

Marta Fernández-Diego¹,
Nolberto Munier²

¹Departamento de Organización de Empresas de la Universitat Politècnica de València;

²Consultor internacional en temas de planeamiento urbano y regional

<marferdi@omp.upv.es>
<nolmunier@yahoo.com>

1. Introducción

El riesgo de incumplimiento de los objetivos de un proyecto es un problema más frecuente y grave de lo que se suele tener en cuenta. Es más, en muchas ocasiones los riesgos de un proyecto no se gestionan, simplemente porque ni siquiera se llegan a considerar [1]. Y, sin embargo, los beneficios de la gestión de riesgos en proyectos son considerables. La gestión de riesgos permite detectar, desde el inicio del proyecto, problemas que de otra forma habrían pasado desapercibidos, y por tanto ayuda eficientemente al director de proyecto a conseguir entregar el producto a tiempo, ciñéndose al presupuesto y con la calidad requerida [2]. Pero si esta gestión de riesgos no se lleva a cabo a lo largo de todo el proyecto, no se alcanzará probablemente con ella todos sus beneficios potenciales.

Este artículo propone una metodología que consiste en construir, a partir de un valor final de riesgo para cada par proyecto/amenaza o alternativa, una matriz de decisión que permita determinar, mediante el empleo de Programación Lineal (PL), cuál es la alternativa más efectiva considerando los riesgos. Por supuesto, en un caso real estas mismas restricciones, más otras, pueden agregarse a la batería de restricciones que tratan aspectos ambientales, económicos, técnicos, financieros, políticos, etc. El resultado reflejará la mejor selección sobre la base de todas las restricciones consideradas en forma simultánea.

La aplicación de la PL a este problema de decisión es algo nuevo en el tratamiento de los riesgos. Abre una serie de posibilidades en el campo de la gestión de riesgos al tratarse de una metodología que permite representar más fielmente que otros métodos las características de un proyecto planteando problemas con todo tipo de restricciones, incluyendo aquellas correspondientes a riesgos, y en consecuencia colocando a estos al mismo nivel que las restricciones económicas, sociales y medioambientales normalmente consideradas, con la idea de elevar esta disciplina de la gestión de riesgos en proyectos. Aunque en definitiva, un nivel todavía más elevado de madurez organizacional en términos de gestión de riesgos correspondería a la gestión integrada de riesgos de la cartera de proyectos y, más allá, la conducción de proyectos por sus riesgos [3].

Selección de alternativas en proyectos considerando los riesgos

Resumen: El tema de la selección de proyectos consiste en determinar qué proyectos de un portfolio llevar a cabo, o qué alternativa adoptar cuando existen restricciones presupuestarias, comerciales, ambientales, técnicas, de capacidad, de localización, etc. Pero desgraciadamente no se da la misma importancia a los diversos riesgos inherentes a todo proyecto o alternativa. Es posible, sin embargo, determinar valores cuantitativos de riesgo para cada par alternativa/amenaza a fin de considerar además restricciones de riesgo en dicha selección.

Palabras clave: Amenaza, gestión del riesgo, programación lineal, proyecto, software libre.

Autores

Marta Fernández-Diego es doctora europea en Ingeniería electrónica y de telecomunicaciones. Después de varios contratos de investigación y desarrollo en universidades y empresas multinacionales de Francia, Reino Unido y España, pertenece actualmente al Departamento de Organización de Empresas de la Universitat Politècnica de València donde imparte entre otras la asignatura de gestión de riesgos en proyectos.

Nolberto Munier es Ingeniero Mecánico, Máster en Dirección y Gestión de Proyectos y Doctorando en Diseño, Fabricación y Gestión de Proyectos Industriales. Ha trabajado extensamente en técnicas de programación lineal y las ha aplicado a la solución de problemas de decisión en proyectos urbanos en varias ciudades de distintos países. Ha desarrollado por otro lado una metodología denominada Simus para la resolución de problemas de naturaleza compleja, con múltiples objetivos y con todo tipo de restricciones. Actualmente es consultor internacional en temas de planeamiento urbano y regional.

El resto del artículo está organizado como sigue. En la **sección 2** se presenta el ejemplo de aplicación. En la **sección 3** se describen con detalle las características del problema que determinan la elección de una u otra alternativa atendiendo a diversos criterios, junto con sus restricciones. Finalmente, una vez resuelto el problema con PL, se discuten los resultados obtenidos.

2. Ejemplo de aplicación

2.1. Antecedentes

En la última década el software libre ha experimentado un auge, incluso desafiando la inercia que persiste todavía en la ingeniería del software controlada básicamente por el software propietario. Aparecen nuevos modelos de negocio y ofertas de productos que permiten una elección real por parte de los consumidores.

Para entender el software libre, hay que empezar por aclarar que la característica fundamental del software propietario es que los derechos de explotación del mismo quedan exclusivamente reservados para el propietario, así como cualquier posibilidad de mejora o adaptación.

La problemática asociada al software, independientemente de si es libre o propietario,

radica en su propia naturaleza. La clave del problema que aborda el software libre está precisamente en la reutilización, en el sentido lógico de poder utilizar partes ya hechas por otros y crear derivados. Para realizar cualquier transformación de una obra ajena es necesaria la autorización de quien sea el titular de los derechos de dicha obra. En vez de usar el simple copyright de las licencias de software propietario que supone "todos los derechos reservados", estas otras licencias de software libre solo se reservan algunos derechos, e informan sobre si el propietario permite o no hacer copias, crear obras derivadas como adaptaciones o traducciones, o dar a las copias o derivados usos comerciales.

Por oposición, la característica fundamental del software libre es que se usa libremente [4]. En concreto permite al usuario el ejercicio de cuatro libertades básicas. Estas libertades son:

- Ejecutarlo para cualquier propósito.
- Estudiar cómo funciona y adaptarlo a sus necesidades.
- Distribuir copias.
- Mejorarlos y liberar esas mejoras al público.

Para cumplir con estas cuatro libertades precisamos de un código fuente¹ abierto. Con "código fuente abierto" nos referimos a que el

código fuente esté siempre disponible junto con el programa.

Y el ejercicio de estas libertades facilita la evolución del software, exponiéndolo lo más posible al uso y a la modificación (al llegar a más gente, está más probado) y quitando restricciones artificiales a la propia evolución (al estar más sujeto al ambiente).

2.2. Planteamiento del problema: alternativas y objetivo

Ante la futura comercialización de modelos de ordenadores con sistemas operativos de software libre preinstalados, un emprendedor, que planea crear una pequeña empresa, se plantea la posibilidad de comprar para su negocio ese tipo de ordenadores con sistemas operativos de software libre preinstalados. Ante tal posibilidad, le interesa decidir entre la alternativa con software propietario o libre atendiendo a criterios de riesgos, consistiendo el objetivo en minimizar el coste de la solución teniendo en cuenta una diferencia estimada de 100 euros a favor del ordenador con sistema operativo de software libre preinstalado.

3. Características del problema

Las características del problema vienen resumidas en la **tabla 1**. Las restricciones del problema, reflejadas también en dicha tabla, vienen explicadas en los siguientes puntos.

3.1. Criterios

En este caso se plantean dos alternativas o proyectos, que se van a analizar sobre la base de criterios que tienen en cuenta los distintos riesgos contemplados en ambos proyectos. En concreto, vamos a considerar tres criterios de selección que corresponden a tres de las amenazas potenciales relacionadas con el software y que marcan las mayores diferencias entre el software libre y el software propietario. Por supuesto, en un caso real pueden existir muchos otros criterios relacionados con la economía, la disponibilidad y experiencia del personal, el medio ambiental, etc., pero todos ellos se consideran en forma simultánea, conjuntamente con los criterios de riesgo. Por lo tanto, las alternativas u opciones tendrán que cumplir simultáneamente con todos ellos.

El riesgo de un proyecto conlleva una desviación con respecto a sus objetivos, en términos de tiempo, coste y funcionalidad, las tres características de un proyecto. En este sentido, el riesgo indica la probabilidad de que una amenaza se materialice poniendo en peligro el resultado del proyecto.

El riesgo puede medirse como la combinación de la probabilidad de que el incidente se materialice y la severidad del impacto [5]. Matemáticamente, este puede expresarse como sigue:

$$Riesgo = Probabilidad \times Impacto \quad (1)$$

Ante la certeza de materialización de la amenaza, el riesgo es igual al impacto, y si la probabilidad de materialización de la amenaza es nula, el riesgo es nulo; sin embargo el riesgo es una combinación de probabilidad e impacto. En estadística, el riesgo se modela a menudo como el valor esperado de un cierto impacto, y combina en un único valor las probabilidades de varias posibles amenazas junto con sus correspondientes impactos potenciales. En consecuencia, cada amenaza contribuye parcialmente a este valor esperado de riesgo.

Las amenazas consideradas, que aparecen como filas en la **tabla 1**, son las siguientes:

Resistencia al cambio

Está claro que todavía existe una gran inercia por el modelo propietario y a pesar de las ventajas que ofrece el software libre, ésta es su principal barrera. La inercia consiste en la resistencia del usuario a abandonar algo que conoce (software propietario), es decir hay una resistencia al cambio (software libre), apoyada por otro lado en las leyes de la física (por ejemplo, la resistencia a iniciar un movimiento).

Aunque los datos dependen de muchos factores, como el tamaño de la empresa, el propósito del software, su alcance, campo de aplicación, etc., se puede considerar que un 85% de las pequeñas empresas se deciden por productos de software propietario por inercia, falta de conocimiento sobre las alternativas de software libre o simplemente por el

miedo de cambiar a un nuevo proveedor, frente al 15% que se aventuraría a algo nuevo. Por lo tanto la probabilidad de resistencia al cambio para el software libre es mayor (85%), que la que presenta el software propietario (15%). Por otro lado, consideraremos en ambos casos que el impacto es total, es decir del 100%, puesto que lo que está en juego es la elección de una alternativa u otra.

Dependencia

Una ventaja no técnica del software libre es la independencia con respecto del proveedor, lo que garantiza la continuidad del negocio aunque desaparezca el creador original.

En principio el software libre surge contra las prácticas abusivas utilizadas por las principales empresas desarrolladoras de software propietario, que obligan de hecho a los usuarios a contratar permanentemente todas las actualizaciones y mejoras. En este sentido el usuario está atado de manos; tiene unos derechos muy limitados sobre el producto adquirido. Pero cuando las empresas recurren al software libre, se liberan de las restricciones que impone el proveedor de software. Precisamente, el software libre aparece para asegurar ciertas libertades al usuario.

Pero no nos limitamos a una dependencia con respecto del proveedor, resulta que esta dependencia se extiende también a los productos que pueden estar relacionados. El producto a menudo funciona mejor con otros productos del mismo proveedor. Con el software libre, por el contrario, los usuarios tienen el control para tomar sus propias decisiones.

Para simplificar el problema se han considerado valores iguales de probabilidad e impacto llegando a un riesgo por dependencia del 16% para el software libre y del 64% para el software propietario.

Falta de seguridad

Hay quien piensa que un sistema operativo de software libre es inherentemente más seguro que uno propietario simplemente por su herencia de Unix, que fue construido precisamente para proveer un alto grado de seguridad. Esta afirmación puede justificarse atendiendo a las siguientes razones:

	x ₁ (Software libre)	x ₂ (Software propietario)	Acción	Operador	B Umbral
Resistencia al cambio	0.85	0.15	MIN	≥	0.15
Dependencia	0.16	0.64	MIN	≥	0.16
Falta de seguridad	0.125	0.375	MIN	≥	0.125

Tabla 1. Características del problema.

Por un lado, un error de codificación puede potencialmente causar problemas de seguridad, como problemas debidos a la falta de validación. Pero el software libre es un software de mayor calidad, puesto que cuanto más gente pueda ver y probar un conjunto de código, más posibilidades hay de detectar algún fallo para corregirlo rápidamente. Es decir que la calidad está asegurada por la revisión pública del software y gracias a la colaboración abierta de un gran número de personas. Por este motivo, el software libre es menos vulnerable a virus y ataques maliciosos. Podríamos estimar que la vulnerabilidad del software libre frente a problemas de seguridad es de un 25%, mientras que para el software propietario dicha vulnerabilidad asciende a un 50%.

Pero por otro lado, el impacto de un problema de seguridad suele ser menor en el caso del software libre, puesto que estos problemas se resuelven normalmente más rápidamente al contar con toda una comunidad mundial de desarrolladores y usuarios para ayudar con eso. En cambio, en el mundo del software propietario, los parches de seguridad tardan considerablemente más en resolverse. Estimando impactos podríamos considerar un 50% para el software libre y un 75% para el software propietario.

En definitiva, considerando el riesgo como una combinación de vulnerabilidad e impacto, resulta un riesgo por falta de seguridad del 12.5% para el software libre frente a 37.5% para el software propietario.

Además, puesto que la transparencia dificulta de hecho la introducción de código malicioso, el software libre tiene mayor seguridad.

3.2. Restricciones

Como estamos hablando en los tres casos de eventos negativos, o lo que es lo mismo amenazas, y no hemos considerado ninguna oportunidad, la restricción que imponemos sobre estos criterios es la de minimizar, o lo que es lo mismo encontrar una solución mayor o igual al valor de riesgo mínimo, dado que no podemos encontrar una solución con riesgo menor que este.

Se podría haber planteado lo opuesto, por ejemplo, a la resistencia al cambio. La inercia

es la medida de la dificultad con la que se acepta un cambio. Mientras no se aplique fuerza alguna, seguimos nuestra propia inercia, que es una oportunidad para el que se ve favorecido. En este planteamiento, la acción adecuada (promover el cambio) hubiese sido la de maximizar, y en ese caso encontrar una solución menor o igual al valor del beneficio máximo dado que no podemos encontrar una solución con beneficio mayor.

4. Resolución por Programación Lineal

La expresión matricial del problema de PL es la siguiente:

$$AX = B \quad (2)$$

En donde:

$$A = \begin{pmatrix} 0.85 & 0.15 \\ 0.16 & 0.64 \\ 0.125 & 0.375 \end{pmatrix}$$

es la matriz de decisión, que aparece recuadrada en la **tabla 1**.

Los componentes a_{ij} de esta matriz son los valores de riesgo que cada amenaza trae aparejada para cada alternativa.

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

es el vector de incógnitas, o sea en este caso la opción a elegir.

$$B = \begin{pmatrix} 0.15 \\ 0.16 \\ 0.125 \end{pmatrix}$$

es el vector de umbrales, es decir establece los límites de cada restricción de acuerdo a lo que se comentó en la **sección 3.2**.

Para cumplir con el objetivo de minimizar el funcional Z; este funcional se expresa como la suma de los productos de costes de cada alternativa por cada una de ellas (o sea las incógnitas x que es lo que se quiere determinar).

Así, asumiendo que el coste de un ordenador con sistema operativo de software libre preinstalado es de 600 euros y que el correspondiente al software propietario es de 700 euros, el funcional será:

$$Z = 600x_1 + 700x_2 \text{ (Mínimo)} \quad (3)$$

Aplicando el método Simplex [6] de PL que

consiste esencialmente en una reiterada inversión de la matriz (2) de acuerdo a ciertas reglas, se obtiene, si existe, la solución óptima del problema. Es decir, la mejor combinación o mejor selección de alternativas que optimiza el funcional (3).

5. Discusión de resultados

5.1. Solución óptima

La solución óptima al problema de PL es la siguiente:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.292 \end{pmatrix} \quad (4)$$

De los dos valores se elige el de mayor valor, porque si bien ambos contribuyen a la obtención del objetivo, en realidad si queremos una opción solamente es evidente que la de mayor valor contribuye más eficientemente que la otra y por tanto es la elegida; por lo tanto en este caso se elige el software propietario (x_2). En nuestro caso ambos valores son muy próximos pero la PL indica que la alternativa con software propietario contribuye más eficientemente a la obtención del objetivo fijado teniendo en cuenta las restricciones de riesgo, y por lo tanto será la elegida.

5.2. Valor dual

Cada problema de PL directo, como el resuelto, presenta una solución que podemos llamar 'su imagen' y que se denomina 'dual'. En el problema dual las columnas son ahora las amenazas y las filas las alternativas. Si bien las variables del problema directo indican cuál de las opciones es la que mejor contribuye al objetivo, las variables del problema dual nos suministran los 'valores de las contribuciones marginales de cada criterio' o 'precios sombra', que es una expresión económica. En esencia, esto significa conocer en cuánto varía el funcional por cada variación unitaria de un criterio, lo cual en definitiva da una idea de la importancia de cada criterio.

En el caso planteado obtenemos los resultados que se muestran en la **tabla 2**.

Resulta que el problema de la inseguridad es el más determinante en la elección de la alternativa, con la resistencia al cambio en un segundo lugar, que intuitivamente podría verse como el más determinante. En cambio, el problema de la dependencia no afecta a la

	Valor igual	Valor marginal
Falta de seguridad	0.125	1683.333
Dependencia	0.207	0,000
Resistencia al cambio	0.150	458.333

Tabla 2. Valor igual y valor marginal.

solución, ya que su valor marginal es nulo. Esta potente herramienta nos permitiría, por ejemplo, analizar cuál sería la diferencia de costes que hace que la solución, y por ende la selección, cambie, y aparezca como más interesante la selección de ordenadores con sistema operativo de software libre preinstalado. Es más, en ese caso se podría observar que la componente de inercia no solo deja de ser clave en el proceso de selección sino que ni influye, y los criterios que interesan para la selección de la alternativa son en este caso la seguridad en primer lugar y el problema de la dependencia en segundo lugar.

6. Conclusiones

La utilización de la PL es una aplicación nueva en el tratamiento de los riesgos en proyectos. Su gran ventaja radica en que es posible representar escenarios reales con un cierto grado de fidelidad, ya que la cantidad de restricciones (y de alternativas) puede medirse en los cientos. Por otro lado, analizando el funcional para varios escenarios es posible deducir cual es la opción más conveniente [7]. Otra gran ventaja es que, de existir una solución, esta es óptima, es decir no se puede mejorar y cumple así con el óptimo de Pareto.

Referencias

- [1] Marta Fernández-Diego, Nolberto Munier. *Bases para la Gestión de Riesgos en Proyectos* 1st ed., Valencia, Spain: Universitat Politècnica de València, 2010.
- [2] Project Management Institute. *Practice Standard for Project Risk Management*, Project Management Institute, 2009.
- [3] M. Fernández-Diego, J. Marcelo-Cocho. Driving IS Projects. In D. Avison et al., eds. *Advances in Information Systems Research, Education and Practice*. Boston: Springer, pp. 113-124, 2008.
- [4] Richard M. Stallman. *Free software, free society: Selected Essays of Richard M. Stallman*. GNU Press, 2002.
- [5] International Organization for Standardization. *ISO 31000:2009 Risk management — Principles and guidelines*, International Organization for Standardization, 2009.
- [6] G.B. Dantzig. *Maximization of a linear function of variables subject to linear inequalities*. Published pp. 339-347 in T.C. Koopmans (ed.): *Activity Analysis of Production and Allocation*, New York-London 1951 (Wiley & Chapman-Hall), 1947.
- [7] Nolberto Munier. *A strategy for using multicriteria analysis in decision-making*. Springer – Dordrecht, Heidelberg, London, New York, 2011.

Nota

¹ Conjunto de líneas de texto con las instrucciones que deben seguirse para ejecutar el programa.



ACTUALIZACIÓN DATOS SOCIO ATI

¿Has cambiado de domicilio, de empresa, y lo has comunicado a la Secretaría General?

¿Recibes el correo postal de la asociación?

¿Te llegan los correos electrónicos enviados por las Secretarías de ATI?

Si has contestado que **NO** a todas estas preguntas, te agradeceríamos que enviaras un mensaje a secregen@ati.es con tus nuevos datos con el fin de tener actualizada tu ficha de socio y, de este modo, nos ayudes a mejorar la comunicación entre la asociación y sus miembros.

* Del mismo modo, si sabes de algún compañero tuyo, miembro de ATI, que no recibirá esta información, te agradeceríamos que se la hagas llegar para que se pueda poner en contacto con nosotros.

ATI Secretaría General | Vía Laietana 46, ppal. 1a. | 08003 Barcelona | 93 412 52 35 | secregen@ati.es | www.ati.es