
Técnicas de dirección de proyectos

PID_00264673

Ramon G. Sedó
Laura Benítez García

Tiempo mínimo de dedicación recomendado: 4 horas



Ramon G. Sedó

Laura Benítez García

La revisión de este recurso de aprendizaje UOC ha sido coordinada por la profesora: Laura Porta Simó (2019)

Tercera edición: septiembre 2019
© Ramon G. Sedó, Laura Benítez García
Todos los derechos reservados
© de esta edición, FUOC, 2019
Av. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea este eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares de los derechos.

Índice

Introducción	5
1. Estructura de descomposición del trabajo o EDT (WBS –work breakdown structure–)	7
1.1. Descripción	7
1.2. Notación	8
2. Análisis de puntos función	9
2.1. IFPUG 4.4	10
3. Program Evaluation & Review Technique (PERT)	11
3.1. Descripción	11
3.2. Notación	11
3.3. Ejemplo	14
4. Método del camino crítico	18
4.1. Descripción	18
4.2. Ejemplo	18
5. El diagrama de Gantt	21
5.1. Descripción	21
5.2. Notación	21
5.3. Ejemplo	22
6. Técnicas de asignación de recursos	23
6.1. Patrón de límites	24
6.2. Histograma de recursos	25
6.3. Planificación de actividades y recursos	26
7. Diagrama de Pareto	28
8. Diagrama de causa/efecto	31
9. Gestión clásica de proyectos y nuevas maneras de plantearse la gestión de proyectos	34
9.1. Scrum. La conceptualización del producto. El día a día en un proyecto ágil	39
9.2. Kanban como sistema de gestión del trabajo en curso	42
Bibliografía	47

Introducción

La **gestión de proyectos** es un conjunto de actividades específicas que se emplean para la administración de los mismos. Estas actividades comprenden diversos aspectos:

- Estimación del esfuerzo necesario para el desarrollo de un proyecto.
- Planificación de tareas y recursos.
- Control de tareas.
- Seguimiento del proyecto.
- Control de las incidencias.
- Control de cambios.

Para el desarrollo de dichas actividades, es necesaria la utilización de técnicas específicas. Las técnicas presentadas en este módulo son las más usadas en la gestión de proyectos. La mayoría de las técnicas presentadas están basadas en las descritas en MÉTRICA versión 3, metodología desarrollada por el Ministerio de Administraciones Públicas español para el proceso de planificación de sistemas de información. El resto de las prácticas están basadas en otras metodologías de la gestión de proyectos, como el Project Management Body of Knowledge (PMBOK), del Project Management Institute (PMI).

1. Estructura de descomposición del trabajo o EDT (WBS –*work breakdown structure*–)

La técnica en la que se basa la planificación de la mayoría de los proyectos es la técnica denominada *estructura de descomposición del trabajo* o EDT¹.

⁽¹⁾En inglés, WBS (*work breakdown structure*).

A partir de la EDT, se deducen los diagramas PERT y Gantt que permiten la planificación del proyecto.

En la organización de un proyecto, esta técnica permite estructurar las actividades y sirve de **lista de comprobación** de las tareas que hay que realizar en el proyecto, así como de herramienta de contabilidad analítica del mismo. Si a cada tarea se le asigna su coste correspondiente, permite llevar un control contable del proyecto.

1.1. Descripción

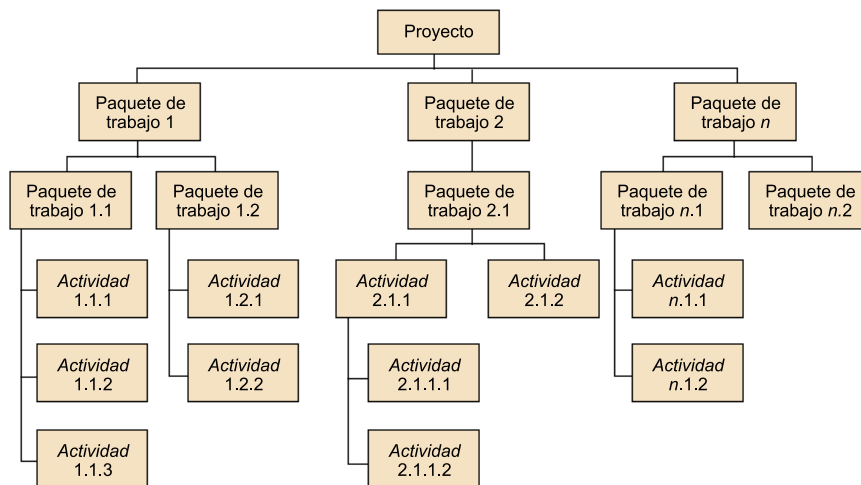
La EDT presenta una descomposición de las actividades de un proyecto según su naturaleza. Se representa como un gráfico jerárquico, un árbol, un agrupador de actividades: desarrollo, calidad, gestión, etc.

Los grupos de actividades sirven de soporte para el seguimiento del proyecto, tanto en tiempo como en costes, y constituyen un histórico útil para proyectos futuros.

El desglose del trabajo en grupos de actividades se realiza hasta que se obtienen actividades que permitan entender el trabajo que hay que llevar a cabo y la estimación de los tiempos y los costes de la realización de dicho trabajo.

El gráfico está ordenado de menor a mayor detalle. Lo construye el equipo de proyecto y asegura que todos conozcan el trabajo requerido.

1.2. Notación



En este ejemplo las actividades o tareas del proyecto se agrupan en n grupos de actividades principales formadas por agrupaciones de actividades específicas.

El paquete de trabajo 1 está formado por dos paquetes de trabajo, 1.1 y 1.2, que engloban tres actividades en el caso del paquete 1.1 y dos actividades en el caso del paquete 1.2.

El paquete de trabajo 1 podría ser el de las actividades de análisis; el paquete 2, el de las actividades de diseño; el paquete 3, el de las de desarrollo, y así sucesivamente.

2. Análisis de puntos función

El análisis de puntos función es una de las técnicas más utilizadas en la estimación de la duración de proyectos para la web.

Las técnicas de estimación tienen como objetivo calcular el coste total del desarrollo de un proyecto.

La estimación del coste de los productos de software es una de las actividades más difíciles y propensas a error. Es difícil hacer una estimación exacta de coste al comienzo de un desarrollo debido al gran número de factores conocidos o esperados que determinan su complejidad y desconocidos o no esperados que van a producirse en cualquier momento, determinando la incertidumbre.

Las técnicas de estimación ayudan en esta tarea y dan como resultado un número de horas de esfuerzo, a partir de las cuales se calculará el coste correspondiente. La estimación nos aportará un número de horas aproximado que habrá que combinar con los recursos para obtener la planificación de actividades en el tiempo y establecer los hitos del proyecto.

Las técnicas de estimación más fiables se basan en el **análisis de puntos función**. La técnica de puntos función permite la evaluación de un sistema de información a partir de un mínimo conocimiento de las funcionalidades y las entidades que intervienen en él.

Las características más destacables de esta técnica son las siguientes:

- es una unidad de medida empírica,
- considera el sistema como un todo que se divide en determinadas funciones,
- es independiente del entorno tecnológico en el que se haya de desarrollar el sistema,
- es independiente de la metodología que vaya a ser utilizada,
- es independiente de la experiencia y del estilo de programación y
- es fácil de entender por el usuario.

El resultado de la aplicación de esta técnica viene dado en **puntos función**, que posteriormente habrán de ser pasados a días de esfuerzo, para lo que sí que deberán tenerse en cuenta la experiencia del equipo de desarrollo y el estilo de programación, la aplicación de una u otra metodología y la tecnología.

Este cálculo de días por puntos función debe basarse en la experiencia adquirida en la valoración y la realización de sistemas anteriores, de manera que, con posterioridad a la finalización de cada proyecto, se debe actualizar el valor de conversión.

Entre las técnicas de estimación basadas en el análisis de puntos función, el **IFPUG 4.4** (*unadjusted functional size measurement method*), también conocido como la norma ISO/IEC 20926:2003 Software engineering, definido por el International Function Point Users Group y evolucionado a partir de la propuesta original de Allan Albrecht en IBM, se está convirtiendo en el estándar de facto en el cálculo de puntos función.

2.1. IFPUG 4.4

Ha sido definido por sus creadores como el "método estándar para medir el desarrollo de software desde el punto de vista del usuario".

El método se basa principalmente en la identificación de los componentes del sistema informático en términos de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio. A cada uno de estos componentes, les asigna un número de puntos por función basándose en el tipo de componente y su complejidad. El sumatorio de esto nos da los puntos de función sin ajustar. El ajuste es un paso final basado en las características generales de todo el sistema informático que se contabiliza.

3. Program Evaluation & Review Technique (PERT)

El objetivo del PERT² es establecer las dependencias entre las distintas tareas del proyecto para saber de qué manera han de encadenarse dichas tareas en la planificación. Estas dependencias o prelacións se establecen a partir de las precedentes.

⁽²⁾Program Evaluation & Review Technique.

3.1. Descripción

El método PERT parte de la descomposición del proyecto en una serie de obras parciales o actividades, entendiendo por **actividad** la ejecución de una tarea que exige para su realización la utilización de recursos tales como recursos humanos, software y hardware...

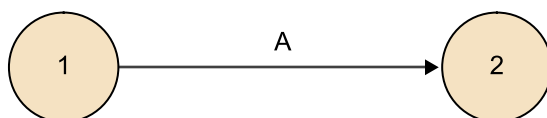
Así, por ejemplo, la elaboración del *briefing*, la recopilación de información, el desarrollo de contenidos, el diseño de la arquitectura del sitio web, el desarrollo gráfico, la programación, la contratación del servicio de hospedaje (*hosting*) y su configuración, la etapa de pruebas y el lanzamiento son actividades que forman parte del proyecto de construcción de un sitio web.

Después del concepto de actividad, el método PERT establece el concepto de suceso. Un **suceso** es un acontecimiento, un punto en el tiempo, una fecha en el calendario. El suceso no consume recursos; sólo indica el principio o el fin de una actividad o de un conjunto de actividades.

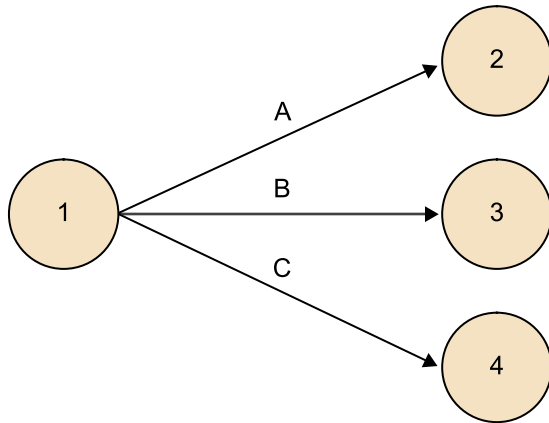
3.2. Notación

Para representar las diferentes actividades en las que se descompone un proyecto, así como sus sucesos correspondientes, se utiliza una **estructura de grafo**. Los arcos del grafo representan las actividades y los vértices, los sucesos.

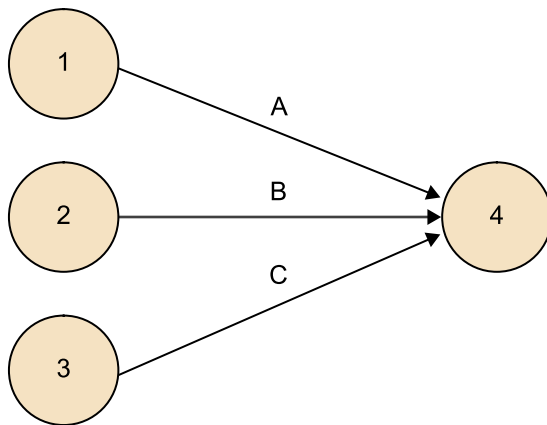
Así, el vértice 1 del diagrama que aparece a continuación indica el suceso inicio de la actividad A y el vértice 2, el suceso fin de dicha actividad. El arco que une los vértices 1 y 2 representa la propia realización de la actividad.



Por otra parte, como se ha dicho, un suceso puede representar el principio o el fin de un conjunto de actividades. En efecto, en la figura siguiente el vértice 1 representa el suceso inicio de las actividades A, B y C.

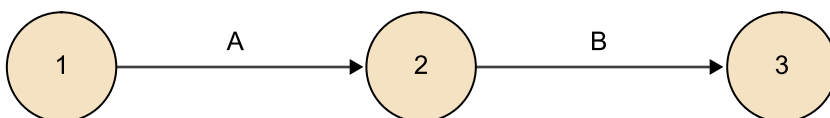


Y en la figura que aparece a continuación, el vértice 4 representa el suceso fin de las actividades A, B y C.

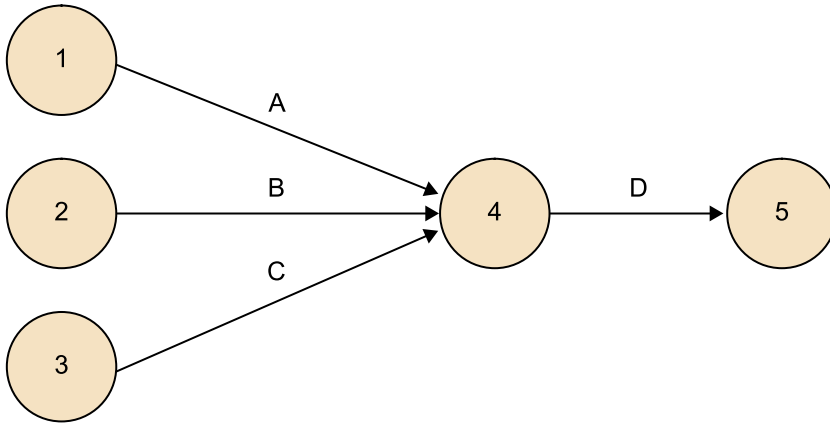


Una vez descompuesto el proyecto en actividades, la fase siguiente del método PERT consiste en establecer las prelacións existentes entre las diferentes actividades. Estas prelacións indican el orden en el que deben ejecutarse dichas actividades. El caso más sencillo son las **prelacións lineales**, que se presentan cuando, para poder iniciar una actividad, es necesario que haya finalizado previamente una única actividad (la precedente).

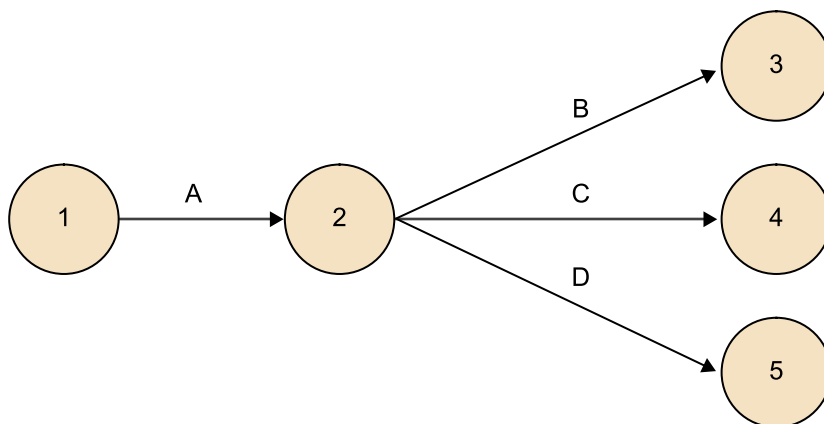
Observando la figura siguiente, se aprecia que para poder iniciar la actividad B es necesario que haya finalizado la actividad A. Es decir, el vértice 2 representa el suceso fin de la actividad A y, a la vez, el suceso comienzo de la actividad B.



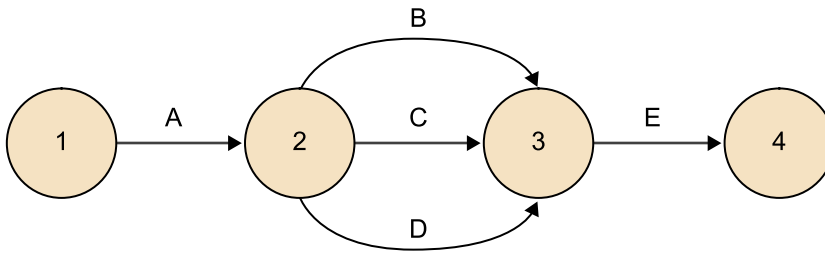
A continuación, estudiaremos el caso de las prelaiones que originan una **convergencia**. En el diagrama siguiente se representa este caso. Para poder iniciar la actividad D es necesario que se hayan finalizado las actividades A, B y C. Es decir, el vértice 4 representa el suceso fin de las actividades A, B y C y, a la vez, el suceso comienzo de la actividad D.



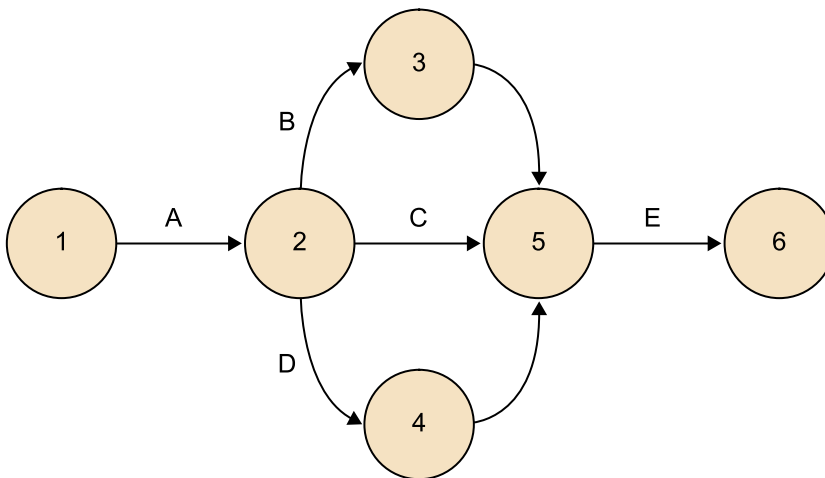
El caso opuesto al anterior es el de las **prelaiones que originan una divergencia**, representado en el diagrama que aparece a continuación. En este caso, para poder iniciar las actividades B, C y D es necesario que se haya finalizado previamente la actividad A.



Consideremos ahora las prelaiones siguientes: la actividad A es anterior a las actividades B, C y D, y dichas actividades (B, C y D) son anteriores a la actividad E. Una forma de reflejarlas es la que viene representada en el diagrama siguiente. En él se muestran las prelaiones anteriores correctamente. Pero esta representación puede resultar confusa, ya que las actividades B, C y D tienen el mismo origen y el mismo destino.



Para resolver este problema, se puede recurrir a las actividades ficticias, como se muestra en el diagrama siguiente, en el que se distinguen perfectamente las actividades B, C y D, ya que las tres actividades, aun naciendo en el mismo vértice, mueren en vértices distintos.



3.3. Ejemplo

Mediante el ejemplo siguiente, se puede realizar la construcción del diagrama PERT de un proyecto completo. Para ello, debe comenzarse por introducir los conceptos de suceso inicio del proyecto y de suceso fin del proyecto.

Se entiende por **suceso inicio del proyecto** aquel que, representando el comienzo de una o varias actividades, no representa, sin embargo, el fin de ninguna. Por el contrario, el **suceso fin del proyecto** es aquel que, representando el fin de una o varias actividades, no representa, sin embargo, el comienzo de ninguna otra actividad. A este suceso se le reconoce en el grafo, puesto que está representado por el único vértice al que llegan, pero del que no salen, arcos.

A continuación, se va a construir el grafo PERT de un proyecto cuyas actividades y las prelacións existentes entre ellas son las siguientes:

- A precede a C, D y E.
- B precede a C.
- C precede a K.
- D precede a F y G.
- E precede a J.

- F precede a I.
- G precede a H.
- H, I y J preceden a L.
- K precede a M.
- L precede a P.
- M precede a N.
- N y P preceden a Q.
- Q precede a R.

Para construir el grafo PERT, el primer paso debe ser recoger de una manera sistematizada la información contenida en el anterior conjunto de prelación-nes. Para ello, existen básicamente dos procedimientos: la matriz de encadenamientos y el cuadro de prelación. La **matriz de encadenamientos** consiste en una matriz cuadrada cuya dimensión es igual al número de actividades en las que se ha descompuesto el proyecto. Cuando un elemento de dicha matriz aparece marcado con una X, ello indica que para poder iniciar la actividad que corresponde a la fila que cruza ese elemento, es necesario que se haya finalizado previamente la actividad que corresponde a la columna que cruza dicho elemento.

	Actividades precedentes																
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R
A																	
B																	
C	X	X															
D	X																
E	X																
F				X													
G				X													
H							X										
I						X											
J					X												
K			X														
L								X	X	X							
M											X						
N													X				
P												X					
Q														X	X		

	Actividades precedentes																
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R
R																X	

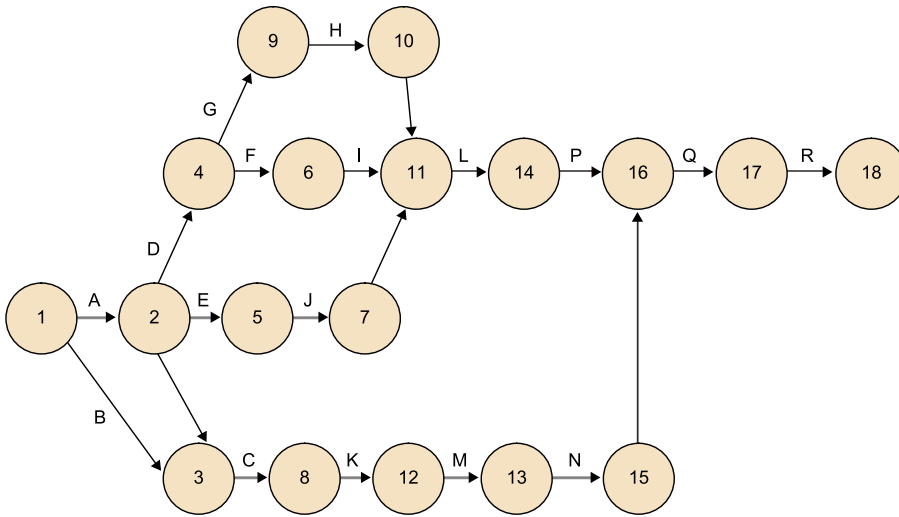
Es interesante observar que aquellas filas de la matriz en las que no aparece ninguna X indican las actividades que no tienen ningún precedente. Se trata de aquellas actividades cuyo suceso inicial coincide con el suceso inicio del proyecto (actividades A y B). Por otra parte, aquellas columnas en las que no aparece ninguna X indican las actividades que no tienen ninguna actividad siguiente, es decir, aquellas actividades cuyo suceso final coincide con el suceso fin del proyecto (actividad R).

El cuadro de prelación está formado por dos columnas. En la primera columna están representadas todas las actividades en las que se ha descompuesto el proyecto. En la segunda columna figuran las actividades precedentes de su homóloga en la primera columna. Con los datos del ejemplo se elabora el cuadro de prelación, que es el siguiente:

Actividad	Precedente
A	
B	
C	A, B
D	A
E	A
F	D
G	D
H	G
I	F
J	E
K	C
L	H, I, J
M	K
N	M
P	L
Q	N, P
R	Q

La actividad o actividades inicio del proyecto se reconocen en el cuadro de prelación por no tener ninguna actividad precedente (actividades A y B). La actividad o actividades fin del proyecto se reconocen por no aparecer en la segunda columna de dicho cuadro (actividad R).

A partir de la matriz de encadenamientos o del cuadro de prelación, se construye fácilmente el grafo PERT correspondiente. Para los datos del ejemplo, el grafo PERT está representado en el diagrama siguiente.



4. Método del camino crítico

El método del camino crítico está formado por una serie de tareas que se deben completar a tiempo para que un proyecto finalice conforme a la programación.

4.1. Descripción

La mayoría de las tareas de un proyecto presentan un cierto tiempo de demora (holgura) y, por lo tanto, se pueden retrasar ligeramente sin afectar a la fecha de fin del proyecto. Las tareas que no se pueden retrasar sin que se modifique la fecha de fin del proyecto se denominan *tareas críticas*.

Para calcular la holgura de las tareas, el método del camino crítico calcula todas las fechas de inicio y finalización teóricas tempranas y tardías para todas las actividades de la red sin considerar limitaciones de recursos.

Las **fechas tempranas** indican la primera fecha en la que se puede iniciar o finalizar una tarea en función de sus relaciones de precedencia o en función de restricciones temporales. Las **fechas tardías** indican la última fecha en la que se puede iniciar o finalizar una tarea en función de estas mismas relaciones y restricciones.

La holgura se determinará realizando un análisis de recorrido hacia delante y un análisis de recorrido hacia atrás.

Las fechas de inicio y finalización tempranas y tardías resultantes indican los períodos dentro de los cuales debería programarse la actividad.

Las tareas que forman parte del denominado *camino crítico* son las tareas sin holgura, las tareas críticas, cuya modificación para resolver sobreasignaciones u otros problemas de la programación afectarán a la fecha de fin del proyecto.

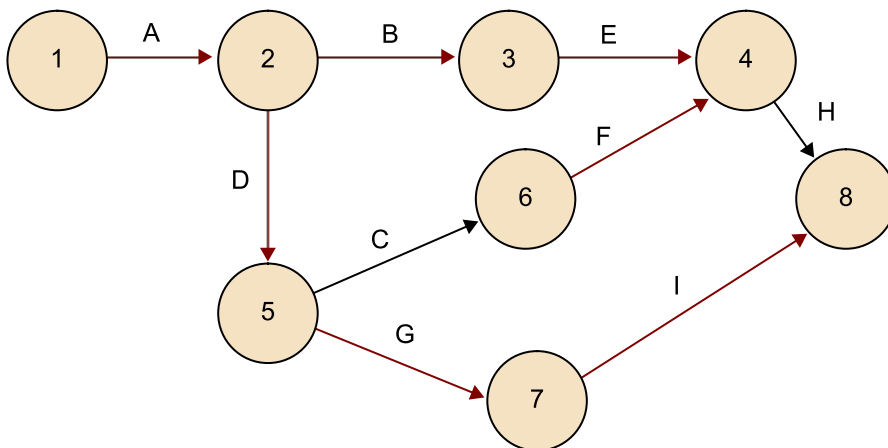
En un proyecto el camino crítico es el recorrido de mayor duración y puede no ser único, es decir, en un mismo proyecto podemos tener más de un camino crítico.

4.2. Ejemplo

Una empresa desea desarrollar un nuevo sitio web para su portal de empleados. Las tareas descritas para realizar este desarrollo y sus duraciones son las siguientes:

Tarea	Después de	Duración de la tarea
A	-	7 días
B	A	3 días
C	D	2 días
D	A	4 días
E	B	2 días
F	C	2 días
G	D	6 días
H	E-F	1 día
I	G	5 días

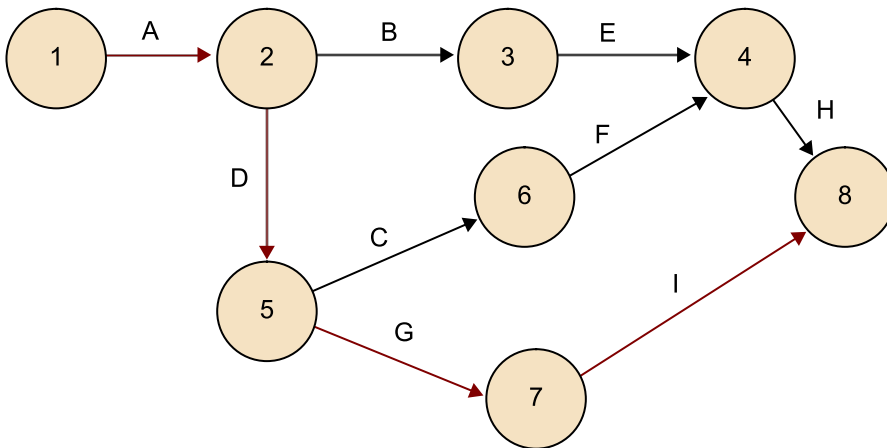
El grafo PERT del proyecto es el siguiente:



Los posibles caminos para llegar del nodo 1 al nodo 8 son los siguientes:

- A-D-G-I = 7 + 4 + 6 + 5 = 22 días
- A-D-C-F-H = 7 + 4 + 2 + 2 + 1 = 16 días
- A-B-E-H = 7 + 3 + 2 + 1 = 13 días

Así pues, tenemos que el camino crítico es el formado por las tareas A, D, G e I.



Si pintamos las tareas dentro de un cronograma obtenemos lo siguiente:

A	█																									
B								█																		
C													█													
D								█																		
E												█														
F														█												
G																█										
H																			█							
I																						█				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22				

Las tareas B, C, E, F y H de este proyecto muestran cierta holgura, puesto que pueden presentar algunas modificaciones temporales sin afectar a la fecha de finalización del proyecto. En cambio, cualquier modificación en la duración de las tareas A, D, G e I influirían directamente en la alteración de la fecha de fin de proyecto.

5. El diagrama de Gantt

El **diagrama de Gantt** o **cronograma** tiene como objetivo la representación del plan de trabajo, mostrando las tareas que se han de realizar, el momento de su comienzo y su terminación y la forma en la que las distintas tareas están encadenadas entre sí.

5.1. Descripción

El gráfico de Gantt es la forma habitual de presentar el plan de ejecución de un proyecto. Recoge en las filas la relación de actividades que hay que realizar y en las columnas, la escala de tiempos que se manejan, mientras que la duración y la situación en el tiempo de cada actividad se representan mediante una línea dibujada en el lugar correspondiente.

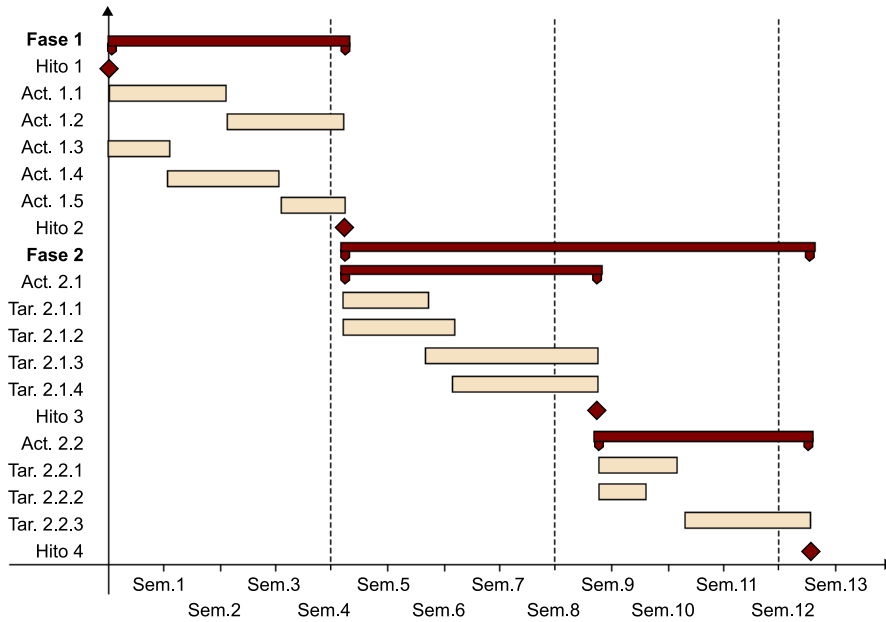
5.2. Notación

- Es un modelo realizado sobre ejes de coordenadas, donde las tareas se sitúan sobre el eje de ordenadas (y) y los tiempos, sobre el de abscisas (x).
- Cada actividad se representa con una barra limitada por las fechas previstas de comienzo y fin.
- Las actividades se agrupan en fases y pueden descomponerse en tareas.
- Cada actividad debe tener recursos asociados.
- Los **hitos** son un tipo de actividad que no representa trabajo ni tiene recursos asociados.
- Las actividades se pueden encadenar por dos motivos:
 - **Encadenamiento funcional o por prelación.** Por ejemplo, un programa no puede probarse hasta que no haya sido escrito.
 - **Encadenamiento orgánico o por ocupación de recursos.** Por ejemplo, un programador no puede empezar un programa hasta que no haya terminado el anterior.

Pueden realizarse actividades en paralelo siempre que no tengan dependencia funcional u orgánica.

5.3. Ejemplo

El diagrama siguiente es un ejemplo típico de un diagrama de Gantt, en el que aparecen fases, actividades, tareas e hitos.



Para que un diagrama de Gantt sea realista y fiable, debe ir acompañado de un gráfico que refleje la actividad de los técnicos (histograma de recursos) que componen el equipo del proyecto.

6. Técnicas de asignación de recursos

La **asignación de recursos** es una tarea fundamental en la planificación, ya que hay que considerar aspectos técnicos de cada recurso, como su disponibilidad, su capacidad de trabajo, sus impedimentos horarios, etc.

Es fundamental que los trabajos se descompongan hasta la unidad mínima de tratamiento, es decir, descomponer el proyecto en fases, las fases en actividades y las actividades en tareas, asignando una tarea a un recurso. No debe caerse en el error de asignar una actividad a varios recursos. De no hacerse así, es muy difícil considerar la plena ocupación de todos los recursos, y se pueden dar situaciones anómalas, como es tener un recurso una ocupación muy baja y otro una ocupación excesiva. Conviene que la planificación esté perfectamente depurada, pues de lo contrario se producen los problemas siguientes:

- Se dilata innecesariamente el plazo de entrega,
- se retiene sin trabajo a un número considerable de usuarios que podrían trabajar en otro proyecto y
- se encarece considerablemente el coste del proyecto, ya que se tienen asignados unos recursos "que no trabajan" todo el tiempo.

Por ello, es necesario tener en consideración los aspectos siguientes a la hora de asignar recursos:

- Cuantificar necesidades y fechas de incorporación de recursos.
- Obtener un patrón que marque los límites del proyecto.
- Considerar la capacidad, los conocimientos y la experiencia de cada recurso.
- Considerar la complejidad, el tamaño y los requerimientos técnicos de cada tarea.
- Asignar tareas sencillas a recursos con poca experiencia. Si se asignan tareas sencillas a recursos con mucha experiencia, se los estará infrautilizando.
- Asignar tareas complejas a recursos con mucha experiencia. Si las tareas complejas le son asignadas a recursos con poca experiencia, perderán mucho tiempo preguntando a sus compañeros y, lo que es más grave, harán perder mucho tiempo al resto del equipo.

- Construir el histograma de recursos para poder ver la coherencia de las asignaciones.
- Tratar de asignar una tarea a un único recurso, descomponiendo cuanto sea necesario.
- Vigilar que no haya vacíos en el histograma.

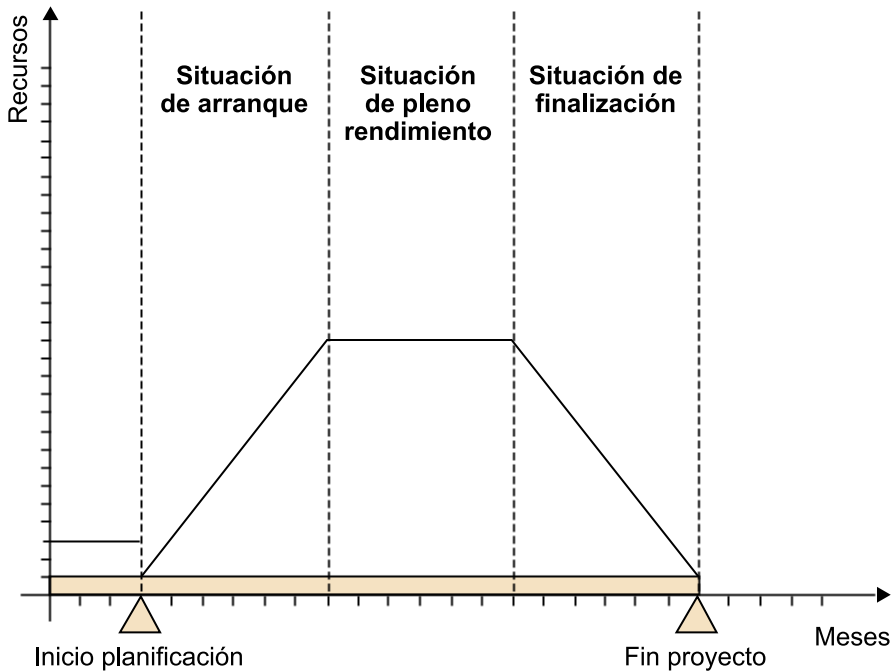
6.1. Patrón de límites

La técnica de patrón de límites tiene como objetivo establecer los límites de recursos aproximados. Para ello, una vez conocidos la estimación, el esfuerzo total y el plazo de entrega, hay que llevar a cabo las operaciones siguientes:

- Establecer el esfuerzo en meses (con decimales).
- Deducir la parte correspondiente a diseño, en la que se deciden los contenidos, los servicios y las funcionalidades, ya que es una fase con un tipo de actividades diferentes al resto.
- Establecer la duración en meses (con decimales). Normalmente este dato se conocerá por el compromiso adquirido.
- Deducir la duración correspondiente al análisis de actividades del proyecto.
- Considerar que todo proyecto tiene tres situaciones claramente diferenciadas:
 - **Arranque.** Durante esta situación se van incorporando paulatinamente recursos al proyecto hasta que se alcanza el número máximo de recursos.
 - **Pleno rendimiento.** Ésta es una situación de estabilidad en cuanto al número de recursos.
 - **Finalización.** Cuando las tareas van terminándose, gradualmente los técnicos comienzan a abandonar el proyecto.
- Seguidamente, distribuir los recursos:
 - Una vez obtenido el número de recursos medios, hay que considerar las tres situaciones que presenta el proyecto: arranque, pleno rendimiento y finalización.
 - Hay que considerar que estas tres fases tienen una duración en tiempo aproximadamente igual.
 - Se debe considerar que el jefe del proyecto está presente durante todo el ciclo de vida.
 - Se ha de extraer de la parte de diseño el esfuerzo correspondiente al jefe del proyecto.

- Hay que dividir la duración estimada (sin la parte de diseño) en tres partes iguales.
- Se tiene que distribuir la mitad del esfuerzo estimado (sin la parte de análisis) en la situación de **pleno rendimiento**.
- Hay que distribuir la otra mitad a partes iguales entre la situación de **arranque** y la de **finalización**.

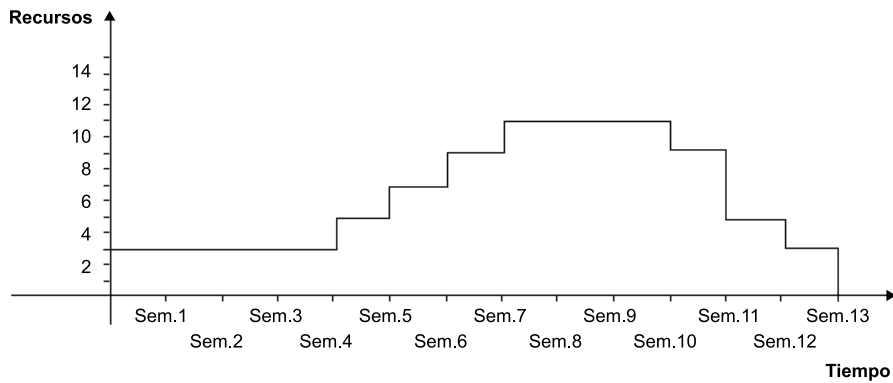
Como consecuencia de ello, se obtiene el gráfico que aparece a continuación y que servirá para establecer los límites de la asignación de recursos.



6.2. Histograma de recursos

El **histograma de recursos** es un gráfico sobre unos ejes de coordenadas, de manera que los recursos se colocan sobre el eje de las ordenadas y el tiempo, sobre el eje de las abscisas.

A medida que se incorporan recursos al proyecto, el gráfico aumenta y, al contrario, disminuye cuando son desasignados.



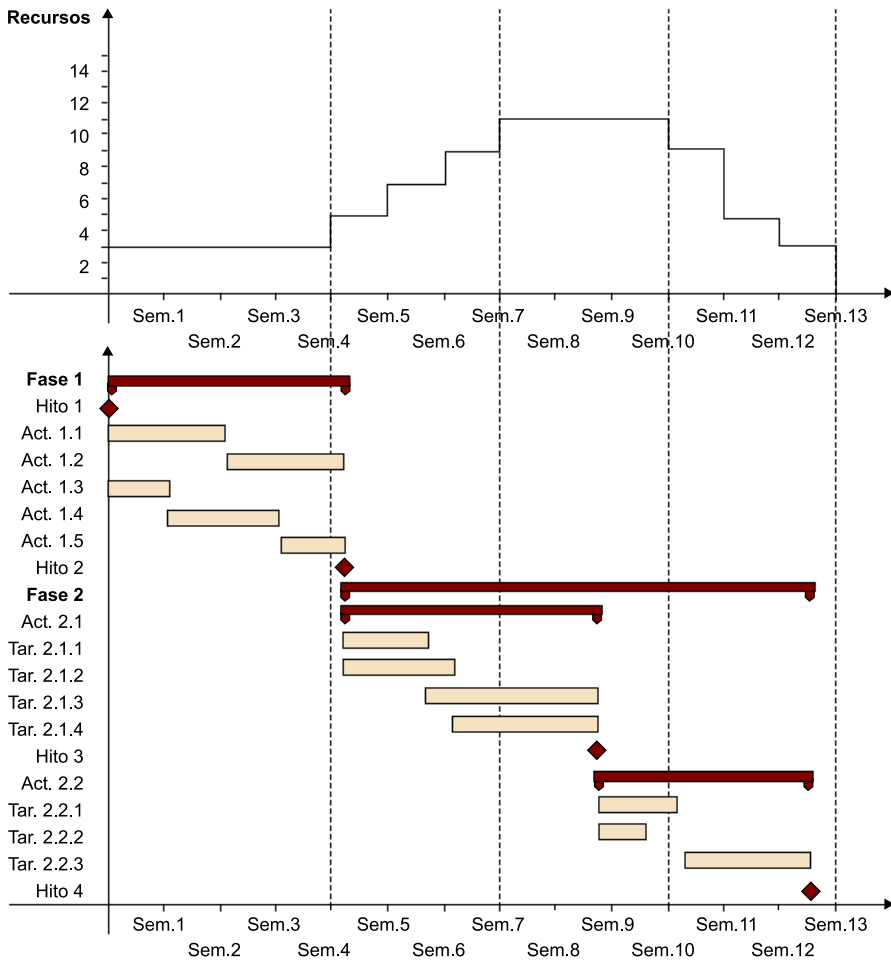
Ejemplo gráfico de un histograma de recursos

6.3. Planificación de actividades y recursos

Para realizar adecuadamente la planificación, es necesario observar lo siguiente:

- Deben construirse paralelamente el diagrama de Gantt y el histograma de recursos.
- Han de realizarse los encadenamientos funcionales a partir del PERT.
- Los recursos deben asignarse:
 - con criterio de capacidad profesional y
 - encadenando sus actividades orgánicamente.
- Ha de considerarse el máximo de recursos que proporciona el patrón de límites para llevar a cabo el histograma de recursos.
- El histograma de recursos deberá reflejar con exactitud los recursos utilizados en el diagrama de Gantt.

A continuación, podemos ver gráficamente cómo han de realizarse la planificación de actividades y la de recursos.



7. Diagrama de Pareto

El **diagrama de Pareto** es una herramienta de representación gráfica que identifica los problemas más importantes en función de su frecuencia de ocurrencia o coste (dinero, tiempo...) y que permite establecer las prioridades de intervención. En definitiva, es un tipo de distribución de frecuencias que se basa en el **principio de Pareto**, a menudo denominado *regla 80/20*, que indica que el 80% de los problemas son originados por un 20% de las causas. Este principio ayuda a separar los errores críticos, que normalmente suelen ser pocos, de los muchos errores no críticos o triviales.

El diagrama de Pareto comunica de forma clara y evidente el resultado del análisis de comparación y priorización de los problemas más importantes.

La construcción del diagrama de Pareto consta de las etapas siguientes:

- 1) **Decidir cómo clasificar los datos.** Se pueden clasificar por tipo de defecto (forma muy usual de hacerlo), por máquina, por fase del proceso, por turno, etc.
- 2) **Determinar el tiempo de recogida de los datos.** Este tiempo de recogida puede hacerse en términos de horas, días, semanas o meses.
- 3) **Obtener los datos y ordenarlos.** En esta fase se debe preparar la hoja de recogida de datos.
- 4) **Dibujar los ejes de coordenadas.** Se colocan, en el eje vertical, la escala de medida de las frecuencias o el coste y, en el eje horizontal, las causas en orden decreciente de la unidad de medida.
- 5) **Dibujar el diagrama.** Es la representación gráfica de los datos recogidos en la hoja.
- 6) **Construir una línea de frecuencia acumulada.**
- 7) **Hacer el análisis de Pareto.** El diagrama pone de relieve los problemas más importantes sobre los que será necesario actuar.

En este ejemplo de diagrama de Pareto, vemos que el 80% de los errores están causados por errores de tipo E, B y C.

Tabla de Pareto			
Tipo de error	Número de errores	% del total	% acumulado del total
E	124	38,0	38,0
B	83	25,5	63,5
C	55	16,9	80,4
F	28	8,6	89,0
D	18	5,5	94,5
A	8	2,5	96,9
G	5	1,5	98,5
H	5	1,5	100
Total	326	100	

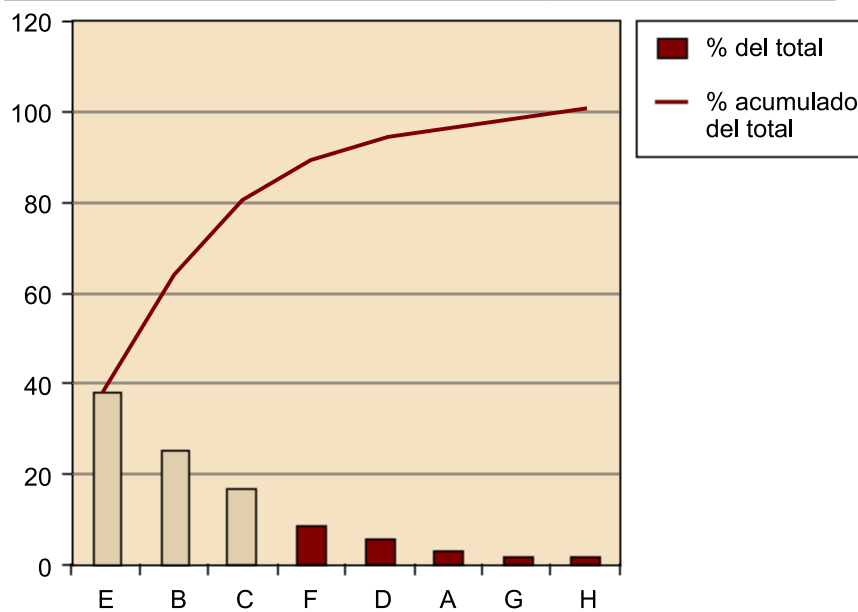


Diagrama de Pareto

Los diagramas de Pareto permiten identificar los mayores problemas y generar nuevos diagramas de Pareto individuales para ellos.

Si se emprenden acciones correctoras, debemos dibujar los diagramas de Pareto antes y después con objeto de comprobar los resultados alcanzados.

Por otro lado, siempre resulta muy útil realizar el análisis observando el coste de los defectos en términos monetarios, sobre todo si se pretenden reducir los costes de la no calidad. Para ello, construimos el diagrama de Pareto en términos de coste de eliminación de cada uno de los defectos o en términos de pérdidas económicas que suponen cada uno de los defectos. Esta forma de proceder nos permite conocer si la identificación y la eliminación de los problemas o defectos posibilita que alcancemos enormes beneficios o, al menos,

que no incurramos en grandes pérdidas. En ocasiones, una cantidad pequeña de defectos provoca grandes pérdidas mientras que, por el contrario, una gran cantidad de defectos puede provocar pérdidas bastante reducidas.

La utilización de esta herramienta presenta las ventajas siguientes:

- Permite observar los resultados de las acciones de mejora implantadas al comparar dos diagramas del mismo fenómeno en momentos distintos de tiempo.
- Es una herramienta polivalente y fácilmente aplicable, no sólo en el control de la calidad, sino en cualquier ámbito.
- Utilizada en presentaciones y reuniones, aumenta la eficacia y la rapidez de la comunicación, ya que permite identificar rápidamente y a simple vista el problema más grave.

8. Diagrama de causa/efecto

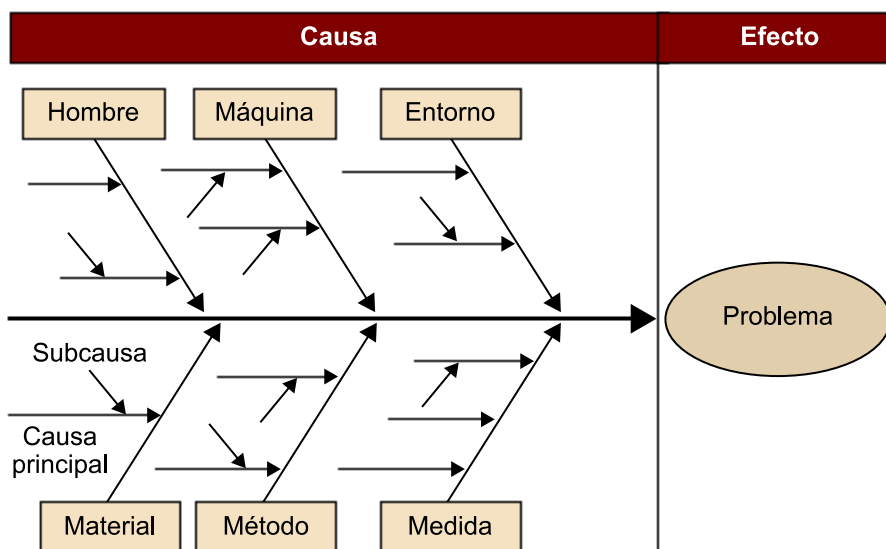
El **diagrama Ishikawa** o **diagrama de causa/efecto** se utiliza para recoger de manera gráfica todas las posibles causas de un problema o identificar los aspectos necesarios para alcanzar un determinado objetivo (efecto).

Kaoru Ishikawa

El ingeniero japonés Kaoru Ishikawa es considerado el padre del análisis científico de las causas de problemas en procesos industriales y dio nombre al diagrama Ishikawa, cuyos gráficos agrupan por categorías todas las causas de los problemas.

El uso de este diagrama, de gran impacto visual, muestra las interrelaciones entre un efecto y sus posibles causas de forma ordenada, clara, concisa y de un solo vistazo, permitiendo una mejor comprensión del efecto en estudio, incluso en situaciones muy complejas, puesto que centra la atención de todos los componentes del grupo de forma estructurada y sistemática.

Por su aspecto, también se denomina *diagrama de espina de pescado (fishbone)*.



Fuente: Wikipedia Commons. Autor: VARGUX

Para desarrollar el diagrama de causa/efecto se deben seguir los pasos siguientes:

1) Definir y determinar claramente el problema o efecto que va a ser analizado. El problema es algo que generalmente queremos mejorar o controlar. Deberá ser específico y concreto: incumplimiento con las fechas de instalación, cantidades inexactas en la facturación, errores técnicos, etc.

2) Identificar los factores o causas que originan el efecto. Ello se hace mediante una tormenta de ideas (*brainstorming*) a partir de la cual se clasifican estas causas según su categoría o naturaleza. Para clasificar las causas encontradas, a menudo se utilizan como referencia las categorías de las **cuatro emes**, definidas por Ishikawa: mano de obra, maquinaria, materiales y métodos. Estas categorías son los rótulos de las espinas.

3) Representación del diagrama. Una vez enumeradas todas las causas, debemos ir colocándolas en el diagrama agrupando las que sean de naturaleza similar.

4) Análisis de las relaciones de causa/efecto que derivan de la construcción del diagrama. Una vez identificadas las causas, debemos identificar las "causas más probables", puesto que no todas las causas que aparecen durante la tormenta de ideas están estrechamente relacionadas con el problema.

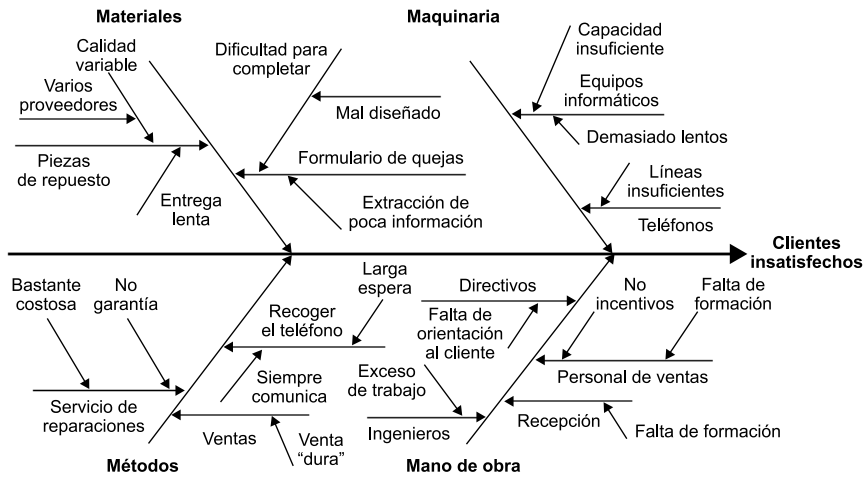
Entre otras aplicaciones, este diagrama puede utilizarse para conocer y afrontar las causas de los defectos, las anomalías o las reclamaciones; reducir costes; obtener mejoras en los procesos; mejorar la calidad de los productos, los servicios y las instalaciones, y establecer procedimientos normalizados (tanto operativos como de control).

A pesar de la aparente sencillez de esta herramienta, su aplicación presenta una serie de ventajas como las siguientes:

- proporciona una metodología racional para la resolución de problemas,
- permite sistematizar las posibles causas de un problema y
- favorece el trabajo en equipo posibilitando que los trabajadores planteen de forma creativa sus opiniones y que la comunicación sea clara y eficaz.

Esta técnica de identificación de las posibles causas de un problema no aporta soluciones. El diagrama ayuda a pasar de opiniones a teorías comprobables, pero sólo la recopilación de datos sobre las causas más probables aportará información de cuál es la solución que se ha de tomar para resolver el efecto o problema.

Ejemplo de diagrama de espina de pescado para "clientes insatisfechos"



9. Gestión clásica de proyectos y nuevas maneras de plantearse la gestión de proyectos

La gestión de proyectos es un conjunto de actividades específicas que se usan para administrar el proyecto.

Este módulo trata de la gestión de proyectos desde un punto de vista clásico, pero hay enfoques y planteamientos nuevos que intentan evitar los errores más habituales que se cometen en la gestión de proyectos: las **metodologías ágiles**.

La complejidad del proyecto multimedia

Cabe recordar que los **productos multimedia**, en general, son proyectos complejos y se basan en la incertidumbre por definición. Aplicado al entorno del desarrollo de productos multimedia, podemos decir que un **proyecto es complejo** cuando tanto la tecnología que se debe utilizar para resolver un problema como los conocimientos sobre los requisitos del proyecto no son conocidos de una manera clara al principio del proceso.

Los requisitos tienden a ser complejos con mucha facilidad. De hecho, podemos hablar de requisitos simples si:

- El cliente los puede capturar todos y transmitirlos al desarrollador, y este es capaz de entenderlos completamente.
- No hay diferentes partes interesadas en el proyecto con intereses divergentes.
- El cliente sabe exactamente lo que necesita.

Como podemos ver, esto es difícilmente alcanzable. Además, debemos tener presente que los productos multimedia son desarrollados por personas que trabajan en equipo. Si las personas ya son complejas desde el punto de vista individual, esta complejidad se incrementa cuando trabajamos en grupo.

En las metodologías, también llamadas *predictives*, se ha intentado luchar contra esta complejidad de modo que tanto los requisitos como la tecnología # los dos, aspectos inherentes al desarrollo de productos multimedia# desapa-

recieran de la ecuación, con la obtención de fases muy grandes de toma de requisitos, largos análisis de estos requisitos y contratos demasiado cerrados y restrictivos como resultado.

Aun con los esfuerzos efectuados, es difícil luchar contra la complejidad de los productos multimedia y estas aproximaciones no han comportado la realización de proyectos más buenos, sino tener proyectos con problemas.

Problemas en el desarrollo de productos multimedia

Los factores habituales por los que un producto multimedia falla son los siguientes:

- Objetivos del proyecto poco realistas o poco articulados.
- Estimaciones poco acertadas de los recursos necesarios para llevar a cabo el proyecto.
- Requisitos del sistema mal definidos.
- Información pobre sobre el estado del proyecto.
- Riesgos no gestionados
- Comunicación insuficiente entre los usuarios, los desarrolladores y los clientes.
- Utilización de tecnología poco madura.
- Incapacidad de gestionar la complejidad del producto.
- Prácticas de desarrollo descuidadas.
- Gestión ineficaz del proyecto.
- Presiones comerciales.

En un estudio realizado por BCS el año 2017, podemos ver que, de los 214 proyectos analizados de todos los ámbitos llevados a cabo con metodologías predictivas, el 23,8% fueron cancelados y el 32,2 se acabaron por encima del presupuesto o del tiempo previsto. Esto quiere decir que cerca de un 55% de los proyectos no tuvieron el final deseado. Otros estudios corroboran estos datos. Si tuviéramos presente la satisfacción final del cliente con el producto que se entrega, seguramente estos porcentajes todavía se incrementarían más.

Algunos de estos problemas son especialmente típicos y preocupantes:

- El cliente tarda mucho tiempo a poder utilizar el resultado del proyecto; mientras, el contexto cambia y los competidores sacan nuevos proyectos. Si se cancela el proyecto, el cliente habrá gastado el dinero para nada.
- El proyecto se ha complicado más de lo que se podía esperar: hay retraso y se deben acelerar las entregas. Empiezan los parches y no hay tiempo para pruebas ni para el control de calidad.
- El equipo hace horas extra y está poco motivado: simplemente se dedica a cumplir órdenes. Cada uno va a la suya.
- ¡El cliente pide cambios! Estos cambios son difíciles de realizar, ya que el proyecto se está acabando y no hay tiempo ni presupuesto suficientes.
- Todo el mundo acaba insatisfecho, tanto el cliente como el equipo.

¿Por qué aparecen estos problemas?

Por un lado, las diferentes fases se alargan más de lo previsto, y siempre salen perjudicadas las fases de pruebas y de posibles cambios.

Por el otro lado, el cliente solo ha ido viendo papel (documentos de diseño) hasta una fase muy avanzada del proyecto. El cliente va a ciegas, gasta dinero en una cosa que no tiene ningún tipo de aplicación. Son documentos, no los pueden utilizar.

Así mismo, siempre aparecen las discusiones entre el cliente y el proveedor. Uno considera que el otro no ha entendido lo que quería y que no se están cumpliendo los plazos previstos. El otro piensa que lo que se le pide no está al alcance del proyecto y que pierde dinero. Todo esto hace que se dedique mucho esfuerzo a objetivos que aportan poco valor.

Es decir, en el desarrollo de productos multimedia se cumple perfectamente el *principio de Pareto*, según el cual el 80% del valor de un producto proviene del 20% de las funcionalidades implementadas. Por lo tanto, invertimos mucho tiempo, esfuerzo y dinero en cosas que no se utilizarán. ¿Debemos hacer pagar esto al cliente?

Para muchos, parece claro que este no es el mejor modo de desarrollar un producto multimedia, por eso surge el desarrollo ágil de proyectos.

El desarrollo ágil

En el año 2011, diecisiete ingenieros de alto nivel en el campo del desarrollo de *software* se reunieron en Utah para explorar y compartir lo que consideraban que tenía que ser el futuro del desarrollo de *software*. Dentro del grupo, había

muchos de los que proponían metodologías emergentes en aquel momento, como Scrum, Extreme Programming, Crystal, Feature Driven Development, entre otros. Juntos, coincidieron en el nombre para su movimiento: *agile* (ágil).

En esta reunión, los integrantes constituyeron la Agile Alliance (Alianza Ágil), y escribieron el *Manifiesto for agiles software development* (popularmente, *The agile manifesto*; en castellano, *El manifiesto ágil*): un conjunto de estamentos que han servido como declaración de valores y principios que deben seguir las metodologías ágiles.

Personas e interacciones por encima de procesos y herramientas.

Software que funciona por encima de documentación exhaustiva.

Colaboración con el cliente por encima de negociación de contratos.

Respuesta al cambio por encima del seguimiento de una planificación.

Los valores ágiles parecen simples y obvios la primera vez que uno los escucha, y continúan sin modificarse desde el día que los fundadores de la Alianza Ágil los publicaron como parte del *Manifiesto ágil*.

Estos doce puntos son los principios sobre los que se basa el desarrollo de software ágil, una buena práctica en los equipos que desarrollan por medio de metodologías ágiles, excepto la de revisar periódicamente la lista de principios para asegurarse de que se continúan siguiendo.

Principios del Manifiesto ágil

- 1) Nuestra prioridad principal es satisfacer al cliente mediante la entrega inicial y continua de software evaluable.
- 2) Son bienvenidos los cambios en los requisitos, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio como una ventaja competitiva del cliente.
- 3) Hay que liberar software a menudo, desde unas cuantas semanas hasta unos cuantos meses, con preferencia por la escala más corta.
- 4) La gente de negocios y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.
- 5) Construimos los proyectos en torno a personas motivadas. Les ofrecemos el entorno y el apoyo que necesitan, y confiamos en que hagan el trabajo.
- 6) El método más eficiente y efectivo para compartir información con el equipo de desarrollo son las conversaciones cara a cara.
- 7) El software que funciona es la principal medida de progreso.
- 8) Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, los desarrolladores y los usuarios deben ser capaces de mantener el ritmo indefinidamente.
- 9) La atención continua a la excelencia técnica y al buen diseño mejoran *ágile*.
- 10) La simplicidad –el arte de maximizar el trabajo no hecho– es esencial.
- 11) Las arquitecturas, los requisitos y los mejores diseños emergen de equipos auto-organizados.
- 12) En intervalos regulares, el equipo reflexiona sobre la manera de ser más efectivo y, en consecuencia, afina y ajusta el contexto.

Podéis encontrar el Manifiesto ágil, en la redacción original, en la siguiente dirección de internet: <http://agilemanifesto.org>.

Colaboración con el cliente por encima de negociación contractual

El valor ágil también busca evitar el gran diseño inicial, pero haciendo énfasis en mantener un diálogo continuo entre el equipo de desarrollo y el cliente, tan abierto y fluido como sea posible. Los contratos son buenos y necesarios, especialmente cuando protegen los intereses de las dos partes, como sucede con los buenos contratos.

Pero muchos de los fundadores de la Alianza Ágil eran consultores y conocían los riesgos que se esconden detrás de los contratos. Uno de los más habituales en nuestra industria es que el proveedor se compromete a realizar una cierta cantidad de trabajo en un tiempo determinado, en el que el provecho que saque del proyecto dependerá de la rapidez y el bajo coste con los que sea capaz de llevar a cabo los requisitos mínimos del cliente. Si alguna vez habéis querido reformar la cocina, seguramente estáis al caso de cuán perjudicial para las dos partes termina siendo este tipo de contrato.

Los autores del *Manifiesto ágil* concluyeron que un proyecto basado exclusivamente en un contrato pone el énfasis en el lugar incorrecto. Ellos prefirieron promover un entorno colaborativo, en el que cliente y equipo trabajan conjuntamente para lograr un objetivo compartido, el éxito del proyecto.

Respuesta al cambio por encima del seguimiento de una planificación

Las organizaciones que basan su modelo en “seguir el plan establecido” acostumbran a tener procesos creados con las mejores intenciones. Para prevenir que el proyecto sufra retrasos, no se permiten cambios de funcionalidad no planificados o cualquier otra función que pueda afectar el plan inicial o, si se da el caso, se someten a unos requisitos duros, como revisar toda la planificación o incluso redactar un nuevo contrato o un anexo al actual. Pero, aun así, esto no funciona. Un control de cambios solo tiene éxito en un entorno en el que los cambios son controlables. En cambio, el desarrollo de *software* es un proceso en el que, a medida que avanzamos, vamos descubriendo cosas que no sabíamos al principio, lo que nos obligará a hacer cambios no esperados.

La planificación en los proyectos de desarrollo de productos multimedia tiene que ser fluida, no fija, para el beneficio del equipo, pero sobre todo para el beneficio del producto que estamos desarrollando y, en última instancia, para el beneficio del cliente. Esta es la razón por la que planificamos para el cambio.

Dentro del ámbito de la gestión ágil de proyectos hay que destacar dos conceptos: SCRUM y Kanban.

9.1. Scrum. La conceptualización del producto. El día a día en un proyecto ágil

Scrum es un marco de trabajo para definir procesos que se caracteriza porque es ligero y fácil de entender.

Scrum no define un proceso concreto, sino que proporciona las herramientas para que cada equipo personalice el marco de trabajo y encuentre el proceso más adecuado a sus circunstancias.

Esta característica hace que Scrum sea apropiado para situaciones complejas, en la que es difícil predecir lo que pasará en el futuro y en las que, por lo tanto, será necesaria una gran capacidad de adaptación, como en el caso del desarrollo de productos multimedia.

Scrum es un método empírico y, por lo tanto, se basa en gestionar el proceso de desarrollo a partir de la experiencia (observación) y no por medio de predicciones. La consecuencia principal de esto es que las decisiones se toman teniendo en cuenta hechos conocidos, en vez de hechos hipotéticos.

Con Scrum estableceremos igualmente una predicción inicial pero durante la ejecución del proyecto nos iremos cuestionando, de una manera regular, la corrección de la predicción y la ajustaremos (por ejemplo, adaptaremos el calendario o modificaremos el conjunto de funcionalidades) para que se adecue a la realidad.

Cómo funciona Scrum

1) El *product backlog*

Un proyecto de Scrum es impulsado por una visión de producto elaborada por el propietario del producto, y se expresa en la **pila de producto**. El *product backlog* es una lista priorizada de lo que se requiere, por orden de valor para el cliente o negocio, con los elementos de más valor en la parte superior de la lista. El *product backlog* evoluciona durante la vida útil del proyecto, y los elementos se añaden, se quitan o son revaluados continuamente.

El equipo Scrum se compone de tres roles de Scrum:

- **El propietario del producto.** Analiza las entradas de lo que debe ser el producto y lo traduce en una visión de producto.
- **El equipo.** Desarrolla el producto previsto por el propietario del producto.
- **El Scrum máster.** Tiene todo lo que se necesita para que el equipo Scrum logre el éxito. Esto pasa por eliminar los obstáculos de organización, facilitar las reuniones y actuar como un guardián para que nadie obstaculice el trabajo del equipo.

2) El *sprint*

Scrum estructura el desarrollo de productos en ciclos de trabajo llamados *sprints*, repeticiones de trabajo que normalmente duran entre una y cuatro semanas. Los *sprints* tienen una duración determinada; nunca se extienden más allá de la fecha fijada al principio, independientemente de que el trabajo planificado por el *sprint* se haya completado o no.

3) Planificación del *sprint*

A comienzos de cada *sprint*, se lleva a cabo la **reunión de planificación del *sprint***. El propietario del producto y el equipo Scrum (con la facilitación del Scrum máster) revisan la **pila de producto** y discuten los objetivos y el contexto de las historias de usuario, mientras que el equipo Scrum selecciona los elementos de la pila de producto que se comprometen a hacer para el final del *sprint*, partiendo de la parte superior de la pila de producto, donde se hallan los elementos de priorización más alta.

Cada elemento seleccionado en la pila de producto se descompone después en una serie de tareas individuales. La lista de tareas se registra en un documento llamado *sprint backlog*.

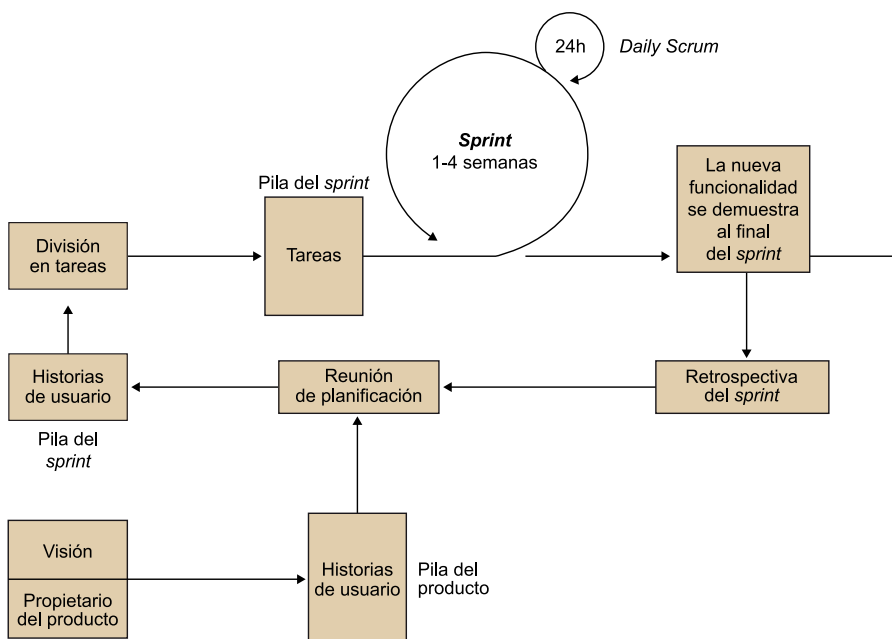
4) Reunión diaria de Scrum

Una vez que ha empezado el *sprint*, el equipo Scrum se dedica a otra de las prácticas principales de Scrum: la **reunión diaria de Scrum**, o *daily stand-up* en la denominación original en inglés. Se trata de un encuentro breve (de quince minutos, como máximo) que se lleva a cabo cada día de trabajo a una hora determinada. Todos los miembros del equipo asisten a él. En esta reunión se presenta la información necesaria para inspeccionar el avance desde el día anterior. Esta información puede dar como resultado una replanificación y debates sobre la funcionalidad después del *daily stand-up*.

5) Revisión del *sprint* y retrospectiva

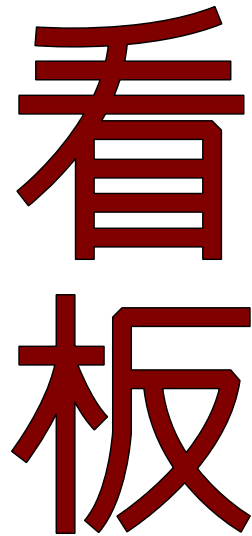
Una vez finalizado el *sprint*, se lleva a cabo la reunión de revisión de este, en la que el equipo Scrum y las partes interesadas inspeccionan lo que se ha hecho durante el *sprint* y lo discuten. En esta reunión, están presentes el propietario del producto, los miembros del equipo y el Scrum máster, además de los clientes, gente de negocios, expertos, ejecutivos y cualquier otra persona interesada.

Después de la revisión del *sprint*, el equipo se reúne para la retrospectiva del *sprint*, una oportunidad para que el equipo discuta lo que ha funcionado bien durante el *sprint* y lo que no funciona, y se pongan de acuerdo en los cambios que hay que intentar hacer en el *sprint* siguiente para ser más productivos.



9.2. Kanban como sistema de gestión del trabajo en curso

Derivado de la combinación de las dos palabras japonesas *Kan*, que significa ‘visual’, y *Ban*, que significa ‘tarjeta’, nace el término *Kanban*, con el que se denomina una metodología de producción u organización del trabajo que se basa en señales visuales para gestionar el esfuerzo y la dedicación de nuestro equipo de producción. Kanban permite al gestor del equipo de producción multimedia identificar aglomeraciones en la producción, mejorar el tiempo de servicio de tareas y optimizar la calidad en el proceso de producción.



Kanban (en *kanji* 看板, donde *kan* (看) significa “visual” y *ban* (板) significa “tarjeta”).

Kanban es un sistema de gestión en el que se produce exactamente la cantidad de trabajo que el sistema es capaz de asumir.

Kanban es un sistema de gestión del **trabajo en curso** WIP, del inglés *work in progress*), que sirve principalmente para asegurar una producción continuada y sin sobrecargas en el equipo de producción multimedia y en el que se produce exactamente la cantidad de trabajo que el sistema es capaz de asumir. También es un sistema de trabajo **justo a tiempo** (*just in time*), que permite evitar excedentes innecesarios de stock, lo que en la gestión de proyectos multimedia equivale a una inversión innecesaria de tiempo y esfuerzo en aquello que no necesitaremos (o, simplemente, es menos prioritario) y así no se sobrecargará al equipo.

Kanban es una aproximación a la gestión del **cambio organizativo**. No es un proceso de desarrollo de productos multimedia o de gestión de proyectos. También es una aproximación a la introducción de cambios en el ciclo de vida de desarrollo de productos multimedia o metodología de gestión de proyectos

ya existente. Con Kanban se empieza con cualquier cosa que esté en curso en la gestión del equipo de producción, no hay que empezar de cero en la organización de una empresa para adoptar este sistema.

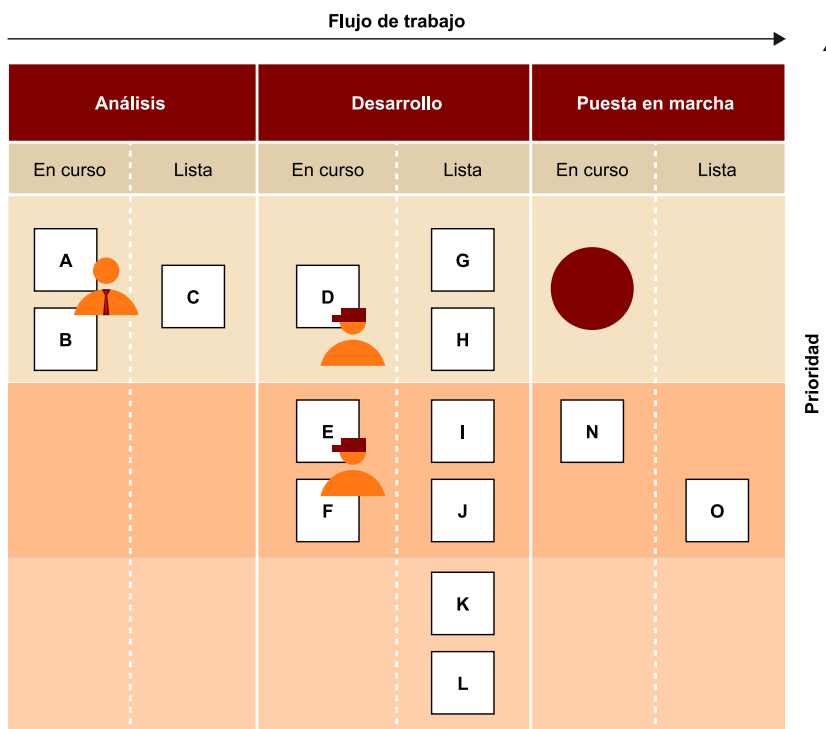
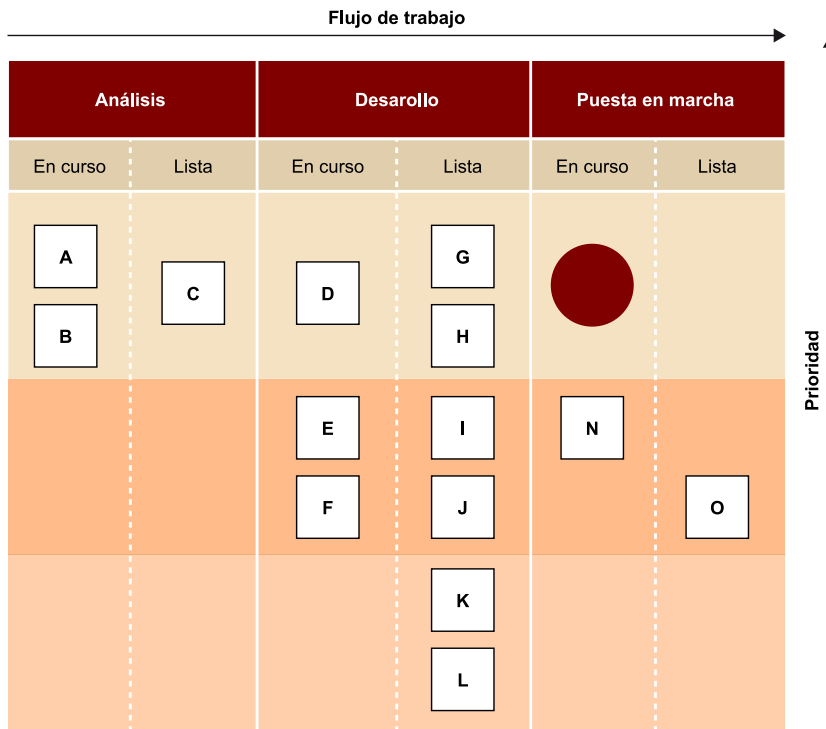
En la gestión del trabajo en curso con Kanban, se busca un concepto clave: **limitar el trabajo en curso**. Está demostrado que si se gestiona mucho trabajo en curso a la vez, los índices de calidad disminuyen drásticamente. En la producción de proyectos multimedia, aumentar el trabajo en curso implica incrementar la cantidad de errores que tendrá este proyecto multimedia, a consecuencia de la poca concentración que los desarrolladores podrán dedicar a las tareas.

Limitar el **trabajo en curso** implica un aumento considerable de la calidad del software desarrollado en la producción de un proyecto multimedia.

Limitar el trabajo en curso mediante la gestión del trabajo con Kanban también tiene otra consecuencia importante, y es que disminuimos el tiempo de servicio de una tarea desde que entra en el sistema hasta que sale de él. Reduciendo la cantidad de trabajo en curso, conseguimos que el enfoque en cada una de las tareas sea mayor y el conjunto del tiempo dedicado a todas sea inferior al utilizado si se asumen todas al mismo tiempo.

Por lo tanto, podemos decir que Kanban, por medio de una disciplina estricta de limitar la cantidad de trabajo que el equipo lleva a cabo a la vez, nos devuelve unos índices de calidad más altos y un tiempo de servicio bastante más bajo.

Un panel Kanban típico se implementaría tal como se muestra en la imagen siguiente:



Representación visual

La representación visual del flujo de trabajo mediante paneles debe ser fiel a la realidad y se debe actualizar constantemente.

En esta imagen vemos un panel constituido por tres columnas que representan las diferentes fases por las que debe fluir una tarea para ser desarrollada (análisis, desarrollo y puesta en funcionamiento). Cada fase está subdividida en dos estados: “En curso” y “Lista para pasar a la fase siguiente” (esta visión está representada por la línea discontinua de cada fase). El estado “En curso” significa que “el equipo trabaja actualmente en esta tarea y en esta fase”, y el estado “Lista” significa que “el equipo ya ha terminado el trabajo a realizar

en esta fase, y la tarea espera que el sistema pueda asumirla para la siguiente fase”. Esta división nos ayuda a localizar atascos en el proceso de producción; lo aprenderemos más adelante en el módulo.

Las filas podrían ser proyectos diferentes en los que trabaja la empresa, pero lo más habitual es que las filas indiquen prioridades, de modo que las tareas superiores son las prioritarias.

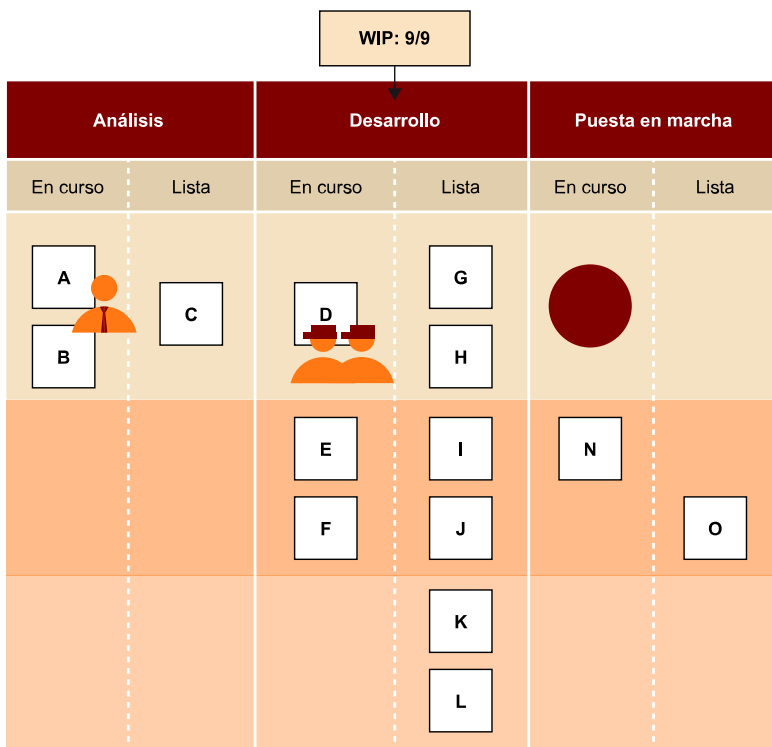
Separar en cada fase las tareas según el estado, “En curso” y “Lista”, nos ayuda a localizar atascos en el proceso de producción de productos multimedia.

Limitar el trabajo en curso (WIP)

Una vez representada la realidad actual de nuestro equipo, debemos actuar y aplicar la primera restricción de Kanban: limitar el WIP. Tenemos que hacer que sea obligatorio que los programadores dejen de programar y se dediquen a acabar las tareas que están bloqueadas. Debemos hacer que los trabajadores de nuestro equipo solo produzcan si el sistema acepta más cantidad de esta tarea producida.

Los cuellos de botella

Cuando la fase “Desarrollo” ha llegado a su límite WIP, tenemos un cuello de botella: la fase “Puesta en marcha” provoca un atasco.



En este momento se pueden tomar tres decisiones diferentes:

1) **La peor.** Aumentar o quitar el límite WIP. Conocemos la teoría y sabemos que Kanban limita el WIP, pero como **no lo podemos hacer**, y es **imposible**, decidimos quitarlo y poner un límite WIP de diez, y más adelante lo pondremos a quince. Realmente, no ayuda en nada a la empresa ni al equipo.

2) **La mala.** Contratar a gente para la fase en el cuello de botella. La solución en un equipo sobrecargado no siempre es poner más gente. Debemos tener presente que incorporar gente nueva requiere dedicarle tiempo, hay un período de tiempo que se debe destinar a formar a la gente nueva (nunca nadie entra sabiéndolo todo, por muy experta que sea la persona) y este tiempo se quitará de la gente que está precisamente en la fase de la que depende el tiempo de servicio de vuestro equipo (en el ejemplo del apartado “Aplicación práctica de Kanban”, el experto en servidores).

3) **La buena.** Poner a uno de los programadores a ayudar en la fase “Puesta en marcha”. Un buen equipo de alto rendimiento es el que está formado por personas expertas en alguna tarea, pero que conocen y se interesan por todas las que le rodean. Esto se llama trabajador T.

En el momento en que al programador le dices “deja de programar y ponte a verificar”, una respuesta muy habitual es “mi trabajo es programar, a mí me pagan para programar y esto es lo que haré”. Hay que tener presente que el objetivo de todo el equipo debe ser **el servicio al cliente** (es decir, **facturar**). Un trabajador que no está dispuesto a asimilar que el trabajo de programar no es el que el sistema de producción necesita en este momento (en que el panel de Kanban refleja que tenemos nueve tareas programadas y solo una facturada) no está haciendo ningún favor a la empresa.

El éxito de Kanban

Implementar Kanban correctamente comporta un alto grado de implicación en el equipo y hacer entender a los trabajadores que su trabajo no ha de ser “programar” o “instalar aplicaciones”, sino hacer que la empresa obtenga beneficios.

Esta cultura tan oriental, que también afecta a la forma de entender los contratos entre cliente y proveedor, es lo que lleva a las empresas que implementan Kanban al éxito.

Bibliografía

Anderson, David J.; Reinertsen, Donald G. (2010). *Kanban. Successful evolutionary change for your technology business* [en línea]. <<http://www.amazon.com/Kanban-Successful-Evolutionary-Technology-Business/dp/0984521402>>

Bataller, Alfons (2010). *La gestió de projectes*. Barcelona: Editorial UOC.

Durán Rubio, S. E. (2003). "Puntos por función. Una métrica estándar para establecer el tamaño del software". *Boletín de Política Informática* (n.º 6). México: Ed. Instituto Nacional de Estadística y Geografía de México.

Krug, Steve (2006). *No me hagas pensar: una aproximación a la usabilidad en la web* Madrid: Pearson Educación.

Ministerio de Administraciones Públicas. *Metodología MÉTRICA versión 3*.

Nielsen, Jakob (2000). *Usabilidad: diseño de sitios web* Madrid: Prentice Hall.

Nielsen, Jakob (2000). *Usabilidad: [prioridad en el diseño web = (prioritizing web usability)]* Madrid: Anaya Multimedia.

Nielsen, Jakob (2000). *Técnicas de eyetracking para usabilidad web* Madrid: Anaya Multimedia.

Pereña Brand, J. (1996). *Dirección y gestión de proyectos* (2.ª ed.). Madrid: Ediciones Díaz de Santos, S. A.

Usability guidelines. Cambridge, MA: Information Services and Technology. Disponible en: <http://ist.mit.edu/services/consulting/usability/guidelines>

