

# Resumen gestión de proyectos

Iñaki Amatria Barral

i.amatria@udc.es

Universidade da Coruña – Enero 2020

Estas notas son, únicamente, un resumen del material de trabajo. Los contenidos de la asignatura se pueden consultar en los siguientes libros:

- Roger S. Pressman, *Ingeniería del software. Un enfoque práctico* (7 edición).
- Mario G. Piattini Velthius, Félix O. García Rubio e Ismael Caballero Muñoz-Reja, *Calidad de sistemas informáticos*.
- Steve McConnell, *Desarrollo y gestión de proyectos informáticos*.

## 1 ¿Es mejorable el proceso software seguido?

Para poder responder a esta pregunta, es necesario saber (1) que meta se quiere alcanzar y (2) cómo de cerca estamos de dicha meta. Es decir, para poder saber si cumplimos con una meta o no, debemos plantearnos unos objetivos. Solo de esta forma podremos observar cual es su grado de cumplimiento.

### 1.1 Objetivos

En general, en todo nuevo proyecto software se persiguen los siguientes objetivos. Con respecto a la gestión de proyectos, buscamos cumplir con (1) el esfuerzo, (2) el tiempo y (3) el coste estimado. Y, con respecto al producto propiamente dicho, nuestro objetivo es el de lograr una alta calidad.

Para los anteriores aspectos es necesario establecer sus medidas correspondientes para poder cuantificar y comparar. Por ejemplo, podemos medir el esfuerzo en términos de horas  $\times$  hombre o días  $\times$  hombre; el tiempo en días o meses; y el coste en euros. Pero, ¿cómo medimos la calidad?

Hay muchas definiciones de calidad, cada una enfatizando un aspecto concreto. Por ello, para cada empresa y, o, producto es necesario establecer lo que se entiende por calidad, haciendo énfasis en un aspecto u otro según sus necesidades, objetivos, clientes... Sin embargo, a todas ellas se aplican las siguientes limitaciones:

- Los parámetros que miden la calidad se establecen por anticipado y es el cliente quien los define.



- Los parámetros deben ser cuantificables y verificables. Es decir, debe existir un procedimiento objetivo y establecido para verificar su valor.

Son ejemplos de parámetros de calidad en software: la satisfacción del usuario con la funcionalidad y la facilidad de uso, el número de errores incluidos o el tiempo medio transcurrido entre fallos.

## 1.2 Grado de cumplimiento de los objetivos

De acuerdo con el Standish Group, *there is less chaos in software development today than there has been since the Standish Group started reporting chaos back in 1994*. Es decir, se va mejorando, pero el porcentaje es bajo.

## 1.3 ¿Cómo es posible mejorarlo?

No hay una bala de plata y las mejoras deben centrarse en tres aspectos claramente interrelacionados: (1) el desarrollo, (2) la gestión y (3) la calidad.

Mejoramos el desarrollo a través de la elección de un ciclo de vida adecuado a cada proyecto. Debemos tener en cuenta aspectos como el tamaño, el riesgo, la complejidad tecnológica y la dificultad de coordinación, entre otros.

Hay margen a la mejora en la gestión a través de la aplicación de aproximaciones sistemáticas, que se basan en la buena estimación del esfuerzo, tiempo y coste, para gestionar los proyectos.

Y, por último, es posible mejorar la calidad si trabajamos con ella desde un principio y a lo largo de todo el proceso de desarrollo software. No buscar la calidad sólo en el último instante, sino desde el principio del proceso.

# 2 Planificación y seguimiento

Los objetivos de un sistema de gestión de proyectos son los de (1) satisfacer la necesidad de información de gestión, (2) fomentar una cultura de gestión de proyectos que contribuya al aumento de la productividad, (3) contribuir a la estructuración y homogeneización del trabajo para facilitar labores de formación, reasignación, gestión y dotar de utilidad a la información histórica; y (4) mejorar la comunicación entre todo el personal involucrado.

## 2.1 Elementos

Los elementos de un sistema de gestión de proyectos son: los cimientos, es decir, una metodología eficaz; los pilares que son la arquitectura y las técnicas de gestión y un techo que marca el entorno organizativo. Sobre estos cuatro elementos, descansa la actividad de gestión de proyectos.

### 2.1.1 Metodología

Los aspectos fundamentales que una metodología de gestión de proyectos debe contemplar son los de (1) definición de los proyectos, (2) modelo genérico del ciclo de vida de los proyectos y (3) metodologías específicas por cada tipo de proyecto.

- **Definición de proyecto.** La definición de proyecto debe adaptarse a las necesidades de la organización. Cada organización establecerá lo que para ella es o no un proyecto. Genéricamente, un proyecto es el conjunto de: (1) recursos, (2) organización, (3) planificación, (4) objetivos, (5) productos y (6) que se mueven en un entorno de riesgo.

Como características, un proyecto es discreto en el tiempo, con inicio y fin bien definidos; complejo, con un conjunto de diferentes tareas interrelacionadas; y único, en relación a su producto final y a su entorno de desarrollo<sup>1</sup>.

De nuevo, y como nota importante, la gestión del proyecto buscará la entrega de sus productos finales cumpliendo las especificaciones, en plazo, dentro del presupuesto y con los niveles de calidad esperados y correspondientes a los estándares profesionales.

- **Modelo del ciclo de vida de un proyecto.** Fases propias a un proyecto son: (1) evaluación y aprobación del proyecto, (2) planificación y puesta en marcha, (3) ejecución del proyecto, haciendo un seguimiento periódico para medir la adherencia del proyecto al plan; y (4) finalización del proyecto.

Además, existen fases adyacentes que son: la gestión de la calidad, gestión de cambios (tema 4) y gestión de riesgos (tema 3).

- **Metodologías específicas.** Durante la ejecución del proyecto haremos uso de técnicas específicas propias del ámbito de ejecución del proyecto. Por ejemplo, en proyectos de desarrollo de software, organizaremos las actividades de acuerdo a un ciclo de vida.

Además, la metodología nos dice que una efectiva gestión de proyectos sucede si se produce una planificación disciplinada. Realizando el trabajo de acuerdo con estándares preestablecidos y tomando acciones correctivas adecuadas cuando las mediciones y evaluación de resultados no sean los esperados.

**Máxima a respetar:** en general, la planificación y su seguimiento no pueden consumir más esfuerzo que el propio proyecto.

### 2.1.2 Conceptos de planificación y seguimiento de proyectos

**2.1.2.1 Tarea** Es la unidad más elemental a nivel de planificación. Puede ser identificada por su duración, consumo de recursos o ambas cosas. Cada actividad es una parte independiente y homogénea del proyecto que, una vez comenzada, se realiza de forma independiente del resto, que le preceden y siguen. Cada actividad conduce a la obtención de un resultado tangible a utilizar para el desarrollo de otras actividades del proyecto.

<sup>1</sup>Un proyecto no es un proceso cotidiano o rutinario.

**2.1.2.2 Evento** También denominados a veces *milestones* o hitos. Es un tipo de actividad especial que no tiene duración y sirve para indicar un acontecimiento, un momento particular e importante del proyecto susceptible a ser modelado. No consume recursos. También se pueden usar ante eventos de terceras partes (subcontratas).

**2.1.2.3 Recurso** Un elemento se considera como recurso si va a estar sujeto a compartición, posiblemente originando conflictos de uso. Los recursos se pueden clasificar en (1) humanos, (2) materiales y (3) maquinaria<sup>2</sup>.

### 2.1.3 Técnicas

**2.1.3.1 Técnicas de representación** Son técnicas que nos permiten visualizar de manera rápida la programación de un proyecto, las interdependencias entre actividades y las desviaciones de diferentes métricas, como el esfuerzo, a través del tiempo.

- **Diagrama de Gantt.** Representa en una escala de tiempos cada una de las actividades mediante barras, que representan su duración en fechas de calendario. Es una representación simplificada de una red de precedencia.
- **Red de precedencia.** Grafo dirigido acíclico que señala las relaciones de secuencia entre las actividades de un proyecto. Un análisis de esta red permite identificar el camino crítico de una planificación.  
Existen fundamentalmente dos técnicas para construir el diagrama de red de un proyecto: PDM<sup>3</sup> y ADM<sup>4</sup>.
- **Histograma.** Es un diagrama de barras que muestra de forma visual la distribución de datos cuantitativos de una misma variable. Es típico el histograma de recursos para mostrar la asignación de recursos a lo largo del tiempo y ver sobre e infrautilizaciones<sup>5</sup>.

**2.1.3.2 Técnicas de estructuración** Nos permiten visualizar la estructura de descomposición de las actividades a realizar y la estructura organizativa del proyecto.

- **Work breakdown structure (WBS).** Técnica que consiste en estructurar las tareas de un proyecto por tipos. Nos ofrece una estructura de desglose de un proyecto en actividades con diferentes niveles de detalle.

<sup>2</sup>Por ejemplo, apero de labranza.

<sup>3</sup>*Precedence diagramming method.* Utiliza nodos, para representar actividades, y vectores, que conectan los nodos representando las dependencias o relaciones lógicas. Permite cuatro tipos de relaciones de precedencia: CC, CF, FC, FF (se verán más adelante). Es la técnica más empleada.

<sup>4</sup>*Arrow diagramming method.* Utiliza vectores, para representar las actividades, y nodos, para indicar las dependencias entre actividades a través de manejo de hitos. Hay tres tipos de relaciones de precedencia: lineales, de convergencia y de divergencia. Es una técnica poco empleada, ya que (1) obliga a manejar dos hitos por cada actividad y (2) para modelar relaciones más complicadas es necesario introducir actividades falsas con duración cero.

<sup>5</sup>Para asegurarse de que un recurso está siendo sobre o infrautilizado, debemos bucear por el plan de proyecto.

- **Organisational breakdown structure (OBS).** Técnica que consiste en estructurar la organización de un proyecto por las unidades organizativas, e incluso personas, que poseen responsabilidad sobre la realización del proyecto. Refleja como están organizadas las diferentes áreas de una organización en términos de responsabilidad funcional.

El cruce entre la WBS y OBS nos ofrece una visión interesante sobre quién hace qué.

**2.1.3.3 Técnicas de programación** Técnicas que ayudan a realizar una programación lo más detallada posible, indicando holguras, tareas críticas, etc.

- **PERT.** Nos permite considerar la probabilidad. Permite estimar la duración de un proyecto partiendo de la secuencia de actividades y de una estimación ponderada de la duración de cada una. A grandes rasgos, los pasos generales de la técnica PERT son los siguientes:

1. Elaboración de la red de precedencia (típicamente ADM) a partir de una WBS elaborada para el proyecto.
2. Cálculo de los tiempos PERT.

$$T_{PERT} = \frac{T_{Optimista} + 4 \times T_{Probable} + T_{Pesimista}}{6}$$

3. Cálculo de los tiempos *early*.
4. Cálculo de los tiempos *late*.
5. Cálculo de las holguras.
6. Determinación del camino crítico<sup>6</sup>.
7. Definición de fechas más tempranas y tardías de comienzo y finalización.

- **CPM.** Permite calcular la lista de actividades con menor flexibilidad en su ejecución. Un retraso en una de ellas implica, obligatoriamente, un retraso en la duración del proyecto. Permite obtener las fechas mínimas esperadas y fechas máximas permitidas, de comienzo y finalización, de las tareas. Los pasos ordenados de la técnica CPM son los siguientes:

1. Elaborar la red de precedencia (típicamente PDM) a partir de una WBS elaborada para el proyecto.
2. Identificación de todos los posibles caminos dentro del grafo dirigido desde el principio al fin del proyecto<sup>7</sup>.
3. Calcular los tiempos totales de cada camino.
4. Identificar el camino crítico: el que tenga mayor duración.

## 2.1.4 Arquitectura

Cuando queremos abordar un nuevo proyecto, debemos hacer un estudio de los diferentes aspectos técnicos y funcionales que nos ofrecen las herramientas de gestión de proyectos para saber cual se ajusta a nuestro problema en particular.

<sup>6</sup>Conjunto de todas las actividades sin holgura.

<sup>7</sup>Fuerza bruta.  $O(2^n)$ .

### 2.1.5 Entorno organizativo

La gestión de proyectos debe ser parte de un procedimiento o manual dentro de la organización. El contenido de ese manual debería contemplar al menos (1) estructura de órganos de responsabilidades, (2) normas y políticas de actuación, (3) conjunto de procedimientos administrativos<sup>8</sup> y (4) la o las técnicas y estándares de planificación a emplear y la forma de operar con ellos.

## 2.2 Pasos básicos para planificar

Los pasos básicos para la correcta planificación de un proyecto son los siguientes:

1. **Definir las actividades (e hitos a manejar).** WBS de acuerdo al estándar metodológico organizativo en función de la tipología del proyecto<sup>9</sup>.
2. **Definir las restricciones lógicas.** Definir las interrelaciones entre las actividades consideradas de acuerdo con la tipología del proyecto y la metodología específica a utilizar<sup>10</sup>. Hay cuatro tipos de restricciones:
  - **CC.** Cuando A comience, B puede comenzar. Por ejemplo, programar las clases y test unitarios.
  - **CF.** Cuando A comience, B puede finalizar. Por ejemplo, turnos de vigilancia.
  - **FC.** Cuando A finalice, B puede comenzar. Es la más usual y también la que más linealidad introduce (i.e. camino crítico). Por ejemplo, pulir antes de pintar.
  - **FF.** Cuando A finalice, B puede finalizar. Por ejemplo, backup viejo de un ordenador y instalación de uno nuevo.

Además del tipo, sobre una restricción se puede aplicar una demora, que puede ser positiva o negativa. Esto es, se puede adelantar o retrasar la actuación de una restricción.

3. **Asignación de recursos a las actividades.**
4. **Obtención del camino crítico.**
5. **Romper el camino crítico**<sup>11</sup>. Las actividades que forman el camino crítico son actividades sin holgura. Por tanto, si reducimos su duración, reducimos la duración del proyecto.
6. **Nivelación de recursos.** Tras definir qué perfiles de recursos se necesitan en cada actividad del proyecto y de asignar recursos a cada actividad, podemos detectar sobrecargas debido a, por ejemplo, (1) la no disponibilidad de recursos en determinados periodos, (2) mayor carga de la soportable por su dedicación en

<sup>8</sup>Como hojas de tiempo, distribución de informes, etc.

<sup>9</sup>Independientemente de la organización, los datos importantes sobre las actividades son (1) id de la actividad, (2) tipo, (3) descripción, (4) duración, (5) calendario, (6) fechas *early* y *late*, etc.

<sup>10</sup>E.g. de acuerdo a una aproximación al desarrollo en cascada, espiral, etc.

<sup>11</sup>Entraremos en bucle entre las tareas 5 y 6 hasta obtener una buena planificación.

una tarea o (3) mayor carga de la soportable por su dedicación en varias tareas al mismo tiempo.

Nuestra tarea es la de proponer posibles soluciones y escoger la que mejor nos convenga. Por ejemplo podemos (1) alargar las actividades al eliminar recursos, (2) cambiar de recursos, (3) introducir recursos que compartan el esfuerzo, (4) segmentar la actividad, (5) modificar su temporalidad, etc.

Además, también es posible que haya “valles” para un recurso en su histograma y que se pudiera llegar a hacer un empleo más eficiente del mismo modificando bien la planificación.

7. **Establecer línea base.** En el caso de que la planificación sea correcta, se establecerá línea base. La línea base es una foto fija de la planificación a efectos de comparación.

## 2.3 Seguimiento y control

### 2.3.1 Objetivos y procedimientos

Según Paulk, los objetivos de seguimiento y la supervisión de un proyecto software son (1) comparar los resultados actuales con los planes previstos, (2) tomar acciones correctivas cuando existan desviaciones significativas con respecto a los planes previstos y (3) acordar los compromisos necesarios con el personal afectado por las acciones correctivas.

Para cumplir con los objetivos de seguimiento de proyectos, es necesario definir una normativa estándar para el seguimiento: mensualmente, trimestralmente, etc.

Normalmente, para automatizar la recogida de datos y aumentar la fiabilidad de los mismos, el esfuerzo real o dedicación a cada tarea de cada proyecto se implementa a través de hojas de trabajo con una periodicidad concreta.

En el momento de realizar el seguimiento de un proyecto se establecerá el momento en el que se efectúa. Es lo que se denomina *time now* o fecha de progreso.

### 2.3.2 Seguimiento

A la hora de actualizar nuestro plan de proyecto con las actividades que hemos realizado hasta el momento, tenemos tres maneras distintas de rellenar la información sobre las tareas realizadas y que están en marcha. Estas tres formas de trabajar son:

- **Grosso-modo.** Simplemente indicamos el porcentaje de realización de una tarea. Lo usaremos para tareas con una gran holgura.
- **Semi-detallado.** Indicamos para cada recurso cuanto esfuerzo total le dedicó a la tarea.
- **Detallado.** Día por día y recurso por recurso indicamos las horas dedicadas a la tarea. Usaremos este método para tareas críticas, ya que son las que más nos interesa controlar. Recordamos que, si se retrasa una tarea crítica, se retrasa el proyecto.



Entonces, y como ya hemos dicho, haremos uso del seguimiento detallado en actividades sin holgura; del grosso-modo para actividades con mucha holgura; y un seguimiento semi-detallado para el resto de tareas.

### 2.3.3 Métricas

**Lord Kelvin**, “cuando puedas medir aquello sobre lo que hablas, y expresarlo en números, sabes algo de ello; si no puedes medirlo, es decir, si no puedes expresarlo en números, tu conocimiento es escaso y no satisfactorio”.

**2.3.3.1 Métricas de progreso** Miden el grado de realización de las actividades de acuerdo con la planificación de las mismas. La diferencia entre lo real y lo planificado es una indicación de la adherencia del proyecto al plan. Desviaciones significativas indican problemas.

La información de progreso es empleada por el jefe de proyecto y los diferentes niveles de la estructura de gestión de proyectos para seguir el progreso del proyecto con respecto al plan.

**2.3.3.2 Métricas de esfuerzo** Miden las dedicaciones de los recursos humanos para la realización de las actividades. La diferencia entre lo real y lo planificado es una indicación de la adherencia del proyecto al plan. Desviaciones significativas indican problemas.

La información de esfuerzo es empleada por el jefe de proyecto y los diferentes niveles de la estructura de gestión de proyectos para seguir el consumo real de dedicaciones de los recursos con respecto al plan<sup>12</sup>.

Una situación prolongada de muy pocos recursos trabajando en un proyecto puede ser el resultado de (1) un progreso inadecuado, (2) un número elevado de conflictos abiertos, (3) incompreensión de los requisitos, (4) un producto de mala calidad, (5) un equipo muy productivo, etc.

Una situación prolongada de exceso de recursos trabajando en un proyecto puede ser el resultado de (1) un problema más complejo de lo esperado, (2) recursos con insuficiente calidad para el trabajo, (3) un progreso inadecuado, (4) un número elevado de conflictos, (5) una subestimación del tamaño del software, etc.

**2.3.3.3 Métricas de coste** Miden los costes reales del proyecto con respecto al plan. La diferencia entre lo real y lo planificado es una indicación de la adherencia del proyecto al plan. Desviaciones significativas indican problemas.

La información de coste es empleada por el jefe de proyecto y los diferentes niveles de la estructura de gestión de proyectos para seguir el gasto real del proyecto con respecto al plan. Las métricas de coste más comunes son:

- **Variación de coste.**

$$CV = BCWP - ACWP$$

<sup>12</sup>Importante destacar las curvas de Rayleigh-Norden del esfuerzo de desarrollo. Asume que el esfuerzo para proyectos de desarrollo de software a lo largo del tiempo se distribuye en una colección de curvas (de Rayleigh), una para actividad del desarrollo.



Donde BCWP es el coste presupuestado del trabajo realizado y ACWP es el coste real del trabajo realizado.

- **Coste estimado a la finalización.**

$$TCE = ACWP + ETC$$

Donde ACWP es el coste real del trabajo realizado y ETC es el coste estimado pendiente hasta finalizar.

- **Variación de programa<sup>13</sup>.**

$$SV = BCWP - BCWS$$

Donde BCWP es el coste presupuestado del trabajo realizado y BCWS es el coste presupuestado del trabajo planificado.

### 3 Gestión de riesgos

De acuerdo con McConnell, se puede obtener un desarrollo rápido, económico, y de calidad basándose en cuatro pilares estructurales. Estos son (1) evitar los errores clásicos, (2) tener unas bases para el desarrollo, (3) hacer gestión de riesgos<sup>14</sup> y (4) usar métodos orientados a la planificación.

Pero cuidado, porque ni los métodos orientados a la planificación más avanzada son capaces de soportar por ellos solos el mejor plan posible. Cometer errores clásicos, no actuar según los buenos principios del desarrollo o no gestionar los riesgos provocará retrasos y costes asociados.

#### 3.1 La gestión de riesgos en el software

Los proyectos software incluyen un amplio rango de riesgos: cambios en los requisitos del usuario, falta de experiencia de gestión, mala estimación en la planificación, etc. Es importante controlar un amplio abanico de riesgos para que no nos controlen a nosotros.

De esta forma, podemos definir la función de la gestión de riesgos como la identificación, evaluación y eliminación de las fuentes de riesgo antes de que empiecen a amenazar la finalización satisfactoria de un proyecto. Según Pressman, se pueden controlar los riesgos a varios niveles, si se está en los niveles 1, 2 o 3 se ha perdido la batalla de la planificación:

- **Nivel 1:** control de crisis. Actuar de bombero apagando el fuego
- **Nivel 2:** arreglar cada error. Detectar y reaccionar rápidamente ante cualquier riesgo, pero solo cuando se ha producido.
- **Nivel 3:** mitigación de riesgos. Planificar con antelación que haría en caso de que ocurriese un riesgo, pero no intento eliminarlos inicialmente.

<sup>13</sup>No sirve para nada.

<sup>14</sup>Riesgo: es un evento, o condición incierta, que, en caso de ocurrir, tiene un efecto negativo sobre los objetivos de un proyecto. Un riesgo tiene una causa y, si ocurre (evento de riesgo), una consecuencia (efecto).

- **Nivel 4:** prevención. Crear y llevar a cabo un plan como parte del proyecto para identificar riesgos y evitar que se conviertan en problemas.
- **Nivel 5:** eliminación de las causas. Identificar y eliminar factores que pueden hacer posible la presencia de algún tipo de riesgo.

Pero, ¿cómo hago gestión de riesgos?. Según Boehm, existen 6 fases principales: (1) identificar los riesgos, (2) analizar los riesgos, (3) priorizar los riesgos, (4) planificar la gestión de riesgos, (5) resolver riesgos y (6) monitorizar los riesgos. A lo largo de las siguientes secciones, veremos una aproximación *lightweight* a la gestión de riesgos de acuerdo con los pasos que propone Boehm.

### 3.1.1 Identificación de riesgos

Consiste en elaborar una lista de riesgos que pueden afectar al proyecto, documentando así cada riesgo potencial. Para identificar riesgos en un proyecto software podemos recurrir a las siguientes técnicas:

- **Resultado insatisfactorio y sus causas.** El riesgo es frecuentemente definido como fuente de resultados insatisfactorios. De modo que, pensar en estos términos para ayudar a identificar riesgos potenciales puede ser una técnica muy beneficiosa. La mayoría de los resultados insatisfactorios tendrán una o más causas de origen, por ejemplo: falta de definición de un ciclo de vida, pobre planificación y seguimiento, pobre gestión de requisitos, etc.
- **Marco clasificatorio de riesgos.** Podemos hacer uso de clasificaciones de riesgos clásicas para identificar riesgos en cada una de las diferentes áreas. Podríamos clasificar los riesgos en: (1) riesgos estratégicos, (2) riesgos comerciales, (3) riesgos contractuales, (4) riesgos de gestión, (5) riesgos de proyecto y (6) riesgos de explotación. Existen otras clasificaciones o marcos pero, lo realmente importante, es disponer de unas tipologías establecidas y que esta aproximación sea lo más metódica posible.
- **Particionar el espacio del problema.** En este caso, esta máxima consiste en examinar individualmente cada tarea principal del proyecto y, por cada una de ellas, identificar aproximadamente 4-6 riesgos principales. Debemos evitar que la lista de riesgos identificados sea una enumeración exhaustiva de pesimismo.
- **Evento de riesgo y efecto.** La identificación de riesgos también se puede apoyar en el estudio de eventos y sus efectos. El mismo evento de riesgo y la misma causa pueden tener a veces efectos muy distintos. Por ejemplo, una avería en la turbina de un motor (causa) hace que falle el motor de un avión (evento de riesgo). Ahora pensamos, ¿qué pasaría si es fin de semana y el avión sobrevuela un polígono industrial? ¿y si sobrevuela un área comercial?
- **Lista de comprobación.** Es una lista que se construye a partir de información histórica, conteniendo los riesgos más habituales en los proyectos de una organización. Su principal ventaja: permite identificar riesgos de forma muy rápida. Su principal inconveniente: es prácticamente imposible tener una lista que incluya todos los posibles riesgos de un proyecto software. Por tanto, usa una

lista de comprobación, pero permitindo despois incluír novos riscos específicos do proxecto.

Como paso final, unha vez identificados os riscos, poderían clasificarse atendendo a las causas orixen que los provocan. Esto suele ayudar en la planificación de la gestión de riesgos para determinar causas a las que prestar mayor atención. Es decir, puede haber causas que deriven en varios riesgos y si elimino la causa, elimino varios riesgos (matar dos pájaros de un tiro).

### 3.1.2 Valoración o cuantificación de riesgos

Una vez identificados los riesgos en la fase anterior, es fundamental su priorización. Hay que atender a los que más pueden afectar al proyecto. Además, con muchos riesgos se podrá convivir sin mayores inconvenientes, ya que no suponen un peligro muy grande. Para su priorización es imprescindible un método de cuantificación o valoración. Una vez cuantificado un riesgo, éste ya es comparable con otros para estudiar la prioridad que posee dentro de todo el proyecto.

De la definición de riesgo, se deriva directamente la medida de la exposición al riesgo, que se define como el producto de:

Exposición al riesgo = Probabilidad de ocurrencia  $\times$  La magnitud de pérdida del riesgo

Por exemplo, si existe un 25% de probabilidade de que ocorra un risco que retrasaría el proyecto en 4 semanas, entonces la exposición ante ese riesgo es de  $0.25 \times 4$  semanas = 1 semana.

Para cuantificar la probabilidad del riesgo podemos usar valores numéricos (e.g. 0.69). Para cuantificar el impacto, tenemos diferentes opciones. Si el riesgo afecta a la planificación, podemos usar unidades temporales (e.g. 2 días). Si el riesgo afecta al presupuesto, el impacto se puede medir en unidades monetarias (e.g. euros). Si el riesgo afecta a la vida, podemos usar la gravedad de la incapacidad o el número de personas afectadas. De todas formas, la mayoría de los riesgos, si no todos, podrían ser expresados en su impacto como una pérdida económica.

Para acotar la subjetividad de estas medidas, podemos (1) solicitar estimaciones a personas familiarizadas con el sistema, (2) hacer uso del método Delphi<sup>15</sup> o (3) usar aproximaciones subjetivas empleando adjetivos y asignando un valor numérico a cada uno de estos adjetivos.

Una vez cuantificados los riesgos, debemos definir los niveles de riesgo aceptables. Así, podremos destacar un conjunto de riesgos clave de entre todos ellos. No solo nos debemos centrar en la ordenación estricta de la lista de riesgos identificados atendiendo a su exposición al riesgo. Algunos riesgos pueden priorizarse independientemente del lugar que ocupen en la lista (i.e. riesgos que producirán pérdidas considerables con una baja probabilidad). En definitiva, usar el sentido común.

Si usamos un método subjetivo de cuantificación<sup>16</sup>, se pueden definir los umbrales de forma sencilla. Por exemplo, los riesgos con exposición baja, vivimos con ellos (nivel 1); los riesgos con exposición media, se define un plan de contingencia (nivel 3); y

<sup>15</sup>Método Delphi: varios miembros de un grupo identifican riesgos y les asignan una probabilidad de ocurrencia y una magnitud de pérdida. Estas estimaciones son discutidas por el grupo y refinadas.

<sup>16</sup>Matriz no cuadrada de probabilidad  $\times$  impacto



riesgos con exposición alta, se definen acciones de prevención, alternativas y plan de contingencia (nivel 5).

### 3.1.3 Análisis de riesgos

Esta fase involucra aspectos relativos al estudio de posibles alternativas, a la definición de estrategias de mitigación a adoptar y a la planificación de las actividades de control de riesgo. Pero antes de poder elaborar planes de acción, tenemos que saber si nuestros riesgos están dentro de nuestro alcance o no. De esta forma, los clasificaremos en tres grupos (1) riesgos gestionables y dentro de alcance, (2) riesgos gestionables y fuera del alcance y (3) riesgos inevitables. De acuerdo con esta clasificación, aplicaremos las siguientes estrategias a los riesgos que caigan en el primer grupo:

- **Transferir.** Sacamos el riesgo del proyecto a través de subcontratas. Aparecen nuevos riesgos pero menos costosos. Podemos mitigarlos con auditorías.
- **Prevenir.** Desarrollamos el proyecto de forma que el riesgo no pueda convertirse en un problema. Consideramos caminos alternativos que eviten el riesgo.
- **Controlar.** Aceptamos la posibilidad de que se dé el riesgo y establecemos medidas para reducir la exposición al riesgo. Además, planificaremos acciones de contención en caso de que se de el evento de riesgo y haremos un seguimiento frecuente de ellos.

Además, para cada riesgo destacado elaboraremos un plan que lo considere. Este plan deberá definir (1) objetivos, (2) acciones concretas, (3) responsabilidades y (4) estimación de plazos, recursos y costes. Posteriormente, todos los planes individuales deben ser integrados.

### 3.1.4 Control y seguimiento de riesgos

A medida que se ejecute nuestro proyecto, haremos una monitorización frecuente de los riesgos. Así, nos aseguraremos de que los planes de gestión de riesgos establecidos en la fase anterior se llevan a cabo y podremos vigilar la modificación de los valores de los parámetros identificativos de los riesgos. Además, durante la ejecución de nuestro plan de proyecto, pueden surgir nuevos riesgos y desaparecer los que antes contemplábamos.

No hay un estándar previamente establecido que nos diga cuando hacer seguimiento de los riesgos. Los riesgos serán revisados y controlados conforme se establezca a nivel organizativo, surge la necesidad de establecer los momentos en que se procederá a la actividad de control de riesgos. Estos momentos dependen en gran medida del ciclo de vida establecido y de los tiempos que éste marque<sup>17</sup>. Como regla general, cada vez que se haga el informe de seguimiento de un proyecto, haremos control y seguimiento de riesgos.

<sup>17</sup>El modelo en espiral ya los establece intrínsecamente.

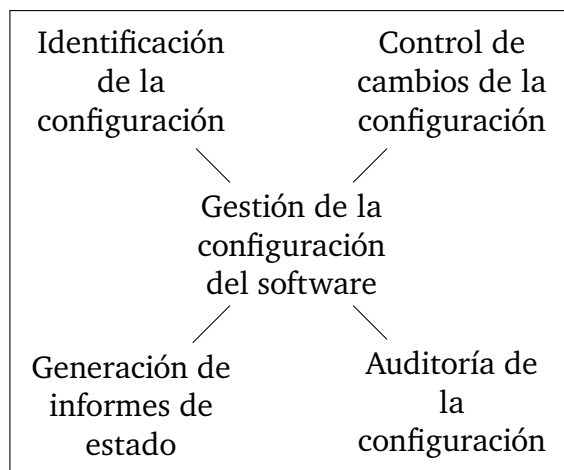


Figura 1: Actividades para lograr los objetivos de la gestión de la configuración del software.

## 4 Gestión de la configuración del software

La gestión de la configuración del software es la actividad que se encarga de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. Su misión se resume en minimizar la confusión, minimizar los errores y maximizar la productividad.

### 4.1 Objetivos de la gestión de la configuración del software

Los dos objetivos principales que persigue la gestión de la configuración del software son (1) facilitar la visibilidad sobre el estado y la evolución del producto software y (2) establecer y mantener la integridad<sup>18</sup> de los productos generados durante un proyecto y a lo largo de todo su ciclo de vida.

### 4.2 ¿Cómo conseguir los objetivos de la gestión de la configuración del software?

Según la IEEE, para alcanzar los objetivos que persigue la gestión de la configuración del software, tenemos que llevar a cabo las tareas de (1) identificar la configuración, es decir, definir los elementos de trabajo y donde encontrarlos, consiste en dar visibilidad; (2) control de cambios de la configuración, para poder trazar la evolución de los productos generados y utilizados; (3) generación de informes de estado, es decir, para cada elemento de la configuración debemos poder saber (a) que cambios se pidieron, (b) cuales se hicieron, (c) si existen pruebas y (d) quien realizó los cambios; y (4) auditorías de la configuración.

<sup>18</sup>Un producto software es íntegro si, y solo si, (1) satisface las necesidades del cliente, (2) cumple los requisitos de rendimiento y (3) se puede trazar su evolución desde que se concibió y a través de todas las fases de su ciclo de vida.

## 4.3 Conceptos básicos

### 4.3.1 Configuración del software

La configuración del software se define como el conjunto de todos los productos tanto generados como utilizados en un proyecto como resultado del proceso de ingeniería del software. Por tanto, es el término que designa el conjunto de todos los elementos de configuración del software de un proyecto.

### 4.3.2 Elemento de la configuración del software

Es cada uno de los componentes de la configuración del software. Es decir, es la unidad de trabajo de la gestión de la configuración del software. Un elemento de la configuración del software debe ser un elemento que se pueda definir y controlar de forma separada. Debe de ser una unidad en sí mismo.

### 4.3.3 Línea base

Es un concepto introducido para facilitar el control de cambios. Permite cambios rápidos e informales sobre un elemento de la configuración antes de que pase a formar parte de una línea base. Desde el punto de vista del proceso, es un punto de referencia en el proceso de desarrollo que queda marcado por la aprobación de uno o más elementos de la configuración mediante una revisión técnica formal.

En el momento en que se establece una línea base, se debe aplicar un procedimiento formal para evaluar y verificar cada cambio.

## 4.4 Actividades relacionadas

### 4.4.1 Control de versiones

Es un software que permite saber, para cada elemento de la configuración, (1) cuál es su última versión<sup>19</sup>, (2) relación entre distintas versiones y (3) dónde están. Esto nos facilita el control de cambios, podemos identificar perfectamente sobre qué versión/es hacer un cambio.

La manera más sencilla de crear una nueva revisión de un elemento de la configuración es realizar una modificación sobre la revisión más reciente. Así, las revisiones van formando una cadena, a la que se suele llamar cadena de versiones. Sin embargo, estas herramientas no nos limitan y nos permiten que varias versiones de un elemento coexistan en un determinado momento<sup>20</sup>. Esto surge de la necesidad de que un objeto satisfaga distintos requisitos al mismo tiempo. Estos nuevos caminos de desarrollo se denominan variantes y pueden ser:

<sup>19</sup>Una versión es una instancia de un elemento de la configuración del software en un momento dado del proceso de desarrollo. Se almacena en un repositorio y puede ser recuperada en cualquier momento para su uso o modificación. Además, cada versión debe poder identificarse de manera única.

<sup>20</sup>Las herramientas de gestión de revisiones no almacenan físicamente todas las versiones. Almacenan solo una de ellas, que puede ser la primera o la última. Sin embargo, nos permiten recuperar cualquier otra versión. Para ello guardan también toda la historia de cambios que han ocurrido sobre el elemento y que lo han hecho pasar de una versión a otra (i.e. deltas).

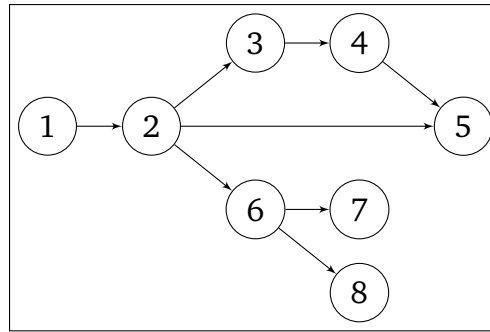


Figura 2: Grafo de revisiones.

- **Temporales:** A veces es necesario que varias personas trabajen simultáneamente sobre la misma versión de un objeto y para que no ocurran conflictos entre ellas se crea una variante distinta para cada persona. Una vez acabadas las modificaciones, es necesario mezclar todas las variantes para que la versión resultante contenga todos los cambios.
- **De usar y tirar:** para explorar diferentes soluciones alternativas en paralelo y quedarse con la mejor.
- **Permanentes:** que a su vez se dividen en (1) variantes de requisitos de usuario, el caso más típico era el del idioma; y (2) variantes de plataforma, una variante por cada sistema operativo o plataforma hardware sobre la que se desee que funcione la aplicación.

Al abrir ramificaciones en el grafo de evolución, en vez de una única configuración, vamos a tener un conjunto de configuraciones. Cada una va a satisfacer las necesidades de un entorno particular o usuario. Cada configuración alternativa se especifica mediante los elementos que la componen y la versión adecuada de cada uno de ellos.

La configuración del sistema que se comercializa o se entrega al cliente se llama release. Las releases deben de identificarse y almacenarse para poder recuperarlas en cualquier momento.

#### 4.4.2 Construcción (building)

Es la actividad que gestiona la compilación y el enlazado de archivos. Es una actividad facilitada por la gestión de la configuración del software. Necesita saber (1) que componetes enlazar, (2) en qué versión y (3) dónde están. Toma esa información de (1) la identificación de la configuración y (2) del control de versiones.

#### 4.4.3 Gestión de problemas

La gestión de la configuración del software facilita la gestión de problemas<sup>21</sup>. Esta actividad gestiona la evolución de los problemas detectados sobre el software, tanto aquellos que se detectan en la fase de pruebas como los informes de problemas que llegan del usuario.

<sup>21</sup>Los cambios en el producto pueden venir dados por cambios en los requisitos o por la aparición de problemas.



Las tareas a realizar en la gestión de problemas son (1) dar de alta los informes de incidencias, valorarlos y priorizarlos; (2) asignar el problema a un responsable, (3) asociar la información del fallo, (4) monitorizar el estado del problema, (5) registro de actividades de corrección de problemas y (6) informar acerca de los problemas, mediante la generación de informes, consultas a bases de datos de problemas y análisis estadísticos.

## 4.5 Otros aspectos relacionados

La gestión de la configuración también tiene una gran influencia en otros aspectos del desarrollo software:

- **Las metodologías** de trabajo deben integrar las actividades de la gestión de la configuración. Además, los productos que generen las fases que establezcan las metodologías son determinantes para establecer la gestión de la configuración.
- **El entorno de desarrollo** debe de estar integrado con las herramientas que facilitan la gestión de la configuración.
- **El entorno organizativo** debe regular los procedimientos de la gestión de la configuración con nuevas políticas, roles y responsabilidades que deberán de integrarse en la organización del proyecto.
- **La calidad**, ya que, directamente, se contribuye a mantener la integridad del producto.

## Anexo

### Aplicación a PYMES

#### Informe de seguimiento

##### 1. Portada

- Nombre y código del proyecto.
- Fechas de seguimiento.
- Quién lo redacta.
- Quién lo aprueba.

##### 2. Índice

##### 3. Introducción

- Breve descripción del proyecto.
- Objetivo del documento.

##### 4. Tareas realizadas desde el último seguimiento

##### 5. Tareas a realizar en el siguiente periodo de seguimiento

## **6. Monitorización de riesgos**

- Enlace a la hoja de cálculo.
- Cambio lo que necesite.

## **7. Gantt de seguimiento**

## **8. Problemas e incidencias**

### **Informe de cierre**

#### **1. Portada**

- Nombre y código del proyecto.
- Quién lo redacta.
- Quién lo aprueba.

#### **2. Índice**

#### **3. Introducción**

- Resumen del sistema.
- Estado actual.
- Revisión de los objetivos.
- Clasificación del proyecto.

#### **4. Resumen de gestión**

- Resumen del trabajo realizado.
- Errores detectados en el trabajo.
- Conclusiones importantes.

#### **5. Consideraciones principales**

- Por cada hamaca hago las consideraciones oportunas.
- Documentación.
- Organización del proyecto.
- Normas y procedimientos.
- Técnicas de desarrollo.
- Esfuerzo, tiempo y coste.

#### **6. Recomendaciones para el sistema de calidad**

- Apartado con formato libre.
- Se proponen mejoras en el sistema de calidad de la organización.

## Exámenes

### Verdadero o falso

Año	Pregunta	Respuesta
2004	Una línea base puede tener sobrecargas	V
2004	La gestión de la configuración del software se encarga de la evolución del software	V
2004	La gestión de la configuración del software está dentro de las disciplinas de control de la integridad de los productos	V
2004	En la gestión de la configuración del software, la visibilidad se define en la fase de identificación de la configuración	V
2007	En el ciclo de vida incremental, si un incremento está mal, afectará a incrementos anteriores y posteriores	F
2007	Si no se hace seguimiento se pierde el control del proyecto sin importar lo buena que sea tu planificación.	V
2012	Para definir calidad es suficiente con establecer por anticipado los parámetros cuantificables y verificables enfocados desde un punto de vista concreto	V
2012	Los objetivos de la gestión de proyectos son la finalización en plazo, dentro del presupuesto y la consecución del nivel de calidad deseado	F
2012	Una variante implica una configuración alternativa	V
2018	Una hamaca puede tener relaciones	F
2018	Si acortamos los caminos críticos dura menos el proyecto	V
2018	Un hito tiene esfuerzo y duración cero	V
2018	De acuerdo con la gestión de la configuración del software, es posible hacer cambios sin tener que formalmente decir que cambios hiciste <sup>22</sup>	N/A
2018	El incremento del grado de cumplimiento en los proyectos de desarrollo se debe a la mejora de la gestión de proyectos	F
N/A	Un ciclo de vida en espiral siempre pasa por las cuatro fases de determinar alternativas, evaluar alternativas, desarrollo del siguiente nivel y planificación de la siguiente fase.	V
N/A	Una efectiva gestión de proyectos sólo requiere poner en ejecución buenas prácticas de desarrollo	F

### Preguntas cortas

**(2007) Las cuatro dimensiones de un proyecto** Personas, proceso, producto y tecnología.

**(2007) Línea base desde el punto de vista del producto** Es un conjunto de elementos de la configuración del software revisados y aprobados que sirven como base para el desarrollo posterior y que solo se pueden cambiar a través de un proceso formal de control de cambios.

<sup>22</sup>Depende de si hemos establecido línea base o no.

**(2012) ¿Cómo modelas la baja de un recurso en Project? ¿En qué afecta a la tarea por defecto?** Para modelar una baja de un recurso basta con asignar días festivos en el calendario propio del recurso. Por defecto Project le reasignará el trabajo para cuando se reincorpore, es decir aumenta la duración de la tarea.

**(2014) ¿Cómo afecta que se quite un recurso a una tarea condicionada por el tiempo?** Se reduce el esfuerzo y coste.

**(2014) ¿Cómo cuantificarías los riesgos de tu empresa?** De la definición de riesgo se deriva directamente la medida de la exposición al riesgo, que se define como el producto de la probabilidad de ocurrencia y la magnitud de pérdida del riesgo.

Para cuantificar la probabilidad del riesgo podemos usar valores numéricos entre 0 y 1 (e.g. 0.79). Para cuantificar el impacto, tenemos diferentes opciones. Si el riesgo afecta a la planificación, podemos usar unidades temporales (e.g. 2 días). Si afecta al presupuesto, el impacto se puede medir en unidades monetarias (e.g. euros). Si el riesgo afecta a la vida, podemos usar la gravedad de la incapacidad o el número de personas afectadas. De todas formas, la mayoría de los riesgos, si no todos, podrían ser expresados en su impacto como una pérdida económica.

Además, podemos trabajar con métricas subjetivas. Estas métricas valoran la probabilidad de ocurrencia y el impacto con adjetivos como poca, media o alta.

**(2016) ¿Es posible que en el seguimiento el esfuerzo real aumente y la duración real de la tarea disminuya?** Sí. Una tarea que solo tenía asignado un recurso la realizan dos. Debido a la dificultad de coordinación hacen un poco más de esfuerzo, pero acaban antes de lo planeado.

**(2016) Como introduces un parón de dos días en Project** Creando dos días festivos en el calendario del proyecto.

**(2018) Tarea condicionada por el esfuerzo de 32 horas hombre de esfuerzo con dos recursos, uno al 100% y otro al 50%. Calcula la duración y horas de trabajo por día** Primero calculamos el tiempo que nos llevará completar en total la tarea:

$$\begin{aligned}
 T &= \frac{E}{R} \\
 &= \frac{32 \text{ horas} \times \text{hombre}}{1.5 \text{ hombre}} \\
 &= 21.33 \text{ horas}
 \end{aligned}$$

Ahora calculamos la dedicación de cada recurso en la tarea y usamos ese resultado para calcular cual es el esfuerzo que realiza en total cada recurso:

$$\begin{aligned}
 D_1 &= \frac{1}{1.5} & D_2 &= \frac{0.5}{1.5} \\
 &= 0.66 & &= 0.33 \\
 E_1 &= E \times D_1 & E_2 &= E \times D_2 \\
 &= 32 \times 0.66 & &= 32 \times 0.33 \\
 &= 21.33 \text{ horas} \times \text{hombre} & &= 10.66 \text{ horas} \times \text{hombre}
 \end{aligned}$$

De esta forma, la asignación diaria de los recursos es:

	Día 1	Día 2	Día 3	
R1	8	8	5.33	21.33
R2	4	4	2.66	10.66
	12	12	8	32

**(2018) Integridad del software** Un producto software se considera íntegro cuando (1) cumple con los requisitos, tanto implícitos como explícitos, del cliente; (2) satisface las necesidades de rendimiento y (3) se puede trazar su evolución desde que se concibió y a través de todas las fases de su ciclo de vida.

**(N/A) Enumera los métodos para identificar riesgos** Para identificar riesgos de una manera efectiva podemos seguir cualquiera de las siguientes alternativas:

1. Pensar en términos de resultados insatisfactorios y sus causas.
2. Marco clasificador de riesgos.
3. Segmentar el espacio del problema.
4. Evento de riesgo y efectos.
5. Listas de comprobación.

**(N/A) Define camino crítico** El camino crítico de una planificación es el conjunto de tareas críticas. Es decir, es la secuencia de tareas que tienen holgura igual a cero. Si se retrasa una de estas tareas, se retrasa todo el proyecto.

**(N/A) Define línea base de un proyecto** La línea base de un proyecto es una foto fija de la planificación a efectos de comparación.

**(N/A) Define línea base desde el punto de vista del proceso** Es un punto de referencia en el desarrollo que queda marcado por la aprobación de uno o más elementos de la configuración del software mediante una revisión técnica formal.