**“Identificación y priorización de Quality Attributes en requerimientos”**

**Alumnos:**

Francisco Bertoni.

Sebastián Villanueva.

**Directores:**

Dra. Claudia Marcos.

Dr. Andrés Díaz Pace.

**1. Motivación**

La calidad de software se define como el grado en el cual el software posee una combinación deseada de atributos [1]. Tales atributos son requerimientos adicionales del sistema  [11] que hacen referencia a características que éste debe satisfacer, diferentes a los requerimientos funcionales.  Estas características o atributos se conocen con el nombre de atributos de calidad, los cuales se definen como las propiedades de un servicio que presta el sistema a sus usuarios [9].

A grandes rasgos, Bass en [10] establece una clasificación de los atributos de calidad en dos categorías:

* *Observables en tiempo de ejecución:* aquellos atributos que se determinan del comportamiento del sistema en tiempo de ejecución (disponibilidad, performance, seguridad, etc.)

* *No observables en tiempo de ejecución:* aquellos atributos que se establecen durante el desarrollo del sistema (modificabilidad, reusabilidad, portabilidad, escabilidad, etc.).

Estos atributos de calidad son especificados a través de requerimientos no funcionales, en adelante denominados como NFRs, estableciendo relaciones importantes sobre el desarrollo y comportamiento de un sistema. De esta manera, juegan un rol critico en guiar el diseño arquitectónico y por lo tanto deben ser consideradas y especificadas lo más temprano posible durante el análisis del mismo [3].

Cada vez se está volviendo más evidente la importancia de la relación entre NFRs correctamente especificados y el éxito de un sistema [4]. Por ejemplo, las consecuencias de una falla de performance pueden variar considerablemente, incluyendo: relaciones dañadas con el cliente, fallas en el negocio, pérdida de ingresos, aumento del costo del proyecto debido a los costos de sumar recursos adicionales para resolver la falla, reducción de la competencia y, potencialmente, contribuir a la caída del proyecto. De esta manera, los NFRs son comúnmente señalados como el factor clave de éxito en la construcción de software de alta calidad, estableciendo un enfoque sistemático y pragmático para introducir calidad en los sistemas de software [5].

Un concepto relacionado con los requerimientos no funcionales es el de *aspectos tempranos* (Early Aspects). Un aspecto temprano es un concern que atraviesa el diseño de un sistema [8] y es encontrado en las especificaciones de los requerimientos u otros primeros documentos producidos en la captura de requerimientos. Muchos aspectos tempranos son similares a requerimientos no funcionales de alto nivel como seguridad, performance, portabilidad y usabilidad [3]. El descubrimiento de los mismos puede llevar al reconocimiento de NFRs.

La técnica para el descubrimiento de aspectos tempranos se denomina minearía de aspectos (aspect mining). Cabe destacar que generalmente en la literatura se utiliza el término minería de aspectos principalmente para referirse a la extracción de aspectos del código fuente de un sistema; sin embargo, se considera que este concepto es más amplio y puede involucrar a cualquier artefacto producido durante el desarrollo. Por ejemplo, los artefactos involucrados para la identificación de aspectos tempranos son las especificaciones de requerimientos.

Justamente una de las técnicas  más difundidas en la etapa de especificación de requerimientos es la de Casos de Uso. Desde la introducción de éstos por Jacobson en el año 1865, la notación se ha vuelto una herramienta casi omnipresente en los proyectos informáticos. Incluso su uso ha abarcado diversas formas, como por ejemplo el reparto de tareas y los recursos necesarios, análisis del dominio, creación de modelos de objetos y componentes, diseño de casos de test y estimaciones [4].

**2. Problemática**

Mientras que los requerimientos funcionales han recibido mucha atención en la literatura, los aspectos no funcionales de la arquitectura han recibido un menor interés [4], sin un claro estándar ni herramienta de ayuda emergente para su identificación.

Los requerimientos no funcionales juegan un rol crítico en el diseño de la arquitectura, por lo tanto la detección temprana de estos atributos de calidad es crucial para tomarlos en consideración, empezando desde las primeras decisiones de diseño. A pesar de que parece una tarea sencilla, puede resultar muy dificultoso e insumir mucho tiempo, especialmente considerando que la mayoría de NFRs  están, de alguna forma, “escondidos” a través de los requerimientos que especifican la funcionalidad, y pueden ser fácilmente ignorados [6]. Tal es así, que Saleh y Al-Zarouni concluyeron que, a diferencia de los requerimientos funcionales, los requerimientos no funcionales son muy difíciles de elicitar [7].

En este sentido, logar una buena identificación de los atributos de calidad en etapas tempranas permitirá, desde un comienzo, considerar aquellos aspectos que guiaran a un sistema de calidad a la hora de tomar las primeras decisiones. Estas decisiones involucran al diseño inicial, arquitectura del sistema y cualquier otro artefacto que guie el proceso de desarrollo.

**3. Solución Propuesta**

Durante la captura de requerimientos difícilmente se logren identificar claramente aquellos atributos no funcionales que guiaran el sistema. Sin embargo, existe la posibilidad de realizar un análisis sobre los documentos generados en esta etapa para identificar aquellos requerimientos no funcionales que han quedado establecidos de forma implícita dentro de los documentos.

Por lo tanto, se propone que partiendo de un conjunto de aspectos tempranos identificados mediante alguna herramienta de detección de Early Aspects, por ejemplo la herramienta Aspect Extractor Tool [X], se identifique el grado de pertenencia de cada aspecto a los diferentes atributos de calidad del sistema, formando lo que se denominará un Quality Attribute Theme (QAT).  Básicamente, un QAT será un subconjunto formado por cada early aspect y los casos de uso sobre los cuales se detectó, complementando con el atributo de calidad más relevante para ese subconjunto. No solo se pretende identificar un único atributo de calidad para cada early aspect, sino que se propone identificar el grado de pertenencia de todos los atributos de calidad del sistema, permitiendo lograr además una priorización de los mismos. Aquel de mayor prioridad será el que finalmente forme parte del QAT.

De esta forma, al analista contará con información importante desde las primeras etapas del desarrollo del sistema, reconociendo los atributos de calidad del mismo, minimizando las consecuencias de no conocerlos a tiempo y posibilitando un mejor tratamiento en las siguientes etapas de desarrollo.

**4. Plan de trabajo**

* Relevamiento bibliográfico de Atributos de Calidad, Escenarios de Calidad, Ontologías y Reconocimiento de texto. Tiempo Estimado: 45 días.
* Análisis de técnicas existentes. Tiempo Estimado: 15 días.
* Definición de la técnica a implementar. Tiempo Estimado: 15 días.
* Definición de las características del proceso de identificación propuesto: 15 días.
* Análisis y Diseño de una herramienta que permita llevar a cabo el reconocimiento y priorización de atributos de calidad. Tiempo Estimado: 30 días
* Implementación de la herramienta. Tiempo Estimado: 60 días.
* Experimentación con ejemplos y análisis de los resultados obtenidos. Tiempo Estimado: 30 días
* Escritura del informe final de tesis y conclusiones. Tiempo Estimado: 60 días

**5. Bibliografía**

[1] IEEE Standard 1061-1992. Standard for a Software Quality Metrics Meth- IEEE-1061 Methodology. New York: Institute of Electrical and Electronics Engineers,  1992.

[2] A requirements model for quality attributes - Isabel Brito, Ana Moreira, João Araújo

[3] Automated classiﬁcation of non-functional requirements - Jane Cleland-Huang Æ Raffa

[4] Modeling Architectural Non Functional Requirements: From Use Case to Control Case

[5] ElicitO: A Quality Ontology-Guided NFR Elicitation Tool

[6] Identiﬁcation of non-functional requirements in textual speciﬁcations: A semi-supervised learning approach - Agustin Casamayor, Daniela Godoy, Marcelo Campo.

[7]  K. Saleh and A. Al-Zarouni, “Capturing Non-Functional Software Requirements Using the User Requirements Notation”, The 2004 International Research Conferenceon Innovations in Information Technology (IIT2004), Dubai UAE, October 2004. (Sacado de [4])

[8]   Baniassan E, Clements P, Araujo J, Moreira A, Rashid A, Tekinerdogan B (2006) Discovering early aspects. IEEE Softw 23(1):61–70 (Sacado de 3)

[9] Quality Attributes - Mario Barbacci, Mark H. Klein, Thomas A. Longstaff, Charles B. Weinstock, December 1995

[10] Bass, L., Clements, P., & Kazman, R. (1998).   Software Architecture in practice.   
Addison-Wesley.

[11] Kazman, R., Clements, P., Klein, M. (2001). Evaluating Software Architectures.   
Methods and case studies. Addison Wesley.