

# Patrones de Estrategia para Proyectos de Medición, Evaluación y Mejora

Belén Rivera, Pablo Becker, Luis Olsina

GIDIS\_Web, Facultad de Ingeniería, UNLPam, General Pico, LP, Argentina  
belenrs@yahoo.com, {beckerp, olsinal@ing.unlpam.edu.ar}

**Abstract.** Para realizar actividades de medición y evaluación, se debe partir de un claro objetivo de proyecto, como por ejemplo, evaluar el estado actual de una aplicación móvil -cuya categoría de entidad es Sistema- respecto a la característica Usabilidad -con foco en Calidad Externa. Para lograr este objetivo, se debe seleccionar una estrategia la cual debe especificar un curso de acción mediante la definición de un proceso con actividades y métodos bien establecidos. Otro objetivo similar puede involucrar la misma categoría de entidad citada pero con respecto a Seguridad, para lo cual se puede emplear la misma estrategia. Pero si el objetivo fuera evaluar, además de la situación actual, la situación ulterior luego de introducir cambios en la aplicación, la estrategia que se seleccionaría es otra. Para este fin, proponemos patrones de estrategia que provean soluciones reusables a problemas recurrentes en proyectos de medición, evaluación y cambio, considerando distintas vistas de calidad, en donde una vista de calidad relaciona un foco con una súper categoría de entidad. Los patrones analizados se ilustran con pruebas de concepto.

**Keywords:** patrón, estrategia, evaluación, cambio, vistas de calidad, proyecto.

## 1 Introducción

En toda organización de software madura, se debe asegurar la mejora continua de la calidad de los productos, servicios, procesos y recursos. Para ello, no solo es necesario contar con actividades de Medición y Evaluación (ME) que permitan conocer la situación real o estimada de un ente, analizando y detectando debilidades y vulnerabilidades, sino que también es preciso contar con actividades que permitan introducir cambios orientados a la mejora (MEC –Medición, Evaluación y Cambio). Un modo ingenieril de organizar las actividades y recursos es mediante la gestión de proyectos.

Considerando la definición de proyecto dada en CMMI [5] y PMBOK [21], la hemos redefinido como “*un esfuerzo temporal y orientado a metas con fecha de inicio y fin establecidos, que considera un conjunto gestionado de actividades, tareas y recursos interrelacionadas para producir y modificar productos de trabajo únicos (esto es, artefactos, servicios o resultados) para satisfacer una necesidad dada de un solicitante*”. Durante la planificación de un proyecto se establecen objetivos y se escogen estrategias a seguir, entre otros aspectos. La estrategia seleccionada debe estar orientada a alcanzar el objetivo del proyecto siguiendo un curso de acción determina-

do. Viendo la definición de estrategia dada en [1], además de otras, la hemos redefinido como “*principios, patrones y marco conceptual de un dominio particular que pueden ser especificados por un conjunto de procesos, además de un conjunto apropiado de métodos y herramientas como recursos principales, con el fin de alcanzar el objetivo del proyecto*”. Así, las organizaciones de software que deseen llevar adelante proyectos de ME o MEC de un modo sistemático y que aseguren repetitividad y consistencia de procesos y resultados deben contar con un enfoque de evaluación.

En este trabajo proponemos un *enfoque genérico e integrado de evaluación de calidad* cuya arquitectura se basa en dos pilares: 1) *un marco de modelado de calidad* [16], que permite especificar vistas de calidad y relaciones entre las mismas, y 2) *estrategias integradas de ME/MEC* [17], que se adecúan a las vistas de calidad que se deseen evaluar según los objetivos del proyecto. A su vez, una estrategia debe integrar al menos tres capacidades, a saber: un marco conceptual de dominio, una especificación de procesos y una especificación de métodos. Este principio de las tres capacidades integradas permite, entre otros beneficios, tener explícitamente definido el dominio de ME y MEC, saber qué actividades genéricas y específicas intervienen, y cómo llevarlas a cabo.

Respecto del primer pilar, en ISO 25010 [10] se representa a las vistas de calidad y relaciones entre las mismas. Por ejemplo, se indica que la Vista de Calidad de Producto software ‘influencia’ a la Calidad de Sistema. Sin embargo, en dicho documento no queda explícito el significado del concepto vista de calidad, el cual definimos como la asociación entre una categoría de entidad del más alto nivel de abstracción (como recurso, proceso, producto, sistema, etc.) y el foco de calidad a evaluar (p. ej. Calidad Interna, Calidad Externa, etc.). Por otro lado, la mayoría de los enfoques de ME actuales, como por ejemplo GQM<sup>+</sup>Strategies [1], no tienen en cuenta a las vistas de calidad, y mucho menos la selección de una estrategia apropiada para la vista o vistas a evaluar.

Respecto a estrategias aplicables a proyectos de ME/MEC existen varias propuestas, como p. ej. GQM<sup>+</sup>Strategies, FMESP [8], GOCAME [17], etc. Si bien cada situación o problema en que la estrategia se aplica al proyecto es única, se pueden identificar sin embargo aspectos comunes. Sólo por mencionar algunas situaciones, hay proyectos que están enfocados en mejorar la calidad de proceso, en tanto que otros en mejorar la calidad de producto, o en mejorar la calidad de un sistema. Por lo tanto, se observa que a menudo los proyectos comparten los mismos objetivos de MEC que involucran a distintas vistas, en donde, cada vista relaciona una súper categoría de entidad (como proceso, producto, sistema) con su respectivo foco de calidad (Calidad de Proceso, Calidad Interna, Calidad Externa). Así, diferentes proyectos aunque con características similares, pueden utilizar la misma estrategia. Por lo que hemos avizorado que una estrategia puede aplicarse a problemas recurrentes de proyectos MEC, y que es posible por lo tanto desarrollar un conjunto de patrones de estrategia.

Un patrón provee una solución, probada y documentada, a problemas comunes en contextos similares. Del mismo modo que en Ingeniería de Software, un conjunto de patrones de diseño dados especifican soluciones reutilizables a problemas de diseño (arquitectural, navegacional, de interface, de programación) particulares, en este trabajo especificamos un conjunto de patrones de estrategia que dan una solución en la

instanciación de una estrategia concreta en proyectos de ME/MEC. Estos patrones pueden ser empleados en proyectos cuyos objetivos involucren la evaluación de una o más vistas de calidad y contexto dados. Los patrones de estrategia se especifican –como en [7]– mediante una plantilla que describe, entre otros aspectos, la intención, motivación (problema), aplicabilidad, estructura (solución), escenarios, etc.

Las contribuciones de este trabajo son: i) *agregar un nuevo componente al marco conceptual de ME* previamente desarrollado [4, 17] que permite modelar vistas de calidad; ii) *robustecer el enfoque genérico de evaluación* propuesto en [16] considerando patrones de estrategia para distintas vistas de calidad; iii) *especificar patrones de estrategia* que den solución a proyectos de ME y MEC; y iv) *ilustrar un patrón de estrategia* y su instanciación.

El resto del artículo está organizado de la siguiente manera: La Sección 2 analiza trabajos relacionados a patrones de estrategia y a estrategias integradas de MEC. La Sección 3 introduce el marco de modelado de calidad en consideración de vistas para distintas categorías de entes y focos de calidad a evaluar. La Sección 4 documenta detalladamente un patrón de estrategia y discute otros escenarios. Finalmente, la Sección 5 resume las principales conclusiones y líneas de avance.

## 2 Trabajos Relacionados

Revisando la literatura respecto a enfoques de ME e incluso considerando trabajos sobre mejora continua, no se observa ni el diseño de una arquitectura ni el uso de un enfoque integral de evaluación de calidad basado en ambos pilares mencionados: 1) *marco de modelado de calidad*, y 2) *estrategias integradas de ME/MEC*. Aunque existen algunos trabajos que tratan de modo parcial uno de los dos pilares; es decir, hay un par de estrategias de ME que soportan, en mayor o menor grado, el principio de ser una estrategia integrada que incluya un marco conceptual, un proceso y métodos especificados. Pero ninguna estrategia se especifica como un patrón de estrategia para ser instanciado en proyectos de ME/MEC. Respecto al otro pilar, el marco de modelado de calidad, hay trabajos que tratan con vistas y modelos de calidad, pero no se enlazan con estrategias. A seguir, se discute con más detalle estos dos aspectos.

Respecto a estrategias integradas, en [1] se presenta a GQM+Strategies la que se construye sobre *Goal Question Metric* (GQM). Ambas estrategias son utilizadas para lograr objetivos de medición y satisfacen el principio de integración de las tres capacidades mencionadas [20]. En GQM y GQM+Strategies una meta organizacional, se descompone en objetivos de medición para los cuales se establecen preguntas y a partir de ellas, se derivan métricas para responder dichas preguntas. GQM+Strategies, a diferencia de GQM, permite alinear los objetivos a distintos niveles organizacionales, otorgando así un mecanismo para alinear objetivos de medición con objetivos tácticos y estratégicos. Ambas estrategias no consideran las vistas de calidad y las relaciones entre ellas ni tampoco se describen a partir de patrones de estrategia. En [13] se propone definir patrones de medición usando al enfoque GQM para definir objetivos, sub-objetivos y métricas a partir de una meta organizacional. La intención de estos patrones de medición es proveer soluciones reusables a problemas similares

encontrados en la creación de programas de medición. Si bien los autores indican que la idea de patrones de medición fue tomada de [7], la especificación del patrón ilustrado no sigue ninguna estructura (p. ej., nombre, intención, problema, solución/estructura, usos conocidos, etc.), ni siquiera la estructura propuesta en [7]. Otra contribución respecto a estrategias es FMESP [8], sin embargo no satisface el principio de las tres capacidades integradas [20], ni presenta especificaciones de patrones de estrategia.

Finalmente, [9] es una propuesta basada en patrones considerando la estrategia GQM. Los autores estructuran en forma de patrones las experiencias almacenadas en lo que llaman Factoría de Experiencias [2] y forma parte del *Quality Improvement Paradigm* (QIP) [3]. Estos patrones facilitan la selección y reuso de las experiencias almacenadas en la Factoría, aspecto de relevancia en el aprendizaje de experiencias pasadas en un ambiente de mejora continua. Los patrones se especifican utilizando una plantilla similar a la utilizada en [7]. Aunque QIP contempla el diseño de patrones para la especificación de experiencias, no es una propuesta semejante a la nuestra dado que no representa una estrategia integrada de ME/MEC. Es decir, QIP es un enfoque de mejora continua no dirigido por objetivos de evaluación y cambio de manera integrada.

En cuanto a trabajos relacionados respecto a vistas de calidad y sus relaciones – aspecto 1) antes mencionado-, podemos citar el aporte realizado en ISO 25010 [10], en donde se representa a diferentes vistas de calidad y relaciones entre las mismas. Sin embargo, en este estándar no se define explícitamente el significado del concepto vista de calidad. Tampoco se especifica un enfoque integral de evaluación que utilice estrategias y vistas de calidad, sino más bien esquematiza a las vistas en el contexto del ciclo de vida de calidad para un producto software evaluado con los modelos de calidad que propone.

Otra iniciativa relacionada a vistas de calidad se discute en [14]. A través del empleo de redes Bayesianas determinan la relación de ‘influencia’ de características de Calidad Externa sobre las de Calidad en Uso tomando como referencia a la norma ISO 9126 [11]. No obstante, el trabajo no presenta ninguna estrategia de MEC para implementar el objetivo del proyecto que involucra estas vistas. Por último, en [16] se propone un marco de calidad denominado 2Q2U (*Internal/External Quality, Quality in Use, Actual Usability, and User Experience*) que extiende a los modelos de calidad definidos en [10], considerando las relaciones de ‘influencia’ y ‘depende de’ para tres vistas de calidad. A su vez, dicho modelo ha sido instanciado utilizando la estrategia integrada SIQinU (*Strategy for Improving Quality in Use*) [12] que busca mejorar, de manera incremental y continua, la Calidad en Uso de una aplicación a partir de la mejora de la Calidad Externa, valiéndose de las relaciones entre las vistas asociadas. Esta estrategia es una instancia de uno de los patrones que propondremos.

### **3 Fundamentos del Enfoque Genérico de Evaluación**

En un proyecto de ME o MEC específico puede intervenir más de una categoría de entidad, p. ej., sistema y sistema en uso que serán evaluadas para los focos de Calidad

Externa y Calidad en Uso respectivamente, siendo el foco el concepto de más alto nivel de abstracción de un modelo de calidad. Como ya se comentó en la Introducción, uno de los dos pilares de nuestro enfoque genérico de evaluación es el marco de modelado de calidad que representa a las distintas vistas de calidad tales como vistas de calidad de recurso, proceso, producto, sistema, sistema en uso, entre otras. Además, este marco sirve como base para la definición de requerimientos no funcionales estructurados en forma de características, sub-características y atributos a ser evaluados.

A seguir, presentamos el marco de modelado de calidad como un componente agregado al marco conceptual C-INCAMI (*Contextual-Information Need, Concept Model, Attribute, Metric and Indicator*) ya existente [17]. Este nuevo componente se denomina *quality view* en la Fig. 1 y permite especificar vistas de calidad al instanciar el concepto *QualityView*, tal como se observa en la Fig. 2. Una vista de calidad es la asociación entre una súper categoría de entidad (*EntitySuperCategory*) y un concepto foco de calidad (*QualityFocus*) a evaluar. Además, se puede representar a las relaciones entre las diferentes vistas de calidad: p. ej., en la Fig. 2 se observa que la Vista de Calidad de Sistema influye (*influences*) a la Vista de Calidad de Sistema en Uso y que esta última depende de (*depends on*) o está determinada por Calidad de Sistema.

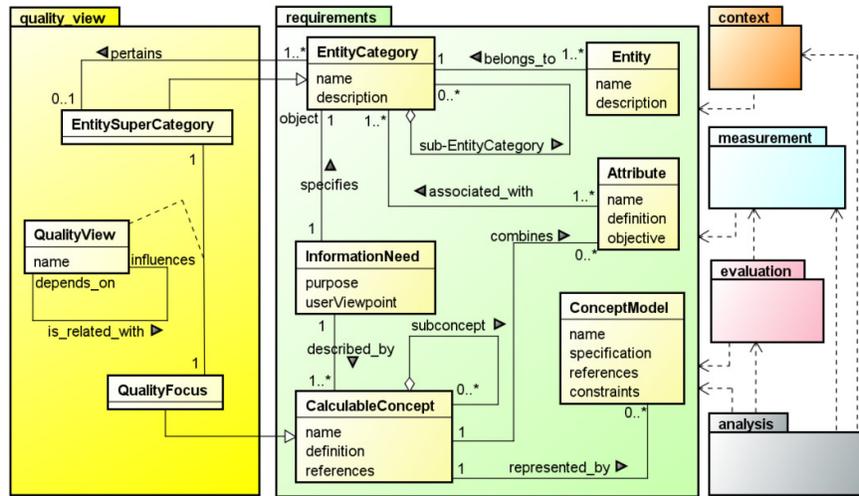


Fig. 1. Vista del marco conceptual C-INCAMI y su relación con el componente *quality view*

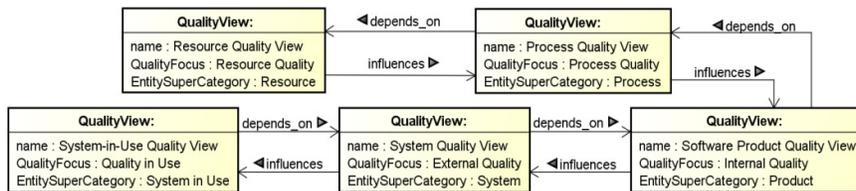


Fig. 2. Una instanciación de *Vistas de Calidad*, a partir del componente *quality view*

Por una parte, una entidad concreta (*Entity*), como p. ej., la aplicación Facebook v14 para Android, pertenece a una *EntityCategory* denominada aplicación móvil de red social, la que a su vez *pertenece* a la *EntitySuperCategory* denominada Sistema. Por otra parte, el foco de calidad (*QualityFocus*) es el concepto calculable (*CalculableConcept*) de más alto nivel como p. ej. Calidad Externa. Así, la asociación entre un foco de calidad y una súper categoría de entidad se llama Vista de Calidad (*QualityView*). Para el ejemplo, la Vista de Calidad de Sistema asocia al foco de Calidad Externa y a la súper categoría Sistema, tal como se especifica en la Fig. 2.

Como indicamos previamente, una vista de calidad ‘depende de’ e ‘influencia’ a otras vistas (ver *influences* y *depends on* en Fig. 1). En la Fig. 2, si consideremos la relación influencia y vistas que pueden intervenir en evaluación para una línea de producción de software, la Vista de Calidad de un Recurso –que relaciona la súper categoría de entidad Recurso con el foco Calidad de Recurso (en donde un recurso puede ser un agente, método, herramienta, etc.)- influencia a la Vista de Calidad de Proceso. A su vez, la Vista de Calidad de Proceso influencia a la Vista de Calidad de Producto software –siendo el código fuente un ejemplo de un ente para la súper categoría producto software-, y ésta a su vez, influencia a la Calidad de Sistema. Además, la Vista de Calidad de Sistema influencia a la Vista de Calidad de Sistema en Uso –siendo ejemplo de ente para Sistema en Uso, una aplicación móvil que es ejecutada por usuarios reales en contextos específicos de uso. En sentido inverso se interpreta a la relación ‘depende de’, es decir, la Vista de Calidad de Sistema en Uso depende de la Vista de Calidad de Sistema, y así sucesivamente.

Finalmente, las estrategias de ME y MEC son el segundo pilar del enfoque genérico de evaluación de calidad. El modelado de vistas de calidad y sus relaciones sirven de fundamento para este pilar dado que las estrategias se seleccionan en base a las vistas de calidad a evaluar y según el objetivo establecido del proyecto. En la Sección 1, se indicó que las estrategias integradas de ME/MEC deben satisfacer tres capacidades simultáneamente [20], a saber: un *marco conceptual* de dominio, una *especificación (de perspectivas) de proceso* y una *especificación de métodos*.

El *marco conceptual* de dominio debería estar construido sobre una base conceptual terminológicamente consistente y robusta (como p. ej. una ontología [19]), en donde se especifique de manera formal y explícita a los componentes, términos, propiedades, relaciones y restricciones para el dominio en cuestión. La segunda capacidad es la *especificación de proceso*, donde se describe principalmente qué hacer. Un proceso bien especificado utiliza perspectivas de proceso como funcional, de comportamiento, entre otras [6], lo que facilita el entendimiento y la comunicación entre los interesados, asegurando además la repetitividad y reproducibilidad en la implementación de las actividades. Por último, la *especificación de métodos* representa cómo se deben llevar a cabo las descripciones de las actividades. Los métodos se asignan a las tareas especificadas en el proceso y pueden ser automatizados por herramientas.

Como parte de nuestra investigación sobre estrategias integradas de ME/MEC hemos desarrollado tanto estrategias para una vista de calidad, como para dos vistas. Por ejemplo, GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*), es una estrategia aplicable a proyectos cuyo propósito puede ser comprender, predecir o controlar los más diversos focos de calidad para una entidad perteneciente

a cualquier súper categoría de entidad [16, 17]; y GOCAME<sup>+</sup>, con iguales características a GOCAME, pero que además contempla objetivos orientados al cambio y mejora continua. Ambas estrategias son aplicables a diferentes vistas de calidad, pero únicamente permiten evaluar una sola vista de calidad a la vez. Es más, GOCAME está diseñada para objetivos de ME, mientras que GOCAME<sup>+</sup> para objetivos de MEC. Otra estrategia es SIQinU [12], la cual relaciona la Vista de Calidad de Sistema en Uso con la Vista de Calidad de Sistema, utilizando las relaciones entre vistas antes comentadas. SIQinU es utilizada para mejorar la Calidad en Uso a través de cambios en atributos de Calidad Externa relacionados al Sistema o aplicación evaluada.

En síntesis, cada estrategia provee un curso de acción específico para resolver un problema u objetivo de ME o MEC, enfocado a una o más vistas de calidad. Para alcanzar el objetivo del proyecto se debe instanciar una estrategia. La propuesta de patrones de estrategia, que se discute en la próxima sección, materializa la idea de proveer soluciones reusables a problemas recurrentes en proyectos de MEC.

## 4 Patrones de Estrategia

Gamma *et al.* [7] mencionan que “*cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se pueda aplicar esta solución un millón de veces, sin hacer lo mismo dos veces*”. Por lo tanto, el fin de los patrones es proveer una solución a un problema recurrente. En este sentido, hemos observado que una estrategia de ME o MEC puede aplicarse a problemas recurrentes de proyectos de ME y MEC, y por ende, es posible desarrollar un conjunto de patrones de estrategia que brinden una solución flexible e instanciable.

Existen diferentes patrones de estrategia. Por ejemplo, para aplicar en proyectos en los cuales sólo se desea comprender el estado actual de una entidad para una sola vista de calidad; en proyectos donde el objetivo es mejorar una entidad introduciendo cambios y teniendo en cuenta relaciones de influencia entre dos vistas de calidad, entre otros casos. En la sub-sección 4.1 se especifica en detalle el patrón de estrategia denominado *Goal-Oriented Context-Aware Measurement, Evaluation and Change for one Quality View* (GOCAMEC\_1V), el cual es aplicable en proyectos de MEC que involucren una sola vista de calidad, refiriéndose a vista de calidad al concepto introducido en la sección previa. Luego, en la sub-sección 4.2 se discuten otros patrones, los cuales surgen a partir del análisis de la Fig. 2.

Cabe mencionar que la especificación del patrón de estrategia GOCAMEC\_1V se realiza mediante una plantilla que consta de los siguientes ítems: i) *nombre*: denominación descriptiva, normalmente expresada en inglés, que permite identificar unívocamente y referenciar al patrón; ii) *alias*: acrónimo o forma abreviada de nombrar al patrón; iii) *intención*: descripción del objetivo del patrón; iv) *motivación*: problema que soluciona el patrón; v) *aplicabilidad*: situaciones en las cuales se puede utilizar el patrón; vi) *estructura*: solución genérica e instanciable que propone el patrón; vii) *usos conocidos*: casos reales de aplicación o uso, y viii) *escenario de uso*: ejemplo concreto de instanciación del patrón. Notar que esta plantilla es una adaptación de aquella utilizada en [7] para especificar patrones de diseño.



Este patrón provee como solución un curso de acción genérico indicando qué actividades deben ser instanciadas durante la planificación del proyecto. A su vez, provee especificaciones de métodos que indican cómo deben llevarse a cabo las mismas. Los métodos podrán ser instanciados durante otras fases del ciclo de vida del proyecto como programación y ejecución.

A seguir se muestran los principales aspectos estructurales de las tres capacidades:

- i. La especificación del marco conceptual (ver Fig. 1 y 3) está definida como una ontología de dominio en la que se incluyen términos, atributos y relaciones –agrupados en componentes- para el área de ME, los que a su vez están enriquecidos con estereotipos de términos de una ontología genérica de proceso [4]. Los conceptos representan metadatos utilizados de modo uniforme y consistente en las actividades, productos de trabajo y métodos de la estrategia. (Notar que las Fig. 1 y 3 no muestran al componente de cambio).
- ii. El proceso o curso de acción genérico se especifica desde distintas perspectivas, a saber: funcional –que describe cuáles son las actividades a realizar junto con sus entradas y salidas-, de comportamiento –que especifica secuencias, paralelismos e iteraciones-, organizacional –que indica roles involucrados en las actividades-, y de información –que detalla la estructura y relaciones de los diferentes productos de trabajo. La Fig. 4 muestra, haciendo uso de la perspectiva funcional y de comportamiento, el proceso genérico provisto como solución. Los nombres de las ocho actividades (A1-A8) de la Fig. 4 deben personalizarse para la vista de calidad que se esté considerando en el proyecto. (Por cuestiones de espacio en este trabajo no se esquematiza el resto de las perspectivas del proceso, ni tampoco la especificación de las actividades a nivel de tareas).

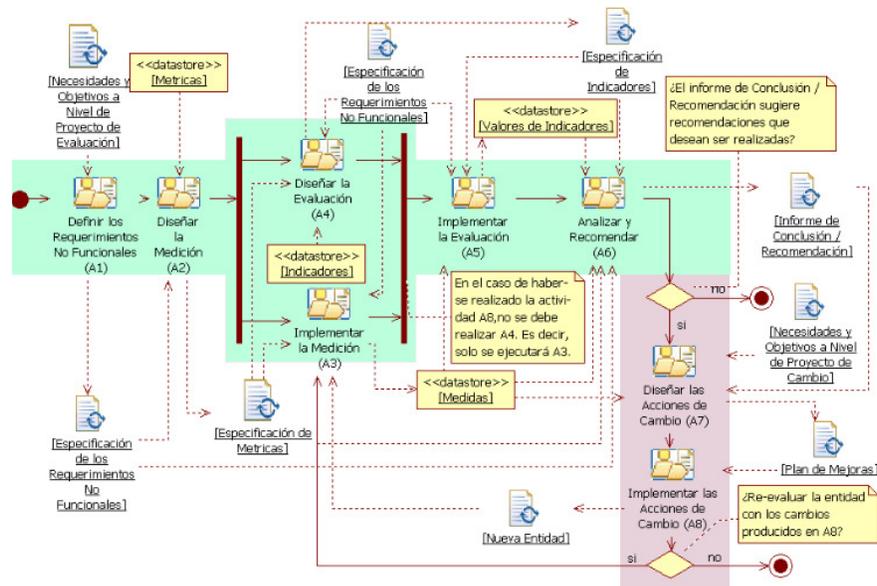


Fig. 4. Vista funcional y de comportamiento del Proceso genérico de GOCAMEC\_IV

- iii. La especificación de métodos indica cómo deben llevarse a cabo las descripciones de las actividades del proceso genérico. La Fig. 5 muestra, a modo de ejemplo, dos plantillas de especificación de métodos: a) *de una métrica indirecta*, como método para realizar una tarea de medición indirecta, y b) *de un indicador elemental*, como método utilizado durante la evaluación elemental. Los campos en las plantillas de métodos se corresponden con términos y atributos de la base conceptual del dominio de MEC (ver Fig. 3).

<p><b>a) Métrica Indirecta</b></p> <p><b>Nombre del Atributo que cuantifica:</b></p> <p><b>Nombre de la Métrica:</b></p> <p><b>Objetivo:</b>            <b>Autor:</b>            <b>Versión:</b></p> <p><b>Procedimiento de Cálculo</b></p> <ul style="list-style-type: none"> <li>• Especificación del Procedimiento:</li> <li>• Fórmula:</li> </ul> <p><b>Escala [Numérica Categorica]</b></p> <ul style="list-style-type: none"> <li>• Nombre del Tipo de Escala:</li> <li>• Tipo de Valor:</li> <li>• Representación:</li> </ul> <p><b>Unidad</b></p> <ul style="list-style-type: none"> <li>• Nombre:    Descripción:    Acrónimo:</li> </ul> <p><b>Herramienta:</b> (Si corresponde usar una)</p> <p><b>Métricas Relacionadas</b></p>	<p><b>b) Indicador Elemental</b></p> <p><b>Nombre del Atributo que interpreta:</b></p> <p><b>Nombre del Indicador:</b></p> <p><b>Autor:</b>            <b>Versión:</b></p> <p><b>Modelo Elemental</b></p> <ul style="list-style-type: none"> <li>• Especificación del Modelo Elemental:</li> <li>• Criterio de Decisión:</li> </ul> <p>Nombre:    Rango de valores:    Descripción:</p> <p><b>Escala [Numérica Categorica]</b></p> <ul style="list-style-type: none"> <li>• Nombre del Tipo de Escala:</li> <li>• Tipo de Valor:</li> <li>• Representación:</li> </ul> <p><b>Unidad</b></p> <ul style="list-style-type: none"> <li>• Nombre:    Descripción:    Acrónimo:</li> </ul>
---	--

**Fig. 5.** Plantillas de especificación de: a) una *Métrica Indirecta* y b) un *Indicador Elemental*

**Usos Conocidos:** Este patrón de estrategia fue utilizado en un proyecto de MEC cuyo objetivo era mejorar atributos de Calidad Externa de un carrito de compras a través de refactorización como método de cambio [18]. Además, este patrón se instanció en un proyecto de mejora para una categoría de ente Recurso [20].

**Escenario de Uso:** (Nota: el siguiente ejemplo surge del caso de estudio documentado en [15]). El objetivo del proyecto de MEC es evaluar la Usabilidad de la aplicación Facebook para dispositivos móviles bajo Android con el fin de detectar problemas de Usabilidad, analizar fortalezas y debilidades de la interface (GUI), recomendar y llevar a cabo acciones de cambio para subsanar las debilidades. Para este fin, el proyecto involucra la evaluación de la Vista de Calidad de Sistema. Específicamente, involucra la evaluación de la aplicación Facebook para dispositivos móviles con Android como ente concreto, la cual pertenece a la súper categoría Sistema. El foco de la Vista es Calidad Externa interviniendo sólo la característica (*CalculableConcept*) Usabilidad y sus sub-características.

Considerando este problema se debe seleccionar un patrón de estrategia aplicable a proyectos donde el propósito sea comprender y mejorar la entidad bajo análisis para una sola vista de calidad. En consecuencia, el patrón GOCAMEC\_1V es el adecuado para este proyecto, debiéndose instanciar para la Vista Calidad de Sistema.

En la etapa de planificación del proyecto, entre otros aspectos, se instancian las actividades que este patrón provee como solución. En este sentido, la actividad A1 de la Fig. 4 pasa a denominarse “Definir Requerimientos No Funcionales para Calidad Externa”, A2 se renombra como “Diseñar la Medición para Calidad Externa”, y así sucesivamente.

<b>Propósito:</b> Mejorar	<b>Punto de vista del usuario:</b> Usuario final
<b>Categoría de Entidad:</b> Aplicación de Red Social	<b>Súper Categoría de Entidad:</b> Sistema
<b>Entidad:</b> Aplicación móvil Facebook, versión 3.8 para Android	
<b>Foco de la Evaluación:</b> External Quality	
<b>Conceptos Calculables (Características):</b> Usability y sus sub-características: Understandability, Learnability, Operability, User Error Protection, User Interface Aesthetics	

**Fig. 6.** Artefacto *Especificación de la Necesidad de Información* producido en A1 instanciada

La actividad A1 de la Fig. 4, ahora instanciada, produce el documento Especificación de los Requerimientos No Funcionales para Calidad Externa, el cual se compone de la Especificación de la Necesidad de Información y del Árbol de Requerimientos. En la Fig. 6 se presenta la Especificación de la Necesidad de Información de MEC generada. Además, en la 1<sup>era</sup> columna de la Tabla 1 se muestra solo un pequeño fragmento, por razones de espacio, del Árbol de Requerimientos instanciado.

En la etapa de programación y ejecución del proyecto, se asignan métodos a las actividades personalizadas, entre otros recursos. Durante la actividad Diseñar la Medición para Calidad Externa (A2), se asignan métricas (métodos) para cuantificar los atributos del Árbol de Requerimientos. Las métricas se recuperan del catálogo de métricas (ver *datastore* en Fig. 4) cuyas especificaciones se basan en plantillas como la mostrada en la Fig. 5 a). Por ejemplo, el nombre de la métrica indirecta que cuantifica al atributo *Permanence of main controls* en la Tabla 1, es *Ratio of Main Controls Permanence* cuyo objetivo es determinar el porcentaje de botones o controles de la barra de controles principales que se mantienen consistentemente en el conjunto apropiado de pantallas (para más detalles sobre la especificación de esta métrica, ver [15]).

El curso de acción establecido en el (proceso genérico del) patrón indica A3 como próxima actividad a instanciar. Esto es, Implementar la Medición para Calidad Externa, la cual permite obtener los valores medidos de los atributos. Para la métrica indirecta mencionada anteriormente, el resultado de la medición obtenida fue 54,90%. Es decir, alrededor del 45% de los botones principales no mantienen su posición de manera consistente dentro de la barra de controles principales del conjunto de pantallas evaluadas para Facebook. El registro de las mediciones (fecha, valor medido, responsable, ID de la métrica, etc.) es almacenado en el repositorio Medidas (ver Fig. 4).

En la actividad A4, denominada ahora Diseñar la Evaluación para Calidad Externa, se seleccionan los indicadores elementales como métodos para interpretar la medida de cada atributo, así como indicadores derivados como métodos para calcular el nivel de satisfacción alcanzado por cada (sub)característica, como por ejemplo *Usability*, *Operability*, entre otras. Los indicadores elementales son diseñados a partir de la especificación dada en la plantilla de la Fig. 5 b). El lector puede consultar [15] para detalles de especificación de los indicadores de este proyecto.

A partir de las medidas obtenidas en A3 y usando los indicadores seleccionados en A4 se debe llevar a cabo la actividad A5, instanciada ahora con el nombre Implementar la Evaluación para Calidad Externa. Esta actividad produce los valores de indicadores elementales y derivados mostrados en la 2<sup>da</sup> y 3<sup>ra</sup> columna respectivamente de la Tabla 1. Notar que en dicha tabla se utilizan distintos colores para identificar el grado de satisfacción alcanzado por cada atributo/sub-característica, siendo por ejemplo el color rojo (valores entre 0-60) el rango de aceptabilidad "Insatisfactorio", lo cual indica que acciones de cambio se deben realizar de manera urgente.

**Tabla 1.** Fragmento del *Árbol de Requerimientos* para el *foco de Calidad Externa* y la característica *Usability*. (EI significa Indicador Elemental y DI significa Indicador Derivado)

	EI	DI
<b>1 Usability</b>		<b>60,50</b> ●
1.1 Understandability		76,10 ●
1.2 Learnability		59,70 ●
1.3 Operability		80,70 ●
⋮	⋮	⋮
<b>1.3.3 Consistency</b>		75,50 ●
1.3.3.1 Permanence of controls		57,30 ●
1.3.3.1.1 Permanence of main controls	54,90 ●	
⋮	⋮	⋮

Culminadas las actividades de medición y evaluación, se continúa con la actividad A6, personalizada para Analizar y Recomendar con respecto al foco de Calidad Externa. La Tabla 2 muestra parcialmente el documento Informe de Recomendación generado en A6. En este informe se especifican los atributos cuyos valores de indicadores no han alcanzado un nivel de aceptabilidad satisfactorio. Además, por cada uno de estos atributos se sugiere una o más recomendaciones que permitan luego llevar a cabo acciones de cambio para la mejora. El informe especifica también la prioridad del cambio, es decir, cuales recomendaciones deben ser consideradas con alta prioridad y cuales con prioridad moderada. La Tabla 2 ilustra la recomendación dada para el atributo *Permanence of main controls*.

**Tabla 2.** Extracto del artefacto *Informe de Recomendación* producido en la actividad A6. Nota: el acrónimo **A** significa prioridad Alta, es decir, se recomienda actuar de un modo urgente

ID	Recomendación (R)	Atributo	Prioridad
<b>R3</b>	1. Asegurar que en la barra de controles principales todos los botones (controles) principales permanezcan consistentemente en el conjunto de pantallas seleccionadas.	<i>Permanence of main controls</i> (1.3.3.1.1)	A

La siguiente actividad es la instanciación de A7, a saber: Diseñar las Acciones de Cambio para Calidad Externa. A7 produce el documento llamado Plan de Mejoras y la Tabla 3 muestra un fragmento del mismo. Este informe indica para cada recomendación documentada cuáles son las acciones de cambio planificadas, las fuentes de los registros de medición que dan soporte al cambio, y cómo pueden llevarse a cabo en la aplicación.

**Tabla 3.** Fragmento del *Plan de Mejoras* producido en la actividad A7

ID	Acción de Cambio (AC)	Fuente para la AC	Atributo	Método
<b>AC1</b>	1. Agregar los controles principales en la barra de controles principales de las pantallas con la debilidad detectada.	El ID de cada pantalla/botón principal con la debilidad detectada consta en el registro de Medidas/Mediciones para 1.3.3.1.1.	<i>Permanence of main controls</i> (1.3.3.1.1)	Refactoring de GUI

Según la Tabla 3, para el atributo *Permanence of main controls* se deben realizar tareas relacionadas a refactorización de GUI, lo cual requiere algún nivel de acceso al

código fuente de la aplicación. (Notar que por tratarse Facebook de un producto propietario, en [15] los cambios no se han podido realizar por no tener acceso al código fuente, diseños u objetos GUI de la aplicación).

Una vez que los cambios fueran ejecutados mediante la instanciación de A8, el curso de acción según GOCAMEC\_1V indica que se puede re-evaluar la nueva versión de la aplicación. Para este fin, A3, A5 y A6 se deben realizar nuevamente. De esta manera se logra comprender el nivel de mejora alcanzado. Por ejemplo, siguiendo la acción de cambio AC1 (Tabla 3), el valor del indicador elemental para el atributo 1.3.3.1.1 pasaría de 54,90 (rojo) a 100% (verde).

## 4.2 Discusión de otros Patrones de Estrategia

Teniendo presente las posibles instanciaciones de Vistas de Calidad de la Fig. 2, y las relaciones existentes entre las mismas, es posible diseñar diferentes patrones de estrategia para proyectos de ME y MEC. Por razones de espacio, sólo se describirán a seguir algunos aspectos como su intención y usos conocidos.

El caso más básico y simple de instanciar es el patrón para evaluar una sola Vista de Calidad (p. ej., Calidad de Recurso, Calidad de Producto software, Calidad de Sistema, etc.) con el propósito de *comprender* o *estimar* una entidad objetivo y su contexto. Para esta situación diseñamos el patrón cuyo nombre conocido previamente era GOCAME (aunque no estaba especificado en la forma de patrón de estrategia) y ahora renombrado como GOCAME\_1V. Así, la intención de este patrón es proveer una solución en la instanciación de una estrategia de medición, evaluación y análisis a ser empleada en un proyecto de ME cuyo objetivo involucre la evaluación de una sola Vista de Calidad para un ente y contexto dado. Este patrón ha sido hasta el presente el más utilizado. Por ejemplo, en la evaluación de una aplicación de mashups [16], un carrito de compras [17], entre otros casos. Su proceso genérico es el sombreado en verde en la Fig. 4, que corresponde a las actividades A1-A6. El patrón similar al mencionado es GOCAMEC\_1V, analizado en la sub-sección 4.1, pero su propósito es además el de *mejorar* la entidad evaluada. Una clara diferencia entre GOCAME\_1V y GOCAMEC\_1V es que este último cuenta con ocho actividades y ciclos de cambio y re-evaluación (actividades A1-A8 en Fig. 4).

Situaciones más complejas son aquellos casos donde intervienen dos o más vistas. Por ejemplo, consideremos un proyecto en el cual se desea mejorar algún aspecto de Calidad Externa de un Sistema. Recordando que la Calidad Externa depende de la Interna (notar en Fig. 2 la relación *depends on* entre la Vista de Calidad de Sistema y la de Calidad de Producto software) es posible introducir cambios en un producto como p. ej. en un diagrama de clases, y que estos mejoren la mantenibilidad del sistema como p. ej. una aplicación Java. Este patrón es denominado GOCAMEC\_2V, y es aplicable a proyectos MEC donde se busque mejorar una vista de calidad introduciendo cambios en otra vista de calidad. Es importante destacar que una restricción en este patrón es que la vista a mejorar debe depender de la vista sobre la cual se realizan los cambios. Un uso conocido de GOCAMEC\_2V está en [12], donde SIQinU fue utilizada para mejorar la Calidad en Uso de una aplicación dedicada al seguimiento de errores e incidentes. Siguiendo el proceso instanciado del patrón GOCAMEC\_2V

primero se detectaron atributos de Calidad en Uso con bajo nivel de satisfacción. A partir de estos problemas se derivaron atributos de Calidad Externa y se introdujeron cambios sobre la aplicación para que afectaran positivamente en la Calidad en Uso.

De manera similar pueden desarrollarse otros patrones que contemplen tres o más vistas considerando las relaciones *influences* y *depends on* en la Fig. 2.

## 5 Conclusión y Trabajos Futuros

Se ha presentado un *enfoque genérico de evaluación de calidad* previamente ideado aunque desarrollado parcialmente. Como contribuciones específicas de este trabajo, listadas en la Introducción, se ha robustecido sus dos pilares, esto es: 1) el marco de modelado de calidad, y 2) las estrategias integradas de ME/MEC. Respecto a 1), se agregó el componente *quality\_view* al marco conceptual C-INCAMI previo. En dicho componente se ha modelado el concepto Vista de Calidad (*QualityView*) como una asociación entre un foco de calidad y una súper categoría de entidad. El componente también especifica las relaciones entre vistas, quedando explícitamente modeladas las relaciones 'influencia' y 'depende de'. Contar con un marco de modelado de calidad que contemple vistas y relaciones entre ellas permite considerar múltiples escenarios, contribuyendo a la selección de estrategias en proyectos concretos de ME/MEC.

Respecto al segundo pilar, se propuso la especificación de patrones de estrategia que brindan una solución en la instanciación de las estrategias en proyectos de ME/MEC. El ítem estructura (solución) de la plantilla se expresa en términos de un curso de acción genérico, un marco conceptual de dominio y especificación de métodos. Los patrones de estrategia pueden ser reusados en proyectos cuyos objetivos involucren la evaluación de una o más vistas de calidad, posibles cambios y mejora continua. Para ilustrar a los patrones, en la Sección 4 se ha especificado al patrón GOCAMEC\_IV que considera una vista y un objetivo de MEC, y se analizaron otros.

A partir de la revisión bibliográfica se detectó que no hay hasta el presente una propuesta de enfoque genérico de evaluación de calidad con las características antes mencionadas, sino más bien que existen esfuerzos aislados o parciales que no aúnan todas estas cuestiones en una arquitectura integral y flexible.

Como trabajo futuro se construirá una ontología de dominio de cambio para ampliar la base conceptual propuesta como solución para aquellos patrones que involucran cambios, y se integrará al marco conceptual C-INCAMI de ME. Por otro lado, se avanzará en el estudio de relacionar vistas de calidad con distintos niveles organizacionales (como operativo, táctico y estratégico). Así, un patrón de estrategia se podrá seleccionar no sólo para instanciar una estrategia que lleve a cabo un objetivo de proyecto ME/MEC considerando vistas de calidad a un mismo nivel, sino también para objetivos alineados a distintos niveles organizacionales, como en GQM<sup>+</sup>Strategies.

## Referencias

1. Basili V., Lindvall M., Regardie M., Seaman C., Heidrich J., Jurgen M., Rombach D., Trendowicz A.: Linking Software Development and Business Strategy through

- Measurement. *IEEE Computer*, 43:(4), pp. 57–65 (2010)
2. Basili V.: Software Development: A Paradigm for the Future. In Proc. of 13<sup>th</sup> Annual Int'l Computer Software & Applications Conference (COMPSAC), IEEE, pp.471-485 (1989)
  3. Basili V.: Quantitative Evaluation of Software Engineering Methodology. Technical Report, TR-1519, Dept. of Computer Science, University of Maryland (1985)
  4. Becker P., Papa M.F., Olsina L.: Process Conceptual Base for Enriching a Measurement and Evaluation Ontology. En CD de la XVII Conferencia Iberoamericana en Software Engineering (CIbSE'14). ISBN: 978-956-236-247-4, pp. 53-66, Pucón, Chile (2014)
  5. CMMI: Capability Maturity Model Integration, CMMI for Dev. Version.1.3. CMU/SEI-2010-TR-033, USA (2010)
  6. Curtis B., Kellner M., Over J.: Process Modelling. *Com. of ACM*, 35:(9), pp.75-90 (1992)
  7. Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, ISBN 0-201-63361-2 (1995)
  8. García F., Piattini M., Ruiz F., Canfora G., Visaggio A.: FMESP: Framework for the Modeling and Evaluation of Software Processes. *Journal of Systems Architecture*, 52:(11), pp. 627-639 (2006)
  9. Houdek F., Kempter H.: Quality Patterns: An Approach to Packaging Software Engineering Experience. ACM SIGSOFT, Software Engineering Notes. pp. 81-88 (1997)
  10. ISO/IEC 25010: Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models (2011)
  11. ISO/IEC 9126-1: Software Engineering Product Quality - Part 1: Quality Model (2001)
  12. Lew P., Olsina L., Becker P., Zhang L.: An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications. In: *Requirements Engineering Journal*, Springer London, 17:(4), pp. 299-330 (2012)
  13. Lindvall M., Donzelli P., Asgari S., Basili V.: Towards Reusable Measurement Patterns. *Software Metrics*, 11th IEEE Int'l Symposium, 8 pgs. (2005)
  14. Moraga M.A, Bertoa M.F., Morcillo M.C., Calero C., Vallecillo A.: Evaluating Quality-in-Use Using Bayesian Networks. In Proc. of QAOOSE 2008, Paphos, Cyprus (2008)
  15. Olsina L., Santos L., Lew P.: Evaluating Mobileapp Usability: A Holistic Quality Approach, In: 14th Int'l Conference on Web Engineering, ICWE 2014, Casteleyn S., Rossi G., and Winckler M. (Eds.): Springer, LNCS 8541, pp. 111-129 (2014)
  16. Olsina L., Lew P., Dieser A., Rivera B.: Updating Quality Models for Evaluating New Generation Web Applications. *Journal of Web Engineering*, Special issue: Quality in new generation Web applications. Rinton Press, US, 11:(3), pp. 209-246, (2012)
  17. Olsina L., Papa F., Molina H.: How to Measure and Evaluate Web Applications in a Consistent Way. *Web Engineering: Modelling and Implementing Web Applications*, Springer HCIS, Rossi G., Pastor O., Schwabe D., Olsina L. (Eds), Ch.13, pp. 385-420 (2008)
  18. Olsina L., Rossi G., Garrido A., Distanto D., Canfora G.: Web Applications Refactoring and Evaluation: A Quality-Oriented Improvement Approach, *Journal of Web Engineering*, Rinton Press, US, 7:(4), pp. 258-280 (2008)
  19. Olsina L., Martín M.: Ontology for Software Metrics and Indicators. *Journal of Web Engineering*, Rinton Press, USA, 2:(4), pp. 262-281 (2004)
  20. Papa M.F.: Toward the Improvement of a Measurement and Evaluation Strategy from a Comparative Study, In LNCS 7703, Springer: Current Trends in Web Engineering, ICWE Int'l Workshops, Grossniklauss M. and Wimmer M. (Eds.), pp. 189-203 (2012)
  21. PMBOK: A Guide to the Project Management Body of Knowledge, 5<sup>th</sup> Edition (2013)