

# El Modelo de Evaluación del proyecto MyFEPS

Amos Sorgen, Paula Angeleri  
Facultad de Tecnología Informática, Universidad de Belgrano, Fco. Lacroze 1947,  
1426 Ciudad Autónoma de Buenos Aires, Argentina  
{amos.sorgen, paula.angeleri}@ub.edu.ar

**Resumen:** La calidad del software se evalúa a partir del análisis de una serie de mediciones de métricas de calidad. Los objetivos de negocio, los entornos de trabajo, las características del sistema, y las normas a seguir son factores que determinan qué, cómo, y cuánto se medirá y cómo se analizarán e interpretarán los resultados de las mediciones.

MyFEPS es un proyecto que tiene como objetivo crear un framework integrativo que, tomando en cuenta todos los factores que influyen en el proceso de evaluación de la calidad de un producto, genera para cada producto específico una metodología de evaluación y calcula su presupuesto. El framework también proporciona herramientas de medición y métodos de análisis. En su núcleo, MyFEPS se basa en un método de evaluación centrado en los agentes interesados en la calidad del producto.

**Palabras Claves:** MyFEPS, Calidad de Producto de Software, Procesos de Evaluación de Calidad, Stakeholder de Calidad.

## 1 Introducción

En los últimos años se han intensificado las actividades de evaluación de la calidad de los productos y artefactos de software [1], necesarias para:

- monitorear el proceso de desarrollo,
- cumplir con requisitos de calidad
- certificar bajo normas internacionales
- evaluar posibles adquisiciones de nuevos productos
- como herramienta de benchmarking para decidir una elección entre varios productos en competencia.

Paralelamente, se complican, encarecen y prolongan cada vez más los procesos de evaluación de la calidad [2] a causa de los siguientes factores:

- Los productos de software son cada vez más sofisticados y más complejos
- Los productos de software se utilizan cada vez más en sistemas críticos
- Las aplicaciones son cada vez más diversas, requiriendo evaluaciones diferentes.
- Surgen nuevas normas internacionales de calidad
- La diversidad de los objetivos de negocio para los que se desarrollan los sistemas.
- La diversidad de los objetivos de la evaluación.

Los ingenieros de software y gerentes de desarrollo que planifican el proceso de evaluación de la calidad deben tomar en cuenta todos los factores que pueden influir en el proceso de evaluación, y dar respuesta a las siguientes preguntas fundamentales:

- A. ¿Qué medir?
- B. ¿Con qué objetivo medir?
- C. ¿Cuánto medir?
- D. ¿En qué contexto medir?
- E. ¿Cómo interpretar los resultados de las mediciones?

Las respuestas a estas preguntas tienen un gran impacto en el costo [3], la duración y los resultados del proceso de evaluación. Y actualmente, son los evaluadores profesionales quienes dan respuesta a estas preguntas, con una cierta arbitrariedad.

El propósito de este trabajo es presentar la solución que usaremos en el marco del proyecto MyFEPS (Metodologías y Framework para la Evaluación de Productos de Software), en desarrollo en la Facultad de Tecnología Informática de la Universidad de Belgrano, cuyo propósito es diseñar e implementar un framework para ayudar a ingenieros y gerentes en todo el proceso de evaluación que incluye desde la planificación hasta la presentación de los resultados. Este framework deberá tomar en cuenta todos los factores que inciden en un plan de evaluación de un producto de software. Su arquitectura y objetivos se presentan en la sección 4 de este artículo<sup>1</sup>.

Lo que los evaluadores profesionales conciben como calidad es sólo “su” perspectiva y no forzosamente la valedera. Otras personas y agentes tienen interés en la calidad del producto y cada uno tiene su propia perspectiva de lo que significa que éste sea de calidad [4], [5]. El problema es cómo dar respuesta a las preguntas fundamentales para que satisfagan a todas estas personas y agentes.

Introducimos para ello el concepto de *stakeholder de calidad*. Análogo al concepto de stakeholder, se define como todo agente que tiene algún interés en la calidad del artefacto o producto que se evalúa. Suponemos que todo agente que tenga algún interés en un artefacto/producto de software tiene también interés en su calidad y viceversa. Por lo que podemos afirmar que en principio todo stakeholder es un stakeholder de calidad y viceversa.

Las diferencias fundamentales con otros trabajos en el área de *Modelos de Evaluación de Calidad* [6], [7], [8], [12], [14], [15], [16], [17], [18] así como las innovaciones que introduce este trabajo, se basan en que presentando un modelo general (no se restringe a un tipo de productos), introduce de forma sistemática y cuantitativa los puntos de vista de todos los *stakeholders de calidad*, proporciona una metodología de agregación para obtener los valores de las características de calidad (basadas en las mediciones de las métricas), da la base para establecer criterios cuantitativos que ayuden a determinar el esfuerzo a invertir en las mediciones, brindando a su vez un modelo para evaluar dichos esfuerzos.

La estructura del artículo que inicia con esta introducción, continúa en la sección 2 definiendo los conceptos que utilizaremos en nuestro modelo de evaluación; la sección 3 describe el modelo de evaluación propuesto; la sección 4 explica cómo se empleará el modelo de evaluación a través de un framework que se obtiene a partir del proyecto MyFEPS y expone sus objetivos y la arquitectura del framework; la

---

<sup>1</sup> Fueron propuestos frameworks con objetivos similares a los de MyFEPS [12], [13].

sección 5 presenta las futuras líneas de investigación; y finalmente sección 6 presenta las conclusiones. Al final se encuentran las referencias.

## 2 Definiciones

Para simplificar, usaremos “producto” para referirnos al producto final y a cualquier artefacto (producto intermedio) en el proceso de desarrollo de software, y “característica” para referirnos a características y subcaracterísticas de calidad.

“*Ítem de calidad*”: cualquier ítem al cual se le pueda asignar un *grado de calidad* (evaluado o medido). Ejemplos: producto, artefacto, propiedad, factor, característica, subcaracterística, métrica.

“*Nivel de evaluación*”: Atributo que se le asigna a los ítems de calidad. Las métricas pertenecen al nivel de evaluación más bajo (cero). En realidad, una métrica no se *evalúa*, sino que se *mide*. Por encima de las métricas están los demás ítems de calidad que sí son *evaluados*. Se evalúa el grado de calidad de todo ítem a nivel N usando los grados de calidad de un subconjunto de ítems a nivel N-1.

“*Rigurosidad*”: Atributo que se le asigna a la evaluación de calidad de un ítem de calidad. Puede tomar un valor entre 0 y 1. Rigurosidad igual a 1 indica que la evaluación para el ítem de calidad en cuestión se debe llevar a cabo con la máxima rigurosidad. Rigurosidad igual a 0 indica que no es necesaria la evaluación. Es un atributo subjetivo y se usa para inferir las “fidelidades” de medición (ver la definición siguiente).

“*Fidelidad*”: Atributo que se le asigna a la medición de una métrica. Puede tomar un valor entre 0 y 1. Fidelidad igual a 1 indica que la medición para la métrica en cuestión se debe realizar con la máxima fidelidad. Fidelidad igual a 0 indica que la métrica en cuestión no se debe medir, y por lo tanto no se debe tomar en cuenta. Ejemplo: Fidelidad F en la medición de la métrica “Coupling” podría ser el porcentaje de clases que se toman en cuenta en la medición, dividido por 100.

“*Factores de Evaluación*”: Todos los factores que haya que tomar en cuenta para poder planificar una evaluación eficaz y eficiente. MyFEPS toma en cuenta los siguientes:

- a. Objetivos de la evaluación
- b. Norma / Modelo de Calidad a usar
- c. Tipo de producto
- d. Tamaño del producto
- e. Complejidad del producto
- f. Objetivos de negocio del producto
- g. Características de los stakeholders

“*Metodología de Evaluación*”: Dado un producto de software a ser evaluado, y dados los demás factores de evaluación, la “*Metodología de Evaluación*” es el conjunto de documentos y métodos que incluye:

- El Modelo de Calidad utilizado
- Las características y subcaracterísticas del modelo designadas para evaluar el producto
- Las rigurosidades con las que se requiere evaluar cada una de las características y subcaracterísticas
- Las métricas elegidas para evaluar las características y subcaracterísticas
- La fidelidad con la que se debe medir cada una de las métricas
- El mapeo de los valores de las mediciones de las métricas a los grados de calidad de las características y subcaracterísticas.

“*Modelo de Evaluación*”: El modelo teórico usado para generar las metodologías de evaluación [7].

“*Configuración de la medición*”: Conjunto de todas las métricas a medir, y la fidelidad con que se debe medir cada una (ver también definición de “fidelidad”).

$$\{M_1, M_2, \dots\}, \{F(M_1), F(M_2), \dots\}, 0 \leq F(M_i) \leq 1 \quad (2.1)$$

$M_i$  es la métrica número “i”.  $F(M_i)$  es la fidelidad con que se mide  $M_i$

### 3 El Modelo de Evaluación

En esta sección presentamos un modelo teórico que sienta las bases para generar metodologías de evaluación de productos de software.

#### 3.1 Unificación de los valores de los grados de calidad

Para todos los ítems de calidad, definimos una escala continua del 0 al 1 para determinar su grado de calidad, siendo 1 el valor máximo y 0 el valor mínimo.

Necesitamos establecer un grado de calidad para las métricas (no sólo para las características). Para lo cual a cada métrica le corresponderá una función que traduzca los valores de la medición de la métrica a su grado de calidad en el rango de 0 a 1. Por ejemplo “tiempo (T)” es el tipo de medida de la métrica “Tiempo de implementación del cambio”, que en la norma ISO/IEC 9126 [6] se usa para evaluar la subcaracterística “*Capacidad para ser modificado*”. Los tiempos se deberán traducir a valores de grados de calidad. Esto se podría hacer por ejemplo con una expresión del tipo  $T_{\min}/T$ , siendo  $T_{\min}$  un tiempo mínimo. De esta forma obtendríamos valores entre 0 y 1, donde 1 representa la mejor “calidad” y 0 la peor.

#### 3.2 Cálculo del Grado de Calidad de un Ítem de Calidad

Sea  $IC_{(N)}$  un ítem de calidad a nivel  $N$  ( $N > 0$ ). Y sea

$$\{IC_{(N-1),i}\} \quad (3.1)$$

el conjunto de los ítems de calidad a nivel  $N-1$  usados para evaluar a  $IC_{(N)}$ .  
Sea

$$\{G(IC_{(N-1),i})\} \quad (3.2)$$

el conjunto de los grados de calidad de los ítems del conjunto  $\{IC_{(N-1),i}\}$

El grado de calidad de  $IC_{(N)}$  es:

$$G(IC_{(N)}) = \sum_i (G(IC_{(N-1),i}) * IR(IC_{(N-1),i})) / \sum_i (IR(IC_{(N-1),i})) \quad (3.3)$$

$(IR(IC_{(N-1),i}))$  es la “Importancia Relativa” del ítem de calidad “i” y en la sección 3.5 se explica cómo se la calcula.

Cuando  $N = 1$  (nivel 1), los valores  $\{G(IC_{(0),i})\}$  son los grados de calidad de las métricas que determinan el grado de calidad de  $IC_{(1)}$ .

Por ejemplo, de acuerdo con la norma ISO/IEC 9126 [6], el grado de calidad de la subcaracterística “*Capacidad para ser modificado*” se determina por los grados de calidad de cinco métricas que a continuación especificamos con ejemplos de valores obtenidos en su medición: 1) Eficiencia en el ciclo de cambio ( $G_1=0.7$ ), 2) Tiempo de implementación del cambio ( $G_2=0.9$ ), 3) Complejidad de modificación ( $G_3=0.3$ ), 4) *Modificabilidad parametrizada* ( $G_4=0.5$ ), y 5) Capacidad de controlar el cambio del software ( $G_5=0.7$ ).

El framework la calculará usando la expresión (3.3), después de haber asignado las Importancias Relativas a cada una de las métricas. Suponiendo que del análisis de los cuestionarios (ver sección 3.5) surge que las importancias relativas de los ítems de calidad son respectivamente  $IR_1=0.9$ ,  $IR_2=0.8$ ,  $IR_3=1.0$ ,  $IR_4=0.9$ ,  $IR_5=0.7$ , el grado de calidad de la “*Capacidad para ser modificado*” sería:

$$G = (0.7 * 0.9) + (0.9 * 0.8) + (0.3 * 1.0) + (0.5 * 0.9) + (0.7 * 0.7) / (0.9 + 0.8 + 1.0 + 0.9 + 0.7) = 0.6$$

En palabras:  $G$  (*Capacidad para ser modificado*) es igual al cociente entre la sumatoria de los productos de los valores obtenidos en cada métrica multiplicado por su importancia relativa, y la sumatoria de las importancias relativas de las métricas respecto de la *Capacidad para ser modificado*

### 3.3 Importancias Relativas de los ítems de calidad asignadas por los stakeholders y el peso relativo de cada stakeholder

Una herramienta central en el modelo de evaluación son los “*Cuestionarios a los Stakeholders*” [8]. Los cuestionarios deben diseñarse de tal forma que de las respuestas que los stakeholders den a sus preguntas, se infiera la importancia relativa de cada ítem de calidad en el proceso de evaluación.

De los cuestionarios a los stakeholders deduciremos, para cada nivel de evaluación, los valores de cada Importancia Relativa que el stakeholder “j” asigna en cada nivel al ítem de calidad “IC”.

$$0 \leq I_j(\text{IC}) \leq 1 \quad (3.4)$$

Se establecen también qué peso relativo tiene cada stakeholder para evaluar la Importancia Relativa del ítem de calidad “IC”:

$$0 \leq P_j(\text{IC}) \leq 1 \quad (3.5)$$

La Importancia Relativa asignada por el Stakeholder “j” al ítem de calidad “IC” que se usará será:

$$\text{IRS}_j(\text{IC}) = I_j(\text{IC}) * P_j(\text{IC}) \quad (3.6)$$

Notas:

- En vez de un simple producto como en (3.6) se podría considerar una función más sofisticada.
- Una “estrategia democrática” puede establecer el valor de  $P_j(\text{IC}) = 1$  para todos los stakeholders que participan en la determinación de la Importancia Relativa del ítem de calidad “IC”.

Siguiendo con el ejemplo de la sección anterior, digamos que tenemos las respuestas a los cuestionarios de dos stakeholders, uno representa a la empresa que encargó el producto a evaluar (stakeholder “1”) y el otro a los desarrolladores del producto (stakeholder “2”). Consideramos que para ambos la subcaracterística “*Capacidad para ser modificado*” es relevante, siendo algo más relevante para la empresa que encargó el producto que para grupo de desarrolladores, es por eso que asignaremos  $I_1 = 1.0$  y  $I_2 = 0.9$ . De los cuestionarios inferimos los pesos que tiene para estos dos stakeholders la característica “*Capacidad para ser modificado*” respecto a otras subcaracterísticas:  $P_1 = 0.7$  y  $P_2 = 0.8$ . Resultando en Importancias relativa asignadas por los stakeholder 1 y 2 a la subcaracterística “*Capacidad para ser modificado*”:

$$\text{IRS}_1 = 1.0 * 0.7 = 0.7, \quad \text{IRS}_2 = 0.9 * 0.8 = 0.72$$

### 3.4 Factores no-humanos

Hasta ahora tomamos en cuenta solo los factores humanos (stakeholders). También se deben tomar en cuenta factores no humanos que tienen efecto sobre las rigurosidades y fidelidades, como por ejemplo reglas de negocio, reglamentaciones, etc.

Para simplificar el modelo de evaluación consideraremos todo factor no-humano como “representado” por un “stakeholder representativo”. Y para ello se esbozarán respuestas a los “Cuestionarios a los Stakeholders” que reflejen las características e imposiciones de estos factores. Por ejemplo si el producto debe respetar una regulación especial, habrá un “stakeholder representativo” que lo represente y éste dará las respuestas a las preguntas del cuestionario como si fuera el agente que redactó la regulación. De estos cuestionarios obtendremos los valores de las Importancias Relativas de los ítems de calidad de acuerdo a dicho “stakeholder representativo”. A los “stakeholders representativos” se les adjudican pesos de stakeholder como los de la expresión (3.5).

### 3.5 Importancias Relativas y Fidelidades

Tomando en cuenta a todos los stakeholders (reales y representativos), la Importancia Relativa de todo ítem de calidad, será:

$$IR(IC) = \sum_j (IRS_j(IC)) / NS(IC) \quad (3.7)$$

en donde el índice de la sumatoria refiere a los stakeholders y NS(IC) es el numero de sumandos de la suma.

El modelo plantea las siguientes suposiciones y restricciones:

- A. IR(IC) será el peso relativo de IC en la evaluación de ítems a un nivel más alto que el suyo (ver sección 3.2 y expresión (3.3))
- B. La rigurosidad con la que se evaluará un ítem de calidad será igual a la Importancia Relativa del mismo.
- C. La fidelidad con la que se medirá una métrica será igual a la Importancia Relativa de la misma.
- D. Ya que sería un derroche de recursos evaluar un ítem con una rigurosidad mayor a la Importancia Relativa del ítem que lo usará en su evaluación, los valores de las Importancias Relativas de los ítems de calidad que participan en la expresión , no deberán superar el valor de  $IR(IC_{(N)})$ ,
- E. El esfuerzo medido en horas-hombre necesario para realizar las mediciones que determinen el valor de una métrica para un producto, es proporcional a la fidelidad con la que se debe medir la métrica y al tamaño del producto.

Para hacer posible esta restricción, tendremos que especificar la forma de medir una métrica con una fidelidad ( $F < 1$ ) de tal manera que el esfuerzo necesario sea proporcional a F.

Todas estas suposiciones y restricciones se deberán probar empíricamente para demostrar que resultan en un buen modelo de evaluación. Si así no fuera, se podrían reemplazar por otras más sofisticadas.

Para hacer posible esta restricción, tendremos que especificar la forma de medir una métrica con una fidelidad ( $F < 1$ ) de tal manera que el esfuerzo necesario sea proporcional a F.

Todas estas suposiciones y restricciones deberán ser probadas empíricamente para demostrar que conllevan a un buen modelo de evaluación. Y podrían en principio ser remplazadas por otras más sofisticadas.

En nuestro ejemplo la Importancia Relativa de la subcaracterística “*Capacidad para ser modificado*” será:  $IR = (0.7 + 0.72) / 2 = 0.71$ . Y este es el valor que se usará para calcular el Grado de Calidad de la característica “*Facilidad de Mantenimiento*” de la cual “*Capacidad para ser modificado*” es una subcaracterística (ver sección 3.2 y expresión (3.3))

### 3.6 Grado de calidad aceptable

En el caso de que la evaluación se haga para certificar/aceptar un producto hay que tomar en cuenta los grados de calidad “acceptables” para las características evaluadas [9]:

$$\{GA(IC_i)\} \quad (3.8)$$

Estos valores influyen en la determinación de las fidelidades necesarias para medir las métricas. Si por ejemplo para una característica es aceptable un grado de calidad medio, no es necesario evaluarla con exagerada rigurosidad y por lo tanto no hace falta medir con máxima fidelidad las métricas que la determinan.

Podemos establecer que si el grado de calidad aceptable para el ítem de calidad  $IC_N$  es  $GA(IC_N) < 1$ , ninguno de los ítems de calidad a un nivel mas bajo ( $IC_{N-1}$ ) que se usan en la evaluación de  $IC_N$ , deberá tener una importancia relativa mayor que  $GA(IC_N)$ . Es decir que  $GA(IC_N)$  es el límite superior para los valores de Importancia Relativa de los ítems de calidad ( $IR(IC_{N-1,i})$ ) que determinan el grado de calidad de  $IC_N$ .

$$IR(IC_{(N-1),i}) \leq GA(IC_N) \quad (3.9)$$

Este límite superior se “propaga” hasta las fidelidades de medición de las métricas.

### 3.7 Esfuerzos y Presupuestos

Si  $E_{\max}(M)$  es el esfuerzo por unidad de tamaño (por ejemplo el precio en dólares por Punto de Función) necesario para medir con máxima fidelidad ( $F = 1$ ) la métrica  $M$  [10], el esfuerzo necesario para medir la métrica  $M$  para un producto que tenga un tamaño  $V$  es:

$$E(M) = E_{\max}(M) * IR(M) * V \quad (3.11)$$

$IR(M)$  viene dada por la expresión (3.7). El esfuerzo total de la evaluación será:



$$ET = \sum_M E(M) \quad (3.12)$$

el índice de la sumatoria son las distintas métricas

Si el esfuerzo se mide en la cantidad de dinero que costaría la medición, y si a ET le agregamos los gastos fijos (independientes de las mediciones), obtenemos una estimación del presupuesto necesario.

#### 4 Arquitectura y Objetivos de MyFEPS

El objetivo del proyecto MyFEPS es construir un framework que asista en todo el proceso de evaluación de productos finales e intermedios de software.

Siguiendo la norma ISO/IEC 14598 [11], reconocemos cuatro fases en el proceso de evaluación: 1) Establecer los requisitos de la evaluación, 2) Especificar la evaluación, 3) Diseñar la evaluación, 4) Realizar la evaluación

La arquitectura del framework (ver figura 1) fue centrada en bases de datos (ver sección 4.1) que constituyen todos los repositorios y las bases de conocimiento necesarios para cumplir el objetivo del proyecto. Los usuarios del framework tienen acceso a ellas a través de módulos especializados (ver sección 4.2).

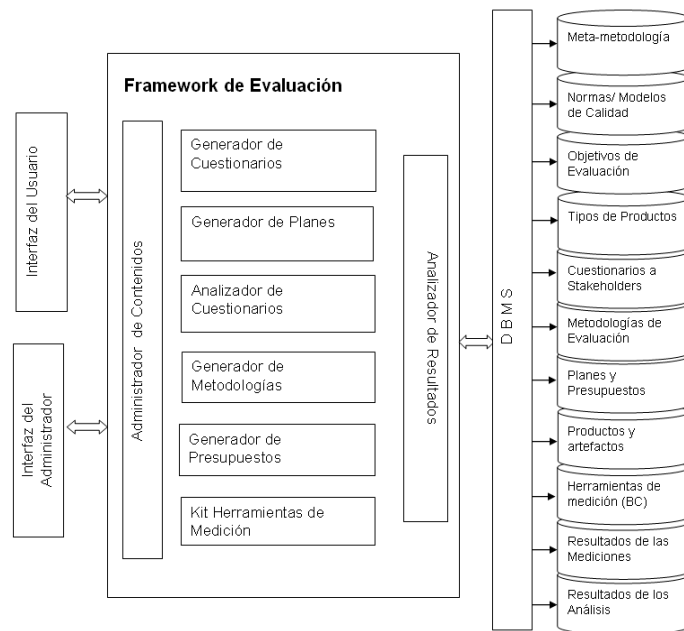


Fig. 2. La arquitectura de MyFEPS

#### **4.1 Repositorios y Bases de Conocimiento**

Meta-metodología (base de conocimientos): Una vez que se conocen los factores de la evaluación, se usa para generar la metodología de evaluación.

Normas / Modelos de Calidad (base de conocimientos): Se usa en la confección de la metodología de evaluación para incluir lo específico de las normas de calidad o modelo de calidad en uso.

Objetivos de Evaluación (base de conocimientos): Se usa en la confección de la metodología de evaluación para incluir los efectos de los objetivos de la evaluación.

Tipos de Productos / Objetivos de Negocio (base de conocimientos). Se usa en la confección de la metodología de evaluación para incluir lo específico del tipo de producto y de los objetivos de negocio.

Productos y Artefactos (repositorio): Productos y artefactos evaluados y a ser evaluados.

Cuestionarios a Stakeholders (repositorio y base de conocimientos): Cuestionarios a los stakeholders generados y base de conocimiento para generarlos.

Metodologías de Evaluación: (repositorio): Metodologías de evaluación generadas.

Planes y Presupuestos: (repositorio y base de conocimientos) Planes y presupuestos generados y base de conocimiento para generarlos.

Herramientas de medición (librería y base de conocimiento): Herramientas de medición y base de conocimientos para asesorar en su uso.

Resultados de las Mediciones (repositorio) de los resultados de las mediciones realizadas.

Resultados de los Análisis (repositorio y base de conocimiento) Se usa para analizar los resultados de las mediciones de acuerdo a la metodología de evaluación.

#### **4.2 Módulos**

Administrador de Contenidos: Para cada repositorio/base de conocimientos hay un módulo que gerencia su contenido.

Generador de Cuestionarios: Genera los cuestionarios para los distintos stakeholders

Analizador de Cuestionarios: Analiza las respuestas obtenidas a partir de los cuestionarios y asiste al módulo Generador de Metodologías en la creación de la metodología de evaluación

Generador de Metodologías: Compone metodologías de evaluación a partir de la meta-metodología, tomando en cuenta los factores de evaluación y el análisis de los cuestionarios

Generador de planes: Genera los planes de evaluación a partir de la metodología de evaluación

Generador de presupuestos: Genera los presupuestos a partir de los planes de evaluación

Kit de Herramientas de Medición: Un kit de herramientas para medir métricas.  
Analizador de resultados: Analiza los resultados de las mediciones y produce informes acerca de la calidad del producto.

#### **4.3 Modo de Uso del framework**

El usuario que desea evaluar su producto ingresa la información acerca de los factores de evaluación (ver sección 2). El framework propone los valores de las Importancias Relativas (ver expresión (3.7)) para todos los ítems que se consideren relevantes. El usuario podrá adoptar los valores propuestos, podrá rechazarlos, o podrá usarlos como los valores propuestos por un “stakeholder representativo” (ver sección 3.4), en cuyo caso les dará pesos a cada uno las Importancias Relativas propuestas por el framework.

El framework genera los cuestionarios para los distintos stakeholders. Los stakeholders completan los cuestionarios. Usando la información contenida en los cuestionarios el framework propone los nuevos valores de las Importancias Relativas. Si se aceptan, el framework genera el plan de evaluación y calcula el presupuesto.

Se llevan a cabo las mediciones indicadas en el plan de evaluación. Se recolectan los resultados de las mediciones en el repositorio “Resultado de las Mediciones” del framework. Finalizadas las mediciones, el framework analiza los resultados y genera los informes referidos al grado de calidad del producto.

## **5 Investigaciones y trabajos futuros**

En el desarrollo del framework será necesario:

- Idear funciones que traduzcan las medidas de las métricas a grados de calidad de las mismas (ver sección 3.1).
- Diseñar los cuestionarios a los stakeholders, de tal forma que las Importancias Relativas que se deduzcan de sus respuestas realmente representen la visión de calidad que tienen los stakeholders (ver sección 3.3).
- Confeccionar directivas de cómo asignar pesos relativos a los stakeholders para ponderar su competencia al estipular Importancias Relativas a los ítems de calidad (ver sección 3.3).
- Idear la forma de medir las métricas con una fidelidad ( $F < 1$ ) de tal manera que el esfuerzo de las mediciones sean proporcionales a las fidelidades (ver sección 3.5).
- Diseñar e implementar los módulos y las bases de datos descriptas en la arquitectura del proyecto (ver sección 4).
- Aplicar el framework en la evaluación de distintos tipos de productos de software y ajustarlo en base a los resultados de las pruebas.

## **6. Conclusiones**

La creciente importancia que cobra la evaluación de la calidad de los productos y artefactos de software y la intensificación de las actividades de evaluación para todos los tipos de productos, creó la necesidad de sistematizar y automatizar el proceso de evaluación en todas sus fases. Nuestro proyecto MyFEPS, suministra una respuesta a esta necesidad.

Una característica fundamental de MyFEPS es, que a diferencia de todos los métodos de evaluación existentes toma en cuenta las perspectivas de calidad de los agentes (personas, instituciones, etc.) interesados en la calidad del producto.

## Referencias

1. Gartner: Magic Quadrant for Integrated Software Quality Suites, <http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA2-8558ENW.pdf>
2. Jones C.: Software Quality in 2010: A Survey of the state of the art, [http://www.semat.org/pub/Main/PubsandRefs/software\\_quality\\_survey\\_2010.ppt](http://www.semat.org/pub/Main/PubsandRefs/software_quality_survey_2010.ppt)  
<http://www.sqgne.org/presentations/2010-11/Jones-Nov-2010.pdf>
3. Schiffauerova, A., Thomson, V. : Cost of quality: Survey of best practices and models, In: Total Quality Management, p.16, V.V. Gopal (editor), 2004.
4. Clarus: Software Quality Attributes: Following All the Steps, <http://www.clarus.com/documents/Software%20Quality%20Attributes.pdf>
5. Kitchenham, B., Lawrence, Pfleeger, S.L.: Software Quality: The Elusive Target, IEEE Software, 13, 12--21, (1996)
6. ISO/IEC 9126-1 Software Engineering - Product Quality – Part1: Quality Model, ISO, 2001.
7. Kan, S.H.: Metrics and Models in Software Quality Engineering, Addison-Wesley Longman Publishing Co., Inc (2002)
8. JISC: JISC E-learning Tools: Software Quality Evaluator Evaluation Criteria Document, [http://www.jisc.ac.uk/uploaded\\_documents/Software%20Quality%20Evaluation%20Criteria%20R01.doc](http://www.jisc.ac.uk/uploaded_documents/Software%20Quality%20Evaluation%20Criteria%20R01.doc)
9. Voas, J.: Developing a Usage-Based Software Certification Process, Computer, 33, 32-- 37, (2000)
10. Boehm, B.W., Abts C., Chulani S.: Software development cost estimation approaches - A survey, Annals of Software Engineering, 10, 177--205, (2000)
11. ISO/IEC 14598 1999-2001. Information Technology - Software Product Evaluation - Parts 1-6, ISO, 1999.
12. Rodríguez, M., Verdugo, J., Coloma, R., Genero, M., Piattini, M.: Metodología para la evaluación de la calidad en los modelos UML, REICIS Revista Española de Innovación, Calidad e Ingeniería del Software, 6, 16--35 (2010)
13. Rodríguez, M., Genero, M., Garzas J., Piattini, M.: KEMIS: Entorno para la Medición de la Calidad del Producto Software, RPM-AEMES, 4, 168--182 (2007)
14. IEEE Std 1061-1992, IEEE Standard for a Software Quality Metrics Methodology, IEEE Press, (1992).
15. Olsina, L., Rossi, G.: Measuring Web Application Quality with WebQEM, IEEE Multimedia, 9, N° 4 (2002).
16. Boloix, G., Robillard, P.N.: A software system evaluation framework, IEEE Comput., 28, 17—27, (1995).
17. Zahedi, F.: A method for quantitative evaluation of expert systems, Eur. J. Oper. Res., 48, 136—147, (1990).
18. Morisio M., Tsoukiàs, A.: IUSWARE: A formal methodology for software evaluation and selection, IEE Proceedings on Software Engineering, 144, 162–174, (1997).