**Aplicación de la estrategia GOCAMECom (Goal-Oriented Context-Aware Measurement, Evaluation and Comparison)**

Germán Vargas - Leonardo Corazza

Ingeniería de Software II – Facultad de Ingeniería – UNLPam – Argentina

corazzal@hotmail.com; [germanv@gmail.com](mailto:germanv@gmail.com)

**Resumen.** Se aplicara el uso de una estrategia de medición y evaluación llamada

GOCAMECom (Goal-Oriented Context-Aware Measurement, Evaluation and

Comparison). Con vista de calidad de producto, con foco en calidad interna, don

de la sub-categoría de entidades a evaluar son ontologías de testing de software.

Se determinara las fortalezas y debilidades de cada una de las ontologías. Para

este trabajo no se desarrollara la nueva ontología, solo se indicarán las fortalezas

que se adoptarían de las ontologías preseleccionadas.

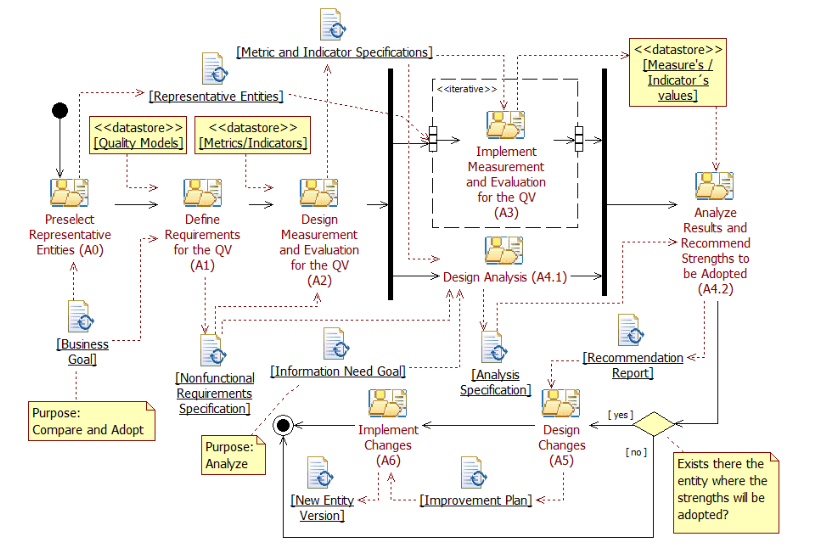
1. Introducción

En toda organización se establecen metas de negocio , es decir propósitos que se desean alcanzar y estos se operacionalizan a través de proyectos que durante su ciclo de vida aplican procesos , métodos , actividades que están contenidas en *estrategias*. Las estrategias en particular definen que hacer para alcanzar el proposito subyacente (en este caso el proposito es comparar y analizar un conjunto de ontologias) estableciendo específicamente qué debe hacerse y cómo debe realizarse. Para ello , resulta de gran utilidad que las estrategias integren tres pilares a saber; una base conceptual de dominio , una especificacion del proceso y especificaciones de metodos.

Este principio de los tres pilares integrados permite, entre otros beneficios, tener explícitamente definido el dominio correspondiente, saber qué actividades se deben realizar, y cómo llevarlas a cabo mediante métodos y para las organizaciones que consideren llevar adelante proyectos de un modo sistematico mediante el empleo de estrategias integradas pueden asegurar repetitividad de procesos y consistencia de resultados.

Partiendo de esta idea , en la presente monografia se selecciona y aplica la estrategia GOCAMECom (Goal-Oriented Context - Aware Measurement , Evaluation and Comparison) para especificar la medicion , la evaluacion elemental y global considerando la vista de calidad de producto , con foco en calidad interna , donde la sub-categora de entidad a evaluar es una ontologia de testing de software.

1. Estrategia a ser aplicada : GOCAMECom (GoalOriented Context-Aware Measurement, Evaluation and Comparison)



**Fig. 1** Especificación del proceso genérico para la estrategia GOCAMECom para la vista de calidad.

**2.1 Descripción del Proceso GOCAMECom**

El Proceso GOCAMECom comienza con la *“Pre-seleccionar Entidades Representativas*” para ser comparadas **(A0)**. La pre-seleccion puede estar basado en métodos tales como juicio de expertos , caso de estudios , entre otros. Hay que tener en cuenta que si las fortalezas se adoptarán en una entidad existente, esta debería ser una de las entidades preseleccionadas. Esta actividad consume “*Meta de Negocios”* y produce *“Entidades representativas”*.

Luego la actividad **(A1)** *“Definir los requisitos no funcionales para la vista de calidad”* , tienen como entrada un modelo de calidad , la información sobre el contexto , y la meta de negocio cuyo propósito es comparar y adoptar. A partir de acá se define el árbol de requisitos no funcionales partiendo de un modelo de requisitos no funcionales , identificando la características raíz que es el foco de evaluación como también las características y atributos asociados a ella. Como producto de trabajo obtenemos el “Documento de requisitos no funcionales”

En **(A2)** *“Diseño de la Medición y Evaluación para la vista de calidad”* se usa como entrada el documento de especificación de requisitos no funcionales y repositorios de métricas e indicadores si los hubiese.

En esta actividad se seleccionan y/o diseñan las métricas que permiten medir los atributos y se definen los indicadores que permiten interpretar los valores medidos (Medidas) para que por medio de los valores de indicadores se pueda conocer el nivel de satisfacción alcanzado para cada uno de los RNFs.

Luego, se puede realizar en paralelo las actividades **(A3)** *“Implementar la Medición y Evaluación para la vista de calidad”* ,

aquí se usa como entrada *“La Especificación de Métricas e Indicadores”* y “*Entidades Representativas”* . Se procede a realizar las mediciones correspondientes a cada uno de los atributos para la instancia que se evalúa. Para lograr esto seguimos los pasos descritos en el documento *“Especificación de Métricas e Indicadores”* a través de un procedimiento de cálculo o medición dependiendo la naturaleza de la métrica.

Una vez que se obtuvo las medidas ,estas se usan como entradas para realizar la evaluación. Para ello se siguen los procedimiento de cálculo presentes en la especificación de los indicadores asociados a cada atributo. Como resultado se obtiene los valores de los indicadores para cada atributo, a partir de aquí se puede calcular los valores de los indicadores de las características.

Una vez que se tienen todos los valores de indicadores de todos los atributos , se debe realizar la evaluación derivada a fin de obtener un valor de indicador para cada característica del árbol de RNFs.

La evaluación derivada nos permite obtener valores comparables de características abstractas que no se pueden medir de forma directa, para esto se tiene en cuenta los pesos y operadores relativos a las características.

Esta actividad como se puede ver en la **Fig 1** , es *<<iterativa>>*, es decir por cada una de las entidades representativas , se va a *“Implementar la Medición y Evaluación”* , que produce un repositorio con las *“Medidas y los valores de los indicadores”.*

La actividad **(A4.1)** , *“Diseñar el Análisis”* tiene como insumo los documentos *“Especificación de requisitos no funcionales“* y *“Especificación de Medición y Evaluación”* . Aquí se establecen los criterios para el análisis de los resultados, en este punto se recomienda tener en cuenta los diferentes *“Criterios de Decisión”* utilizados en los indicadores ya que resultan útiles para determinar que condiciones nos permiten clasificar un atributo y/o características como una debilidad o fortaleza.

Como salida de las actividades **(A3)** y **(A4.1)** *“Repositorio de las medidas y valores de los indicadores”* Y *“Especificación del Análisis”* respectivamente , estas salidas se usan como insumo para la nueva actividad **(A4.2)** denominada “Analizar los resultados y recomendar fortalezas a ser adoptadas” , en esta actividad se busca recomendar cambios sobre los atributos relacionados a ellas. Resulta de utilidad fijar prioridades a las recomendaciones de cambio , ya que en general el tiempo en los proyectos es limitado. Como resultado esta información se vuelca en *“Informe de conclusión y recomendación”* .

Una vez finalizada la actividad **(A4.2)** , en caso de no existir recomendaciones de cambios en la Entidad y/o en su Contexto, por ejemplo, debido a que el nivel de satisfacción alcanzado es óptimo, el proceso finaliza. Por el contrario, si existen recomendaciones de cambio debido a debilidades detectadas, se continúa el flujo de ejecución con la actividad *“Diseñar los Cambios”* **(A5)**. A partir del *“Informe de conclusión y/o Recomendación”*, se plantean las acciones de cambio sobre las debilidades detectas, como también los métodos a utilizar para llevarlas a cabo. Vale la pena aclarar que se pueden dejar fuera algunas de las debilidades detectadas teniendo en cuenta diferentes factores como pueden ser el tiempo y el presupuesto disponible entre otros. Como salida de esta actividad se obtiene *“Plan de mejora”*.

*“Implementar los cambios”* es la actividad codificada como **(A6)**. Partiendo de lo expresado en el *“Plan de Mejora”*, se procede a realizar los cambios sobre la Entidad. El resultado es una nueva versión de la Entidad bajo estudio.

1. Aplicación de la estrategia GOCAMECOm
2. **Especificación de la Necesidad de información**

* *Declaración de la meta de negocio:* *Comparar* un conjunto de ontologías de testing de software con el objetivo de *adoptar* las mejores características en una nueva ontología de testing de software a desarrollar.
* *Vista de calidad:* Vista de calidad de producto.
* *Categoría de entidad:* Ontologías de testing de software.
* *Foco de calidad:* Foco en calidad interna.
* *Entidad:* TestTDO 1.1, RooST, ATopDomainOntologyForSoftwareTesting
* *Especificación del proceso:*  GOCAMECom: (Ver sección 2 , Fig 1)

1. ***Pre-seleccionar Entidades Representativas***

La selección de las entidades competidoras fueron otorgadas por los docentes que componen la cátedra de Ingenieria de Software II.

Como resultado de la elección 3 entidades fueron seleccionadas:

1. TesTDo
2. RoosT
3. A Top Domain Ontology For Software Testing
4. **Definir los requisitos no funcionales para la vista de calidad**

Esta actividad produce el *“Documento de requisitos no funcionales”* que incluye *Las Metas de negocio* , *la especificación de la necesidad de información* , *información sobre el contexto* y el á*rbol de requisitos no funcionales.*

Por su parte el árbol de requisitos no funcionales comienza con su característica raíz “Calidad interna” , seguido de sub-caracteristicas y atributos. En particular nos centraremos en evaluar dos sub-caracteristicas; *Calidad Estructural Ontológica Y Calidad De Reuso Ontológico* que a su vez están relacionados con atributos como se mostrara en la siguiente tabla.

|  |  |
| --- | --- |
| **1.Interna Quality** |  |
| **1.1 Ontological Structural Quality** | …an ontology is well structured, i.e., has defined terms availability, defined properties availability, specified axioms availability and it is properly balanced with regard to types of relationships. |
| 1.1.1 Defined Terms Availability | …an ontology has defined terms. |
| 1.1.2 Defined Properties Availability | …an ontology has defined properties. |
| **1.1.3 Balanced Relationships Availability** | …an ontology has a balance between the amount of non-taxonomic and taxonomic relationships in addition to the former are defined. |
| 1.1.3.1 Defined Non-Taxonomic Relationships Availability | …an ontology has a balance between the amount of non-taxonomic and taxonomic relationships. |
| 1.1.3.2 Balanced Non-Taxonomic/Taxonomic Relationships Availability | …an ontology has a balance between the amount of non-taxonomic and taxonomic relationships. |
| **1.2 Ontological Reuse Quality** |  |
| **1.2.1** Level of Reuse by Semantic Extensibility |  |
| **1.2.3** Level of Complete Reuse of Terms | … an ontology completely reuses terms defined in other ontologies |

**Tabla 1 :** Árbol de requisitos no funcionales

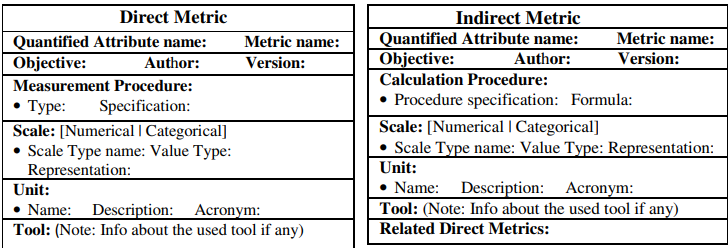
1. ***Diseño de la Medición y Evaluación para la vista de calidad***

La actividad **(A2)** *“Diseño de la Medición y Evaluación para la vista de calidad”* se usa como entrada el documento de *“Especificación de requisitos no funcionales”* y *“Repositorios de métricas e* indicadores” si los hubiese.

En esta actividad se seleccionan y/o diseñan las métricas que permiten medir los atributos y se definen los indicadores que permiten interpretar los valores medidos (Medidas) para que por medio de los valores de indicadores se pueda conocer el nivel de satisfacción alcanzado para cada uno de los RNFs.

Para las (especificaciones de métodos), GOCAME cuenta con plantillas creadas a partir de la base conceptual  que permiten especificar métricas e indicadores.

La salida de esta actividad es la *“Especificación de Métricas e Indicadores” ,* esto se encuentra en el Apéndice [A] y Apéndice [B] respectivamente .



**Fig 2** : Plantilla de Métricas Directas e Indirectas

En el Apéndice A se especifican el resto de las métricas utilizadas para este caso de estudio, mientras que en el Apéndice B se encuentran los demás indicadores elementales. Cabe mencionar que todos los indicadores (elementales y derivados) tienen una escala porcentual y se han utilizado los mismos criterios o niveles de aceptabilidad. Específicamente, se utilizaron tres niveles de aceptabilidad: “Satisfactorio” cuyo rango de valores está entre [90-100%]; “Marginal” [60-90); e “Insatisfactorio” [0-60).

1. **Implementar la Medición y Evaluación para la vista de calidad**

Aquí se usa como entrada *“La Especificación de Métricas e Indicadores”* y “*Entidades Representativas”* como se describe en la **Fig 1** actividad **(A0)**. Se procede a realizar las mediciones correspondientes a cada uno de los atributos para la instancia que se evalúa y luego, con las medidas obtenidas, se procede a calcular los indicadores elementales y derivados. Recordemos que esta actividad es *<<Iterativa>>*  por cada Entidad Representativa se debe implementar la actividad definida como **(A3)** , dichas entidades son: TesTDo - ROoST - A Top Domain Ontology For Software Testing.

Para lograr esto seguimos los pasos descritos en el documento *“Especificación de Métricas e Indicadores”* ***(A2)*** a través de un procedimiento de cálculo o medición dependiendo la naturaleza de la métrica.

Una vez que se obtiene todas las Medidas, estas se utilizan como entrada para realizar la *Evaluación*. Para ello, se siguen los Procedimientos de Cálculo presentes en la especificación de los Indicadores asociados a cada Atributo Apéndice B. Como resultado se obtienen los Valores de Indicadores para cada Atributo. A partir de ellos podemos calcular los Valores de Indicadores de las Características.

Ya con los valores de indicadores de todos los atributos, se debe realizar la evaluación derivada a fin de obtener un valor de indicador para cada característica del árbol de RNFs.

A continuación en las siguientes tablas se podrá ver los resultados obtenidos para cada una de las entidades evaluadas.

1. Conclusión (Análisis)

As stated in the Introduction, a general benefit of having the suitable software testing ontology is that of minimizing the current heterogeneity, ambiguity and incompleteness problems in concepts, i.e., mainly in terms, properties and relationships. We have initially confirmed that there exists heterogeneity, ambiguity and incompleteness for concepts dealing with testing activities, artifacts and methods. Consequently, we are looking for the suitable software testing ontology for populating the ‘testing’ conceptual component (Fig. 1), which in turn should appropriately relate some of its concepts with FRs and NFRs components’ concepts. Note that an additional finding recorded in the last column of Table 6 is that none conceptualized ontology links its testing conceptual component or module with FRs and NFRs components.

In this study, we have documented the conducted SLR (as part of the A3’s second sub-activity) following a well-established SLR process. This process specification promotes the validation of the ‘SLR Protocol’ artifact not only in the A1 activity but also in A2.1 by performing a pilot test. In addition, we have paid thorough attention to consistency checking in the A2.2 sub-activity. Hence, we have tried to minimize the threats of validity for getting evidence that this type of research methodology entails.

Referencias

1. Arnicans G., Romans D., Straujums U.: Semi-automatic Generation of a Software Testing Lightweight Ontology from a Glossary Based on the ONTO6 Methodology, Frontiers in Artificial Intelligence and Applications, V.249, pp. 263-276, (2013)
2. Asman A., Srikanth R. M.: A Top Domain Ontology for Software Testing, Master Thesis, Jönköping University, Sweden, pp. 1-74, (2016)
3. Bai X., Lee S., Tsai W. T., Chen Y.: Ontology-Based Test Modeling and Partition Testing of Web Services, In IEEE Int’l Conference on Web Services (ICWS'08), pp. 465-472, (2008)
4. Barbosa E. F., Nakagawa E. Y., Riekstin A. C., Maldonado J. C.: Ontology-based Development of Testing Related Tools, 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08), pp. 697-702, (2008)
5. Becker P., Papa F., Olsina L.: Process Ontology Specification for Enhancing the Process Compliance of a Measurement and Evaluation Strategy, CLEI eJnal., 18:(1), pp. 1-26, (2015)
6. Brereton P., Kitchenham B., Budgen D., Turner M., Khalil M.: Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software, 80:(4), pp. 571–583, (2007)
7. Cai L., Tong W., Liu Z., Zhang J.: Test Case Reuse Based on Ontology, In15th IEEE Pacific Rim International Symposium on Dependable Computing, pp. 103-108, (2009)

[Apéndice/s]