

## Estudio de métricas para el control de proyectos software

**Concepción Presedo**  
Dept. de Lenguajes y Sistemas  
Informáticos  
EUITI Bilbao  
UPV/EHU  
48012 Bilbao  
[conchi.presedo@ehu.es](mailto:conchi.presedo@ehu.es)

**J. Javier Dolado**  
Dept. de Lenguajes y Sistemas  
Informáticos  
Facultad de Informática  
UPV/EHU  
20009 Donostia- San Sebastián  
[javier.dolado@ehu.es](mailto:javier.dolado@ehu.es)

**Amaia Aguirregoitia**  
Dept. de Lenguajes y Sistemas  
Informáticos  
EUITI Bilbao  
UPV/EHU  
48012 Bilbao  
[amaia.aguirregoitia@ehu.es](mailto:amaia.aguirregoitia@ehu.es)

### Resumen

Este artículo establece un marco de selección de métricas para controlar la gestión del proyecto de desarrollo software y propone un conjunto inicial de indicadores. La validez de la selección propuesta se evalúa mediante un estudio realizado en varias empresas del País Vasco y se presentan las conclusiones alcanzadas.

*Palabras clave:* Gestión del proyecto software, Control del proyecto software, Métricas proyecto software, Estudio.

### 1. Introducción

Actualmente, la ingeniería del software centra su atención en el *proceso de desarrollo global*. Este es el propósito del desarrollo de software estructurado, que desglosa el proyecto software en una serie de *fases* distintas, cada una de ellas con unos objetivos bien definidos cuyo logro se puede verificar, lo que garantiza una base sólida para las fases posteriores. Esto permite la planificación global de cómo se va a desarrollar el software, así como la consideración de qué producto se va a desarrollar [13].

Una de las principales dificultades en el estudio de este tipo de desarrollo software es el análisis adecuado e *integral* de las propiedades del producto y del proceso durante la ejecución del proyecto, que permita a sus usuarios potenciales – como son los gestores del proyecto – tomar decisiones basadas en información fiable.

También existe una necesidad urgente de ofrecer técnicas y herramientas de apoyo para el

control de estos proyectos [12]. Para que estas técnicas y herramientas – por ejemplo representaciones visuales o simulaciones – resulten efectivas deben ser capaces de obtener la información sobre el estado del proyecto que se necesite en cada momento.

Por todo ello, es crucial la selección de unas métricas apropiadas que reflejen de manera fidedigna el estado del proyecto desde una perspectiva global y posibiliten su análisis para mejorar el control e incrementar la productividad y la calidad del producto obtenido.

En este artículo se establece un marco de selección y se propone un conjunto inicial de métricas para controlar el proyecto de desarrollo software que se valida mediante un estudio realizado en el País Vasco. En los dos apartados siguientes se describen los fundamentos y los estudios previos que nos servirán como marco para la selección de las métricas y en el cuarto apartado se proponen los indicadores. En el quinto se detallan las características de las empresas encuestadas así como los resultados del estudio propio. Por último, se describen las conclusiones y el trabajo futuro.

### 2. Gestión y control de proyectos software

En este apartado se proporciona la información necesaria para entender los conceptos utilizados en el resto del artículo.

La *gestión de proyectos* es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto con el objetivo de cumplir las necesidades y expectativas de los interesados [10]. La gestión involucra las

actividades llevadas a cabo con el propósito de planificar y controlar las actividades de otros para alcanzar los objetivos que éstos no podrían lograr actuando en solitario. Para cumplir las necesidades y expectativas de los interesados se han de equilibrar demandas contrapuestas en tres vertientes:

- Alcance, tiempo, coste y calidad.
- Interesados con diferentes necesidades y expectativas.
- Requerimientos identificados (necesidades) y no identificados (expectativas).

Las distintas funciones de gestión pueden ser clasificadas como planificación, organización, dotación de personal, dirección y control. *Organizar* puede describirse como establecer las relaciones entre las distintas unidades de trabajo para el cumplimiento de los objetivos y la concesión de responsabilidades y autoridad para alcanzar esos objetivos. *Dotar de personal* consiste en seleccionar y formar a los individuos para los puestos de la organización y *dirigir* es crear una atmósfera que ayude y motive a la gente a alcanzar los objetivos finales deseados [14].

La *planificación del proyecto* desempeña un papel clave en la gestión de proyectos y tiene fundamentalmente un propósito: finalizar con éxito un proyecto mediante el establecimiento de un plan que se utiliza como base para su ejecución. Para la elaboración del plan es esencial subdividir el proyecto en actividades gestionables, programar cada actividad, evaluar los recursos, determinar las responsabilidades de cada actividad y definir el uso de los recursos y el presupuesto por período de tiempo.

El *control* de la ejecución del plan es lo que posibilita que el proyecto finalice con éxito y consiste en identificar y corregir aquello que no progresa de acuerdo al plan. El control realiza preguntas del tipo ¿Está el proyecto dentro de los tiempos y costes previstos? ¿Existen problemas potenciales que puedan retrasar el cumplimiento de los requisitos dentro del presupuesto y calendario previstos? Su objetivo es identificar las excepciones en el momento en que ocurran. Idealmente, el tiempo de respuesta debería ser corto y se deberían considerar acciones para evitar que problemas similares a los detectados vuelvan a ocurrir.

El control también proporciona planes y propuestas para la eliminación de las divergencias

entre los planes y/o estándares y los resultados reales [14].

## 2.1. Medidas para el seguimiento del programa

Una de las propuestas para apoyar el control efectivo de los proyectos de desarrollo software es el uso de principios básicos de ingeniería, con especial atención al seguimiento y análisis del producto real y los estados de los procesos, la comparación de los estados reales con los planeados, y el inicio de cualquier acción correctiva necesaria durante la ejecución del proyecto [11]. Cuando los gestores comparan los logros con los planes, con frecuencia se centran en el calendario y el coste de las tareas, pero existen otras cuestiones importantes que también deben ser consideradas en el control de proyectos. Estas cuestiones son la gestión de los cambios mediante el control de los requerimientos, la gestión de riesgos y la gestión de los errores mediante verificación y validación.

## 2.2. Medidas para la gestión de cambios

Una de las responsabilidades del gestor del proyecto es conseguir la conformidad con los requerimientos. Los requerimientos son un reto desde las primeras fases del proyecto ya que su especificación proporciona la base para el resto del proyecto software. Decidir con precisión qué construir y documentarlo, así como superar las dificultades de la adherencia de problemas específicos a esos requerimientos puede ser un gran problema y los problemas en los requerimientos son persistentes, dominantes y costosos [7].

Una propuesta de requerimientos completa define el proceso pero debería definir los productos a ser producidos y los criterios de aceptación. No se deberían medir solamente los entregables por fase reales frente a los planificados sino también debería asegurarse su aceptación.

Las especificaciones de requerimientos deberían estar organizadas para su referencia y revisión y deberían utilizarse con frecuencia para verificar la implementación, una actividad esencial para garantizar la satisfacción con el desarrollo.

Así, la gestión de las modificaciones en las especificaciones del producto software es otra de las facetas fundamentales en el control del proyecto software. La autorización y aceptación de las modificaciones y su implementación debería controlarse y también sus implicaciones. Esto asegura que los cambios en las especificaciones son analizados, clasificados y evaluados y debería también garantizar la integridad del sistema modificado y su documentación.

### 2.3. Medidas para la gestión de riesgos

Cualquier proyecto implica problemas potenciales que deberían ser identificados así como sus probabilidades de ocurrir y las pérdidas que supondrían. Estos problemas potenciales se denominan *riesgos* y los riesgos materializados son problemas, es decir, situaciones no deseadas que requieren tiempo y recursos para ser corregidas.

Es importante utilizar procedimientos sistemáticos para evaluar lo que podría ir mal, las probabilidades asociadas y la severidad de los impactos, desde una perspectiva proactiva.

### 2.4. Medidas para la gestión de errores

Existen distintas técnicas para revisar el producto software y proporcionar visibilidad sobre los resultados del producto. La verificación y validación pueden definirse como el proceso de asegurar que cada fase del ciclo de vida del desarrollo implementa correctamente las especificaciones de la fase previa y que cada producto software obtenido satisface sus requerimientos. Las *pruebas* son la ejecución controlada del código del programa en busca de errores. Las *revisiones formales* son revisiones planificadas y periódicas de los productos obtenidos llevadas a cabo por desarrolladores, clientes, usuarios y gestores para evaluar el progreso. Las *inspecciones* y *walk-throughs* son revisiones sistemáticas de los productos software obtenidos realizadas por los pares con el propósito de encontrar errores [14].

Independientemente del método que se utilice, deberían llevarse a cabo revisiones de todos los productos software obtenidos y debería hacerse un informe con medidas de los problemas, su tipo,

alcance y criticidad. El análisis de estos informes con información detallada sobre las revisiones y los errores es de vital importancia para mejorar tanto el producto como el proceso y para proporcionar la confianza de que el producto cumple con los estándares establecidos.

## 3. Estudios previos

En este apartado se presentan los resultados de los estudios previos en el área de la gestión de proyectos software que nos proporcionan el marco para la selección y definición de las métricas de las representaciones.

Como hemos visto, en el control de proyectos software existen cuatro áreas que es beneficioso e importante medir: el seguimiento del calendario y estimación del proyecto, la gestión de los cambios en los requerimientos, la identificación y el análisis de riesgos, y la identificación, clasificación y corrección de los errores. A continuación, se resumen los aspectos más relevantes que guiarán la definición de las métricas del sistema en cada una de estas áreas.

### 3.1. Seguimiento del programa

De acuerdo con estos estudios, entre un 26% y un 34% de los proyectos IT fracasan (proyectos cancelados más proyectos entregados sin la funcionalidad adecuada). En 2007, el 37% de los encuestados definieron la “capacidad para cumplir los objetivos de presupuesto” como pasable o pobre y un 41.7% estimó el “cumplimiento de los objetivos de calendario” como pasable o pobre. Por tanto, estimar el calendario y cumplirlo fueron los problemas de desempeño más críticos en los proyectos software entregados [5].

### 3.2. Gestión de cambios en requerimientos

De acuerdo con un estudio de 2007, el alcance y los cambios en los requerimientos fueron las principales razones para la cancelación del proyecto en un 33% de los encuestados [5].

Otros estudios establecen que los proyectos con una mayor habilidad en la gestión y desarrollo de los requerimientos se correspondieron con unos resultados más exitosos.

### 3.3. Gestión de riesgos

Desde hace tiempo, los proyectos software se consideran como proyectos de alto riesgo propensos al fracaso [1] y se reconoce que la gestión de riesgos es una de las mejores prácticas de la industria del software para reducir el factor sorpresa [4]. Mediante la observación se ha probado que los gestores de proyecto más exitosos son también buenos gestores del riesgo [3]. Un proceso de gestión de riesgos formal proporciona múltiples beneficios tanto al equipo del proyecto como a la organización en conjunto. La gestión de riesgos en los proyectos software es importante para evitar desastres, centrar y equilibrar el esfuerzo evitando repetir el trabajo y estimular situaciones win-win [2].

### 3.4. Gestión de errores

Un aspecto clave a la hora de producir software fiable es el análisis por fases de la distribución de errores como se propone en los tipos de fallos definidos en [8].

Este análisis implica la existencia de procedimientos de adquisición de datos y revisiones técnicas – evaluaciones de un producto software llevadas a cabo por un equipo o personal cualificado que determina su idoneidad para el uso pretendido e identifica desviaciones de los estándares y especificaciones – que podrían por ejemplo, clasificarse en revisiones de requerimientos, análisis, diseño y documentación según lo establecido en la propuesta [9].

## 4. Métricas propuestas

La determinación del estado de un proyecto involucra determinar el estado individual y las relaciones entre los atributos del proyecto (esfuerzo, coste, calendario y el estado actual de los productos obtenidos). La lista de atributos medidos debería ayudar a realizar el control, monitorización y seguimiento del plan (proceso), mejorar el aseguramiento y control de la calidad del producto, identificar las situaciones de riesgo excepcionales (proceso-producto) y evaluar la utilización, productividad y eficiencia de los recursos (recurso). Así, las *métricas* seleccionadas deben cubrir tres áreas - *producto*, *proceso* y *recurso* - y la lista debe incluir información que

asegure que quedan cubiertos los aspectos más relevantes para el éxito del proyecto.

Es difícil imaginar un proyecto en el que no sea importante algún nivel de medición y control sobre cada uno de los siguientes *atributos* para la consecución de un resultado exitoso [6]: *esfuerzo* (tiempo gastado durante las actividades del trabajo), *programa* (logro de los hitos objetivamente medidos), *coste* (gastos en las distintas clases de recursos, incluidos los recursos humanos), *progreso* (productos completados, aceptados y valores de referencia), *características del producto* (requerimientos implementados y en funcionamiento), *atributos de calidad del producto* (defectos, fiabilidad, disponibilidad, tiempo de respuesta, rendimiento y otros que se especifiquen) y *riesgo* (estado de los factores de riesgo y acciones para mitigarlos).

Las métricas propuestas cubren todos estos atributos y miden los siguientes aspectos del proyecto de desarrollo software:

1. Errores, fracasos (producto)
2. Modificaciones, fiabilidad (producto)
3. Otros fallos (producto)
4. Finalización (producto)
5. Complejidad (proceso)
6. Control de la gestión (proceso)
7. Evaluación de riesgos, beneficios y costes (proceso)
8. Recursos humanos (recursos)
9. Otros recursos (recursos)

La selección de *indicadores* incluye la medición de esfuerzos (en horas-persona) y costes (en dinero) tanto reales como planificados, número de entregables aceptados por el usuario y desviaciones, esfuerzo reutilizado de otros proyectos o inactividad en recursos humanos, número de modificaciones en el producto e información sobre su evaluación (solicitadas, rechazadas y aceptadas), esfuerzo dedicado a la detección y corrección de errores, número de revisiones finalizadas, e información sobre la detección de errores para evaluar la calidad del producto (por ejemplo errores detectados antes y después de la entrega). Asimismo, muchos de los indicadores incluyen mediciones desglosadas por fase (por ejemplo desglose en requerimientos, diseño, codificación y documentación).

En la *Tabla 1*, se indica el conjunto de métricas propuesto junto con las áreas y aspectos tratados por cada una de ellas.

Métrica	PRD	PRC	REC
Esfuerzo real en horas-persona		6,7	8
Esfuerzo planificado en horas-persona		6,7	8
Desviación esfuerzo		6,7	8
Coste real en dinero		6,7	8,9
Coste planificado en dinero		6,7	8,9
Desviación coste		6,7	8,9
Fechas corrección errores por fase	1	6,7	8
Horas-persona corrección errores por fase	1	5,6,7	
Horas-persona reutilizadas		6,7	8
Horas-persona en paradas		6,7	8
Entregables propuestos	4	6	
Desviación en tiempo por entregable	4	6,7	
Revisiones finalizadas por fase	4	6	
Tareas realizadas en gestión de riesgos		6,7	
Horas-persona tareas de gestión riesgos		6,7	8
Nº cambios solicitados por fase		6	
Nº cambios introducidos por fase	2,4	5,6	
Nº cambios rechazados por fase		6	
Horas-persona cambios por fase		5,6,7	8
Errores detectados antes de la entrega	1,2,3	6	
Nº Errores detectados tras la entrega	1,2,3	6	
Errores detectados tras la entrega (%)	1,2,3	6	
Tiempo medio de resolución de errores	1	5,6	

Tabla 1. Métricas propuestas y aspectos que miden

Estas métricas se evaluaron y contrastaron con la realización de un estudio cuyos resultados detallamos en el siguiente apartado.

## 5. Estudio propio.

En este apartado se presentan los resultados de un estudio propio realizado en varias empresas del País Vasco con el propósito de adquirir conocimientos de primera mano respecto a sus procedimientos y a las métricas que utilizan en la gestión de proyectos. La información obtenida permitirá verificar la validez del conjunto de métricas seleccionado.

A continuación se resume el perfil de las empresas estudiadas y se detallan los resultados obtenidos.

### 5.1. Perfil de las empresas

Las empresas estudiadas cumplimentaron vía Web una encuesta con 34 preguntas basadas en el desarrollo del software que fueron clasificadas dentro de las cuatro grandes áreas detectadas por los estudios previos.

Una vez cumplimentadas las encuestas se eliminó del estudio a una de las empresas, puesto que no se dedicaba al desarrollo software.

En cuanto al perfil de las empresas consideradas válidas, el 80% resultaron ser PYMEs según la definición por plantilla y facturación de la recomendación de la Comisión Europea.

El 40% de las empresas tienen una plantilla de menos de 10 trabajadores y se dedican al desarrollo de software libre, mientras que el 60% restante desarrolla software propietario. De estas últimas, una es una multinacional con miles de trabajadores.

Cabe destacar que en ninguno de los casos estudiados la empresa utiliza el modelo de desarrollo en cascada. Así, en el 60% de los casos el modelo de desarrollo es RAD (desarrollo rápido de aplicaciones) mientras que en el resto varía entre iterativo e incremental (iterativo con intenciones de modificarse en el futuro), espiral (evaluación y disminución constante de riesgos), metodologías ágiles (Scrum y eXtreme Programming) o modelo propio de la empresa.

En cuanto a los métodos y certificaciones para alcanzar la excelencia en el proceso (por ejemplo ISO/IEC 9000, CMMI, SEI TSP...) creemos que debido a que en su mayoría las empresas son PYMEs han optado por no escoger ninguna metodología/ certificación debido a su alto coste. Así, el 60% de las empresas encuestadas no utiliza ningún método. Del resto, el 20% utiliza Metrica3, el 10% EFQM y la multinacional utiliza varios estándares (CMMI, ITIL y otro propio de la compañía).

### 5.2. Seguimiento del programa

Como se ha indicado anteriormente, el seguimiento del programa se controla principalmente mediante métricas que miden el coste y el esfuerzo durante el desarrollo software, así como su desviación sobre lo planificado.

La mayoría de las métricas propuestas en el cuestionario son controladas con frecuencias variables por más del 50% de las empresas encuestadas. Menos de la mitad controlan el coste y el esfuerzo planificado por persona, la desviación total del coste y el coste planificado por tareas.

Estos resultados permiten afirmar que existe un control sobre costes y esfuerzos aunque debería existir un mayor control en los costes planificados.

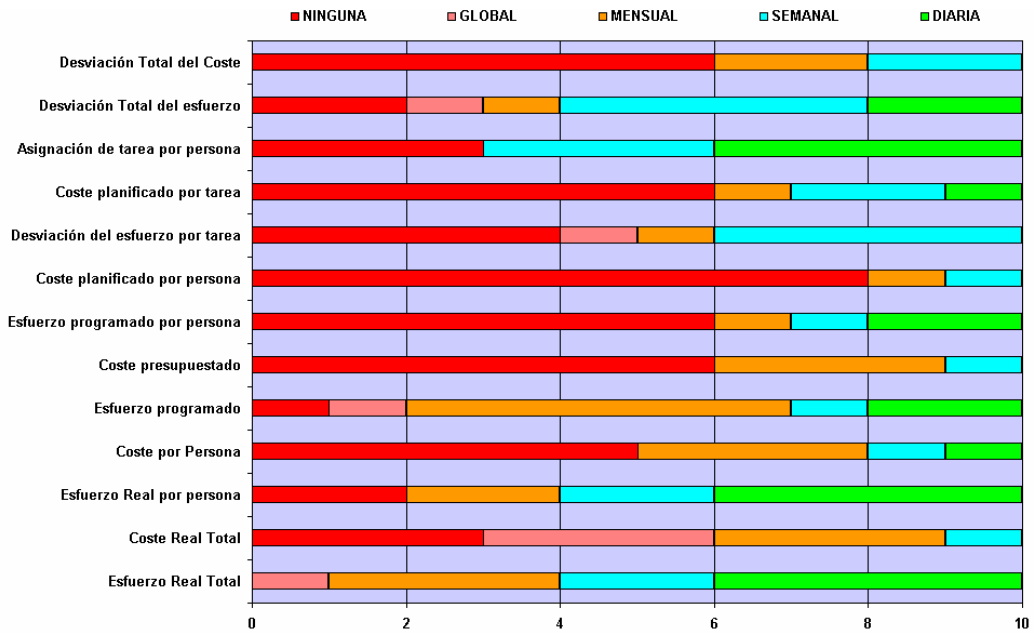


Figura 1. Frecuencia de control de cada métrica

### 5.3. Gestión de cambios en los requerimientos

En cuanto al control de cambios el 90% de las empresas encuestadas controla las modificaciones y la fase en la que se han producido y un 60% utiliza además algún otro tipo de clasificación para gestionarlas.

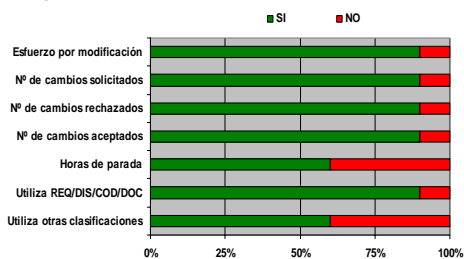


Figura 2. Control de modificaciones

Asimismo, se realiza algún tipo de seguimiento para saber qué y cuántas solicitudes se aprueban/rechazan/solicitan y se utiliza algún procedimiento para registrar la cantidad de tiempo

dedicado a cada modificación como se muestra en la Figura 3.

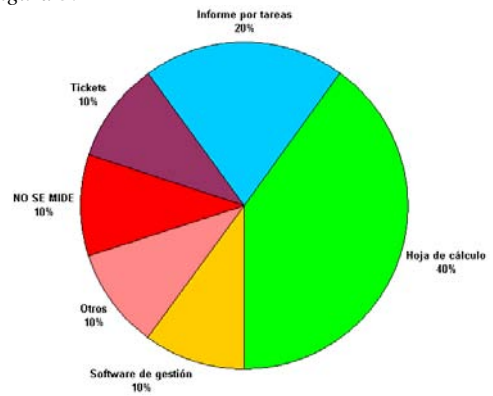


Figura 3. Técnicas en el control de modificaciones

### 5.4. Gestión de riesgos

Se realiza control de riesgos y también un seguimiento de las actividades definidas para mitigar el riesgo en el 70% de los casos.

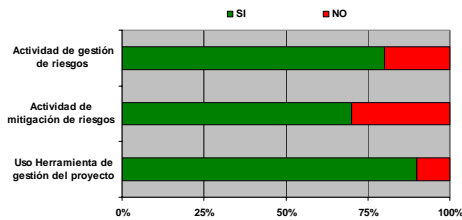


Figura 4. Control de riesgos

Tanto en la gestión de riesgos como en las demás áreas consideradas se emplean herramientas de apoyo adecuadas que facilitan el desarrollo y el control del proyecto disminuyendo el riesgo de fracaso. Todas las empresas encuestadas utilizan al menos una de estas herramientas.

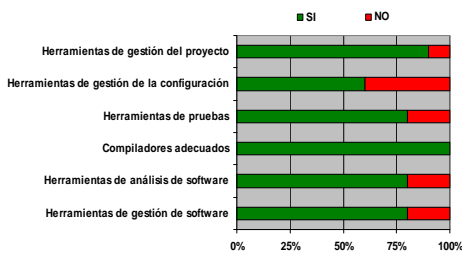


Figura 5. Uso de técnicas y herramientas de apoyo

### 5.5. Gestión de errores

El 90% de las empresas encuestadas mide el esfuerzo dedicado a la detección y corrección de los errores, mientras que el 60% utiliza algún método de detección (prevención de errores) que además considera efectivo.

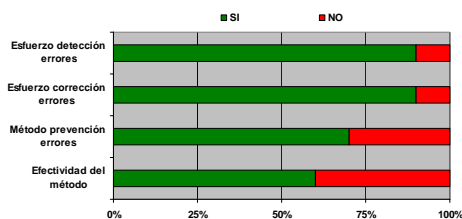


Figura 6. Control de errores

En cuanto al tipo de errores que se identifican y controlan, todas las empresas encuestadas controlan los principales tipos de errores en los

requerimientos, lógica y ejecución del producto software.

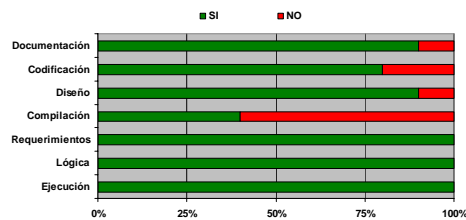


Figura 7. Tipos de errores controlados

En cuanto a la frecuencia en el control de los errores, solamente un 10% de las empresas deja todo el control para el final del proyecto mientras que las demás controlan los distintos tipos de errores en cada fase con periodicidades variables como se muestra en la Figura 8.

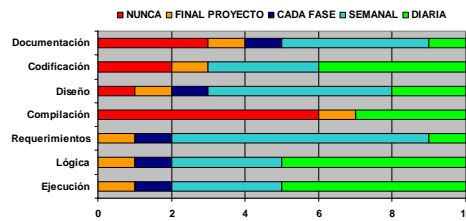


Figura 8. Frecuencia en el control de cada tipo de error

En cuanto a las técnicas utilizadas en la detección y corrección de errores todas las empresas encuestadas utilizan al menos una con mayor o menor eficacia.

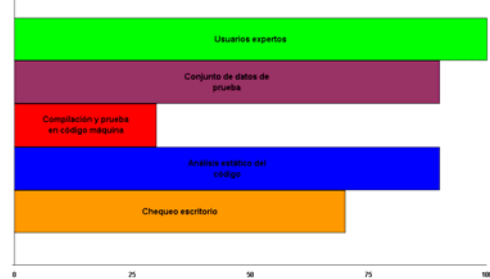


Figura 9. Uso de técnicas de detección de errores

Todas controlan el porcentaje de errores detectados por cada técnica utilizada sobre el total de errores encontrados. Las medias porcentuales

obtenidas en los cuestionarios se pueden ver en la *Figura 10*.



Figura 10. % de errores detectados con cada técnica

### 5.6. Conclusiones del estudio

El control de costes y esfuerzos y sus desviaciones no fueron tan estrictos como se recomienda (como se muestra en la *Figura 1. Frecuencia de control de cada métrica*), mientras que la clasificación y el control de errores es bastante completa en casi todos los casos.

Los resultados también muestran que en estas organizaciones se utilizan comúnmente diferentes técnicas de detección de errores, así como la gestión y actividades de mitigación de riesgos, y varias herramientas de soporte al desarrollo de software.

### 6. Trabajo futuro

A la vista de los resultados del estudio, es evidente que la mayoría de los indicadores sobre los que se preguntó a las empresas están siendo controlados actualmente, con mayor o menor frecuencia lo que respalda la validez de las métricas propuestas. El trabajo futuro consistirá en refinar esta selección inicial y utilizarla posteriormente en temas de control de proyectos software mediante representaciones visuales y simulación.

### Referencias

[1] Bannerman, P. (2008). Risk and risk management in software projects: A reassessment.

The Journal of Systems & Software, Vol. 81, No. 12, pp. 2118-2133.

[2] Boehm, B. (1989). Tutorial: Software Risk Management: IEEE Computer Society.

[3] Boehm, B. (1991). Software risk management: principles and practices. IEEE Software, Vol. 8, No. 1, pp. 32-41.

[4] DeMarco, T. & Lister, T. (2003). *Waltzing with Bears: Managing Risk on Software Projects*: Dorset House Publishing Co., Inc. New York, NY, USA.

[5] El Emam, K. & Koru, A. (2008). A Replicated Survey of IT Software Project Failures. IEEE Software, Vol. 25, No. 5, pp. 84-90.

[6] Fairley, R. (2009). *Managing and Leading Software Projects*: Wiley-IEEE Computer Society Press.

[7] Faulk, S. (1997). *Software Requirements: A Tutorial*. In *Software Engineering*. IEEE Computer Soc. Press, Los Alamitos, USA. pp. 82-101.

[8] IEEE (1988). IEEE Std 982.1-1988 IEEE Standard Dictionary of Measures to Produce Reliable Software.

[9] IEEE (1997). IEEE Std 1028-1997 IEEE Standard for Software Reviews and Audits.

[10] IEEE (2004). *A guide to the project management body of knowledge*. Project Management Institute, Pennsylvania, USA.

[11] Liggesmeyer, P., Heidrich, J., Münch, J., Barthel, H. & Zeckzer, D. (2009). Visualization of Software and Systems as Support Mechanism for Integrated Software Project Control. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends*. San Diego, CA: Springer-Verlag. pp. 846-855.

[12] Rome, I., Kusters, R., Cowderoy, A., Heemstra, F. & Trienekens, J. (1998). Project control for 2000 and beyond In *Proceedings of the European Software Control and Metrics Conference ESSCOM-ENCRESS 98*. Rome, Italy: Shaker Publishing BV. pp. 1-3.

[13] Rook, P. (1986). Controlling software projects. *Software Engineering Journal*, Vol. 1, No. 1, pp. 7-16.

[14] Thayer, R.H. (2001). *Software engineering project management*, 2nd edition: Wiley-IEEE Computer Society Press.