

# Metodología Para Identificar Problemas en Equipos que Desarrollan Software Científico-Técnico.

Salamon Alicia, Maller Patricio, Mira Natalia, Boggio Alejandra, Giró Juan, Cuozzo Jose, Coenda Francisco, Pérez Sofía.

**Resumen**— El presente trabajo es una investigación sobre un enfoque nuevo para elicitar y estructurar la información que permite detectar actividades críticas en equipos que desarrollan software científico-técnico a partir de la experiencia acumulada de sus integrantes. Para llevar a cabo dicho proceso de elicitación de las características propias que resultan críticas en el desarrollo de este tipo de software, se utilizan las técnicas Card Sorting, Repertory Grid y el Método de Ponderación Lineal para el procesamiento de la grilla. Esta problemática es objeto de estudio a nivel internacional con la utilización de encuestas aplicadas a diferentes equipos de desarrollo de software científico-técnico. La experiencia que se presenta en este trabajo con resultados iniciales se llevó a cabo en el ámbito de la Dirección de Análisis Operativo (D.A.O.) de la Fuerza Aérea Argentina.

**Palabras Claves**— Clasificación de Tarjetas, Ponderación Lineal, Repertory Grid, Desarrollo de Software científico-técnico.

**Abstract**- This paper shows a research of new approach to structure and elicit information that detects critical activities on technical-scientific software development team from the experience accumulated. To carry out this elicit process of it owns characteristic that are critical in development of such software; we use Card Sorting, Repertory Grid and Linear Weighting Method to process the grid. This problem is internationally study with surveys applied to different technical-scientific software development teams. The experience show in this paper was conducted in Dirección de Análisis Operativo de la Fuerza Aérea Argentina.

**Key Word** - Card Sorting, Linear Weighting, Repertory Grid, Technical-Scientific Software.

## I. INTRODUCCIÓN

La experiencia indica que las diferencias en los procesos de construcción de software empresarial y software científico son tales que los modelos y prácticas utilizados en el primero no son fácilmente transferibles al segundo sin un esfuerzo importante de adaptación (tayloring) [1] [13].

Este trabajo presenta una experiencia que contribuye a la elicitación de esas características que resultan críticas en el desarrollo de software científico-técnico, así como un conjunto de técnicas aplicadas a la actividad de descubrimiento de las mismas.

Algunas diferencias entre el software científico y software empresarial [12] [13]:

Tabla 1. Diferencia entre software científico y software empresarial

Software científico	Software Empresarial
El software es un medio para un fin.	El software es un fin en sí mismo.
Al comienzo del proyecto los objetivos son una idea vaga pero a medida que avanza el proyecto se van definiendo.	Desde el comienzo del proyecto los objetivos se definen de manera clara y concisa son los que guían el proyecto.
Generalmente los desarrolladores y usuarios son los mismos, es difícil encontrar un mercado externo para sus productos.	Generalmente los desarrolladores y los usuarios del sistema son distintas personas y existe un amplio mercado externo para sus productos.
Las pruebas y aseguramiento de la calidad son costosas. Se utiliza simulación para llevarlas a cabo ya que las pruebas del experimento físico la mayoría de las veces no se realiza por el costo que ello significa.	Las pruebas y aseguramiento de la calidad son más fáciles de llevar a cabo que en el software científico.

En investigaciones a nivel internacional se han utilizado las encuestas, siendo una herramienta de uso común cuando se plantea identificar áreas susceptibles de mejoras en el desarrollo de software [6], aún así, no se puede negar que el uso de esta técnica como única herramienta presenta serios inconvenientes [3] [7] [10].

En este sentido, resulta fundamental enfocarse en las dificultades que se presentan en los procesos de desarrollo como una forma para mejorar posteriormente los mismos.

El equipo de investigación aborda dicha problemática a partir de las percepciones de los integrantes de los equipos que desarrollan software científico-técnico en la Dirección de Análisis Operativo de la Fuerza Aérea Argentina, en virtud de su experiencia acumulada en el dominio científico-técnico. Estos grupos conocen los procesos de desarrollo en los cuales trabajan, y por lo tanto las situaciones críticas a partir de las cuales estos desarrollos no llegan a tener éxito en todos los casos.

Es por esto que, para generar un Plan de mejoras en los equipos que desarrollan software científico-técnico, resulta razonable contar con un conjunto de técnicas aplicadas para caracterizar el entorno de trabajo, que permitan obtener, refinar y representar las percepciones de los individuos involucrados en los procesos en cuestión.

Utilizando la técnica de Repertory Grid (Kelly) (Grilla de Repertorio) [5] se pretende obtener las percepciones individuales de los ingenieros especialistas en el dominio que desarrollan software en los equipos de desarrollo de la D.A.O.

A continuación se expone la metodología utilizada, luego se describe la experiencia en sí misma y se concluye con los resultados del trabajo.

## II. HERRAMIENTAS Y METODOLOGÍA

El trabajo tiene como objetivo en esta etapa inicial obtener las áreas críticas en los procesos de desarrollo de software científico-técnico y la priorización de las mismas, a partir de las percepciones de cada uno de los integrantes de los equipos de desarrollo, a fin de que éstas puedan ser el foco de posteriores actividades de mejora y optimización. Se busca el aporte a la calidad de los procesos y productos resultantes en este dominio.

La “racionalidad limitada” planteada por Simon [11], sumada a la complejidad del entorno científico-técnico, nos conduce a representar los procesos mentales de individuos pertenecientes a equipos de desarrollo de software, en este caso a través de mapas cognitivos para capturar su visión acerca de aspectos específicos en dicho dominio. Se pretende como objetivo conocer los elementos con mayor incidencia crítica en el éxito de los proyectos de construcción de software científico-técnico.

En una primera instancia se aplicó la técnica de Card Sorting (Clasificación de Tarjetas) [2] como herramienta de exploración para obtener las dificultades más importantes que se presentan en los procesos de desarrollo de software.

### A. Card Sorting

Esta técnica de categorización de contenidos se basa en la observación de cómo los participantes agrupan y asocian entre sí un número predeterminado de tarjetas etiquetadas con las diferentes categorías temáticas.

De esta manera, partiendo del comportamiento de los propios participantes, es posible organizar y clasificar la información conforme a su modelo mental.

Con la aplicación del concepto de clasificación se logró manejar y reducir la información compleja acerca del desarrollo de software con la finalidad de mejorar la comprensión del entorno. Una categoría cognitiva se trata como un conjunto de concepciones intelectualmente organizadas que referencian la realidad del sujeto como ser las dificultades que el desarrollador asume, considera, percibe críticas en su proceso de desarrollo, y que conforman así los grupos cognitivos.

Esto es, el desarrollador modela o representa mentalmente el proceso de desarrollo de software del cual es parte, en este caso su entorno específico. Los grupos mencionados con los que se trabaja comparten las dificultades que implica la construcción de software en el dominio científico-técnico del ámbito de la D.A.O. (Fuerza Aérea Argentina).

A partir de la propuesta de Kelly (1955) se utilizó la técnica Repertory Grid [5] que permite a los desarrolladores identificar y expresar en forma explícita los criterios o características que forman parte de sus procesos de desarrollo, obteniendo de esta manera los constructos personales a través del método del Contexto Mínimo [4].

### B. Repertory Grid

Esta técnica se utiliza como una forma de entrevista estructurada, orientada a explicitar y analizar los constructos con los que una persona organiza su mapa conceptual. Kelly afirma que las personas buscan encajar las experiencias del mundo en constructos cognitivos.

Se define como un constructo a una categoría descriptiva que permite clasificar los acontecimientos y que representa la percepción de la realidad del sujeto. Estas definiciones parten de las entrevistas para luego construir una matriz de datos que se somete a varios análisis para revelar su estructura implícita, esta matriz se conoce como Matriz de Kelly.

La técnica en cuestión es un instrumento de evaluación de las dimensiones y estructura del significado personal que se deriva de la “Teoría de los Constructos Personales”, estas dimensiones reciben el nombre de elementos, que son los aspectos relevantes en el contexto que se está evaluando. Los mismos deben ser representativos de lo que está sujeto a análisis, homogéneos y lo más precisos posibles.

La Matriz de Kelly es un sistema bidimensional de elementos y constructos donde los mismos están interceptados entre sí. Los elementos están ubicados en la parte superior de la matriz, columnas y los constructos en las filas.

En base a la teoría de Kelly los constructos son bipolares ya que son el principal mecanismo utilizado para organizar, simplificar e interpretar el “entorno”. En la matriz de la tabla 2 se observan los constructos bipolares emergentes (positivos) a la derecha y los constructos bipolares opuestos (negativos) a la izquierda de cada fila.

Como se mencionó anteriormente los subsistemas están relacionados entre sí, en este caso se realizó a través del Método de Contexto Mínimo donde Kelly plantea la tríada. La misma consiste en presentar al individuo tres elementos y se le pide que describa una característica común a dos de ellos (medida de similitud) y que a su vez los diferencia del tercero

Tabla 2. Ejemplo Teórico de Matriz con Constructos Bipolares

	Elemento 1	Elemento 2	
Constructo Negativo 1			Constructo Positivo 1
Constructo Negativo 2			Constructo Positivo 2

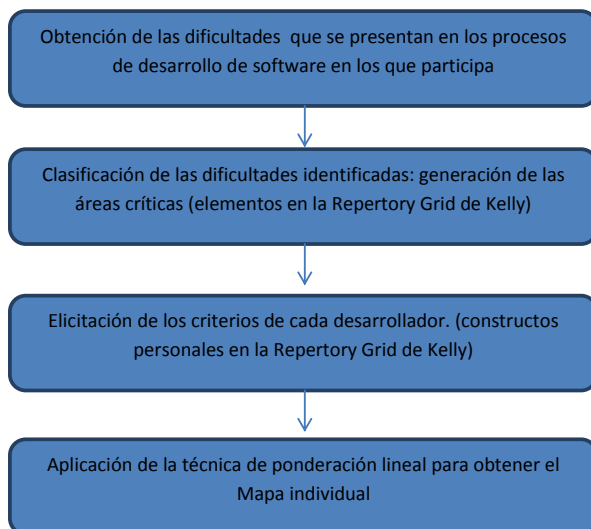


Fig 1. Procedimiento aplicado por el equipo de investigación.

(medida de contraste), de esta manera se toman estas medidas como valores de bipolaridad que refleja la posibilidad de moverse en la misma dimensión entre dos valores identificados ayudando a validar los constructos expuestos.

Paso siguiente se completa la matriz con valores, de acuerdo a una escala seleccionada entre varias, donde se categoriza cada elemento en función de la valoración realizada sobre cada constructo [14].

Por último se analizan los resultados obtenidos en cada una de las matrices mediante el método de ponderación lineal, con el cual se llega a un mapa cognitivo de cada individuo, exponiendo el orden de los elementos en función de la criticidad de su impacto en el resultado del proceso de construcción de software científico-técnico.

### C. Ponderación Lineal

El procedimiento de ponderación lineal es probablemente el más conocido y utilizado de los métodos de multicriterio. Con el mismo se obtiene una puntuación global por simple suma de las contribuciones obtenidas de cada atributo.

En el caso que se tengan varios criterios con diferentes escalas, dado que los mismos no son sumables en forma directa, se requiere un previo proceso de normalización para que pueda efectuarse la suma de contribuciones de los atributos.

Este es un método depende de la asignación de pesos a los criterios, en el cual consiste en construir una función de valor lineal para cada una de las alternativas, es decir, obtener para cada alternativa la suma de los productos de las calificaciones obtenidas con respecto a cada criterio por el peso atribuido a cada criterio particular. Los criterios  $W_j$  pueden ser evaluados de forma cuantitativa o cualitativa. Requieren determinar si son crecientes o decrecientes, máximos o mínimos.

$$S_i = \sum_j^0 A_{ij} W_j \quad (1)$$

donde:

$S_i$  = Resultado función valor para la alternativa  $i$ .

$A_{ij}$  = calificación de la alternativa  $i$  según el criterio  $j$ .

$W_j$  = Ponderación del objetivo  $j$ .

### III. EXPERIENCIA

La experiencia se llevó a cabo con grupos de desarrollo de software científico-técnico de la D.A.O. dos grupos de la ciudad de Río IV y uno del Centro de Desarrollo de Entrenadores y Simuladores de Vuelos de la ciudad de Córdoba, [9].

Con respecto a estos grupos se observa en ellos las siguientes particularidades:

- Es difícil determinar quién conforma el equipo de desarrollo. En los mismos se identifican :
    - programadores comprometidos,
    - programadores ocasionales,
    - referentes que responden consultas frecuente,
    - no programadores que colaboran ofreciendo orientación o la creación de nuevos enfoques científicos para un problema particular que enfrenta el proyecto.
  - Los integrantes de los grupos adoptan un rol flexible.
  - Actividades como aseguramiento de la calidad, coordinación y análisis de requerimientos, entre otras funciones especializadas, están ausentes en la mayoría de los proyectos.
  - Sin embargo, como los proyectos evolucionan, crecen, las funciones de administración de sistemas y base de datos deben ser formalmente asignada a un miembro.
  - Los proyectos crecen en dos direcciones:
    - hacia los recursos humanos técnicos: contratación de un administrador de base de datos o diseñadores de interfaces de usuario.
    - hacia los recursos humanos científicos: más investigadores ingenieros especialistas se involucran en el proyecto, en general, con el fin de profundizar la investigación.
  - Es difícil encontrar objetivos claramente establecidos en este dominio.
  - Los programadores plantean que están involucrados con:
    - los problemas de investigación.
    - con la complejidad de la plataforma software.
- y como complemento:
- la documentación de requisitos en ambientes científicos-técnicos son escasos.
- Los proyectos “parecen no finalizar”, y aunque los entrevistados utilizan términos tales como "ochenta por ciento hecho ", no parecen tener en claro la cuantificación del progreso, pero tienen la sensación de que hay siempre un poco más de trabajo que hacer.
  - Algunos proyectos peligran ser abandonados por razones de presupuesto, interés, formación.

- Hay poca planificación previa, predomina una adaptación permanente a un enfoque.
- Algunos de los proyectos comenzaron como proyectos paralelos o porciones de código escritos para solucionar parte de un problema, su autor las comparte con un grupo más amplio, y de allí comienza su interés, por lo que cobra vida propia y de interés para sus integrantes.
- Otros proyectos fueron construidos para solucionar un problema general, se tomaron como base y se fueron ampliando y modificando por otros desarrolladores para hacer frente a las variaciones del mismo problema, o de problemas más específicos, años después.
- Los integrantes de los grupos a veces tienen sólo un esbozo de las direcciones futuras del proyecto.
- Su estrategia parece ser la de tomar el desarrollo de una característica a la vez, y pasar luego a la característica que "naturalmente" debe seguir.
- Es difícil establecer una clara línea en los límites de estos productos. Sus módulos tienden a ramificarse, permaneciendo formalmente siendo parte del mismo paquete.
- Pueden interactuar con productos externos para abordar cuestiones más amplias de investigación.

Para el desarrollo de esta experiencia se realizó una intervención grupal en la que se reunió a los integrantes de los equipos de desarrollo mencionados, entregándose para un ejercicio individual tres tarjetas a fin de identificar y consignar los problemas o dificultades que se han presentado en el desarrollo de software que llevan a cabo [1].

Posteriormente se realizó una tarea grupal donde se dieron a conocer todas las dificultades mencionadas, se discutieron las mismas, fue muy clara la existencia de vivencias compartidas entre los entrevistados y se reagruparon en cuatro áreas críticas que fueron rotulados por ellos mismos de la siguiente manera:

- metodología,
- soporte,
- gestión y
- capacitación.

Posteriormente, mediante la Repertory Grid [5] [4] cada entrevistado evaluó la situación actual a partir de su comprensión individual, práctica basada en la Teoría de los Constructos Personales para capturar la interpretación propia de la experiencia de cada integrante.

Para esta actividad:

- se contó con los elementos seleccionados anteriormente (áreas críticas).
- se obtuvieron los constructos a partir de cada integrante a través del método del Contexto Mínimo, es decir, seleccionando tres de las columnas disponibles en forma aleatoria, determinar el criterio que hace que dos de ellas sean semejantes entre sí y diferentes a la tercera. Este

proceso se repitió tantas veces como cada participante estimó suficiente.

- se evaluaron los constructos obtenidos frente a los elementos a partir de la experiencia personal del entrevistado en la actividad objeto de estudio, utilizando una escala del 1 al 6, donde éste último era el mayor peso del criterio en el polo positivo.
- se procesaron y analizaron los datos obtenidos usando Ponderación Lineal [8]. Para cada constructo:

$$S_j = \sum_{i=1}^0 w_i r_{ij} \quad (2)$$

Donde  $i$  representan las filas,  $j$  las columnas,  $w$  es la ponderación del constructo y  $r$  es el valor que el entrevistado le ha dado al constructo en función del elemento, es decir, la intersección de la fila y la columna de la matriz.

#### IV. RESULTADOS

Se realizó el ejercicio individual a los quince ingenieros especialistas que desarrollan software científico técnico en el ámbito de la D.A.O.

En la primera etapa se obtuvieron cuatro áreas críticas del proceso de desarrollo de software consensuadas por los integrantes de equipos de desarrollo de software. En la última etapa luego de elicitar los constructos y de realizar la valoración de cada constructo con respecto de cada área crítica para cada participante, se compararon los resultados obtenidos en cada Repertory Grid. De esta forma se pudo obtener el mapa individual de cada uno de los desarrolladores que participaron en la experiencia. En la Tabla 3 se muestra un ejemplo de la matriz obtenida de uno de los participantes, donde:

$S_{\text{Metodología}} = 231$

$S_{\text{Capacitación}} = 182$

$S_{\text{Soporte}} = 199$

$S_{\text{Gestión}} = 207$

Lo que significa que para este integrante de equipo de desarrollo el área crítica de Metodología es prioritario con respecto a las demás áreas.

En esta matriz individual queda evidenciado que el entrevistado considera que las dificultades más críticas que se presentan en los procesos de desarrollo de software en los que participa son de orden metodológico en primer lugar.

Analizando los mapas individuales de cada participante se observa una similitud en los resultados que indica la visión y experiencia compartidas por los miembros de los grupos estudiados.

Tabla 3. Ejemplo de matriz obtenida de uno de los participantes. RP: Resultado Parcial – P: Ponderación – M: Metodología – C: Capacitación – S: Soporte – Gestión

	P	M	RP	C	RP	S	RP	G	RP	
Falta de experiencia	9	6	54	1	9	6	54	1	9	Mayor capacitación
Conocimientos básicos	8	1	8	6	48	2	16	6	48	Capacitación especializada
Objetivos estratégicos confusos	3	1	3	6	18	1	3	6	18	Objetivos estratégicos claros
Ausencia procedimientos y pautas de trabajo	10	5	50	5	50	1	10	6	60	Procedimientos y pautas de trabajo claras
Falta de planificación de las actividades	7	5	35	1	7	6	42	1	7	Llevar metodologías de ejecución
Inestabilidad de los plazos	6	6	36	2	12	6	36	1	6	Estabilidad de los plazos
Escasez de herramientas y recursos	4	4	16	5	20	5	20	5	20	Documentar de manera ágil
Resistencia al cambio por parte de las personas	5	4	20	2	10	2	10	5	25	Personas proactivas
Alta rotación de cargos directivos	2	4	8	1	2	1	2	6	12	Poca rotación de cargos directivos
Presupuestos bajos	1	1	1	6	6	6	6	2	2	Presupuestos más altos
			231		182		199		207	

## V. CONCLUSIONES

Mediante la utilización de un conjunto de técnicas cognitivas, se obtuvieron las percepciones individuales de los miembros de tres grupos de desarrollo de software de la D.A.O. (Fuerza Aérea Argentina) acerca de sus procesos de desarrollo de software científico-técnico. Se logró identificar las características o aspectos de estos procesos que deben ser foco de las actividades de mejora de forma prioritaria a fin de contribuir a la calidad desde la perspectiva de cada uno de los participantes.

La investigación actual en el dominio continúa, apunta a realizar intervenciones grupales y la agregación de los mapas individuales obtenidos, a fin de obtener valoraciones y priorizaciones grupales de los criterios, así como al descubrimiento de subgrupos dentro de los mismos, que compartan valoraciones similares, siempre apuntando a lograr la mejora de los procesos de desarrollo en este ámbito.

## REFERENCIAS

- [1] P. Maller et. al. *New Practices to Structure and Elicit Improvement Opportunities in Scientific Software Development Teams*. GDN 2012. Submitted for publication.
- [2] G. Rugg and P. McGeorge, "The Sorting Techniques: A Tutorial Paper On Card Sorts, Picture Sorts And Item Sorts," Expert Systems, 1997
- [3] Jo Erskine Hannay, Hans Petter Langtangen, Carolyn MacLeod, Dietmar Pfahl, Janice Singer, Greg Wilson. *How Do Scientists Develop And Use Scientific Software?*. In: SECSE '09 Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, 2009
- [4] José L. Zanazzi, Alicia G. Salamon, Gabriela Cabrera, Analía González, Beatriz Pedrotti. *La Investigación Operativa Soft En La Estructuración De Problemas Vinculados Con La Orientación Vocacional*. XXIV ENDIO - XXII EPIO, Rio IV, 2011
- [5] Kelly, G. *The Psychology Of Personal Constructs*. New York. Norton, 1955
- [6] Kjetil Moløkken and Magne Jørgensen. *A Review Of Surveys On Software Effort Estimation*. ISESE '03 Proceedings of the 2003 International Symposium on Empirical Software Engineering.
- [7] Luke Nguyen-Hoan, Shayne Flint, Ramesh Sankaranarayanan. *A Survey Of Scientific Software Development*. In: ESEM '10 Proceedings of the

- 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010.
- [8] Tkach, R.J., and Simonovic, S. 1997. *A new approach to multi-criteria decision making in water resources*. Journal of Geographic Information and Decision Analysis, vol 1
- [9] P. Maller, A. Salamon, N. Mira, A. Boggio, J. Giró, J. Cuozzo, F. Coenda, S. Perez. Workshop WDSCT, Córdoba, 2011.
- [10] Salo O., Abrahamsson P., *Agile Methods In European Embedded Software Development Organisations: A Survey On The Actual Use And Usefulness Of Extreme Programming and Scrum*. Software, IET, vol.2, no.1, pp.58-64, February 2008
- [11] Herbert A. Simon. *Models of Bounded Rationality, Vol. 3: Empirically Grounded Economic Reason*. The MIT Press, June 1997.
- [12] Segal J. y Morris C. "Developing Scientific Software". IEEE Software Computer Society Washington, DC, USA 2008.
- [13] Basili V. et. al. "Understanding the High-Performance-Computing Community: A Software Engineer's Perspective". IEEE Software Computer Society, 2008.
- [14] Fransella F., Bell R. y Bannister D. *A Manual for Repertory Grid Technique*. Ed. John Wiley & Sons, Ltd. 2004.