

Análisis de las soluciones guiadas por búsqueda para el problema de selección de requisitos

Isabel María del Águila, José del Sagrado y Alfonso Bosch

Departamento de Informática, Universidad of Almería,
Crtra. de la Playa s/n, 04120 Almería, Spain
{imaguila, jsagrado, abosch}@ual.es

Resumen La Ingeniería del Software guiada por búsqueda propone la utilización de algoritmos de optimización en los problemas de Ingeniería del Software. Este área de investigación ha sido muy prolífica durante los últimos años, formulando y dando soluciones a problemas de optimización multiobjetivo, como el de la selección de requisitos. Pero cuando los desarrolladores de software tienen que tomar la última decisión acerca de cuál es el conjunto de requisitos a implementar, de entre las soluciones ofrecidas por los métodos multiobjetivo, necesitan revisar y analizar una gran cantidad de datos. Para ayudar en este proceso de toma de decisiones, este trabajo propone un conjunto de indicadores de calidad que facilitan el análisis del problema a nivel de requisitos, soluciones y clientes. El proceso de análisis utilizado combina estos indicadores de calidad con resúmenes estadísticos y visualización de datos. El caso de estudio abordado muestra la forma en la que el proceso de análisis ayuda en la definición de criterios de selección de soluciones, apoyándose en el estudio y visualización de los indicadores de calidad propuestos.

Keywords: Ingeniería del Software guiada por búsqueda, problema de selección de requisitos, óptimos de Pareto, procesos de toma de decisiones

1 Introducción

La complejidad computacional de los problemas de Ingeniería del Software ha hecho que los equipos de desarrollo de software tengan que aplicar algoritmos no deterministas para obtener soluciones aceptables a los problemas formulados dentro de este dominio. Esta circunstancia fue el origen de Ingeniería del Software guiada por búsqueda (Search-Based Software Engineering - SBSE). Este campo de investigación propone la aplicación de algoritmos de optimización basados en búsqueda para resolver los problemas de Ingeniería del Software [14].

Los investigadores en este área de conocimiento han sido muy prolíficos durante los últimos años [15, 12]. Una vez que un problema de Ingeniería del Software se formula como un problema de optimización, puede ser resuelto aplicando distintas técnicas metaheurísticas. A pesar de los éxitos alcanzados por los algoritmos propuestos, sigue siendo un reto hacer realmente útil esta disciplina

dentro del mundo comercial de la Ingeniería del Software. De hecho los trabajos publicados no indican claramente como transferir los métodos propuestos a los proyectos software en desarrollo. Por ejemplo, si nos centramos en uno de los problemas básicos de esta disciplina, el problema de selección de requisitos o problema de la siguiente versión (Next Release Problem (NRP) [1]), que propone un modelo de decisión para la priorización de requisitos basado en los beneficios de los clientes y el coste de su desarrollo; aunque hay numerosos trabajos que ofrecen buenas soluciones utilizando distintas aproximaciones [17], sólo unos pocos transfieren sus resultados en una herramienta software de gestión o modelado de proyectos software que facilite su incorporación dentro de los procesos del ciclo de vida de la Ingeniería del Software [22, 7, 20].

En tareas de decisión donde aparecen objetivos contradictorios los métodos de optimización multiobjetivo son una buena alternativa. Este es el caso que se presenta cuando el equipo de desarrollo tiene que decidir cuál es el subconjunto de requisitos que tiene que contemplarse en las siguientes etapas del desarrollo de la siguiente versión de una aplicación software. La idea es combinar los métodos computacionales con la experiencia de los expertos humanos para obtener mejores conjuntos de requisitos de los que se producirían sólo mediante el juicio experto. La optimización multiobjetivo para este problema, introducida por [25], ha sido tratada desde distintos puntos de vista, bien mediante la interacción entre requisitos y restricciones de implementación [21], o considerando múltiples objetivos como el coste-beneficio [25], o empleando distintas medidas de evaluación [10, 11], o aplicando distintos algoritmos genéticos NSGA-II, MOCell and PAES [9], o finalmente aplicando algoritmos de colonia de hormigas [19]. Sin embargo, sigue siendo un reto definir como el ingeniero del software ha de tratar la información ofrecida por estos algoritmos para decidir cuáles serán finalmente los requisitos cubiertos en la siguiente iteración del desarrollo del proyecto software. En la práctica no está claro como enlazar las soluciones basadas en búsqueda con los procesos de Ingeniería del Software. Es decir, no hay un método claro para seleccionar cuál de las soluciones mostradas en los frentes de Pareto debe ser seleccionada como la más rentable para el proyecto, acorde con la situación actual del proyecto.

En este punto es donde los ingenieros del software tienen que tratar y revisar datos de distintas fuentes, sus decisiones han de basarse en el riesgo asociado con cada requisito, el coste de desarrollarlo, los beneficios que supondrá, fechas límite, interrelaciones, restricciones de recursos, o algunos otros factores que afecten a las tareas de gestión. [16]. Algunos autores utilizan técnicas de aprendizaje y minería de datos para estudio de las soluciones ofrecidas por herramientas de planificación de versiones [8]. Pero, los ingenieros del software prefieren tener a su disposición métricas o indicadores de calidad que puedan avalarles y ayudarles en sus decisiones, sobre este aspecto [3].

Este trabajo busca una solución al reto de la interconexión de los métodos propuestos por la Ingeniería del Software guiada por búsqueda y el mundo de la industria del desarrollo de software mediante el análisis de los datos ofrecidos en estos métodos, centrándose la definición de una serie de indicadores o métricas

de calidad para el problema de la selección de requisitos. Este análisis se ha implementado utilizando el marco metodológico ofrecido por R [5]. Los indicadores de calidad definidos proporcionan, de un sólo vistazo, el resumen de los datos, tanto del propio problema, como de las soluciones obtenidas con los algoritmos guiados por búsqueda.

El trabajo se estructura en cinco secciones. En la sección 2 se describe y define formalmente el problema de selección de requisitos. La toma de decisiones en este ámbito se aborda, desde el punto de vista de la Ingeniería del Software, en la sección 3. En esta sección también se formulan las métricas que se utilizan para fundamentar el proceso de toma de decisiones. Este proceso se ilustra sobre un caso de estudio en la sección 4. Por último, la sección 5 recoge las conclusiones.

2 Formulación del problema

Desde los años 90, la Ingeniería de los Requisitos se considera como un proceso esencial en el desarrollo de sistemas y especialmente en el desarrollo de sistemas software. Las decisiones tomadas durante estas primeras etapas del desarrollo son cruciales y difíciles, puesto que se maneja información incompleta, imprecisa y sujeta a numerosos cambios [18]. Además, estas decisiones son fundamentales, puesto que tendrán una gran influencia a lo largo del ciclo de vida completo del sistema software [13].

La planificación de versiones, es decir, la definición de cómo va a ir evolucionando un producto software a lo largo de su vida útil, es una tarea fundamental cuando se emplean enfoques incrementales para la construcción del producto software. Estas decisiones suelen tener que considerar varios objetivos diferentes e incluso conflictivos, como las interacciones entre las características candidatas, las preferencias de los clientes o las limitaciones en los recursos. En otras palabras, la planificación de versiones, en general, implica la optimización basada en múltiples criterios [25, 24].

Es en este campo de la Ingeniería del Software guiada por búsqueda se han realizado numerosos trabajos que resuelven el problema de seleccionar cuáles de las características o requisitos deben ser cubiertos por el producto software que se está desarrollado. Los primeros autores que formularon el problema de la selección de requisitos (NRP) fueron Bagnall et al.[1]. La definición completa del problema de selección de requisitos, incluyendo todas las interacciones entre requisitos, aparece en [19] y la resumimos a continuación.

Sea $\mathbf{R} = \{r_1, r_2, \dots, r_n\}$ el conjunto de requisitos que aún no han sido diseñados e implementados. Estos requisitos representan mejoras o nuevas funcionalidades al sistema actual propuestas por un conjunto de m clientes o partes interesadas, $\mathbf{C} = \{c_1, c_2, \dots, c_m\}$. Estas partes interesadas no son iguales, en términos de relevancia, para el proyecto. Cada parte o cliente i tiene un peso asociado w_i , que mide su importancia dentro del proyecto: $\mathbf{W} = \{w_1, w_2, \dots, w_m\}$.

Cada requisito $r_j \in \mathbf{R}$ tiene un coste o esfuerzo e_j , que representa el esfuerzo que se tiene que emplear en su desarrollo: $\mathbf{E} = \{e_1, e_2, \dots, e_n\}$. El valor del requisito r_j para el cliente i se representa con un valor v_{ij} . Cuanto más alto es el

valor de v_{ij} mayor es la prioridad que da el cliente i al requisito r_j . Un valor cero para v_{ij} representa que el cliente i no ha sugerido el requisito r_j . Estos valores de importancias v_{ij} pueden estructurarse en una matriz $m \times n$.

Tabla 1: Asignación del nivel de prioridad, esfuerzo e interacciones funcionales.

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}	r_{20}	w_i
c_1	4	2	1	2	5	5	2	4	4	4	2	3	4	2	4	4	4	1	3	2	1
c_2	4	4	2	2	4	5	1	4	4	5	2	3	2	4	4	2	3	2	3	1	4
c_3	5	3	3	3	4	5	2	4	4	4	2	4	1	5	4	1	2	3	3	2	2
c_4	4	5	2	3	3	4	2	4	2	3	5	2	3	2	4	3	5	4	3	2	3
c_5	5	4	2	4	5	4	2	4	5	2	4	5	3	4	4	1	1	2	4	1	4
Ef.	1	4	2	3	4	7	10	2	1	3	2	5	8	2	1	4	10	4	8	4	

$r_3 \odot r_{12} \quad r_{11} \odot r_{13} \quad r_4 \Rightarrow r_8 \quad r_4 \Rightarrow r_{17} \quad r_8 \Rightarrow r_{17} \quad r_9 \Rightarrow r_3 \quad r_9 \Rightarrow r_6 \quad r_9 \Rightarrow r_{12} \quad r_9 \Rightarrow r_{19} \quad r_{11} \Rightarrow r_{19}$

La satisfacción, s_j , o valor añadido por la inclusión de r_j en la siguiente versión del software, se mide como la suma ponderada de los valores de importancia, $s_j = \sum_{i=1}^m w_i * v_{ij}$.

El conjunto de satisfacciones se denota $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$. Los requisitos no son independientes unos de otros, es habitual que presenten interacciones que se clasifican en dos categorías. El primer grupo son las interacciones funcionales: *implicación*, *combinación* y *exclusión*. El segundo grupo son las interacciones que implican un cambio en el valor que aporta o en la cantidad de recursos necesarios para ser abordados: *basadas-en-beneficio* and *basadas-en-coste*. Las interacciones funcionales se representan con un grafo $G = (\mathbf{R}, \mathbf{I}, \mathbf{J}, \mathbf{X})$ donde:

- \mathbf{R} (el conjunto de requisitos) es el conjunto de nodos
- $\mathbf{I} = \{(r_i, r_j) \mid r_i \Rightarrow r_j\}$ cada par $(r_i, r_j) \in \mathbf{I}$ es una interacción de implicación que se representa con un enlace dirigido $r_i \rightarrow r_j$
- $\mathbf{J} = \{(r_i, r_j) \mid r_i \odot r_j\}$ cada par $(r_i, r_j) \in \mathbf{J}$ es una interacción que se representa con un enlace dirigido doble $r_i \leftrightarrow r_j$
- $\mathbf{X} = \{(r_i, r_j) \mid r_i \oplus r_j\}$ cada par $(r_i, r_j) \in \mathbf{X}$ es una interacción de exclusión que se representa con un enlace tachado.

Por ejemplo, sea el conjunto de requisitos $\mathbf{R} = \{r_1, r_2, \dots, r_{10}\}$ y el conjunto de interacciones funcionales $\mathbf{I} = \{(r_1, r_3), (r_1, r_6), (r_2, r_4), (r_2, r_5), (r_4, r_6), (r_5, r_7), (r_7, r_8), (r_7, r_9), (r_8, r_{10}), (r_9, r_{10})\}$, $\mathbf{J} = \{(r_3, r_4)\}$, $\mathbf{X} = \{(r_4, r_8)\}$. Se pueden representar en forma de grafo tal como se muestra en la Figura 1.

El segundo grupo de interacciones se modela mediante un par de $n \times n$ matrices simétricas en las que los elementos de la diagonal son cero:

- ΔS , cada $\Delta s_{ij} \in \Delta S$ es el incremento o decremento de s_j cuando r_i y r_j se incluyen en la misma versión,
- ΔE , cada $\Delta e_{ij} \in \Delta E$ es el incremento o decremento e_j cuando r_i y r_j se incluyen simultáneamente,

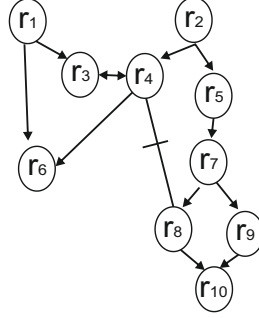


Fig. 1: Grafo de interacciones funcionales, $G = (\mathbf{R}, \mathbf{I}, \mathbf{J}, \mathbf{X})$.

Cuando se especifica la siguiente versión de software debemos seleccionar el conjunto de requisitos $\hat{\mathbf{R}}$ incluidos en \mathbf{R} , que maximizan la satisfacción y minimizan el esfuerzo de desarrollo. La satisfacción y el esfuerzo de desarrollo de la siguiente versión se obtienen respectivamente:

$$\text{sat}(\hat{\mathbf{R}}) = \sum_{j \in \hat{\mathbf{R}}} s_j \quad \text{eff}(\hat{\mathbf{R}}) = \sum_{j \in \hat{\mathbf{R}}} e_j \quad (1)$$

donde j es la abreviatura del requisito r_j . Para cada nueva versión existe un límite de coste o recursos B , que no debe superarse. Por tanto, el problema de selección de requisitos para la siguiente versión del software se puede formular como un problema de optimización:

$$\text{maximizar } \text{sat}(\hat{\mathbf{R}}), \quad \text{minimizar } \text{eff}(\hat{\mathbf{R}}) \quad (2)$$

sujeto a la restricción $\sum_{j \in \hat{\mathbf{R}}} e_j \leq B$ dado un determinado límite de esfuerzo B , donde $\hat{\mathbf{R}}$ cumple con las interacciones funcionales entre los requisitos. Por tanto, dos objetivos conflictivos, maximizar la satisfacción de los clientes o personas interesadas y minimizar el esfuerzo de desarrollo, se optimizan al mismo tiempo dentro de un determinado límite de esfuerzo y manteniendo las interacciones funcionales. La solución a este problema consiste en un conjunto de soluciones conocido como conjunto óptimo de Pareto [4, 6, 23].

3 Toma de decisiones para el equipo de desarrollo

En todas las etapas de un proyecto de desarrollo y evolución de software es necesario tomar decisiones. El impacto en la calidad del software de las buenas decisiones es mucho mayor en las etapas iniciales del ciclo de vida de los proyectos. Por tanto las decisiones en las tareas relacionadas con los requisitos serán de gran importancia para el éxito final del proyecto software [3].

La priorización de requisitos es un proceso de toma de decisiones que permite a los gestores del proyecto centrar su atención en aquellas características del producto que añadirán más valor al beneficio global del proyecto. Esta tarea de

gestión se emplea para determinar cuáles de los requisitos candidatos deberán incluirse en una determinada versión. Los requisitos pueden evaluarse utilizando diversos criterios, [16]:

- *Valor*: Se basa en el estudio coste-beneficio relativo al valor de los requisitos. Este enfoque se utiliza para mejorar la solución existente o cuando se está contruyendo el proyecto de forma incremental.
- *Riesgo*: Los requisitos seleccionados son aquellos que presentan el mayor riesgo de fallo. Se decide tratar estos requisitos primero para asegurar que si falla el proyecto, el fallo se producirá antes de realizar una gran inversión.
- *Dificultad de desarrollo*: Los requisitos más fáciles de implementar se seleccionan primero. Este enfoque se utiliza cuando se trata de proyectos piloto de desarrollos novedosos.
- *Posibilidades de éxito*: Es común que en proyectos polémicos e incipientes la selección se centre en requisitos en los que es fácil y rápido conseguir éxito.
- *Normativa y política de la organización*: En este caso se priorizan los requisitos que deben implementarse para cumplir con la normativa o las políticas de la empresa u organización.
- *Interacciones entre los requisitos*: Un requisito candidato puede no tener un valor alto por sí sólo, pero sí tener interacciones con otros de alta prioridad.
- *Acuerdo con las personas interesadas*: Se utiliza en combinación con alguno de los demás factores, busca llegar a acuerdos con las personas interesadas llegando a compromisos.
- *Urgencia*: Se priorizan los requisitos basándose en el factor tiempo.

La inclusión de todas estas dimensiones en el problema de optimización podría complicar en exceso los algoritmos metaheurísticos. Como alternativa, planteamos un proceso de selección o priorización en diversas etapas [20]. La primera etapa se ocupa de la recopilación de los requisitos y los datos proporcionados por todos los interesados en el proyecto, de cara a la formulación del problema para el proyecto concreto. La segunda se centra en la ejecución de las técnicas y algoritmos de búsqueda, generando como salida el conjunto de soluciones no dominadas para el problema. La etapa final consiste en el análisis de estas soluciones teniendo en cuenta algunos de los factores citados en los párrafos anteriores. La tarea de análisis se realiza, principalmente, apoyándose en la experiencia de los ingenieros del software, que son los encargados de tomar la decisión final. Estos decisores deberían ayudarse con indicadores de calidad, calculados empíricamente, que pudieran recoger todos los factores a tener en cuenta, a la hora de tomar su decisión. Este tipo de medidas aparecen en ciertos trabajos de la Ingeniería de Requisitos [8, 2], donde se utilizan métricas para analizar los resultados de las tareas de priorización; el flujo de trabajo de estas etapas se muestra en la Figura 2.

Sin embargo tratar con esta gran cantidad de datos puede ser difícil y, habitualmente, es necesario recurrir a la abstracción. En este trabajo proponemos un conjunto de indicadores de calidad para el problema de la siguiente versión, con el objetivo de permitir que, de un vistazo, los ingenieros del software fun-

damenten sus decisiones en estas medidas. Los indicadores de calidad se definen en tres niveles: requisito, solución y persona interesada o cliente.

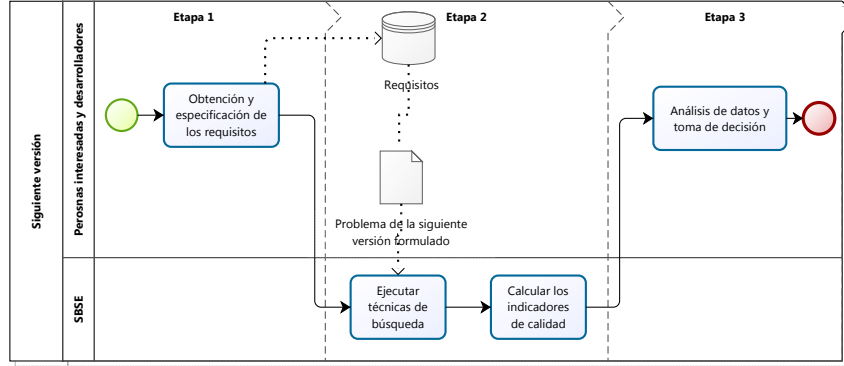


Fig. 2: Etapas en la toma de decisiones de la priorización de requisitos

A nivel de requisitos, además del esfuerzo de desarrollo y la satisfacción, se definen dos indicadores de calidad:

- *Productividad*: es el ratio entre la satisfacción y el esfuerzo. Da un valor del rendimiento que produciría la inclusión de un requisito (r_j) en la solución:

$$pro(r_j) = s_j / e_j. \quad (3)$$

- *Contribución*: es la parte de la satisfacción s_j asociada a un requisito concreto r_j que aporta o corresponde al cliente c_i . Se define como :

$$con(r_j, c_i) = (w_i \cdot v_{ij}) / s_j. \quad (4)$$

Estas medidas se extienden a las soluciones del problema como sigue

- *Productividad*: es el ratio entre satisfacción y esfuerzo asociado a una solución. Para una solución $\mathbf{S} \subseteq \mathbf{R}$, se define como:

$$pro(\mathbf{S}) = sat(\mathbf{S}) / eff(\mathbf{S}). \quad (5)$$

- *Contribución*: es la parte de la satisfacción $sat(\mathbf{S})$ asociada a una solución $\mathbf{S} \subseteq \mathbf{R}$ que corresponde a un cliente c_i . Se define como:

$$con(\mathbf{S}, c_i) = \frac{\sum_{j \in \mathbf{S}} w_i \cdot v_{ij}}{sat(\mathbf{S})}. \quad (6)$$

Además, desde el punto de vista del ingeniero del software se emplea otra métrica ligada a los clientes:

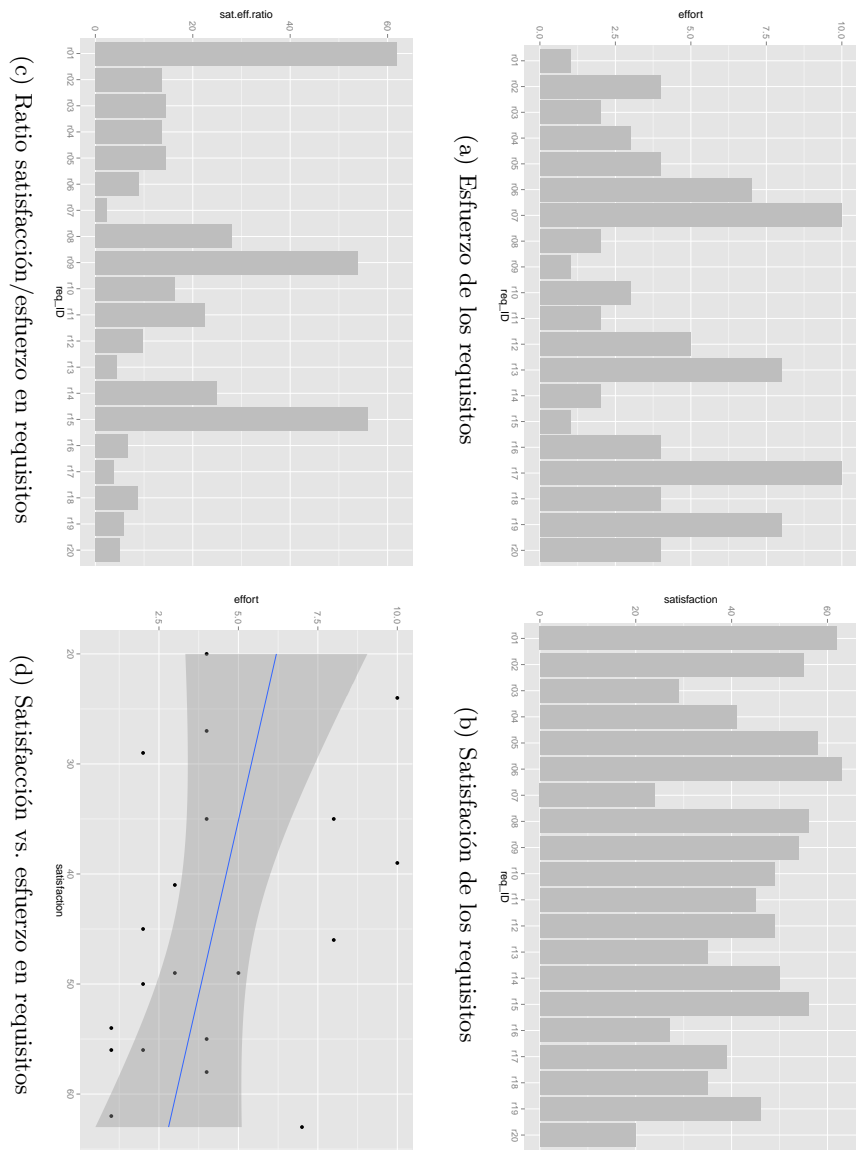


Fig. 3: Distribución del esfuerzo y la satisfacción en el conjunto de requisitos.

- *Cobertura*: mide hasta que punto una solución cubre las propuestas realizadas por un determinado cliente. Se define como:

$$cob(\mathbf{S}, c_i) = \frac{\sum_{j \in \mathbf{S}} v_{ij}}{\sum_{j=1}^n v_{ij}}. \quad (7)$$

4 Análisis de un caso

En esta sección se analizan las soluciones no dominadas que encuentran los algoritmos guiados por búsqueda para resolver el problema de selección de requisitos. La idea es ofrecer soporte a los ingenieros del software, para ayudarles en la tarea de seleccionar una de las soluciones, que se convertirá en el objetivo de desarrollo de la siguiente versión del software del proyecto. Esta ayuda conjuga la información sobre los requisitos de partida, las soluciones no dominadas encontradas y los clientes o personas interesadas que propusieron los requisitos. El análisis del ingeniero del software se basa en estos tres aspectos que se visualizarán de diferentes formas.

Tabla 2: Resumen de las estadísticas

	asociadas a los requisitos			asociadas a las soluciones		
	esf.	sat.	prod.	esf.	sat.	prod.
Mínimo	1.00	20.00	2.40	0.00	0.00	0.00
1° Cuartil	2.00	35.00	6.50	14.50	312.50	12.88
Mediana	4.00	47.50	13.71	37.00	544.00	14.70
Media	± 4.25	± 44.65	± 18.80	± 36.56	± 514.70	± 18.70
des.stan.	2.88	12.89	18.07	23.97	244.63	11.60
3° Cuartil	5.50	55.25	23.12	54.50	710.00	20.34
Máximo	10.00	63.00	62.00	85.00	893.00	62.00

El caso de estudio considerado, corresponde al conjunto de requisitos utilizado en [18], así como sus interacciones. Ambos se muestran en la Tabla 1. El análisis del ingeniero del software empieza por los requisitos. Además de las estadísticas (ver Tabla 2), se puede utilizar un panel de gráficos que resume de forma visual el esfuerzo de los requisitos, su satisfacción y el ratio satisfacción/esfuerzo que representaría la métrica de productividad (ver Fig. 3). La satisfacción de los requisitos respecto al esfuerzo presenta una correlación negativa -0.353 . El valor negativo sugiere que la satisfacción decrece conforme el esfuerzo crece, Esta relación se ve más claramente en la Fig. 3(d), que muestra el gráfico de dispersión junto con la regresión lineal para esas medidas. La banda sombreada muestra el error de la desviación estándar para la estimación lineal. Cabe destacar que es muy ancha debido a la dispersión de los datos.

Una vez que el ingeniero del software ha obtenido una visión de lo que pasa a nivel de requisitos, es el momento de pasar a analizar las soluciones. Igual que con los requisitos, se parte del resumen de los datos estadísticos (ver Tabla 2) junto con la representación gráfica de esta información. El panel de gráficos para las soluciones se muestra en la Fig. 4. Está compuesto por el conjunto de soluciones no dominadas (el frente de Pareto), la productividad de las soluciones, el esfuerzo, la satisfacción y el esfuerzo frente a la satisfacción. La satisfacción de las soluciones con respecto al esfuerzo presenta una correlación positiva alta

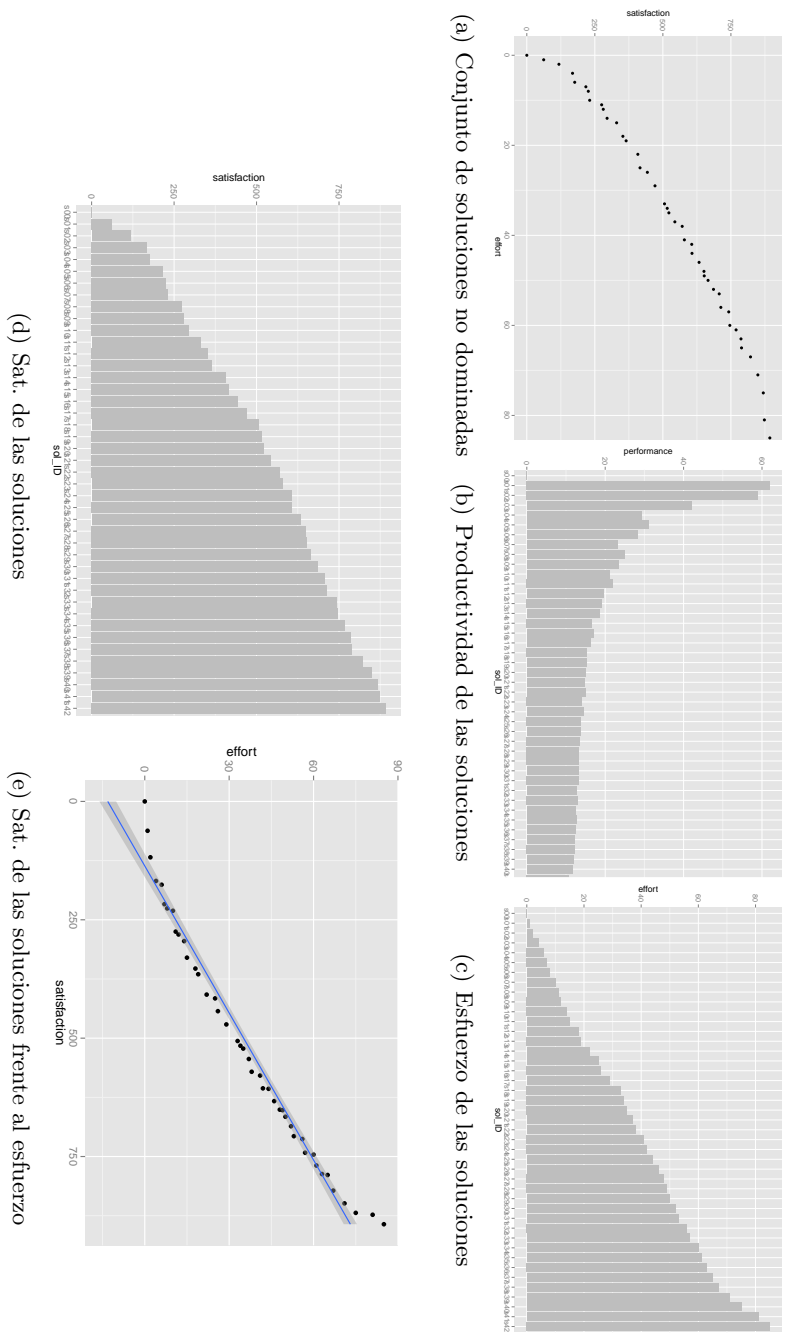


Fig. 4: Esfuerzo y la satisfacción del conjunto de soluciones no dominadas.

0.986. Es decir, el esfuerzo y la satisfacción crecen conforme más requisitos se van considerando en la solución. Esta relación se muestra en la Fig. 4(e), donde se representa el gráfico de dispersión y la regresión lineal entre estas medidas. Nótese que la banda del error de la desviación estándar relativa a la estimación lineal es mucho más delgada que en el caso de los requisitos, debido a la densidad de los datos. Al mismo tiempo, la productividad de las soluciones disminuye (véase Fig. 4(b)), pero este decremento se ralentiza a partir de la solución s_{12} que es cuando la productividad de la solución está en torno a 20. La productividad de las soluciones decrece un 80.77% desde s_{01} a s_{12} , y el 19.23% desde s_{12} a s_{42} . Por tanto, desde el esfuerzo 18 (esfuerzo de s_{12}) hasta el esfuerzo 85 (esfuerzo de s_{42}), las soluciones muestran una productividad similar y este límite de esfuerzo podría ser utilizado como criterio de decisión para seleccionarlo como valor a considerar en la siguiente versión.

La influencia que las personas interesadas o clientes tienen sobre los requisitos y las soluciones se muestra en la Fig. 5. En primer lugar, la satisfacción a nivel de requisitos se analiza en las figuras Fig.5(a), en la que la satisfacción de cada requisito se representa utilizando la contribución de cada cliente, y en la Fig.5(c), donde se visualiza la valoración de la satisfacción de los requisitos hecha por cada cliente. En segundo lugar, la satisfacción a nivel de la solución se analiza en la Fig.5(b), donde se muestra la satisfacción de la solución descompuesta a nivel de cliente y en la Fig. 5(d) se representa la valoración de las satisfacciones de las soluciones para cada cliente, calculada como a la agregación de sus valoraciones de satisfacción de los requisitos. La satisfacción de las personas interesadas aumenta conforme se incrementa el esfuerzo de las soluciones.

El ingeniero del software no sólo analiza la información anterior, sino que también tiene que seleccionar una solución (o un pequeño conjunto de soluciones), de entre el conjunto de soluciones no dominadas, que alcance una alta satisfacción de los cliente con un cierto límite de esfuerzo. Es decir, para un valor dado de esfuerzo los ingenieros del software preferirán las soluciones con una alta satisfacción (y por tanto alta productividad). También es importante que las soluciones alcancen un mínimo nivel de cobertura de todos los clientes. Este límite puede ser utilizado como otro criterio de decisión a considerar a la hora de realizar la decisión final. Retomando la solución s_{12} , donde se ha identificado un punto de inflexión, el mínimo nivel alcanzado en la cobertura se detecta para el cliente 4, con un valor de 0.31. Podemos asumir que el equipo de desarrollo quiere alcanzar para cada cliente al menos un 50% de cobertura, es decir un valor de 0.5. En ese caso, s_{18} cumple con el criterio. A partir de aquí y de cara a considerar otras soluciones alternativas, se puede volver a revisar el rendimiento de las soluciones (Fig. 4(b)) y buscar soluciones de productividad similar. Como resultado de este proceso aparecen la solución s_{22} (con valor de productividad similar a s_{18}) y s_{31} , s_{33} con una productividad ligeramente menor (en torno 13), pero con una mayor cobertura para los clientes 0.77 (ver Tabla 3).

Finalmente, las soluciones seleccionadas se caracterizan en términos de los requisitos que las componen. Esto permite identificar el núcleo de los requisitos que se comparten en todas ellas y que mantienen las dependencias entre los requisitos.

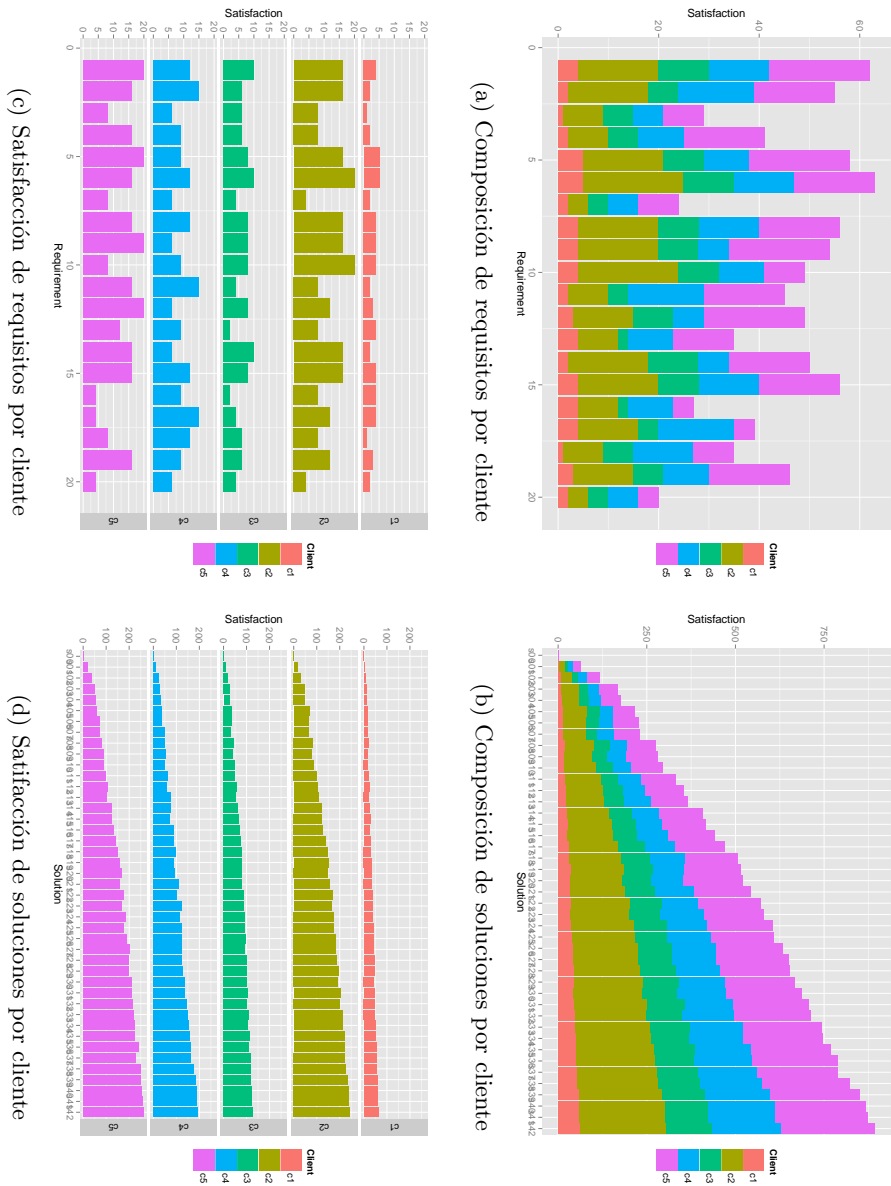


Fig. 5: Satisfacciones por cliente en los conjuntos de requisitos y de soluciones.

En este caso, el núcleo de estos requisitos es el conjunto $\{r_1, r_2, r_5, r_{10}, r_{14}, r_{15}\}$. Estos requisitos se caracterizan por un bajo esfuerzo y su alta satisfacción, es decir, una buena productividad.

Tabla 3: Productividad y cobertura de clientes para soluciones seleccionadas.

Solución	Prod	Cobert. c_1	Cobert. c_2	Cobert. c_3	Cobert. c_4	Cobert. c_5	Esf.
s_{18}	15.33	0.50	0.61	0.63	0.51	0.56	33
s_{22}	15.03	0.60	0.69	0.69	0.52	0.67	38
s_{31}	13.34	0.76	0.84	0.83	0.71	0.80	53
s_{33}	13.02	0.77	0.87	0.88	0.77	0.83	57

5 Conclusiones

Este trabajo estudia cómo los ingenieros del software procesan y analizan el conjunto de soluciones no dominadas ofrecidas por los métodos de Ingeniería del software guiada por búsqueda que pueden aplicarse al problema de selección de requisitos. El objetivo es ayudar a los ingenieros a tomar la decisión definitiva sobre cual será el conjunto de requisitos a desarrollar en las siguientes etapas del proyecto. Para ello, combinamos diferentes medidas y técnicas de visualización. Hemos definido varios indicadores de calidad que tratan distintos niveles (requisitos, soluciones y clientes) junto a un conjunto de resúmenes estadísticos que se pueden utilizar para cuantificar estos procesos. Algunos de estos indicadores (principalmente la productividad a los tres niveles tratados) sirven para crear gráficos de ayuda y para identificar criterios de selección de la solución. Creemos que los ingenieros del software y los equipos de desarrollo pueden beneficiarse de estas ayudas numéricas y visuales, en el momento en el que realizan el análisis de los factores en la tarea de la selección de los requisitos de la siguiente versión.

Los resultados del caso de estudio sugieren que los requisitos de bajo esfuerzo y alta satisfacción se deben considerar en primer lugar, lo que coincide con la formulación del problema de selección de requisito. Además, con la presentación del caso se ilustra el posible proceso a seguir para establecer cuáles son los criterios de selección de la solución y cómo aplicarlo para fijar un pequeño número de soluciones e identificar el conjunto principal de requisitos, siempre dentro de los límites de satisfacción y esfuerzo fijados por el equipo de desarrollo.

References

1. Bagnall, A.J., Rayward-Smith, V.J., Whittle, I.: The next release problem. *Information & Software Technology* 43(14), 883–890 (2001)
2. Borgianni, Y., Cascini, G., Rotini, F.: Business process reengineering driven by customer value: a support for undertaking decisions under uncertainty conditions. *Computers in Industry* 68(0), 132 – 147 (2015)
3. Bourque, P., Fairley, R.E., et al.: Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press (2014)
4. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York (2007)
5. Dalgaard, P.: *Introductory Statistics with R*. Springer-Verlag New York (2008)

6. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, New York (2001)
7. Denzinger, J., Ruhe, G.: Decision support for software release planning using e-assistants. *Journal of decision systems* 13(4), 399–421 (2004)
8. Du, G., Ruhe, G.: Two machine-learning techniques for mining solutions of the releaseplannerTM decision support system. *Information Sciences* 259(0), 474 – 489 (2014)
9. Durillo, J.J., Zhang, Y., Alba, E., Harman, M., Nebro, A.J.: A study of the bi-objective next release problem. *Empirical Software Engineering* 16(1), 29–60 (2011)
10. Finkelstein, A., Harman, M., Mansouri, S.A., Ren, J., Zhang, Y.: "fairness analysis" in requirements assignments. In: *Proceeding of 16th IEEE International Requirements Engineering Conference (RE 2008)*, Barcelona, Spain. pp. 115–124 (2008)
11. Finkelstein, A., Harman, M., Mansouri, S.A., Ren, J., Zhang, Y.: A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirement Engineering* 14(4), 231–245 (2009)
12. de Freitas, F.G., de Souza, J.T.: Ten years of search based software engineering: A bibliometric analysis. In: *Proceedings of the Third International Conference on Search Based Software Engineering*. pp. 18–32. SSBSE'11, Springer-Verlag, Berlin, Heidelberg (2011)
13. Glass, R.L.: Addison-Wesley Professional (October)
14. Harman, M., Jones, B.F.: Search-based software engineering. *Information and Software Technology* 43(14), 833–839 (December 2001)
15. Harman, M., Mansouri, S.A., Zhang, Y.: Search-based software engineering: Trends, techniques and applications. *ACM Comput. Surv.* 45(1), 11:1–11:61 (Dec 2012)
16. IIBA, A.: guide to the business analysis body of knowledge (babok guide). International Institute of Business Analysis (IIBA) (2009)
17. Pitangueira, A.M., Maciel, R.S.P., Barros, M.: Software requirements selection and prioritization using {SBSE} approaches: A systematic review and mapping of the literature. *Journal of Systems and Software* 103(0), 267 – 280 (2015)
18. Ruhe, G.: *Product Release Planning: Methods, Tools and Applications*. CRC Press (June 2010)
19. del Sagrado, J., del Águila, I.M., Orellana, F.J.: Multi-objective ant colony optimization for requirements selection. *Empirical Soft. Eng.* pp. 20(3), 577-610 (2015)
20. del Sagrado, J., del Águila, I.M., Orellana, F.J.: Metaheuristic aided software features assembly. In: *ECAI*. pp. 1009–1010 (2012)
21. Saliu, M.O., Ruhe, G.: Bi-objective release planning for evolving software systems. In: *Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering*, Dubrovnik, Croatia. pp. 105–114 (2007)
22. Software Engineering Decision Support Lab, U.o.C.: ReleaseplannerTM
23. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2, 221–248 (September 1994)
24. Zhang, Y., Harman, M.: Search based optimization of requirements interaction management. In: *Proceedings of Second International Symposium on Search Based Software Engineering (SSBSE 2010)*, Benevento, Italy. pp. 47 –56 (sept 2010)
25. Zhang, Y., Harman, M., Mansouri, S.A.: The multi-objective next release problem. In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2007)*, London, England, UK. pp. 1129–1137 (2007)