Tema 3: Planificación de Sistemas Software

BLOQUE I: INTRODUCCIÓN Y PARADIGMAS DE DESARROLLO EN INGENIERÍA DEL SOFTWARE

Ingeniería del Software
Grado en Ingeniería Informática
Curso 2024/2025

David Cáceres Gómez

UNIVERSIDAD
D
CÓRDOBA

Índice

1. Introducción

La planificación de un sistema software se encuentra enmarcada dentro de lo que se denomina gestión de proyectos. Esta disciplina es fundamental en la ingeniería del software, ya que aunque una buena gestión no garantiza el éxito del proyecto, si se ha demostrado que la mala gestión generalmente conduce al fracaso del proyecto, produciendo entregas fuera de plazo, costes mayores que los estimados e incumplimiento de los requisitos.

Los gestores de software tienen el mismo trabajo que otros de otras áreas, pero es importante reconocer que la ingeniería del software presenta características únicas:

- El producto es intangible.
- No existen procesos de software estándar.
- A menudo los proyectos software grandes son únicos.

En este contexto, en este tema nos centraremos en tres aspectos clave de la gestión: la planificación temporal, la gestión de riesgos y la estimación de los costes.

Un proyecto se define como un conjunto de etapas, actividades y tareas que tiene como finalidad alcanzar un objetivo que implica un trabajo no inmediato, a un plazo relativamente largo. Por lo tanto, un proyecto se caracteriza por varias características esenciales:

- Implica un principio y un final.
- Utiliza diversos recursos finitos y cuenta con un presupuesto.
- Tiene actividades únicas y esencialmente no repetitivas.
- Tiene un objetivo.
- Requiere un jefe de proyecto y personal de desarrollo cuyos roles y estructura de equipo deben definirse y desarrollarse.
- Tiene que planificarse.
- Debe medir su progreso frente al plan.
- Suele coexistir con otros proyectos y competir por los recursos.

2. Plan del Proyecto

El contenido del plan del proyecto es variable en cada proyecto, pero es recomendable incluir, al menos, los siguientes elementos:

- Un resumen del proyecto que pueda ser comprendido por cualquier persona. Debe indicar los productos entregables de forma que, cuando se produzcan se pueda comprobar que se ajustan al plan.
- Una lista de los hitos alcanzables.
- Los procedimientos y estándares que se van a aplicar.
- Una especificación del proceso de revisión que determine quién, cómo y cuándo se puede revisar la planificación del proyecto y con que objeto.
- Un diagrama de descomposición del trabajo (WBS).
- Una lista del personal del proyecto y su asignación en relación al WBS.
- Una red de actividades que muestre la secuencia de tareas en el tiempo y su relación entre ellas.
- Los responsables de todas y cada una de las actividades.
- Los presupuestos de esfuerzo y costes y los calendarios y plazos para todas las actividades.

Un proyecto de software debe seguir una estructura bien definida que abarca varios contenidos esenciales [1]. A continuación se detallan los elementos clave que debe incluir un proyecto de software:

PLAN DEL PROYECTO SOFTWARE

- INTRODUCCIÓN
  - Alcance y propósito del documento
  - Objetivos del proyecto
    - Objetivos
    - Funciones
    - Aspectos de funcionamiento
    - Restricciones técnicas y de gestión
- ESTIMACIONES DEL PROYECTO

    - Datos históricos utilizados para las estimaciones
    - Técnicas de estimación
    - Estimaciones
- RIESGOS DEL PROYECTO
    - Análisis del riesgo
        - Identificación
        - Estimación del riesgo
        - Evaluación
    - Gestión del riesgo
        - Opciones de aversión al riesgo
        - Procedimientos de supervisión del riesgo
- AGENDA
    - Estructura de descomposición de trabajos del proyecto
    - Red de actividades
    - Diagrama de Gantt
    - Tabla de recursos
- RECURSOS DEL PROYECTO
    - Personal
    - Hardware y software
    - Recursos especiales
- ORGANIZACIÓN DEL PERSONAL
    - Estructura de equipos (si existe)
    - Información de gestión
- MECANISMOS DE SEGUIMIENTO Y CONTROL
- ANEXOS

3. Planificación Temporal

Principios de la planificación temporal:

- **Descomposición del proyecto:** Es fundamental dividir el proyecto en un número manejable de tareas más pequeñas y específicas.
- **Interdependencia:** Se deben determinar las dependencias de cada tarea, ya que algunas tareas no pueden comenzar hasta que otras hayan finalizado.
- **Asignación de tiempo:** Cada tarea debe tener asignado un tiempo específico para su ejecución, incluyendo una fecha de inicio y otra de finalización.
- **Validación del esfuerzo:** A medida que se asigna tiempo a las tareas, el gestor del proyecto debe asegurarse de que los técnicos y humanos necesarios estén disponibles en el momento adecuado.
- **Responsabilidades definidas:** Cada tarea que se programe debe asignarse a un miembro específico del proyecto.
- **Resultados definidos:** El resultado de cada tarea, normalmente un producto, deberá estar definido. Los productos resultantes de las tareas suelen combinarse en entregas mayores, lo que facilita la evaluación del progreso del proyecto en su conjunto.
- **Hitos definidos:** Todas las tareas o grupos de tareas deben asociarse con algún hito del proyecto. Se considera un hito cuando se ha revisado la calidad de uno o más productos y se han aceptado.

Características comunes de los métodos de planificación:

- Identificar las tareas del proyecto y sus dependencias.
- Establecimiento de estimaciones de tiempo para las tareas.
- Estimación de los recursos necesarios para llevar a cabo cada tarea.
- Calcular los tiempos límite que definen un espacio temporal para cada tarea.

Dado que la planificación se basa en estimaciones, es importante que se actualice de forma continua a medida que avanza el proyecto y se obtienen nuevos datos.

Existen diversas representaciones visuales que pueden utilizarse para la planificación: basadas en calendario (diagrama de Gantt) o redes de actividades (diagrama de precedencias).

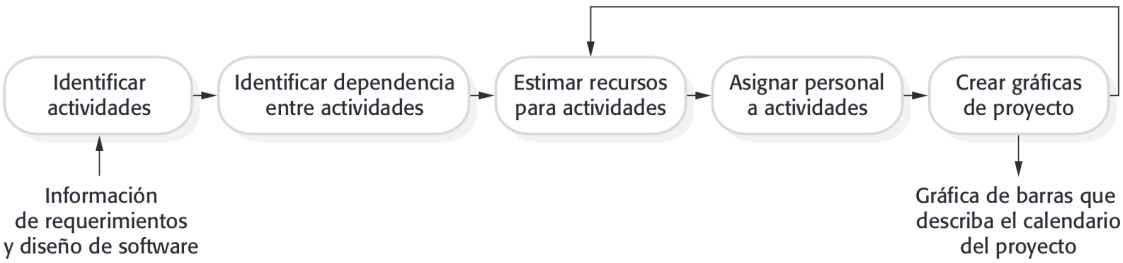

Figura 1: Proceso de planificación temporal del proyecto [2]

3.1. Diagrama de Gantt

El diagrama de Gantt es uno de los métodos más antiguos y, sin duda, uno de los más utilizados en la planificación y gestión de proyectos. Su simplicidad y claridad lo convierten en una herramienta valiosa para visualizar el cronograma de trabajo.

Se representa en forma de cuadro de doble entrada:

- **Eje horizontal:** Representa el tiempo, generalmente dividido en días, semanas o meses, dependiendo de la duración del proyecto.
- **Eje vertical:** Enumera las tareas del proyecto, dispuestas en orden jerárquico o secuencial.
- **Tareas:** Cada tarea se muestra como un rectángulo colocado a la altura correspondiente en el eje vertical. La longitud de cada rectángulo indica la duración de la tarea, desde su inicio hasta su finalización.

A pesar de su utilidad, el diagrama de Gantt presenta algunos **inconvenientes**:

- **Falta de interdependencias:** No muestra de manera explícita las relaciones entre las diferentes tareas, lo que puede dificultar la identificación de dependencias críticas en el proyecto.
- **Complejidad en proyectos grandes:** En proyectos con muchas tareas, el diagrama puede volverse complicado y difícil de crear, lo que puede llevar a una visualización desordenada y confusa.

El diagrama de Gantt también ofrece varias **ventajas** significativas:

- **Facilidad de comprensión:** Su diseño simple y visual lo hace fácil de entender para todas las partes interesadas, independientemente de su nivel de experiencia en gestión de proyectos.

- **Versatilidad:** Además de representar el cronograma de tareas, se puede utilizar para visualizar la utilización de recursos, mostrando quién está asignado a qué tarea y cuándo, lo que facilita la gestión de recursos humanos y materiales.
- **Seguimiento del progreso:** Permite a los gestores de proyectos monitorear el avance de las tareas a lo largo del tiempo, lo que ayuda a identificar retrasos y ajustar el cronograma según sea necesario.

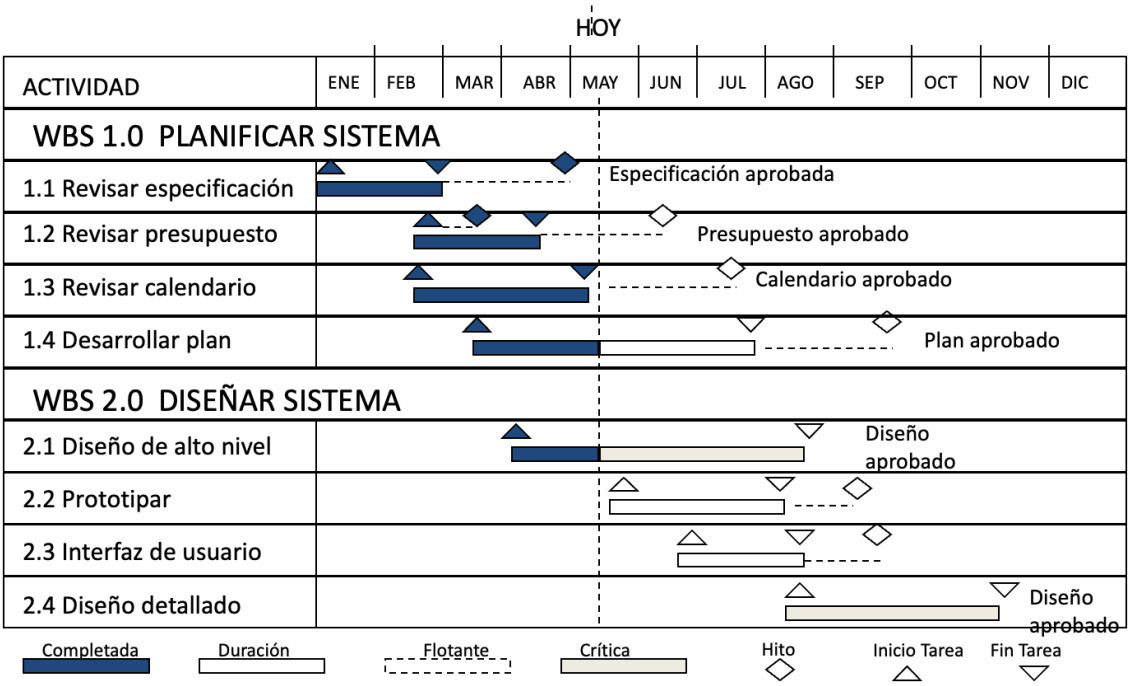

Figura 2: Diagrama de Gantt de ejemplo

3.2. Diagrama de Precedencias o Actividades

El diagrama de precedencias es una herramienta de gestión de proyectos que representa gráficamente las actividades y sus dependencias. Cada actividad tiene una duración definida, precedentes (actividades que deben completarse antes) y un producto o entregable. Los hitos son puntos en el tiempo que marcan eventos importantes, como el inicio o fin de una actividad, y no tienen duración.

El modelo de precedencias define relaciones entre actividades (Fin a Inicio, Inicio a Inicio, etc.) y permite calcular la duración total del proyecto.

El diagrama de precedencias es un modelo simplificador, con las siguientes características principales:

- **Duraciones fijas:** Cada actividad tiene una duración establecida, facilitando la planificación del tiempo.
- **Limitaciones de inicio:** No permite indicar el comienzo de una actividad en un punto específico de la ejecución de otra. Para solucionar esto:
  - **Dividir actividades:** Las actividades complejas se pueden segmentar en partes más pequeñas.
  - **Uso de herramientas:** Por ejemplo, en software como *Project*, se puede establecer que una actividad no comience hasta que su actividad precedente esté cumplida en un 90 %.
- **Grafo ordenado totalmente:** Las actividades y sus relaciones están organizadas en un grafo que muestra claramente la secuencia de tareas.
- **Representación de tareas:** Las tareas se representan como nodos en el diagrama.
- **Relaciones entre tareas:** Las flechas que conectan los nodos representan las relaciones de precedencia, indicando el flujo desde la tarea antecesora a la sucesora.

Todos los nodos tienen el mismo tamaño y pueden contener mucha información sobre la tarea. En la Figura 3 se puede ver la estructura de la actividad en el grafo, que contiene:

- **Descripción de la actividad:** nombre dado a la actividad.
- **Etiqueta actividad:** número que identifica a cada actividad.
- **Duración:** tiempo que calculamos que se tardará en completar la tarea.
- **Inicio temprano:** fecha más temprana en que puede comenzar la tarea.
- **Final temprano:** fecha más temprana en que puede finalizar la tarea.
- **Inicio tardío:** fecha más retrasada en la que se puede comenzar sin que afecte a la fecha de terminación del proyecto.
- **Final tardío:** fecha más retrasada en la que puede terminar la tarea sin afecte a la fecha final del proyecto.
- **Máximo tiempo disponible (MTD):** tiempo máximo que puede durar una tarea en caso de comenzar en su Inicio temprano y concluir en su Final tardío.
- **Holgura (H):** tiempo que disponemos para jugar con el inicio de la tarea, sin afectar al proyecto.

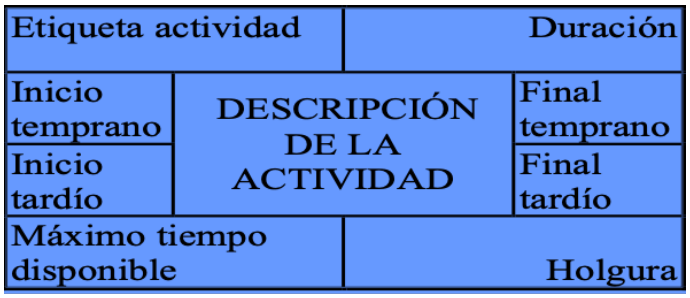

Figura 3: Estructura de una actividad [1]

A continuación, se presenta un ejemplo de diagrama de precedencias. En la Tabla 1, se muestra cómo se plantea el problema, indicando las actividades del proyecto, sus actividades precedentes y la duración de cada una. En la Figura 4, se puede observar el diagrama construido a partir de la información de la tabla anterior.

Actividad	Precedente	Duración
A	-	3
B	A	4
C	B	5
D	A	5
E	C, F	7
F	D	8
G	E	5
H	G	7
I	H	1
J	H	2
K	H	10

Tabla 1: Actividades con sus precedencias y duración

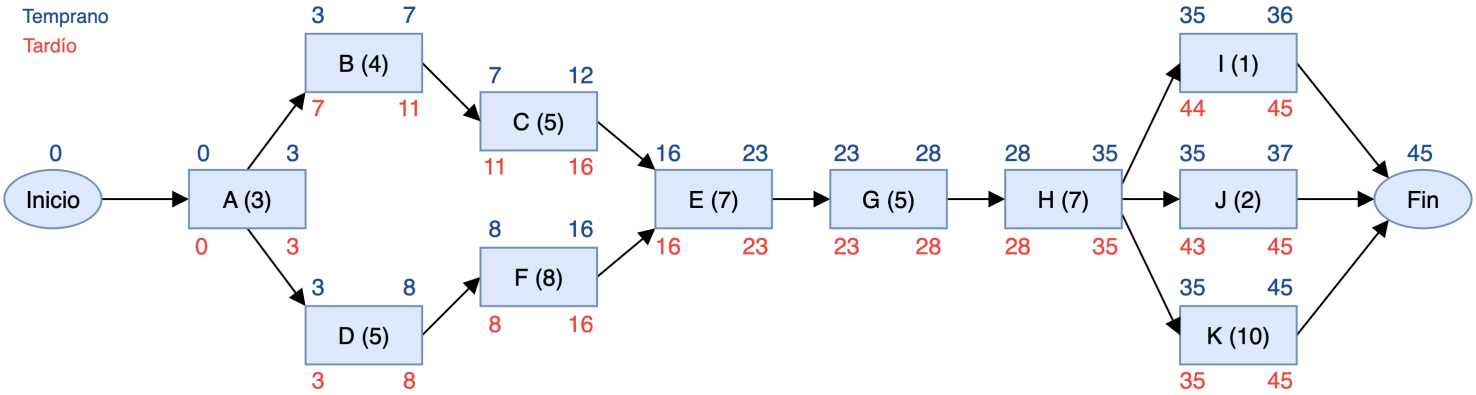

Figura 4: Diagrama de Precedencias de ejemplo

Para el cálculo de las fechas para cada actividad en el proyecto de ejemplo, se han seguido los siguientes pasos:

1. Partimos de la tabla de precedencias.
2. Asignamos como **inicio temprano** "0" a todas las actividades sin predecesor.
3. El **final temprano** de cada actividad es el inicio temprano más su duración.
4. Si la actividad **tiene predecesoras**, y todas estas tienen calculado su final temprano, asignamos como **inicio temprano el máximo** de todos ellos.
5. Obtenemos la **fecha de final** del proyecto, siendo la máxima fecha de final temprano.
6. A todas las actividades que no tengan sucesoras se le asigna esta fecha como **final tardío**.
7. El **inicio tardío** se calcula restando al final tardío la duración.
8. Aquellas **actividades con sucesoras**, se les asigna como final tardío el **mínimo de los inicios tardíos** de estas.

Para obtener el máximo tiempo disponible (MTD) y la holgura (H) de cada actividad:

$$MTD = Final\ Tardio - Inicio\ Temprano \tag{1}$$

$$H = MTD - Duración \tag{2}$$

Camino Crítico

El camino crítico es el conjunto de actividades o tareas que tienen **holgura cero**, lo que significa que cualquier retraso en estas actividades afectará directamente la duración total del proyecto. Este camino se define como:

- Comienza con una actividad que no tiene predecesoras, atraviesa el grafo a través de actividades con holgura cero y finaliza en una actividad que no tiene sucesoras.
- Si la duración de las actividades es mínima, entonces el camino crítico está claramente definido.
- Las actividades que forman parte del camino crítico se denominan actividades críticas. Un retraso en cualquiera de ellas resultará en un retraso en la finalización del proyecto.

El camino crítico de este ejemplo sería: **A-D-F-E-G-H-K**

4. Análisis del Riesgo

El análisis del riesgo tiene que ver con la identificación de los riesgos y los planes para minimizar sus efectos en el proyecto.

Podemos clasificar los riesgos en tres categorías:

- **Riesgos del proyecto:** Identifican problemas de presupuesto, de agenda, del personal, de los recursos , del cliente y sus requisitos, del tamaño y complejidad del proyecto. Afectan al coste y duración del proyecto
- **Riesgos del producto:** Problemas en el diseño, implementación, interfaz, incertidumbre técnica, problemas de obsolescencia o de utilización de tecnología punta. Afectan a la calidad del software resultante
- **Riesgos del negocio:** Cambio en la dirección, cambios de estrategia del negocio, cambios en el mercado, perdidas en la empresa. Afectan al equipo de desarrollo y a la realización del proyecto en sí.

Riesgo	Tipo	Descripción
Abandono del Personal	Proyecto	El personal experimentado deja el proyecto antes de su finalización
Cambios de dirección	Proyecto	Cambios en la dirección del proyecto y en la estructura y prioridades del equipo de desarrollo
HW no disponible	Proyecto	El Hardware necesario para la realización del proyecto no está disponible en la fecha acordada
Cambio en los requisitos	Proyecto y producto	El número de cambios en los requisitos es mucho mayor al esperado
Retrasos en la especificación	Proyecto y producto	La especificación de las interfaces del sistema no están disponibles en la fecha acordada
Tamaño subestimado	Proyecto y producto	El tamaño del proyecto se ha subestimado
Herramientas CASE no disponible	Producto	Las herramientas CASE previstas para el desarrollo del proyecto no están disponibles a tiempo
Cambios en la tecnología	Negocio	La tecnología sobre la cual iba a construirse el proyecto es substituida por una nueva tecnología
Producto de la competencia	Negocio	Un producto competitivo aparece en el mercado con objetivos similares al que se está construyendo

Tabla 2: Ejemplo de riesgos

Proceso de gestión del riesgo

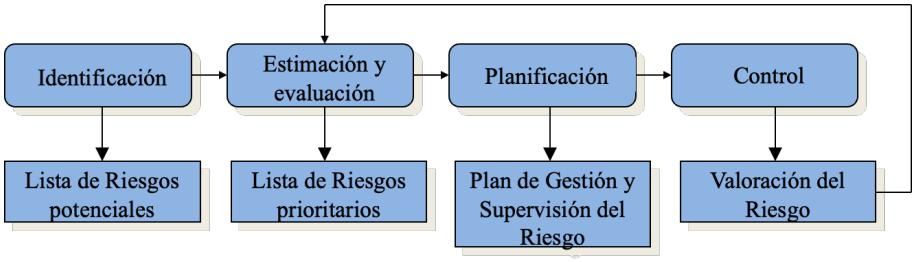

Figura 5: Proceso de gestión del riesgo [2]

Identificación del riesgo

- Identifica los posibles riesgos del proyecto, del producto y del negocio.
- Consiste en determinar para cada tipo de situación cuales son los posibles riesgos que pueden afectar al desarrollo del proyecto.

Estimación y evaluación del riesgo

- Determina la probabilidad y el impacto de cada riesgo.
- La probabilidad puede ser expresada de forma cuantitativa o cualitativa: muy bajo (<10 %), bajo (del 10 al 25 %), moderado (del 25 al 50 %), alto (del 50 al 75 %) o muy alto (>75 %).
- El impacto tiene que ver con sus consecuencias: catastrófica (amenazan la supervivencia del proyecto), grave (causarían grandes demoras), tolerable (demoras dentro de la contingencia permitida) o insignificante, y con la duración de las mismas.

Planificación del riesgo

- Traza un plan para evitar o minimizar la ocurrencia de un riesgo.
- Considerar cada riesgo con probabilidad alta/muy alta a partir de un impacto tolerable y moderada a partir de un impacto serio y desarrollar estrategias para gestionar dicho riesgo.
- Se plantean escenarios "What-if".
- Se diseñan estrategias para manejar el riesgo:
    - **Estrategias de evitación:** buscan reducir la probabilidad de que el riesgo ocurra.

    - **Estrategias de minimización:** buscan reducir el impacto del riesgo si este ocurre.
    - **Planes de contingencia:** son estrategias para gestionar el peor escenario posible.

- Todas estas tareas se engloban y detallan en el plan de gestión y supervisión del riesgo.

Control del riesgo

- Controla la ocurrencia de riesgos a lo largo del proyecto.
- Asegura el cumplimiento de las tareas para evitar el riesgo y para minimizar su impacto en caso de que ocurra.
- Revisa periódicamente cada uno de los riesgos identificados para decidir si su probabilidad de ocurrencia ha aumentado o disminuido.
- Revisa también si las consecuencias del riesgo cambian.
- Los riesgos considerados deben ser discutidos en las reuniones periódicas de progreso del proyecto.

5. Técnicas de Estimación

Las técnicas de estimación en el desarrollo de software están diseñadas para estimar el tiempo y esfuerzo necesarios para completar un proyecto. Estas técnicas son fundamentales en la planificación del proyecto, ya que permiten una mejor organización de los recursos y una gestión adecuada del tiempo. Se basan en una serie de datos históricos, obtenidos a partir de métricas sobre el proceso y el proyecto, que ofrecen una perspectiva sobre experiencias previas. Esto permite que las estimaciones se basen no solo en conjeturas, sino en la realidad de proyectos anteriores, lo cual proporciona una mayor precisión en las previsiones.

La experiencia previa del equipo juega un papel importante en la estimación, ya que conocer la dinámica del trabajo y las posibles dificultades ayuda a prever con mayor exactitud el tiempo y esfuerzo requeridos. Un equipo con experiencia es capaz de identificar las tareas más complicadas y las áreas que pueden representar un mayor riesgo o consumir más tiempo. De esta manera, se obtiene una estimación más realista, lo que influye positivamente en otras acciones de la planificación, como la asignación de tareas y la gestión de recursos.

El proceso de estimación requiere no solo experiencia, sino también acceso a información histórica fiable, así como el compromiso de todas las partes involucradas en el proyecto. Este compromiso es esencial, ya que las estimaciones influyen en los plazos y costos finales. Sin embargo, a pesar de todos los esfuerzos, cualquier estimación siempre conlleva un nivel de riesgo e incertidumbre. Este riesgo está asociado a factores como la complejidad del proyecto, el tamaño del mismo y el grado de especificación de los requisitos, que pueden variar a lo largo del desarrollo.

En metodologías evolutivas o incrementales, las estimaciones se pueden ajustar conforme cambian los requisitos del proyecto. A medida que el desarrollo avanza, es posible revisar y modificar la planificación, lo que ayuda a mitigar el riesgo asociado a cambios imprevistos. Aunque la estimación de costes/esfuerzo no es una ciencia exacta, existen pasos sistemáticos que pueden seguirse para lograr estimaciones más fiables y con un nivel de riesgo aceptable.

Existen varios enfoques para realizar estimaciones en proyectos de software:

1. **Estimación tardía:** ofrece mayor fiabilidad, aunque puede resultar poco práctica debido al retraso en su obtención.
2. **Basada en proyectos anteriores:** no garantiza que se repitan las mismas condiciones de los proyectos pasados.
3. **Uso de técnicas de descomposición:** estrategia que sigue el principio de "divide y vencerás", donde el proyecto se fragmenta en partes más manejables para estimar mejor cada componente.
4. **Uso de métodos empíricos**: basado en la combinación de datos históricos para generar estimaciones más fundamentadas.

Cabe destacar que las técnicas de descomposición y los métodos empíricos no son excluyentes, sino complementarios, ya que pueden utilizarse conjuntamente para lograr estimaciones más precisas.

Las estimaciones suelen basarse en diversas métricas, que se dividen en tres categorías principales:

- Sobre el proceso, que consideran aspectos como las metodologías y herramientas utilizadas.
- Sobre el proyecto, que evalúan factores relacionados con el equipo, como el número de personas y la productividad.
- Sobre el producto, que abarcan características como el tamaño y la complejidad del software a desarrollar.

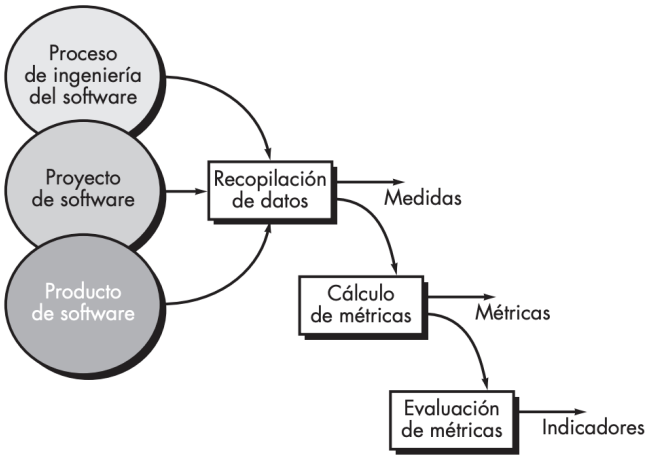

Figura 6: Proceso de recopilación de métricas del software [3]

Las **métricas de software** son herramientas útiles para evaluar de forma cuantitativa diversos aspectos del desarrollo del proyecto. Su principal objetivo es generar indicadores que ayuden a mejorar del rendimiento a largo plazo en el proceso o permitan ajustar el flujo de trabajo en el proyecto. Sin embargo, es importante interpretarlas con precaución, ya que ofrecen solo una visión parcial de la realidad. No existe una única métrica universalmente válida, por lo que deben relativizarse según el contexto, como el lenguaje de programación utilizado. Además, estas métricas no deben emplearse para criticar a individuos o equipos.

La clasificación histórica de las métricas de software se organiza en diferentes categorías:

- **Orientadas al tamaño:** miden aspectos como errores o defectos por líneas de código (LOC).
- **Orientadas a la función:** como los puntos de función (FP), que han generado cierta controversia.
- **Orientadas a casos de uso**: se relacionan de manera indirecta con el tamaño del sistema y las pruebas.
- **Para sistemas orientados a objetos:** incluyen factores como la herencia, cohesión y acoplamiento.

Las **técnicas de estimación** basadas en **descomposición** consisten en dividir el proyecto en partes más pequeñas para facilitar la estimación. Esta descomposición puede realizarse a nivel del problema, dividiendo el sistema en módulos, o a nivel del proceso, separando las tareas involucradas. La precisión de estas técnicas depende en

gran medida de una correcta estimación del tamaño del proyecto. El valor estimado final se calcula a partir de tres escenarios: optimista, más probable (con mayor peso) y pesimista.

Los **métodos de estimación empíricos** se basan en fórmulas que predicen el esfuerzo a partir de mediciones como líneas de código (LOC) o puntos de función (FP), utilizando técnicas de regresión. Estos modelos toman como datos de entrada una muestra de proyectos finalizados para generar sus predicciones. Sin embargo, el modelo puede requerir recalibración al compararse con resultados reales. Un ejemplo destacado de esta categoría son los modelos de estimación de costes *COCOMO* (Boehm, 1981) y *COCOMO II* (Boehm, 2000), los cuales ofrecen una jerarquía de modelos con distintos fines. Estos modelos se basan en métricas de tamaño y asignan niveles de dificultad (simple, medio, difícil) a diferentes partes del sistema.

Escenarios alternativos en la planificación del desarrollo de software incluyen diversas opciones estratégicas:

- **Adquisición vs. Desarrollo de software:** Esta decisión depende de la criticidad del desarrollo y los costes implicados. Se analizan distintas alternativas y se estima la probabilidad de cada escenario. El coste total se calcula multiplicando la probabilidad por el coste de cada alternativa.
- **Externalización del desarrollo (outsourcing):** Implica contratar parte del desarrollo a una empresa externa, con el objetivo de reducir costes y mejorar la calidad. La decisión puede ser de carácter estratégico, enfocada en reducir costes, o táctica, al recurrir a una empresa especializada. Requiere un análisis financiero, y aunque permite reducir la necesidad de recursos internos, se pierde control sobre el proceso.

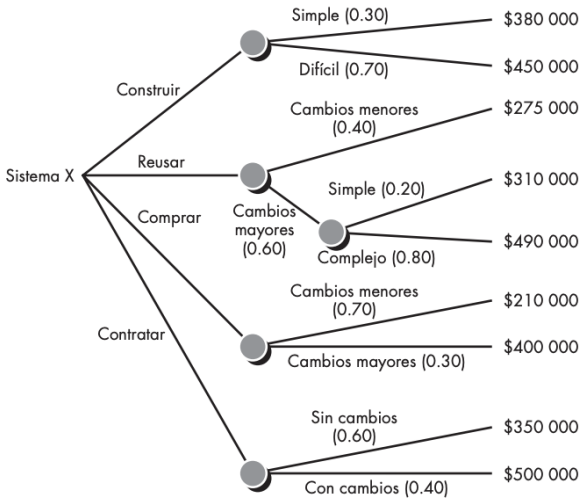

Figura 7: Árbol de decisiones para apoyar la decisión hacer/comprar [3]

Referencias

[1] Irene T. Luque Ruiz, Apuntes de la Asignatura Ingeniería del Software (2013).

[2] I. Sommervile, Ingeniería del Software, 9th Edition, Pearson, 2011.

[3] Roger S. Pressman, Ingeniería del Software, Un Enfoque Práctico, 9th Edition, Mc Graw Hill, 2021.