

Identificación de interacciones entre las características de calidad del software

Gabriel Alberto García-Mireles¹, M^a Ángeles Moraga², Félix García² y Mario Piattini²

¹Departamento de Matemáticas, Universidad de Sonora
Blvd. Encinas y Rosales s/n col. Centro, 83000 Hermosillo, Sonora, México
mireles@gauss.mat.uson.mx

²Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, España
{MariaAngeles.Moraga, Felix.Garcia, Mario.Piattini}@uclm.es

Resumen. Las interacciones entre atributos de calidad del software pueden ser conflictivas, situación que podría dificultar que el software cumpla con las expectativas de los participantes. Los conflictos entre los requisitos de calidad pueden identificarse al revisar tablas de interacción que muestren el tipo de contribución potencial que tendría una característica de calidad sobre otra. Por tanto, realizamos una revisión de la literatura para recopilar las tablas de interacción publicadas. Tras la revisión de los artículos seleccionados, proponemos una técnica para construir tablas de interacción, la cual fue aplicada en los modelos de calidad de usabilidad, mantenibilidad y seguridad. Con la intención de corroborar la información de las tablas, realizamos una encuesta exploratoria cuyo objetivo es determinar si las empresas advierten las interacciones, y en su caso, conocer cómo las gestionan. Los resultados señalan que este es un tema de interés y que la práctica actual requiere de soporte metodológico adecuado.

Palabras clave: requisitos de calidad del software, interacción entre características de calidad, conflicto entre características de calidad, tablas de interacción.

1 Introducción

La calidad del software es un aspecto fundamental que debe ser abordado durante el desarrollo del producto. La calidad del software se define como “el grado en el cual el software posee una combinación de atributos deseados (por ejemplo: fiabilidad, interoperabilidad)” [1]. Un atributo de calidad, por su parte, es una propiedad de un producto o servicio por la cual los participantes relevantes pueden juzgar la calidad [2]. Esos atributos deseados se pueden agrupar en características que se han descrito en diversos modelos de calidad del software, tales como el ISO/IEC 25010 [3], ISO/IEC 9126 [4], Dromey [5], McCall [6] y Boehm [7].

Una meta importante en el desarrollo de software es lograr un equilibrio adecuado entre los atributos de calidad [8]. Durante el proceso de obtención de requisitos se

presenta la ocasión para identificar las posibles interacciones entre los requisitos de calidad, en particular las conflictivas. Sin embargo, la identificación de los conflictos entre atributos de calidad es un problema complejo. Dicha complejidad puede estar causada por las diferencias en el significado del término calidad que cada participante del proyecto concibe o por la escasa cantidad de métodos, técnicas y herramientas diseñadas para gestionar el conflicto [9, 10]. La mayoría de los recursos tecnológicos para el análisis comparativo, o dependencias, entre características de calidad, se concentran en las etapas de análisis de requisitos y evaluación del diseño arquitectónico [11], etapas del proceso software en las cuales es deseable resolver los problemas.

A pesar de las limitaciones encontradas en esta área de estudio, las interacciones entre atributos de calidad se deberían considerar en los proyectos software. La gestión pobre (o inexistente) de este tipo de interacciones puede ser una causa de los fracasos en proyectos [8]. Sin embargo, las interacciones rara vez se gestionan y esta situación es problemática debido a que la identificación del conflicto es el primer paso para tomar decisiones entre alternativas de solución [12, 13]. Además, se considera una práctica de riesgo sólo analizar por separado cada una de las características de calidad sin considerar la posible interacción entre ellas [14].

La forma tradicional de abordar las interacciones negativas, o conflictos, es a través de análisis comparativos (*tradeoffs*). Los enfoques existentes, como WinWin [8] y el marco NFR [15, 16] proporcionan estrategias para identificar el conflicto. El primero utiliza tablas en las cuales se relacionan los conflictos potenciales que una solución arquitectónica tendría sobre una característica de calidad particular. En el segundo enfoque se introduce el concepto de meta suave (*softgoal*) para modelar el tipo de contribución que tendrá una meta de calidad, o mecanismos de solución, sobre los objetivos globales del software. Otros investigadores han propuesto tablas de dependencia como mecanismo para identificar conflictos potenciales entre características de calidad [10].

Dada la importancia del tema, en este trabajo realizamos una revisión de las propuestas de tablas de interacción (o dependencia) entre características de calidad. El objetivo es identificar las características de calidad estudiadas y el enfoque usado para justificar dicha interacción. Con base en la revisión de la literatura, proponemos una técnica para derivar tablas de interacción para modelos de calidad de producto software y elaboramos un conjunto de tablas de interacción para las características de calidad de usabilidad, mantenibilidad y seguridad. Realizamos además una encuesta exploratoria a profesionales cuya intención es corroborar la información contenida en las tablas que proponemos y obtener información respecto de la gestión de interacciones entre características de calidad.

La sección dos describe los trabajos relacionados con las interacciones entre características de calidad. La sección tres presenta el proceso para elaborar las tablas de interacción y algunos ejemplos relacionados con la mantenibilidad y usabilidad. La sección cuatro muestra los resultados preliminares de la encuesta. Finalmente, se exponen las conclusiones y trabajo futuro.

2 Trabajos relacionados

En el campo de ingeniería de requisitos se abordan los requisitos de calidad o requisitos no funcionales (NFR). Estos últimos constituyen las justificaciones para las decisiones del diseño y restringen la manera en que la funcionalidad requerida por el software puede ser desplegada [17, 18]. Además el impacto de los NFR puede ser local o sistémico. A diferencia de los requisitos funcionales, cuya inclusión en el software puede ser evaluada con una escala binaria, los NFR están incluidos en el software en diferentes grados, y por tanto, la escala de evaluación tendría que considerar, al menos, los rangos aceptables para cada característica de calidad de interés en el proyecto.

Los requisitos de calidad pueden producir situaciones de conflicto. Esto es, situaciones en las cuales es imposible, o muy difícil satisfacer simultáneamente un par de requisitos de calidad en un contexto específico del sistema [19, 20]. Algunos investigadores señalan que el conflicto es inherente a las características de calidad, mientras otros apuntan que depende de la arquitectura e implementación del software [21]. El origen del conflicto puede deberse al uso de un vocabulario inconsistente durante la ingeniería de requisitos, volatilidad de los requisitos y problemas para identificar su impacto en el producto software, complejidad del dominio, clientes y usuarios que proponen requisitos conflictivos, cambio en el personal asignado al proyecto, participantes no identificados y cambio en las expectativas del proyecto [10]. Otros factores que influyen en la identificación del conflicto son el nivel de abstracción en el cual están descritos y las dependencias identificadas entre ellas [10]. Algunas razones para que el conflicto se presente son: los requisitos necesitan el mismo recurso, un requisito describe una tarea que es dependiente de otro requisito y/o los requisitos describen consecuencias de otro [22].

Una manera de calificar las interacciones entre características de calidad es utilizar tablas de interacción. En ellas, cada interdependencia, o interacción, se califica como positiva, negativa o no especificada [10]. Las interacciones negativas son las más estudiadas, dado que el impacto en los costes del proyecto es mayor con respecto de las interacciones positivas [22]. De hecho, algunos investigadores consideran que las tablas de interacción son una solución práctica, ya que estas pueden guiar la inferencia entre interacciones potenciales entre requisitos de calidad [8].

La búsqueda que se ha realizado sobre tablas de interacción entre características de calidad se ha centrado en una revisión en la base de datos SCOPUS así como las referencias bibliográficas de los trabajos de Berander et al., [19] y Mairiza et al. [20]. A continuación se describen los elementos más importantes de cada uno de ellos.

Diversos investigadores han propuesto tablas de interacción entre características de calidad. Berander et al. [19], definen el conjunto de interacciones con base en los puntos de vista de gestores, desarrolladores y usuarios. Mairiza et al. [20], realizan un análisis exhaustivo de la literatura y reportan 20 requisitos de calidad que han sido definidos y, algunos de ellos evaluados según sus interacciones. Ellos proponen un esquema de clasificación, según la consistencia de la evidencia publicada, en el cual la interacción se denomina absoluta, relativa o nunca en conflicto. Con base en esa taxonomía, determinan que los informes de las interacciones entre requisitos de cali-

dad de seguridad y rendimiento son los más frecuentes en la literatura y corresponden a la categoría de conflicto absoluto. Además, señalan que cerca del 75% de las interacciones entre requisitos de calidad no se conocen porque no hay información en la literatura.

Egyed y Grünbacher [23] elaboraron un modelo de conflicto potencial y cooperación basado en el ISO/IEC 9126 [4]. Con base en la clasificación de requisitos del proyecto y las dependencias entre artefactos del proyecto se identifican las interacciones entre requisitos de calidad. Por su parte Zulzalil et al. [24] exploran la interacción entre atributos de calidad para aplicaciones web. Adaptan la definición de términos del ISO/IEC 9126 y realizan encuestas para determinar la interacción entre requisitos de calidad. Sus resultados son similares a los encontrados en la literatura. Boehm e In [8] elaboraron matrices de dependencia en donde la interacción se califica considerando tanto el atributo de calidad primario como la estrategia arquitectónica. Henningson y Wohlin [21] exploraron la percepción de los profesionales respecto de la interacción entre características de calidad y concluyeron que no es posible optimizarlos todos ellos simultáneamente. Además, señalan que en la literatura no se enuncian claramente las supuestas relaciones entre características de calidad. Sadana y Liu [25] apuntan que la identificación del conflicto se realiza ad hoc. Para generar sus tablas de dependencia, con base en el ISO/IEC 9126, proponen un proceso en el cual las comparaciones se realizan según el nivel de abstracción de la descripción de las características de calidad. Finalmente, Svahnberg y Henningson [26] consolidan el análisis de literatura con resultados de experimentos para analizar el impacto de las arquitecturas en los atributos de calidad. Su análisis muestra las dependencias que existen entre características de calidad con ciertas evidencias empíricas.

En resumen, aunque existen algunas evidencias para comprender las interacciones entre requisitos de calidad [20], se requieren más estudios empíricos para conocer su efecto en el producto y el proceso software [9, 17]. Los enfoques empleados para investigar las relaciones entre características de calidad son el análisis de literatura, encuestas y experimentos. La mayoría de los atributos de calidad analizados se basan en el ISO/IEC 9126, en particular en las características de funcionalidad, fiabilidad, rendimiento, usabilidad, mantenibilidad y portabilidad, pocos trabajos analizan las interacciones que podrían surgir entre las subcaracterísticas de dicho modelo. Finalmente, las tablas de interacción revisadas contienen de 5 a 20 características de calidad.

3 Elaboración de tablas de interacción entre características de calidad

En esta sección presentamos los aspectos que consideramos para elaborar las tablas de interacción entre características de calidad del producto, generalmente descritas en modelos de calidad, y su relación con los requisitos de calidad que podrían abordar los proyectos software. Además, se incluye el procedimiento usado para generar tablas de interacción entre usabilidad, mantenibilidad y seguridad.

La calidad del producto software cubre, al menos, dos dominios de estudio. El que surge en el ámbito operativo del proyecto, relacionado con la identificación y gestión de requisitos de calidad, donde estos últimos surgen de la consideración de las propiedades operativas que el software y su ambiente deben tener; y, el que se presenta en un nivel más alto de abstracción, en el cual se estudian las características de calidad del producto [27]. El modelo conceptual preliminar que hemos propuesto se presenta en la figura 1, el cual captura los conceptos relevantes para entender la relación entre los modelos de calidad y los proyectos software. La motivación para la elaboración de este modelo se basa en la escasez de estudios empíricos sobre la interacción en la gran mayoría de atributos de calidad.

En términos generales, los modelos de calidad están descritos en un nivel de abstracción más alto que la especificación detallada de cada requisitos de calidad abordado en el proyecto software. La relación entre esos dos dominios se presenta cuando los modelos de calidad se utilizan para clasificar los requisitos de calidad del proyecto. Otras relaciones relevantes son las que se establecen entre las interacciones, o conflicto, y las tablas de interacción. En estas últimas, la tabla de interacción proporciona información para que, de acuerdo a la clasificación de requisitos de calidad, se identifiquen conflictos potenciales. Se sabe que el uso de las tablas de interacción para identificar relaciones entre requisitos de calidad puede generar relaciones espurias entre estos últimos y que deben ser analizadas por el ingeniero de software [8, 23]. Si el equipo de desarrollo considera que efectivamente ha surgido un conflicto, entonces las características de calidad involucradas alimentarán las tablas de interacción. Las notas de la figura se refieren a las características de calidad que abordamos en este proyecto. En lo que sigue, centraremos la atención en los aspectos de interacción entre características de calidad, en el ámbito de las tablas de dependencia.

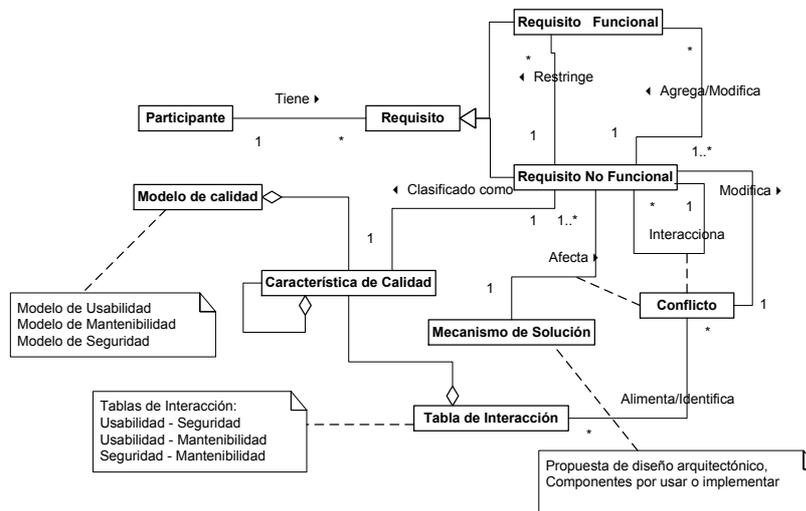


Figura 1. Modelo conceptual de las interacciones entre características de calidad

Como parte de la propuesta para gestionar las interacciones entre características de calidad, presentamos la técnica para elaborar las tablas de interacción base. Estas se construyen tomando en cuenta las propuestas publicadas y se adaptan a los modelos de calidad usabilidad, mantenibilidad y seguridad. En este artículo denominamos modelo de calidad a la estructura jerárquica compuesta por una característica abstracta la cual se descompone en elementos más concretos hasta que estos puedan ser medidos en el producto software.

El proceso realizado para obtener las tablas de interacción base está compuesto por:

1. *Analizar el modelo de calidad.* Analizar el vocabulario, estructura y metas del modelo.
2. *Determinar cobertura.* Con base en los términos que denominan a las características de calidad, identificar en la bibliografía las tablas de dependencia que las incluyen.
3. *Revisar la interacción entre características de calidad.* A partir de las referencias identificadas en el paso previo, revisar el tipo de interacción registrado para cada par de característica de calidad (usabilidad – mantenibilidad, usabilidad – seguridad, mantenibilidad – seguridad).
4. *Elaborar tabla de interacción.* Establecer las dependencias entre cada par de subcaracterísticas de los modelos de calidad con base en la calificación de la interacción más pesimista.

3.1 Analizar el modelo de calidad

La investigación presentada en este artículo forma parte del proyecto industrial MEDUSAS (Mejora y evaluación del diseño, usabilidad, seguridad y mantenibilidad del software) cuya meta es construir un ambiente basado en el ISO25000 para soportar el control de calidad y el aseguramiento de la calidad del software [28]. El proyecto tiene entre sus objetivos el desarrollo de una metodología de evaluación de calidad de productos software así como la elaboración de modelos de calidad específicos para la usabilidad, mantenibilidad y seguridad, heurísticas, listas de comprobación y medidas, que pueden usarse tanto en el código fuente como en los modelos UML generados en la etapa de diseño de software.

En este proyecto se elaboraron los modelos de calidad para la usabilidad, mantenibilidad y seguridad con base en la documentación del ISO25010 y con otros modelos de calidad publicados. Cada uno de ellos incluye los términos usados y sus definiciones, organizados en una estructura jerárquica. Por tanto, en este primer punto, se identificaron los términos y definiciones de los modelos de calidad de usabilidad, mantenibilidad y seguridad.

El modelo de mantenibilidad se compone de reusabilidad, analizabilidad, modificabilidad, modularidad y facilidad de prueba. Además, se identifican los atributos relacionados con cada una de las subcaracterísticas, tales como aumentabilidad, reestructuración, consistencia, estabilidad, simplicidad, autodescripción y legibilidad.

Esta lista de atributos facilita la identificación de métricas y las heurísticas para mejorar la mantenibilidad del software.

El modelo de usabilidad se basa en el análisis de 42 características identificadas en la literatura y contrastadas con el ISO25010. Con base en los estándares, establecen dos características principales: usabilidad interna/externa y usabilidad en uso. En la primera categoría se incluyen las subcaracterísticas pertinencia del reconocimiento, facilidad de aprendizaje, operabilidad, protección contra errores del usuario, estética de la interfaz de usuario y accesibilidad. En la característica de usabilidad en uso se considera eficacia, eficiencia, satisfacción (subjetiva), consecución de objetivos y confianza.

El modelo de seguridad considera dos categorías principales: protección ante ataques y protección ante accidentes. En el primer caso se incluyen las características de autenticidad, confidencialidad conformidad, detección de ataques, disponibilidad, integridad, no repudio y trazabilidad. Autenticidad, confidencialidad e integridad tienen definido un nivel más de componentes, en el cual se incluyen, en total, nueve subcaracterísticas. La categoría de protección ante accidentes sólo define las características de conformidad, daño comercial, daño ambiental, seguridad y salud del operador, y finalmente, seguridad y salud pública.

Como resultado de la elaboración de los modelos de calidad, listas de comprobación, heurísticas y mediciones, también surge la necesidad de gestionar apropiadamente las situaciones de conflicto entre características de calidad.

3.2 Determinar cobertura

Para cada uno de los modelos de calidad se buscaron los términos que han sido estudiados en las tablas de interacción. Los resultados se presentan en una matriz en donde las columnas corresponden a los términos del modelo de calidad, y las filas corresponden a las diferentes propuestas de interacciones entre características de calidad encontradas. La tabla 1 muestra los resultados para el modelo de calidad mantenibilidad.

Robinson et al. [10] y Boehm e In [8] señalan que una solución práctica a este problema se puede alcanzar considerando las características de calidad de más alto nivel de abstracción. Dado que el análisis de la literatura muestra que pocas interacciones han sido estudiadas, en la primera columna se agrega la característica principal. En el caso de la tabla 1, es mantenibilidad. Las columnas siguientes están etiquetadas con los nombres de las subcaracterísticas de calidad denominadas reusabilidad, analizabilidad, modificabilidad, modularidad y facilidad de prueba. Los resultados muestran que el estudio se ha realizado en el nivel de la característica de nivel superior (mantenibilidad) y que las subcaracterísticas rara vez han sido analizadas. Reusabilidad y facilidad de prueba han sido consideradas en al menos dos referencias bibliográficas. En el caso de los modelos de calidad usabilidad y seguridad, encontramos que las propuestas de interacción entre subcaracterísticas de calidad aún no han sido publicadas.

Tabla 1. Referencias que abordan el modelo de calidad de mantenibilidad

Característica →	Mantenibilidad	Reusabilidad	Analizabilidad	Modificabilidad	Modularidad	Facilidad de prueba
Referencia						
Berander et al. 2005 [19]		X	X	X		X
Mairiza et al. 2010 [20]	X	X				X
Egyed y Grünbacher, 2004 [23]	X					
Zulzalil et al., 2008 [24]	X					
Boehm e In, 1996 [8]	X	X			X	
Henningsson y Wohlin, 2002 [21]	X					

3.3 Revisar la interacción entre características de calidad

Las referencias identificadas en el paso anterior muestran las características de calidad estudiadas. Con base en las tablas de referencias y modelos de calidad, se pueden revisar las interacciones. Por ejemplo, la tabla 2 muestra las interacciones registradas para los modelos de usabilidad y de mantenibilidad. Sólo se han incluido las subcaracterísticas de usabilidad reportadas en las referencias bibliográficas.

La interacción positiva entre dos características de calidad (+) indica que se ha considerado que mejorar el nivel de calidad en una de las características tiene un efecto positivo sobre la otra característica. La interacción negativa (-) indica que los trabajos analizados han considerado que la mejora del nivel de calidad de una característica particular dificulta que otra característica de calidad pueda ser mejorada. La no interacción (O) indica que, los estudios han explorado la relación pero aún no se han reportado interacciones entre las característica de calidad bajo análisis. Finalmente, las celdas en blanco indican que no se sabe el tipo de relación que existe dado que aún no hay evidencias para calificar esas relaciones. Las celdas con el símbolo (/) están señalando que los autores de las tablas de interacción tienen distinta percepción del efecto de la relación entre las características de calidad en cuestión, es decir, existe una inconsistencia en la literatura al respecto.

La tabla 2 muestra que existen conflictos potenciales entre las características de calidad de operabilidad cuando se compara con la analizabilidad, modificabilidad y facilidad de prueba. Otro conflicto es el que se presenta entre eficiencia (usabilidad externa) y la reusabilidad. Las relaciones (marcadas con “O”) han sido estudiadas y no se han reportado conflictos entre ellas. Las relaciones entre usabilidad interna y mantenibilidad, así como la de usabilidad externa y facilidad de prueba, también se han reportado como dependencias sinérgicas. Sin embargo, los autores de tablas de interacción no señalan las condiciones bajo las cuales se mantienen las relaciones

conflictivas o sinérgicas, ni el contexto particular de los proyectos en los cuales fueron determinadas.

Tabla 2. Tabla temporal de dependencia entre usabilidad y mantenibilidad

Mantenibilidad →	Mantenibilidad	Reusabilidad	Analizabilidad	Modificabilidad	Modularidad	Facilidad de prueba
Usabilidad						
Usabilidad interna	O/+	O	O	O		O
Facilidad de aprendizaje						
Operabilidad		O	-	-		-
Usabilidad externa	O	O	O	O		O/+
Eficiencia		-	O	O		O

3.4 Elaborar tablas de dependencia

Existen pocos estudios que analizan las interacciones entre características de calidad [20]. Por otra parte, los modelos de calidad de MEDUSAS contienen subcaracterísticas que no han sido estudiadas desde la perspectiva de la interacción. Por tanto, para mitigar esta carencia de información, se toma en cuenta la estructura jerárquica del modelo y los enfoques usados en el campo de la medición del software para agregar resultados de mediciones básicas. Además, como se está trabajando en un nivel de abstracción alto para generar las tablas de dependencia base, con esta estrategia garantizamos que todas las subcaracterísticas estén calificadas. Así, en el caso de usabilidad y mantenibilidad, se genera la tabla 3. Aquí solo se presentan las filas que corresponden a la usabilidad interna del modelo de calidad de usabilidad de MEDUSAS.

Como se observa en la tabla 3, aunque diferentes autores han calificado de manera distinta la interacción entre usabilidad interna y mantenibilidad (O/+), la calificación de las características que no han sido estudiadas se obtuvo del valor más pesimista (en este caso, O). Por tanto, la operabilidad tiene conflicto potencial con analizabilidad, modificabilidad y facilidad de prueba. El resto de interacciones pueden considerarse que no tienen efecto.

Tabla 3. Tabla de interacción entre usabilidad y mantenibilidad (parcial)

Mantenibilidad →	Mantenibilidad	Reusabilidad	Analizabilidad	Modificabilidad	Modularidad	Facilidad de prueba
Usabilidad interna	O/+	O	O	O	O	O
Usabilidad interna	O/+	O	O	O	O	O
Pertinencia de reconocimiento	O	O	O	O	O	O
Facilidad de aprendizaje	O	O	O	O	O	O
Operabilidad	O	O	-	-	O	-
Protección contra errores	O	O	O	O	O	O
Estética de la interfaz	O	O	O	O	O	O
Accesibilidad	O	O	O	O	O	O

4 Exploración de la gestión de interacciones

Realizamos una encuesta cuyo objetivo, en primer término, es corroborar que las empresas de desarrollo de software advierten las interacciones. En segundo término, si las interacciones son identificadas, nos interesa comprender cómo son gestionadas. Para obtener datos iniciales sobre este tema, se realizó un cuestionario con preguntas abiertas en el cual se investigan la interacción percibida entre características de calidad y las estrategias de solución empleadas para resolver las interacciones negativas. Una lista parcial de las preguntas se encuentra en [29], las cuales se basaron en el modelo conceptual presentado en la figura 1.

El cuestionario fue contestado por los gestores de los proyectos de las tres empresas participantes en el proyecto MEDUSAS [28]. Las empresas colaboradoras son pequeñas, dos de ellas desarrollaron sistemas de información, mientras que la restante desarrolló una herramienta software para soporte a la medición. El tamaño del código generado está en el rango de 1,4 KLOC a 160KLOC y la duración de los proyectos está en el rango de 3 a 36 meses. La gestión del desarrollo se basó en SCRUM y al final de cada iteración se realizaban las mediciones en el código de la seguridad y mantenibilidad.

En el caso particular de las interacciones entre características de calidad de seguridad y mantenibilidad, las tres empresas encuestadas proporcionaron resultados distintos. Mientras que una de ellas señala que no pudo identificar interacciones, otra indica que las relaciones entre subcaracterísticas de calidad son independientes. Finalmente, otra proporciona un esquema en el cual señala que ellos percibieron diferentes tipos de interacciones entre características de calidad.

La tabla 4 muestra resultados parciales de las interacciones identificadas por una de las empresas. En particular, señalan tres interacciones negativas: confidencialidad – reusabilidad, detección de ataques - reusabilidad, detección de ataques - modificabi-

lidad. La primera interacción la justifican con el argumento de que los componentes considerados como confidenciales podrían interferir con los criterios de reusabilidad. Respecto de la detección de ataques, y su interacción con la modificabilidad y reusabilidad, señalan que la superficie de ataque se incrementa cuando el componente es más reusable. Apuntan, además, que las dependencias negativas fueron identificadas en componentes aislados del software y que las mediciones tomadas de seguridad y mantenibilidad, así como las reuniones periódicas y las heurísticas de MEDUSAS, les permitieron resolver los conflictos.

Tabla 4. Interacciones entre mantenibilidad y seguridad, según una organización

Mantenibilidad →	Reusabilidad	Analizabilidad	Modificabilidad	Modularidad	Facilidad de prueba
Seguridad					
Protección contra ataques	+	+		+	+
Autenticidad		+			
Confidencialidad	-	+		+	+
Conformidad		+			
Detección de ataques	-	+	-	+	+
Disponibilidad				+	
Integridad				+	+
No repudio		+			
Trazabilidad		+			+

La tabla de interacciones entre seguridad y mantenibilidad generada con el proceso descrito en la sección tres, muestra que la interacción ha sido calificada por unos autores independiente (O) y por otros, positiva (+). Además, las referencias bibliográficas consideradas no abordan interacciones entre las subcaracterísticas de mantenibilidad ni seguridad. Puesto que decidimos tomar la opción más pesimista, toda la tabla tiene calificación “O”. Lo que significa que no se han identificado efectos entre subcaracterísticas. La tabla generada coincide con una de las encuestas de este estudio exploratorio. Sin embargo, la tabla 4, que muestra la percepción de los participantes de otra empresa, tiene diferencias significativas con la tabla de interacción obtenida.

Como se señaló al inicio de la sección 3, el propósito de las tablas de interacción es tener una calificación inicial de las interacciones entre características de calidad. La inconsistencia entre distintas referencias es un elemento que requiere estudios más detallados para establecer las condiciones bajo las cuales se mantienen dichas interacciones. Se necesita información de la argumentación y evidencias empíricas para antes de hacer una modificación a dichas tablas. Estas podrán modificarse conforme

las lecciones aprendidas en el proyecto y si las evidencias señalan que las interacciones conservan un patrón distinto al establecido, adjuntar la información contextual pertinente. Los resultados de las encuestas, de carácter exploratorio, muestran que los profesionales de la informática encuestados tienen distintas perspectivas de las interacciones entre características de calidad. Necesitamos más información contextual y de medición para estar en condiciones de combinar los resultados de la encuesta con lo que existe en la literatura. De hecho, es un área que requiere de más estudios para entender el impacto de las interacciones en el proyecto software [26].

5 Conclusiones

Las características de calidad del producto software desempeñan un papel relevante para que el proyecto sea exitoso. Entre ellas existen interacciones que los ingenieros de software deberían identificar y resolver aquellas que son conflictivas. La identificación temprana de las interacciones negativas puede disminuir los costes asociados en la especificación de requisitos inalcanzables, tanto técnicamente como por cuestión de esfuerzo y costes.

Una forma de aproximarse al problema de las interacciones negativas entre características de calidad es el uso de tablas de interacción. En ellas se califican las dependencias como positivas, negativas o independientes. Existe poca información respecto de los criterios empleados para construirlas y los autores señalan que se han basado en la literatura, encuestas y experimentos.

En este artículo se ha propuesto una técnica para elaborar tablas de interacción base considerando modelos de calidad del producto software. Dado que existen pocos estudios relacionados con las interacciones, el procedimiento propuesto considera cómo derivar una calificación a partir de la estructura jerárquica de los modelos de calidad. También se realizó una encuesta para conocer la percepción de los profesionales de la informática respecto de la percepción de las interacciones entre características de calidad y la gestión de las mismas. El resultado muestra que una de las empresas coincide en percepción con la tabla de interacción construida con el proceso propuesto en este artículo. Sin embargo, los resultados de las dos empresas restantes invitan a continuar con un estudio más profundo de este tema.

Como trabajo futuro estamos considerando los procesos asociados a la identificación de interacciones negativas en el proyecto software. En particular estamos investigando los factores relevantes que contribuyen a ese tipo de interacción y su relación con las distintas etapas del ciclo de vida del software. Con esta información se podrían actualizar las tablas de interacción con base en los resultados de los proyectos que una organización emprenda. También, se necesita investigar el impacto que podría tener la aplicación de las interacciones positivas en el proyecto software. Adicionalmente, se necesitan establecer los mecanismos más apropiados para generalizar los resultados de cada proyecto en el contexto de la empresa. Actualmente se está construyendo un marco de procesos cuyo objetivo es facilitar la gestión de las interacciones entre características de calidad, en el cual se proporcionarán procedimientos para

identificar el conflicto en el ámbito del proyecto y consolidar los datos empíricos en tablas de interacción para su uso en futuros proyectos.

Agradecimientos. Este trabajo ha sido financiado por los proyectos GEODAS-BC (Ministerio de Economía y Competitividad y Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01) y PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN y Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01).

Referencias

1. Barbacci, M., et al. *Quality Attributes (CMU/SEI-95-TR-021)*. 1995 [cited 2012; Available from: <http://www.sei.cmu.edu/library/abstracts/reports/95tr021.cfm>].
2. CMMI, P.T. *CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033)*. 2010 [cited 2012; Available from: <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>].
3. ISO, *ISO/IEC FCD 25010: Systems and software engineering - system and software product quality requirements and evaluation (SQaURE) - System and software quality models*. 2010.
4. ISO, *ISO/IEC 9126 Software Engineering - Product Quality," in Part 1 - Quality model*. 2001.
5. Dromey, R.G., *Model for software product quality*. IEEE Transactions on Software Engineering, 1995. **21**(2): p. 146-162.
6. McCall, J., P. Richards, and W. GF, *Factors in software quality. Concept and definitions of software quality: final technical report* N.T.I.S. RAD-TR-77-369, Editor. 1977: Springfield.
7. Boehm, B.W., J.R. Brown, and M. Lipow, *Quantitative evaluation of software quality*, in *Proceedings of the 2nd international conference on Software engineering*. 1976, IEEE Computer Society Press: San Francisco, California, USA. p. 592-605.
8. Boehm, B. and H. In, *Identifying quality-requirement conflicts*. IEEE Software, 1996. **13**(2): p. 25-35.
9. Paech, B. and D. Kerkow, *Non-Functional Requirements Engineering - Quality is essential*, in *10th Anniversary International Workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'04)*, B. Regnell, E. Kamsties, and V. Gervasi, Editors. 2004, Universität Duisburg-Essen p. 237-250.
10. Robinson, W.N., S.D. Pawlowski, and V. Volkov, *Requirements Interaction Management*. *ACM Computing Surveys*, 2003. **35**(2): p. 132-190.
11. Barney, S., et al., *Software quality trade-offs: A systematic map*. Information and Software Technology, 2012. **54**(7): p. 651-662.
12. Berntsson Svensson, R., T. Gorschek, and B. Regnell, *Quality requirements in practice: An interview study in requirements engineering for embedded systems*, in *Requirements Engineering: Foundation for Software Quality*. 2009. p. 218-232.
13. Cysneiros, L.M. and J.C.S.P. Leite, *Nonfunctional Requirements: From Elicitation to Conceptual Models*. IEEE Transactions on Software Engineering 2004. **30**: p. 328-349.
14. Kazman, R. and M. Klein, *Performing architecture tradeoff analysis*. International Software Architecture Workshop, Proceedings, ISAW, 1998: p. 85-88.

15. Chung, L. and B.A. Nixon, *Dealing with non-functional requirements: three experimental studies of a process-oriented approach*, in *17th international conference on Software engineering* 1995. p. 25-37.
16. Myopoulos, J., L. Chung, and B. Nixon, *Representing and using nonfunctional requirements: A process-oriented approach*. IEEE Transactions on Software Engineering, 1992. **18**(6): p. 483-497.
17. Chung, L. and J.C.S. Do Prado Leite, *On non-functional requirements in software engineering*, in *Conceptual Modeling: Foundations and Applications*, A. Borgida, et al., Editors. 2009, Springer Berlin Heidelberg. p. 363-379.
18. Landes, D. and R. Studer, *The Treatment of Non-Functional Requirements in MIKE*, in *Proceedings of the 5th European Software Engineering Conference*. 1995, Springer-Verlag. p. 294-306.
19. Berander, P., et al. *Software Quality Attributes and trade-offs*. 2005 [cited 2012 Nov, 2012]; Available from: http://www.uio.no/studier/emner/matnat/ifi/INF5180/v10/undervisningsmateriale/reading-materials/p10/Software_quality_attributes.pdf.
20. Mairiza, D., D. Zowghi, and N. Nurmuliani, *Towards a catalogue of conflicts among non-functional requirements*, in *Fifth International Conference on Evaluation of Novel Approaches to Software Engineering*. 2010. p. 20-29.
21. Henningsson, K. and C. Wohlin, *Understanding the Relations between Software Quality Attributes - A Survey Approach*, in *12th International Conference for Software Quality*. 2002: Ottawa, Canada. p. 1-12.
22. Dahlstedt, Å. and A. Persson, *Requirements Interdependencies: State of the Art and Future Challenges*, in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Editors. 2005, Springer Berlin Heidelberg. p. 95-116.
23. Egyed, A. and P. Grünbacher, *Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help*. IEEE Software, 2004. **21**(6): p. 50-58.
24. Zulzalil, H., et al., *A Case Study to Identify Quality Attributes Relationships for Web-based Applications*. IJCSNS International Journal of Computer Science and Network Security, 2008. **8**(11): p. 215-220.
25. Sadana, V. and X.F. Liu, *Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements*, in *Computer Software and Applications Conference. COMPSAC 2007. 31st Annual International I.C. Society*, Editor. 2007. p. 215-218.
26. Svahnberg, M. and K. Henningsson, *Consolidating Different Views of Quality Attribute Relationships*, in *WoSQ'09*. 2009. p. 46-50.
27. Kitchenham, B., et al., *The SQUID approach to defining a quality model*. Software Quality Journal, 1997. **6**(3): p. 211-233.
28. AQC. *MEDUSAS: Mejora y Evaluación del Diseño, Usabilidad, Seguridad y Mantenibilidad del Software*. 2009; Available from: <http://www.alarcosqualitycenter.com/index.php/i-mas-d/proyectos/medusas>.
29. García-Mireles, G.A., et al., *A framework to support quality trade-offs from a process-based perspective*, in *EuroSPI2013, in press*. 2013.