

KEMIS: ENTORNO PARA LA MEDICIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE

Moisés Rodríguez¹, Marcela Genero³, Javier Garzás^{1,2}, Mario Piattini³

¹*Kybele Consulting S.L.; Madrid, España*

²*Kybele Research, Dpto. de Lenguajes II, Universidad Rey Juan Carlos; Madrid, España,*

³*Grupo ALARCOS, Dpto. de Tecnologías y Sistemas de Información; Universidad de Castilla-La Mancha; Ciudad Real, España.*

¹Moises.Rodriguez, Javier.Garzas}@kybeleconsulting.com, ²Javier.Garzas@urjc.es,

³Marcela.Genero, Mario.Piattini}@uclm.es

Abstract Establishing measurement systems is a basic piece of software quality control, even more with the current trend of “outsourcing” the software development, which frequently is developed by external teams or software factories. Besides, to be efficient a measurement system of software product quality must have a high automation level, which allows its frequent use without consuming excessive time. Pursuing such goals, Kybele Consulting has proposed KEMIS, a measurement environment for software product quality. KEMIS uses free software tools, like Maven 2 and its measurement and reporting plugins, which allow the automatic calculation of metrics, the customization of results and the definition of the periodicity of measurements. The main goal of this paper is to describe the main characteristics of KEMIS.

Resumen Establecer sistemas de medición es una pieza básica del control de la calidad del software, más aún dentro de la actual tendencia a externalizar (outsourcing) el desarrollo, que frecuentemente es realizado por equipos o fábricas de software externos. Además, para que un sistema de medición de la calidad del producto software sea eficiente debe tener un alto nivel de automatización y posibilitar su uso de manera frecuente sin un excesivo consumo de tiempo. Persiguiendo éstos objetivos la empresa Kybele Consulting ha propuesto KEMIS, un entorno de medición para la calidad del software. KEMIS utiliza herramientas de software libre como Maven 2 y sus plugins de medición y reporte, lo que permite automatizar la obtención de las métricas, personalizar los resultados, y definir la periodicidad con la que se realizan dichas mediciones. El objetivo del presente artículo es presentar las principales características del entorno KEMIS.

Palabras Clave KEMIS, Calidad de Producto Software, Outsourcing, Externalización, Maven 2, PSM, Plugins de Medición.

1. INTRODUCCIÓN

Hoy en día las organizaciones, en la búsqueda de mayores ventajas competitivas, están invirtiendo cada vez más en automatizar sus procesos de negocio y en desarrollar nuevos servicios basados en tecnología software, lo que requiere una confianza creciente en los productos de desarrollo software, considerando el rol estratégico que estos tienen en las organizaciones. Por todo ello, el incremento y el control de la calidad de los productos software se ha convertido en una necesidad vital.

A la vez los productos software son cada vez de mayor tamaño, más sofisticados y complejos (Wong, Boehm et al. 2004)

(Wong, Verner et al. 2005), creando el reto de desarrollar dichos productos dentro de las restricciones de tiempo sin sacrificar la calidad.

Para poder asegurar que un proceso o sus productos resultantes son de calidad es necesario asignar valores, descriptores, indicadores o algún otro mecanismo mediante el cual se pueda llevar a cabo dicha evaluación. Para ello es necesario implantar un proceso de medición del software, que en general, persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en los proyectos y poder mejorar los procesos y productos (Fenton and Pfleeger 1997).

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo de software y los proyectos de mantenimiento (Briand, Morasca et al. 1996) y pueden ser utilizadas por profesionales e investigadores para tomar mejores decisiones (Pfleeger 1997).

La construcción de un entorno que permita llevar a cabo la medición, requiere tanto de un soporte metodológico como de un soporte tecnológico (Lavazza 2000). Por tanto, para que las métricas puedan ser utilizadas para evaluar los productos software de un modo práctico, eficiente y exacto, es necesario contar con herramientas que permitan automatizar la adquisición, la presentación y el análisis de los valores obtenidos para dichas métricas (Giles and Daich 1995).

En el presente artículo se describe el proyecto KEMIS (*Kybele Environment Measurement Information System*) el cual proporciona un entorno de software libre para la adquisición, presentación y análisis de métricas de calidad de producto software de manera automática. Además KEMIS presenta un soporte metodológico basado en las mejores prácticas de marcos de medición como Goal Question Metric (GQM), Goal Question Indicator Metric (GQ(I)M), Goal-Driven Software Measurement (GDSM), Practical Software Measurement (PSM), y está perfectamente alineada tanto con los estándares específicos de medición (ISO 15939, ISO 9126, IEEE Std 1061-1998, etc.) como con las actividades de medición de los principales modelos de procesos software y de madurez (CMMI, ISO 12207, ISO 15504).

El resto del artículo está organizado de la siguiente forma: en la sección 2 se ofrece una visión global del proyecto KEMIS exponiendo los objetivos que pretende alcanzar y el soporte metodológico y tecnológico que utiliza para ello. Posteriormente, en la sección 3, se estudia la arquitectura de KEMIS y las fases de implantación del entorno. En la sección 4 se exponen las principales métricas e indicadores de calidad del produc-

to estudiados en el proyecto. A continuación, la sección 5 muestra los beneficios de KEMIS tanto en empresas que externalizan su desarrollo como en fábricas de software. En la sección 6 se presentan un conjunto de recomendaciones y buenas prácticas propuestas por Kybele Consulting para la medición de la calidad del producto software. Finalmente en la sección 7 expondremos nuestras conclusiones, junto con las aportaciones de KEMIS y el trabajo futuro.

2. PROYECTO KEMIS

KEMIS "*Kybele Environment Measurement Information System*" es un entorno desarrollado por Kybele Consulting que proporciona, por un lado, un conjunto predefinido de aplicaciones de software libre, junto con su configuración e instalación, que permiten implantar un sistema de medición de la calidad software a nivel operativo, táctico y estratégico, y por otro, un soporte metodológico basado en PSM para la evaluación de la calidad del producto software.

Si bien el uso de métricas, revisiones, inspecciones o controles de calidad son actividades clásicas y con años de madurez, en la actualidad lo que hace que éstas se conviertan en práctica de uso, útiles y de la que se obtengan amplios beneficios es que se consideren los siguientes objetivos:

- 1) **Definir unos objetivos claros y medibles**, determinando qué datos son necesarios y por qué, evitando así riesgos como que el exceso de información haga perder el objetivo final del programa de medición.
- 2) **Realizar las mediciones de manera periódica y frecuente**, permitiendo tomar acciones correctivas en el momento oportuno, proporcionando la base de acciones preactivas más que reactivas.
- 3) **Automatizar el proceso de medición**, para posibilitar los anteriores

objetivos, ya que una medición muy costosa en tiempo o muy manual puede obstaculizar que se muestre de manera clara la información de carácter estratégico o que por su coste en tiempo baje su periodicidad y frecuencia.

- 4) **Definir diferentes niveles de abstracción**, que eviten perderse en los detalles y muestren a todos los niveles (operativos, tácticos y estratégicos) la información precisa para la toma de decisiones.

A continuación se presenta el modo en que KEMIS cubre los cuatro objetivos anteriores apoyándose en una serie de metodologías, técnicas y herramientas de medición que facilitarán a las empresas el proceso de medir la calidad de sus productos software.

2.1. Apoyo Metodológico

Para cumplir con el primer objetivo (definir unos objetivos claros de medición), KEMIS está basada en PSM “*Practical Software and System Measurement*”. PSM propone un proceso de medición que permite dirigir los objetivos técnicos y de negocio de una organización, y recoge las mejores prácticas utilizadas por los profesionales de la medición dentro de las comunidades del software, la adquisición de sistemas y la ingeniería.

Como se puede observar en la Figura 1, PSM sirvió de base al estándar ISO/IEC 15939, *Software Engineering - Software Measurement Process* (ISO/IEC 2002). Este estándar describe el proceso de medición en términos de propósito y resultados de un conjunto de actividades y tareas asociadas. PSM aporta detalles adicionales para llevar a cabo con éxito las tareas presentadas en el ISO/IEC 15939. El propósito y resultados del proceso de medición propuesto en el ISO/IEC 15939 han sido añadidos a la ISO/IEC 12207 y a la ISO 90003: aplicación de la ISO 9001 al software. Además la ter-

minología y conceptos de la ISO/IEC 15939 se han incluido en las normas ISO/IEC 15288, ISO/IEC 9126 y ISO/IEC 14598.

El estándar ISO/IEC 15939 fue también utilizado para la elaboración de la sección “*Measurement and Analysis*” (MA) de CMMI (Chrissis, Konrad et al. 2003). El área de proceso MA resume las actividades necesarias para llevar a cabo el proceso de medición y aporta una metodología para valorar si el proceso de medición se realiza de acuerdo al estándar.

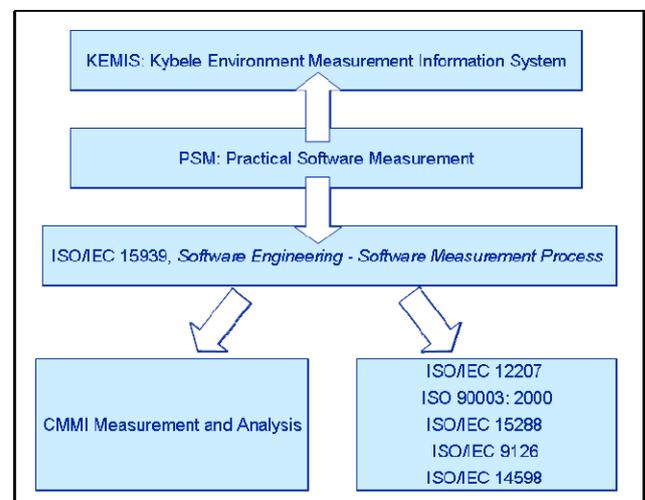


Figura 1. Relaciones entre Metodologías y Estándares de medición.

La aproximación para la selección y especificación de métricas propuestas por PSM se basa en la relación directa entre la necesidad de información y las medidas que soportan la información requerida.

Por todas estas razones, Kybele Consulting ha seleccionado PSM como proceso de referencia para llevar a cabo la implantación de la infraestructura KEMIS, considerando todas aquellas categorías y métricas relacionadas con la calidad del software y extendiéndolas con el conocimiento previo y los resultados conocidos en anteriores proyectos.

A partir de PSM y los *plugins* de medición existentes para Maven 2⁹, KEMIS propone una adaptación que permite relacionar las métricas e indicadores propuestos por PSM con los resultados obtenidos mediante la ejecución de los *plugins*. La Tabla 1 nos muestra la correlación establecida entre las métricas que propone PSM para estudiar la calidad del producto software y los datos que nos aportan los *plugins* de medición.

Se puede observar que los *plugins* de Maven 2 cubren las métricas propuestas por PSM respecto a componentes, líneas de código, defectos y complejidad ciclomática. Pero además, los *plugins* de medición aportan nuevas métricas como son:

- Líneas de código duplicado.
- Líneas de comentarios en el código.
- Responsabilidad de los paquetes.
- Independencia de los paquetes.
- Inestabilidad de los paquetes.
- Dependencias cíclicas entre paquetes.

En la Tabla 1 se muestran con fondo blanco las métricas e indicadores propuestos por PSM y la adaptación realizada por Kybele Consulting para relacionarlos con los *plugins* de medición. Con fondo gris, se representa la ampliación que aportan los *plugins* de medición a lo propuesto en PSM y la creación de nuevos indicadores por parte de Kybele Consulting que han sido integrados con el soporte metodológico.

2.2. Automatización de la medición

Para alcanzar el segundo y tercer objetivo (realizar las mediciones de una manera automatizada, periódica y frecuente), la infra-

estructura del entorno KEMIS se basa en Maven 2¹⁰, una herramienta software para la gestión y comprensión de proyectos Java que mediante sus *plugins* de medición permite obtener un conjunto de métricas de manera automática. Además, Maven 2 se basa en el concepto de integración continua (Fowler 1999; Pérez 2005), y mediante la herramienta Continuum, permite realizar una planificación de los procesos de medición.

Por otro lado, nuestra experiencia en auditorías de calidad nos indica que la aplicación de Maven 2 junto con sus *plugins* de medición es ideal para llevar a cabo la mejora continua y el control periódico de calidad de cada uno de los proyectos. Tal y como observamos en la Figura 2, la primera vez que se evalúa la calidad de un proyecto, es mayor el esfuerzo que hay que realizar utilizando entornos basados en Maven (ejecución en modo *batch*) respecto a otros entornos interactivos. Sin embargo, mientras en las siguientes mediciones el esfuerzo realizado utilizando entornos interactivos es el mismo, el esfuerzo necesario utilizando Maven es prácticamente nulo, obteniendo los resultados automáticamente a pesar de los cambios realizados en el proyecto.

⁹ Existen una gran cantidad de *plugins* para Maven 2 y entre ellos, son cada día más los que se dedican a estudiar la calidad del código y generar documentación al respecto.

¹⁰ <http://maven.apache.org/>

Issue - Category - Measure Mapping				MAVEN 2 Plugins				
Common Issue Area	Measurement Category	Measures	Indicators	PMD RULESET	CheckStyle CATEGORÍA	CPD	JavaNCSS	JDepend
Product Size and Stability	Physical Size and Stability	Components					Total number of packages Total number of classes Total number of methods	Classes per package
		Lines of Code	Software Size By Configuration Item Software Size Lines of Code				NCSS total NCSS per package NCSS per class NCSS per method	
Product Quality	Functional Correctness	Defects	Status of Severity 1 Defects Defect Density Defect Classification Defect Density Distribution	Basic JSP rules Basic Rules Braces Rules Clone Implementation Rules Controversial Rules Coupling Rules Design Rules Finalizer Rules Import Statement Rules J2EE Rules JavaBean Rules Jakarta Commons Logging Rules Java Logging Rules Naming Rules Strict Exception Rules String and StringBuffer Rules Unused Code Rules	Javadoc Comments Naming Conventions Headers Imports Size Violations Whitespace Modifiers Block Checks Coding Class Design Miscellaneous			
Supportability Maintainability		Cyclomatic Complexity	Software Complexity Units with Complexity > 10				CCN: Cyclomatic Complexity Number	
		Duplicate Code	% Duplicate Code			Sum Total lines		
		Comments	% Comments			Sum Total Comment lines		
		Responsibility	Responsibility Average				Sum Total Responsibility	
		Independence	Independence Average				Sum Total Independence	
		Instability	Instability Average				Sum Total Instability	
		Cycles	Sum Cycles				Sum Total Cycles	
Product Quality Product Size and Stability	Physical Size and Stability Functional Correctness Supportability - Maintainability	Defects Cyclomatic Complexity Lines of Code	Software Quality					

Tabla 1. Relación entre PSM y los plugins de medición.

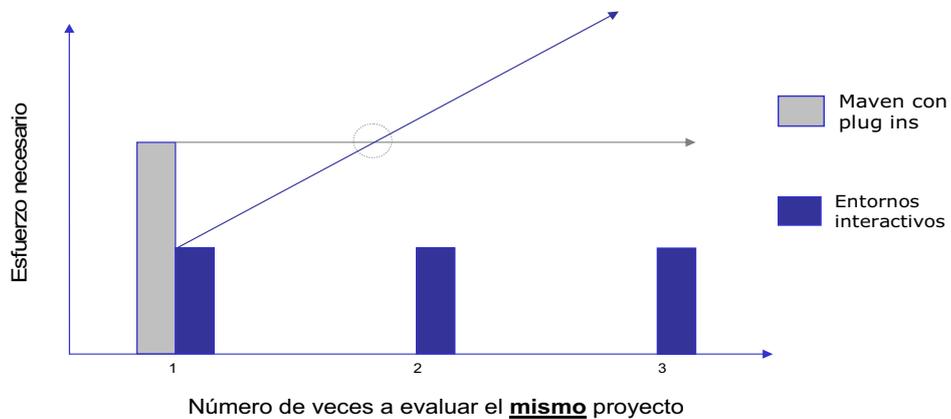


Figura 2. Evolución del esfuerzo para evaluar la calidad de un mismo producto

2.3 Presentación de Resultados

Una vez automatizado el proceso de medición de la calidad, debemos estar atentos a otra amenaza: la automatización puede generar tanta información en tan poco tiempo que en ocasiones puede ser peor que no tener información. Esto implica la necesidad de disponer de herramientas de síntesis, históricos, indicadores, tendencias, escalado de información, etc. (lo que en ocasiones se conoce como cuadro de mando).

Por ello, para cumplir el cuarto objetivo (obtener distintos niveles de abstracción en la presentación de resultados) y permitir realizar un análisis intuitivo de la calidad del producto software, KEMIS propone un conjunto de informes en los que recoge los principales indicadores de calidad. Para ello utiliza una base de datos MySQL donde almacena los resultados más representativos obtenidos por los *plugins* de medición y un servidor en el que se registran los informes, permitiendo después consultar y presentar automáticamente la información sobre la calidad del producto. En la sección 4 se muestran los indicadores de alto nivel obtenidos.

3. ARQUITECTURA DEL PROYECTO KEMIS

En la Figura 3 se observa la arquitectura del entorno KEMIS. En primer lugar, destaca la división de la imagen en dos mitades separadas por una línea discontinua. Esta separación corresponde a las dos fases en las que se realiza la implantación del entorno:

- **Infraestructura de medición básica (nivel operativo):** La primera fase corresponde a la instalación y configuración de Maven 2, así como de los *plugins* de medición. Al terminar esta primera fase, el usuario dispone de un entorno que le permite obtener

métricas de calidad de manera periódica y automática.

- **Infraestructura de medición avanzada (niveles táctico y estratégico):** La segunda fase corresponde a la instalación y configuración del entorno para la generación de informes. Al terminar esta segunda fase, el usuario dispone del entorno de medición KEMIS completo, lo que le permite, además de realizar las actividades de la primera fase, obtener informes personalizados con los principales indicadores de calidad del producto.

El funcionamiento de la infraestructura global queda resumido en los siguientes pasos:

1. Mediante Continuum se planifica una ejecución periódica de Maven 2.
2. Maven 2 se encarga de una ejecución asíncrona y predefinida de comandos configurados en un archivo ("pom.xml").
3. Maven/Continuum recuperan mediante el sistema de gestión de la configuración (SGC) los archivos del proyecto con los que debe trabajar.
4. Una vez recuperados los archivos fuentes del sistema de control de versiones, Maven realizará las acciones definidas en su configuración.
5. Los *plugins* de Maven 2 realizan el cálculo de métricas sobre los archivos obtenidos del repositorio y guardan el resultado en formato XML.
6. Los informes resultados en XML son estudiados para extraer de ellos los datos necesarios y almacenarlos en la base de datos.
7. Mientras, se crean las plantillas para los informes de calidad del

- producto software que mostrarán los principales indicadores.
8. Dichas plantillas se integran y configuran en la herramienta JasperServer.
 9. JasperServer extrae del almacén los datos necesarios para la creación de los informes mediante consultas SQL.

10. Finalmente, JasperServer genera los informes con los datos y formato indicados por el usuario.

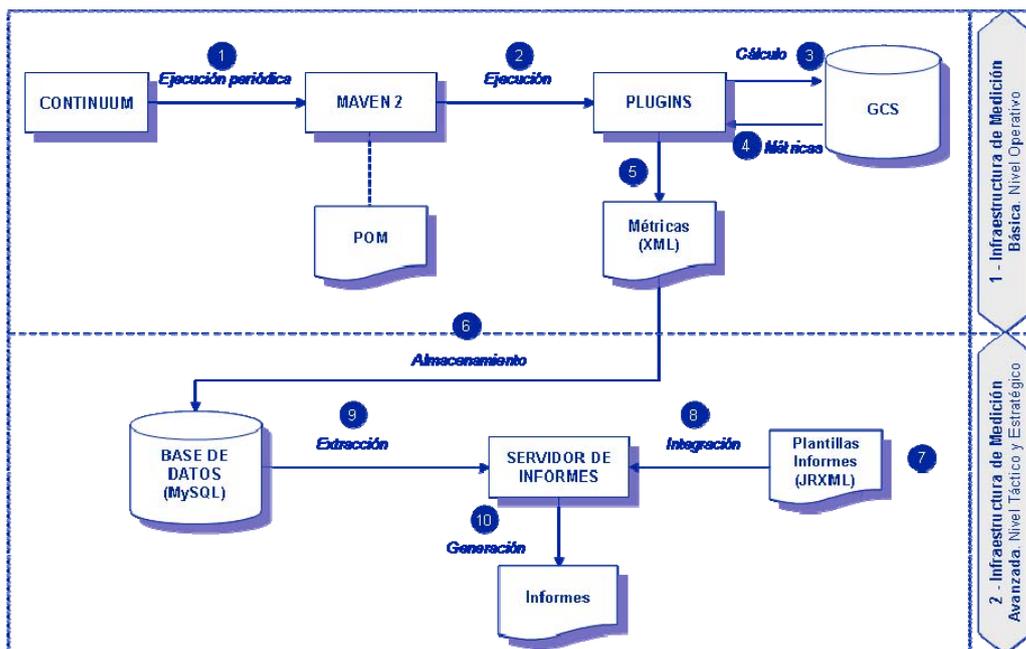


Figura 3. Arquitectura del entorno KEMIS.

CATEGORÍA	IDENTIFICADOR DE LA MÉTRICA
CÓDIGO INNECESARIO	Método finalize vacío Finalize sólo llama a "super" finalize Atributo privado no utilizado Variable local no utilizada Método privado no utilizado Parámetro no utilizado Modificador no utilizado Sentencia condicional vacía Bucle "while" vacío ...
DOCUMENTACIÓN	Javadoc en métodos Estilos javadoc Javadoc en clases e interfaces Estándares documentación ...
ESTILO PROGRAMACIÓN	Evitar asignaciones en operandos Llamar a "super" en un constructor

CATEGORÍA	IDENTIFICADOR DE LA MÉTRICA
	Bucles "for" que deberían ser "while" Sentencias "if" comprimibles Evitar asignaciones en parámetros Densidad en switch Evitar inicializadores estáticos Etiqueta default no está al final de sentencia switch Convenios para nombrar clases ...
MANTENIBILIDAD	Número elevado de imports Evitar un elevado número de "if" anidados Evitar lanzar ciertos tipos de error Evitar lanzar nullpointerexception Evitar literales duplicados Método muy largo Demasiados parámetros Clase muy larga Complejidad ciclomática ...
POSIBLES ERRORES	Método finalize sobrecargado El método finalize debe declararse "protected" Evitar llamar a finalize dentro del código Asignación del valor null Bloques "catch" vacíos Variable incremental entrelazada Sobrecribir equals y hashcode Bloqueo por doble comprobación Return en bloques finally ...
RENDIMIENTO	Atributo final podría declararse como static Llamada a método toarray optimizable Evitar concatenar cadenas en un stringbuffer Inicialización explícita
MÉTRICAS	Número de clases por paquete Número de líneas de código exceptuando comentarios Numero de bloques javadoc Líneas de documentación Complejidad ciclomática Bloques de líneas duplicadas Responsabilidad Independencia Nivel de abstracción Inestabilidad Dependencias cíclicas ...

Tabla 2. Categorías y métricas obtenidas mediante los plugins de medición.

4. MÉTRICAS E INDICADORES OBTENIDOS CON KEMIS

Mediante los *plugins* de medición disponibles para Maven 2, una vez implantada la infraestructura de medición básica de KEMIS, es posible obtener un amplio listado de métricas.

4.1 Clasificación de Métricas

La Tabla 2 presenta la clasificación de las métricas en un conjunto de categorías, así como algunas de las métricas pertenecientes a dichas categorías.

4.2 Indicadores

Debido a la gran cantidad de información generada mediante los *plugins* de medición y la dificultad de manejar dichos resultados en la manera que son generados, KEMIS propone y proporciona una serie de informes que recogen los principales indicadores de calidad del producto. Estos indicadores se encuentran clasificados en las siguientes categorías:

4.2.1 Componentes

Los indicadores pertenecientes a esta categoría sirven para dar un punto de vista global acerca del tamaño del proyecto bajo estudio en cuanto a número de elementos se refiere. A continuación se enumeran los informes que se podrán obtener de la presente categoría:

- Evolución de la cantidad total de paquetes que componen un proyecto determinado. Este indicador presenta el tamaño en paquetes de un proyecto y además muestra la variación de paquetes que ha sufrido a lo largo del tiempo.
- Evolución del número total de clases que contiene un proyecto determinado. Este indicador estudia el tamaño de un proyecto en cuanto a número de clases y además revela la varia-

ción de clases que va sufriendo el proyecto en las distintas mediciones que se realizan sobre el mismo.

- Variación del número total de métodos o servicios que presenta un proyecto determinado. Gracias a este indicador se puede observar como ha evolucionado el proyecto en cuanto a número de métodos.
- Proyectos por encima o debajo de un determinado umbral en cuanto a número de componentes. Este indicador estudia los proyectos que el usuario posee almacenados en su base de datos de mediciones que superan o no alcanzan un determinado umbral respecto al número de paquetes, clases o métodos.

4.2.2 Líneas de Código

Los indicadores pertenecientes a esta categoría sirven para dar un punto de vista global acerca del tamaño del proyecto bajo estudio en cuanto a número de líneas de código (NCSS) se refiere. A continuación se enumeran los informes que se podrán obtener de la presente categoría:

- Variación del número total de líneas de código (NCSS) que presenta un determinado proyecto. Este indicador muestra la evolución del número de líneas de código a lo largo de las distintas mediciones realizadas.
- Proyectos por encima o debajo de un determinado umbral de líneas de código. Este indicador estudia aquellos proyectos almacenados en el repositorio que superen o no alcancen un determinado número de líneas de código indicado por el usuario.

4.2.3 Defectos

Los indicadores pertenecientes a esta categoría sirven para dar un punto de vista global acerca de la calidad del código del proyecto

bajo estudio en cuanto a número de defectos se refiere. Para esta categoría destacan entre otros los siguientes informes:

- Total de defectos del código fuente del proyecto. Este indicador estudia el total de defectos encontrados en el código de un proyecto y además muestra la variación de defectos que ha sufrido el proyecto a lo largo de las distintas mediciones realizadas por el usuario.
- Proyectos que se encuentran por encima de un determinado umbral de defectos. Este indicador estudia aquellos proyectos que superen un número determinado de defectos totales. En la Figura 5 se puede observar un ejemplo del informe que se obtendría para este indicador.
- Evolución de la densidad de defectos detectados en el código. Este indicador estudia la densidad de defectos que presenta el proyecto bajo estudio relacionando el total de defectos con el total de líneas de código (NCSS). Además muestra como ha variado dicha densidad de defectos a lo largo de las distintas mediciones.
- Proyectos cuya densidad de defectos se encuentra por encima o debajo de un umbral. Este indicador muestra aquellos proyectos que superan o no alcanzan un determinado umbral especificado por el usuario respecto al valor de la densidad de defectos.

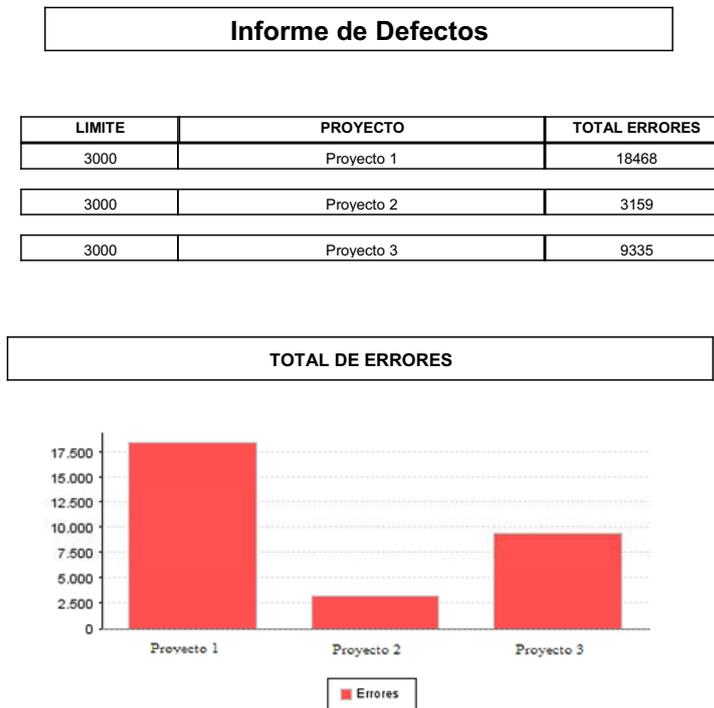


Figura 4. Ejemplo de informe de defectos.

4.2.4 Complejidad Ciclomática

Los indicadores pertenecientes a esta categoría sirven para dar un punto de vista global acerca de la calidad del código del proyecto bajo estudio respecto a la complejidad ciclomática que presenta. A continuación se enumeran los informes que se podrán obtener de la presente categoría:

- Variación de la complejidad ciclomática total de un determinado proyecto. Este indicador estudia la complejidad ciclomática total de un proyecto seleccionado por el usuario. Además muestra la evolución de dicha complejidad a lo largo de las distintas mediciones realizadas.
- Proyectos que superan o no alcanzan un valor límite de complejidad ciclomática total. Este indicador nos permite conocer aquellos proyectos que tienen una complejidad ciclomática total superior o inferior a un límite determinado.
- Evolución de la complejidad ciclomática media de las funciones de un determinado proyecto. Este indicador estudia la complejidad ciclomática media de las funciones de un proyecto seleccionado por el usuario. Además muestra la evolución de dicha complejidad a lo largo de las distintas mediciones realizadas.
- Proyectos que superan o no alcanzan un valor límite de complejidad ciclomática media por función. Este indicador nos permite conocer aquellos proyectos que tienen una complejidad ciclomática media por función superior o inferior a un límite determinado por el usuario.

4.2.5 CÓDIGO DUPLICADO

Los indicadores pertenecientes a esta categoría sirven para dar un punto de vista global acerca de la calidad del código del proyecto bajo estudio respecto a la cantidad de código

duplicado que presenta. Destacan los informes de:

- Evolución de la cantidad total de líneas de código duplicado. Este indicador estudia el número total de líneas de código duplicado que presenta un determinado proyecto seleccionado por el usuario. Además muestra la variación de líneas de código duplicado que ha tenido el proyecto a lo largo del tiempo.
- Variación de la cantidad total de bloques de código duplicado. Semejante al indicador anterior, salvo que trata al código duplicado por bloques en lugar de por líneas.
- Evolución de la densidad de código duplicado. Este indicador estudia la densidad de código duplicado que presenta el proyecto bajo estudio relacionando el total de líneas de código duplicado con el total de líneas de código (NCSS). Además muestra como ha variado dicha densidad a lo largo de las distintas mediciones.

4.2.6 RESUMEN SOBRE CALIDAD DE SOFTWARE

Indicador global cuya función es resumir el resultado de los indicadores anteriores, mostrando mediante un único informe aquella información considerada más importante para determinar la calidad de los proyectos estudiados. En la Tabla 2 podemos observar la información que presentará este indicador:

Producto	Tamaño	Defectos	Densidad defectos	N° de clases	Complejidad media
Proyecto 1	12345	5000	0,4	30	21,3
Proyecto 2	32523	45653
...

Tabla 3. Indicador global de Calidad del Software.

5. BENEFICIOS DE KEMIS

De manera general, KEMIS puede implantarse en la organización que quiera usarlo tanto para evaluar la calidad de los productos software que recibe, como de aquellos que han sido desarrollados internamente. Además, dada la creciente necesidad de certificarse en modelos de calidad de procesos como CMMI, las empresas necesitan implantar un sistema de medición tanto del proceso de desarrollo como del producto software. KEMIS provee a la organización de la infraestructura necesaria para llevar a cabo la medición de calidad del producto.

En el caso de fábricas de software, permite controlar la calidad de los desarrollos desde las primeras etapas, lo que se traduce en desarrollos más eficientes, de mejor calidad final, y más económicos. KEMIS está pensado para poder ser utilizado también por los desarrolladores, que pueden ir autoevaluando su trabajo mediante la infraestructura de medición básica. Todo ello permite crear una disciplina de control de la calidad frecuente, predictiva y con visibilidad a todos los niveles de la organización.

En el caso de empresas que han externalizado sus desarrollos software, KEMIS controla la calidad de las entregas realizadas por los proveedores, permitiendo evaluar grandes cantidades de desarrollos antes de los pasos a producción.

6. RECOMENDACIONES O BUENAS PRÁCTICAS SOBRE LOS ENTORNOS DE MEDICIÓN

A continuación presentamos un conjunto de buenas prácticas o recomendaciones que por nuestras experiencias nos parecen importantes, destacando:

- La evaluación cuantitativa del producto:
 - Es un método necesario no sólo en el desarrollo, también

si éste se externaliza, más aún si somos responsables de la puesta en producción.

- Permite identificar síntomas para poder aplicar soluciones correctas.
 - Aporta argumentos sólidos para defender y concienciar sobre la necesidad de mejora del proceso.
- La periodicidad del muestreo la dicta la complejidad del método/infraestructura de medición. Cuanto mas costoso resulta el proceso de medición, se llevarán a cabo menos mediciones y de menor calidad.
 - Una gran cantidad de pasos a producción, o un importante proceso de mejora en, por ejemplo, una fábrica software, necesitará de una alta periodicidad en la medición, así podremos controlar los efectos de las mejoras (“que arreglar algo no rompa otra cosa”), etc.
 - Hoy en día es fácil recopilar un gran volumen de datos sobre la calidad de los productos software, sin embargo, la dificultad está en saber que hacer con ellos.
 - Es necesario decidir qué mostrar, cuándo y a quién. Niveles de abstracción e información.
 - El proceso de definición de métricas es incremental y ha de ser mejorado continuamente, según las necesidades concretas del usuario y el producto.
 - También existe la opción de externalizar la propia actividad de evaluar la calidad del producto.
 - Nunca se debe perder de vista el objetivo final, que será alinear los objetivos de negocio con planes de mejora, incrementar la productividad y rentabilidad del desarrollo, aumentar

la calidad del software y la satisfacción del usuario, etc.

7. CONCLUSIONES Y TRABAJOS FUTUROS

En el desarrollo software actual, la medición es una pieza básica a la hora de controlar la calidad del producto, más aún si el desarrollo es realizado por equipos externos, donde el receptor del software debe establecer sus propios sistemas de control de la calidad. Es en este punto donde surge nuestro proyecto KEMIS, como una solución para aquellas empresas que desean realizar una medición de la calidad de sus productos software, especialmente aquellas que han externalizado el desarrollo de dichos productos y necesitan un análisis de calidad automatizado, planificable e intuitivo.

Actualmente existen proyectadas una serie de tareas para el futuro del proyecto KEMIS entre las que destacamos el hecho de integrar nuevos *plugins* de medición y documentación dado el auge y frecuencia con la que hoy en día se están desarrollando dichos *plugins*. Así como, almacenar mayor cantidad de datos procedentes de los resultados de la medición que nos permitan definir y generar nuevos indicadores e informes.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto GOLD (Ministerio de Educación y Ciencia TIN2005-00010/ - España) y el proyecto ESFINGE (Ministerio de Educación y Ciencia (FEDER) TIN2006-15175-C05-05).

REFERENCIAS

Briand, L., S. Morasca, et al. (1996). "Property-Based Software Engineering Measurement." IEEE Transactions on Software Engineering 22(1): 68-86.

Chrissis, M. B., M. Konrad, et al. (2003). CMMI: Guidelines for Process Integration and Product Improvement, Addison Wesley Professional.

Fenton, N. and S. Pfleeger (1997). Software Metrics: A Rigorous Approach. Londres, Chapman & Hall.

Fowler, M. (1999). Refactoring. Improving the Design of Existing Code, Addison-Wesley.

Giles, A. and G. Daich (1995). "Metrics Tools." Crosstalk, The Journal of Defense Software Engineering Febrero.

ISO/IEC (2002). "ISO/IEC 15939 International Standard Software engineering - Software measurement process."

Lavazza, L. (2000). "Providing Automated Support for the GQM Measurement Process." IEEE Software 17(3).

Pérez, J. (2005). "Integración Continua utilizando herramientas OpenSource." Agile-spain, from http://javahispano.net/frs/download.php/126/PonenciaIntegracionContinua_AgileSpain.pdf.

Pfleeger, S. L. (1997). "Assessing Software Measurement." IEEE Software March/April: 25-26.

Wong, B., B. Boehm, et al. (2004). Second Workshop on Software Quality. 26th international conference on Software engineering, St. Louis, MO, USA, ACM Press

Wong, B., J. Verner, et al. (2005). Third Workshop on Software Quality. 27th international conference on Software engineering, St. Louis, MO, USA, ACM Press

SITIOS WEB RECOMENDADOS

- <http://www.aemes.org/>, Asociación Española de Métricas del Software.
- <http://www.cosmicon.com/>, Sitio Web de COSMIC "Common Software Measurement International Consortium".
- <http://www.ifpug.org/>, Sitio Web de IFPUG "International Function Point Users Group".
- <http://www.isbsg.org/>, Sitio Web de ISBSG "International Software Benchmarking Standards Group".
- <http://www.nesma.nl/english/index.htm>, Sitio Web de NESMA "Netherlands Software Metrics Users Association".
- <http://www.ukσμα.co.uk/>, Sitio Web de UK-SMA "United Kingdom Software Metrics Association".