso de Desarrollo de Software para las Empresas Cubanas". *Revista de Ingeniería Industrial*. XVII(1): 31-35, 2006.
23. ANDRÉ, M. *Roles definidos por el Proceso Unificado de Rational. Reporte de Investigaciones del CEIS-ISBN: 959-261-179-3*. 2005.
24. BECK, K. *Extreme Programming Explained. Embrace Change*. [s.l.]: Addison-Wesley, 1999.
25. COCKBURN, A. *Crystal Clear: A Human-Powered Methodology for Small Teams*. [s.l.]: Addison-Wesley, 2005.

