

## Persistencia de mediciones como apoyo a la gestión de proyectos de software

Greiner, C. (1); Dapozo, G.(1); Acosta, J. (1); Domínguez, M. (1) Chiapello, J. (1); Estayno, M.(2);

(1) Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura  
Universidad Nacional del Nordeste

{gndapozo, cgreiner}@exa.unne.edu.ar

(2)Departamento de Informática. Facultad de Ingeniería. Universidad Nacional de Lomas de Zamora  
mestayno@gmail.com

### RESUMEN

La medida de la calidad del software es una necesidad para las empresas de Software y Servicios Informáticos (SSI), porque representa una ventaja estratégica al proporcionar el conocimiento de los procesos productivos y permitir mejorar las tareas menos eficientes. La calidad del software está estrechamente vinculada con la medición del mismo. La implementación exitosa en una organización de una nueva práctica o el uso de una herramienta depende en gran medida de la automatización de la misma, a fin de no significar una pérdida de tiempo o uso de recursos adicionales. En este trabajo se presenta una línea de investigación que apunta a la elaboración de metodologías de medición de atributos, recolección de datos, y diseño e implementación de un repositorio de mediciones, que contribuya a una más eficiente gestión de proyectos de software y a la toma de decisiones en aspectos referidos a la calidad de software.

**Palabras clave:** Calidad de software. Herramientas de medición de software. Sistematización de la medición. Gestión de proyectos de software.

### Contexto

La línea de I/D presentada forma parte de las actividades definidas en el marco del proyecto F007-2009: "Modelos y métricas para la evaluación de la calidad de software", acreditado por la Secretaría General de Ciencia y Técnica de la Universidad Nacional del Nordeste (UNNE).

En el proyecto se trabajan diversas líneas, entre ellas una orientada a la calidad del producto software. El objetivo fundamental es brindar soluciones a las empresas de desarrollo de software de la región NEA en términos de modelos y herramientas que colaboren en la gestión de proyectos de software, particularmente en el proceso de mejora de atributos de calidad del producto final, mediante modelos y herramientas de automatización de medición de software.

### Introducción

El software se ha convertido en parte de la vida cotidiana. Hay una constante interacción con él, y los usuarios necesitan garantías de que el software utilizado cumple con determinados estándares de calidad, los cuales le garantizan un correcto

desempeño y el cumplimiento de sus requerimientos. A lo largo de los años se ha buscado permanentemente la forma de garantizar la corrección del software que se construye. Esto es tanto más importante cuanto más crítica sea la aplicación de que se trate, es decir, cuyas fallas pueden ocasionar daños de importancia (pérdida de vidas humanas, grandes pérdidas financieras, catástrofes nucleares, etc.). Por ello la implementación de procesos de mejora, gestión de atributos de calidad, mediciones de software y certificaciones de distintos estándares juegan cada vez un papel más importante en el ámbito del desarrollo de software.

La medida de la calidad del software es una necesidad para las empresas de Software y Servicios Informáticos (SSI), porque representa una ventaja estratégica al proporcionar el conocimiento de los procesos productivos y permitir mejorar las tareas menos eficientes.

La calidad del software está estrechamente vinculada con la medición del mismo. Autores como Piattini et.al [1] aseguran que la medición de atributos internos del software es el primer indicador de cumplimiento de atributos externos, como la mantenibilidad, funcionalidad etc.

En [2] se señala que atributos como la mantenibilidad y comprensibilidad son evaluados utilizando métricas de software, las cuales proveen un modo de representar en números atributos abstractos como la complejidad y el tamaño. Los mismos autores mencionan que la utilización de una sola métrica es insuficiente para analizar efectivamente atributos de calidad, por lo que sugieren utilizar un conjunto de métricas para evaluar cada atributo externo de calidad.

En [3] se destaca que el esfuerzo del testing debe estar enfocado en las clases cuyas métricas están por fuera de los valores recomendados. Identificar estas clases constituye un desafío y una ardua tarea, para lo cual se han probado muchas métricas, técnicas y modelos.

Por lo tanto en esta línea de investigación se han enfocado los esfuerzos al estudio de modelos y métricas de atributos de calidad del producto, y a la automatización de esta tarea, con el propósito de apoyar la gestión de proyectos de software para el desarrollo de productos de calidad.

### **Métricas de software orientado a objetos**

En el proceso de desarrollo de software, la medición de los atributos que definen la calidad del mismo permite a los desarrolladores analizar la eficacia de los modelos de análisis, diseño y código, así también como de la efectividad de los casos de prueba y la calidad general del software mientras se diseña o construye [4].

Las métricas para aplicaciones Orientadas a Objetos (OO) deben ajustarse a las características que distinguen al software de este paradigma del software convencional, por lo que hacen hincapié en conceptos tales como encapsulamiento, herencia y polimorfismo. Sin embargo, comparten los objetivos de las métricas convencionales: comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado a nivel del proyecto [5].

Dependiendo de la tecnología utilizada para el desarrollo del software, se encuentran disponibles diversas herramientas para la medición de atributos en aplicaciones OO, con diferentes características y nivel de desarrollo. Las mismas ayudan a comprender mejor el código fuente de los proyectos, entender su arquitectura, auditar la calidad del código, detectar las posibilidades de refactorización y reducción de la complejidad

### **Métricas de Software OO en JAVA**

Existe una gran variedad de herramientas para la medición de atributos en programas escritos en Java, que trabajan de forma independiente, o como plugin de otros entornos de desarrollo, tales como Eclipse metrics plugin, Chidamber and Kemerer Java Metrics, Coqua, JRefactory, EclipsePlug-in to calculate CKmetrics, JDepend, etc.

Entre ellas, destaca RefactorIT, que está disponible como un plugin de Eclipse, NetBeans, JDeveloper, o JBuilder. Su valor deriva de la baja curva de aprendizaje y no tiene costo adicional. Presenta en forma gráfica las dependencias entre paquetes, y permite exportar las mediciones en archivos de diferentes formatos, como CSV, HTML y XML.

### **Métricas de Software OO en .NET**

Del relevamiento y análisis de herramientas que permiten la medición de aspectos inherentes al diseño e implementación de software del entorno .NET, se seleccionaron SourceMonitor y NDepend. Estas implementan un gran número de métricas, incluidas las de Chidamber y Kemerer, uno de los conjuntos de métricas de software OO a los que se hace más ampliamente referencia [6]. Ambas permiten comparar las mediciones de diferentes versiones, proporcionando información sobre la evolución entre etapas de refactorización del código. También permiten exportar el resultado de las mediciones a archivos de distintos formatos, como HTML y XML.

### **Métricas de Software OO en PHP**

PHP es uno de los lenguajes más utilizados en el desarrollo de aplicaciones web. Forma parte de las tecnologías LAMP (Linux Apache, MySQL, PHP/Python/Perl), siendo esta plataforma una de las más extendidas en cuanto software open source para desarrollo web se refiere [7], y en sus últimas actualizaciones se introdujo un modelo de objetos, lo cual dio lugar a la necesidad de mediciones de atributos OO en aplicaciones PHP. Un relevamiento de las herramientas de medición para aplicaciones de estas características permitió establecer que PHPDepend es la que mayor cantidad de información brinda, sin embargo, no aporta una cantidad significativa de información en comparación a herramientas para otros lenguajes y entornos de desarrollo.

### **Métricas de software como apoyo a la gestión de proyectos**

Los ambientes cambiantes en el área de desarrollo de software y la competencia globalizada han cambiado el modo en que se lleva a cabo actualmente la administración de proyectos de software. Las nuevas tecnologías, la mejora de la calidad, generación y distribución de conocimiento, coordinación de varios proyectos, etc. evidencian la necesidad de nuevas herramientas e iniciativas de apoyo a la administración de proyectos, las cuales deben adoptarse considerando las características y objetivos propios de la organización. Las tecnologías de información juegan un papel clave en esta evolución.

La mejora en las prácticas de la administración de proyectos de desarrollo de software, dadas las características de la competitividad actual, solo puede lograrse por medio de la integración de múltiples herramientas, el uso de nuevas técnicas, etc, sin olvidar el fomento al aprendizaje organizacional [8].

Para la implementación exitosa en una organización de una nueva práctica o el uso de una herramienta se debe tener en cuenta la obligatoriedad del carácter sistemático y automatizado que esta debe presentar. Incorporar la medición en un proyecto de software, no debe significar una pérdida de tiempo o uso de recursos adicionales, sino que debe ser vista como una actividad que aporte información automática, rápida, sencilla y estandarizada de las distintas etapas por las que pasa un proyecto software para contribuir a la toma de decisiones.

Esta línea de investigación se enfoca en metodologías de medición de atributos, recolección de datos, estado de proyectos etc., y al diseño e implementación de un repositorio de mediciones, que contribuya la toma de decisiones en la gestión de proyectos de software.

## Líneas de investigación y desarrollo

Las principales líneas del proyecto están orientadas hacia el análisis, estudio y discusión de modelos de evaluación de calidad, estándares, metodologías y herramientas, especialmente enfocadas a la aplicabilidad en las pymes de software.

En particular, se trabaja actualmente en las siguientes líneas:

- Medición de atributos de calidad OO en aplicaciones desarrolladas en los lenguajes de programación Java, PHP y .NET
- Herramientas software que automatizan el cálculo de las métricas OO y generan información para su análisis.
- Persistencia de los valores de las distintas mediciones de los atributos de calidad definidos, integrándolos en una base de datos.
- Conformación de una base de conocimiento para contribuir a la toma de decisiones en la gestión de proyectos de software.

## Resultados y Objetivos

En esta etapa del desarrollo del proyecto algunos resultados obtenidos fueron:

- Se elaboró una metodología para la medición de atributos de calidad y determinación de nivel de complejidad en aplicaciones orientadas a objeto desarrolladas en Java, utilizando la herramienta RefactorIT [9].
- Se realizó un relevamiento y análisis de herramientas de medición de atributos de calidad de aplicaciones OO desarrolladas en .NET [10].
- Se elaboró una metodología de medición de atributos característicos de la programación orientada a objetos sobre aplicaciones escritas en PHP 5, con el objetivo de conocer qué tan bien se implementan los conceptos de la POO en aplicaciones PHP. La misma fue aplicada para evaluar cuatro frameworks de desarrollo PHP, que soportan POO, utilizando la herramienta PHPDepend cuyos resultados se describen en [11].

Los trabajos que se encuentran en curso son:

- Diseño y desarrollo de una herramienta de medición de atributos de calidad orientados a objeto para evaluación de código PHP ya que, como se mencionó en [11], las herramientas relevadas proveen información generalizada sobre un conjunto de atributos, que no permite identificar fehacientemente el elemento que presenta alguna falencia. No especifican la clase o método exacto que contiene el código que mayormente influye en el resultado negativo de la medición, representado con un

valor de medición fuera del rango de aceptación. Por tal motivo se propone el desarrollo de una herramienta de medición con el objetivo de proporcionar a los desarrolladores y directores de proyectos, información sobre el estado de calidad del software de sus proyectos y un informe detallado señalando las porciones de código donde se producen resultados de mediciones fuera de rango.

- Se encuentra en etapa avanzada de desarrollo una aplicación orientada a la persistencia de las mediciones de complejidad de software orientadas a la generación de un repositorio de mediciones provenientes de diversos proyectos, cuyas características se describen más adelante.
- Finalizada la herramienta, se espera contar con la colaboración de empresas de desarrollo del medio, para la recolección de datos reales, y su posterior análisis y emisión de informes, que contribuyan a la administración de proyectos de software.

## Persistencia de las mediciones

Con este objetivo se encuentra en desarrollo una herramienta que permite registrar mediciones de atributos de calidad de software en un repositorio de datos cuantitativos, de modo tal de generar un registro histórico como apoyo a la gestión de proyectos de desarrollo de software, que posee las siguientes funcionalidades:

- **Importar medición:** Permite la selección del archivo que contiene la medición que provee la herramienta de cálculo de las métricas. Analiza la sintaxis de la información adaptando al modelo subyacente, y registra en la base de datos.
- **Informes:** Permite la generación de reportes basados en las mediciones registradas.
- **Gestión:** Permite la administración de las métricas disponibles, como ser su edición o consulta.

Las mediciones de los atributos de calidad son generadas por herramientas existentes en el mercado para el cálculo automatizado. Actualmente la aplicación permite la manipulación de información generada por *RefactorIT*, para software desarrollado en java. Merced a la arquitectura e implementación de patrones detallados más adelante es posible extender la aplicación permitiendo la importación de mediciones generadas por otras herramientas de medición como por ejemplo las que ofrece PHP Depend, entre otras.

Los informes proveen información detallada, señalando clases con posibles dificultades y sugerencias para su corrección, e información

resumida, indicando en qué porcentaje las clases se encuentran dentro de los valores recomendados.

Esta información permite un análisis puntual del código, permitiendo detectar situaciones que pueden afectar la calidad del software. De este modo, se podrá tomar mejores decisiones respecto a la implementación de mejoras correctivas.

Dada la necesidad de hacer sistemas más modulares, que permitan la reutilización de componentes, reduciendo de ese modo el costo de desarrollo, en el diseño de la presente aplicación se prestó especial cuidado de la arquitectura de la misma. Una arquitectura bien implementada permite mantener un bajo acoplamiento entre los elementos del sistema, con muy alta cohesión, dado que se define una clara estructura de los componentes, sus formas de comunicarse y las relaciones que existen entre ellos.

Para el desarrollo de esta aplicación se optó por el Modelo Vista Controlador (MVC), patrón de arquitectura de software que separa los datos, la interfaz de usuario y la lógica de control de una aplicación en tres componentes distintos. Esta arquitectura permite desarrollar software cuyo mantenimiento y evolución son fáciles de realizar, contribuyendo con estas características a la calidad del software.

También se tuvo en cuenta el uso de patrones, utilizándose los siguientes:

- **DAO:** (Objeto de Acceso a Datos) permite abstraer a la aplicación de la tecnología de persistencia subyacente a través de una interfaz común, logrando un bajo acoplamiento e independencia con respecto al modo y fuente de acceso a datos (base de datos, archivos, servicios externos, etc.).
- **Strategy** (patrón de diseño): Permite definir un grupo de clases que representan un conjunto de posibles comportamientos, los cuales pueden ser fácilmente intercambiables en una aplicación, modificando la funcionalidad en cualquier instante. Este patrón se utiliza en la funcionalidad de importación de las mediciones, brindando flexible respecto a la incorporación de mediciones provenientes de otras herramientas, no previstas inicialmente.

Para la implementación de la aplicación se eligió el lenguaje de programación orientado a objetos Java debido que ofrece portabilidad y robustez, y la persistencia de los datos se realiza en una base de datos relacional, con el sistema de gestión de bases de datos MySQL, efectuando el mapeo de los datos con el ORM Hibernate.

Se utilizan un conjunto de librerías para optimizar el desarrollo, entre ellas se cuentan: Hibernate, Spring, Jasper Reports, IReport, Jdom y Swing.

## Formación de Recursos Humanos

En esta línea desarrollan su plan de actividad tres becarios de pregrado de la SECYT-UNNE, una estudiante de la Licenciatura en Sistemas de Información de la UNNE quien desarrolla su Trabajo Final de Aplicación y una docente de la misma carrera avanza con su tesis de Maestría Ingeniería de Software.

## Referencias

- [1] Piattini, M.; Garzas, J.; García, F.; Genero, M. "Medición y estimación del software". ISBN: 9789701514139 - Editorial ALFA OMEGA. 2008.
- [2] Herbold, S.; Grabowski, J.; Waack, W. "Calculation and optimization of thresholds for sets of software metrics". Empir Software Eng. 2011.
- [3] Kpodjedo, S.; Ricca, F.; Galinier, P.; Guéhéneuc, Y.; Antoniol, G. "Design evolution metrics for defect prediction in object oriented systems". Empir Software Eng. 2011.
- [4] Pressman, R. "Ingeniería de Software. Un enfoque práctico". McGraw-Hill – 2005.
- [5] González, D. "Las Métricas de Software y su uso en la Región". Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Universidad de las Américas Puebla. 2001.
- [6] Chidamber, S., Kemerer, C., "A Metrics Suite for Object-Oriented Design." M.I.T. Sloan School of Management E53-315, 1993.
- [7] Fernández, A. Comparativa proyectos PHP para e-commerce. Revista Todo Linux. N°. 112, 2010, págs. 26-30.
- [8] Caballero Cervantes, O.H. "Tecnologías de información y herramientas para la administración de proyectos de software". Revista Digital Universitaria. 2006 • Volumen 7 Número 6 • ISSN: 1067-6079.
- [9] Demchum, D., Greiner, C., Dapozo G.; Cuenca Pletsch L., Estayno M. "Medición de Atributos de Calidad en Aplicaciones Orientadas a Objeto". Publicado en la revista "La UTN en el NEA: Investigación y Desarrollo en la Regional Resistencia". 2011.
- [10] Chichi López, Elías D.; Greiner, Cristina L.; Dapozo, Gladys N. Métricas Técnicas de Software. "Análisis comparativo de herramientas de medición". Reunión de Comunicaciones Científicas y Tecnológicas 2012. Universidad Nacional Del Nordeste (UNNE). Modalidad: Póster y Exposición oral. Con referato. 27, 28 y 29 de Junio de 2012. Campus Universitario. Corrientes. Argentina.
- [11] Acosta, J.; Greiner, C.; Dapozo, G.; Estayno, M. "Medición de atributos POO en frameworks de desarrollo PHP". CACIC-2012. Publicado en anales. ISBN 978-987-1648-34-4.
- [12] Stelting, S.; Maassen, O. "Patrones de diseño aplicados a java". ISBN: 9788420538396. Prentice Hall. 2003.
- [13] Spring Source Feature Tour – Sitio oficial del proyecto. <http://www.springsource.org/features>