

Tema 4: Planificación

GESTIÓN Y MANEJO DE PROYECTOS INFORMÁTICOS
MÁSTER EN INGENIERÍA INFORMÁTICA

Fernando Berzal Galiano



Índice

- La crisis del software
- Planificación temporal
- Descomposición en tareas
- Camino crítico
 - CPM
 - PERT
- Holguras [slack]
- Prioridades
 - MMF
 - Modelo en espiral
- Seguimiento del proyecto
- Objetivos y restricciones

Planificación

Antes de empezar, ¿qué es la Ingeniería del Software?

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, la explotación y el mantenimiento de programas informáticos.

Planificación

- Principal causa de la crisis del software:
 - Máquinas más complejas.
- Máquinas más complejas → Mayor capacidad de cómputo
- Mayor capacidad de cómputo → Programas más complejos.

- Los programas complejos son difíciles de manejar.
 - Difícil encontrar errores.
 - Necesidad de estructura.

https://en.wikipedia.org/wiki/Software_crisis

Planificación

Los proyectos de software, al menos tal y como lo ve el gerente no técnico, tiene algo de este carácter; suele ser inocente y sencillo, pero es capaz de convertirse en un monstruo de horarios perdidos, presupuestos desorbitados y productos defectuosos. Así que escuchamos gritos desesperados por una “silver bullet”, algo para hacer que los costos de software bajen tan rápido como los costos de hardware.

-- **Frederick P. Brooks, Jr.:**

“No Silver Bullet: Essence and Accidents of Software Engineering”,
IEEE Computer 20(4):10-19, April 1987

Planificación

Problemas causados por la crisis “crónica” del software

- Por lo general, un proyecto software consume un 50% más de tiempo del planeado inicialmente.
 - En proyectos grandes es aun peor.
- El 75% de los grandes sistemas presentan fallos (no funcionan como se esperaba o, simplemente, no funcionan).
- El 25% de los grandes proyectos acaban cancelándose.

W. Wayt Gibbs: “Software's Chronic Crisis”
Scientific American, September 1994

Planificación

La crisis “crónica” del software

- La probabilidad de que un proyecto grande fracase es mayor que la de un proyecto pequeño.
 - El motivo del fracaso suelen ser motivos sociológicos y de comunicación.
- Las opciones de éxito de que un proyecto se complete con éxito desaparecen casi por completo en proyectos de gran escala.

Planificación

¿Por qué se retrasan los proyectos?

- **Plazos** poco realistas (impuestos desde fuera).
- **Cambios** en los requisitos del cliente (que no se reflejan en cambios en la planificación).
- Subestimación [honesta] del **esfuerzo** requerido.
- **Riesgos** no considerados al inicio del proyecto.
- Fallos de **comunicación**.
- Errores de **gestión** al no reconocer que el proyecto se está retrasando con respecto al plan (y ausencia de acciones correctivas).

Planificación

- "Los horarios excesivos o irracionales son probablemente la influencia más destructiva de todo el software". -- Capers Jones.
- Una programación demasiado optimista no resulta en una programación real más corta, sino en una más larga.
- Cada hora de planificación ahorra un día entero de tiempo y esfuerzo perdido. -- Steve McConnell

Planificación

Tareas de planificación

- Establecer el ámbito del proyecto.
- Determinar su viabilidad.
- Analizar posibles riesgos.
- Determinar los recursos necesarios.
- Estimar coste y esfuerzo.
- Desarrollar una planificación temporal del proyecto.

Planificación temporal

Principios a seguir en la planificación

- **Compartimentación**
 - Definición específica de tareas

- **Interdependencias**
 - Identificación de relaciones entre tareas

- **Recursos**
 - ¿Están disponibles los recursos necesarios?

Planificación temporal

Principios a seguir en la planificación

■ Responsabilidades

- Asignar un responsable concreto para cada tarea.

■ Resultados

- Definición de los resultados esperados de cada tarea.

■ Hitos

- Revisión de puntos clave del proyecto.

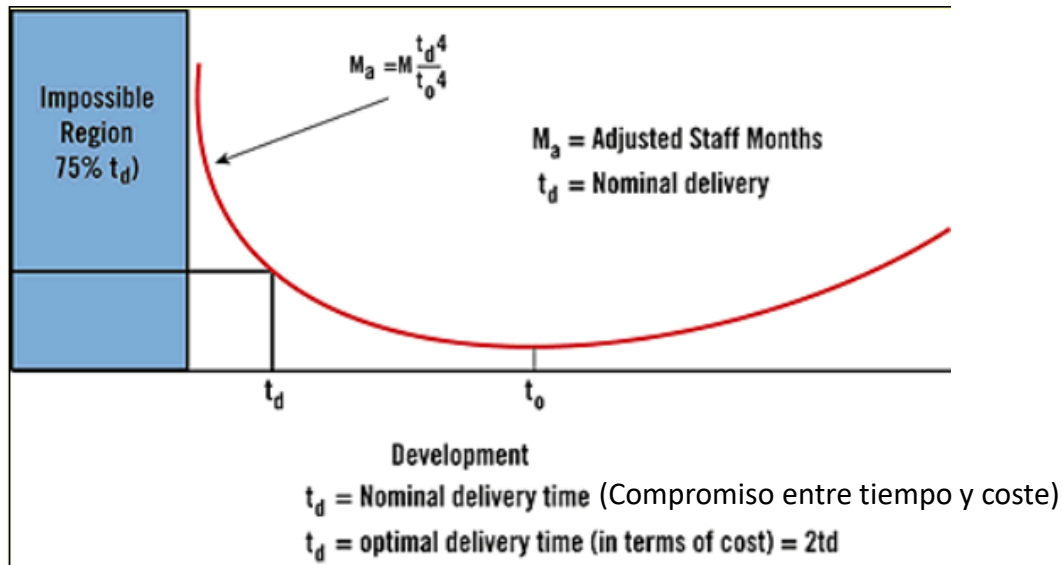
Planificación temporal

Región imposible y Curva PNR

- **Región imposible:**
 - No se pueden comprimir los horarios más allá de este punto.
- **Curva Putnam-Norden-Rayleigh (PNR):**
 - Gráfica tiempo – personas mes.
 - Trata de determinar la mejor combinación tiempo/ personas mes.

Planificación temporal

Curva PNR [Putnam-Norden-Rayleigh]



Estimating Internet Development

<https://www.drdoobs.com/estimating-internet-development/184414639>

Planificación temporal

Región imposible:

- ¿Cuál es la relación entre el tiempo de desarrollo y el número de personas-mes necesarias?

$$T > 2.15 \sqrt[3]{PM}$$

- El 99% de los proyectos completados obedecen esta regla.

Barry Boehm: "Software Engineering Economics", 1981

Planificación temporal

Asignación de recursos

Regla 40-20-40

40-50% actividades iniciales [front end]

- Comunicación con el cliente y análisis (10-25%)
- Diseño (20-25%)
- Revisiones y modificaciones

15-20% actividades de construcción (implementación)

30-40% pruebas

Roger S. Pressman:

"Software Engineering: A Practitioner's Approach", 7th edition, 2009

Planificación temporal

Asignación de recursos

Fase	Tiempo	Esfuerzo
Análisis de requisitos	12%	6%
Diseño preliminar	8%	8%
Diseño detallado	15%	16%
Implementación	30%	40%
Pruebas del sistema	20%	20%
Pruebas de aceptación	15%	10%

Software Engineering Laboratory (SEL)
Relationships, Models, and Management Rules
NASA Software Engineering Laboratory, SEL-91-001, 1991.

Descomposición en tareas

- El plan de proyecto perfecto es posible si primero se documenta una lista de todas las incógnitas – Bill Langley.
- Pero... ¿es posible listarlas absolutamente todas?

Descomposición en tareas

WBS [Work Breakdown Structure]

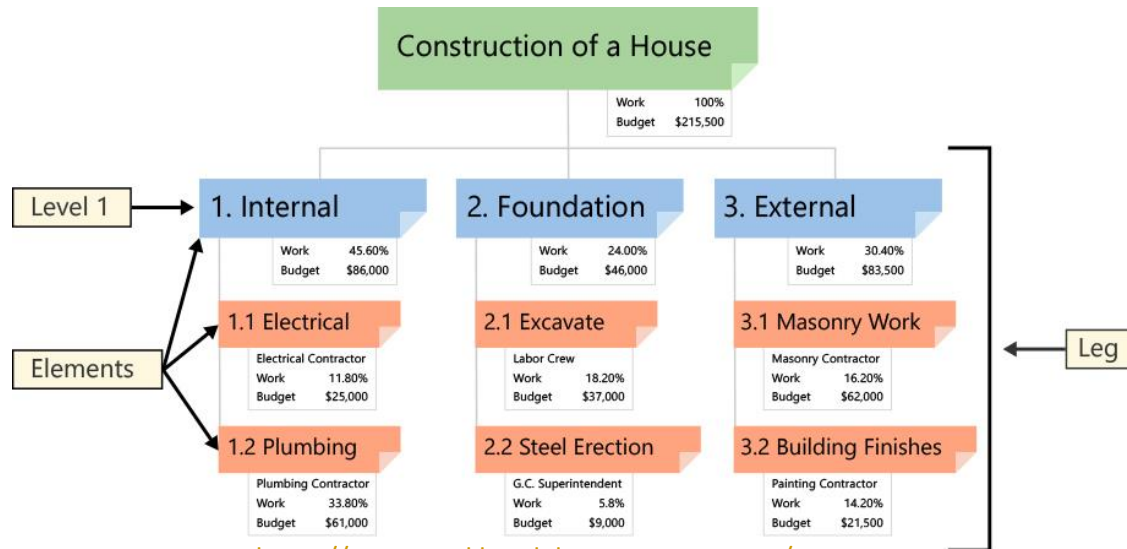
- Descomposición jerárquica orientada a la generación del producto de las tareas del trabajo que debe ser ejecutado por el equipo del proyecto.

- Integra:
 - Alcance del proyecto.
 - Coste.
 - Calendario.

Descomposición en tareas

WBS [Work Breakdown Structure]

- Orientado al producto

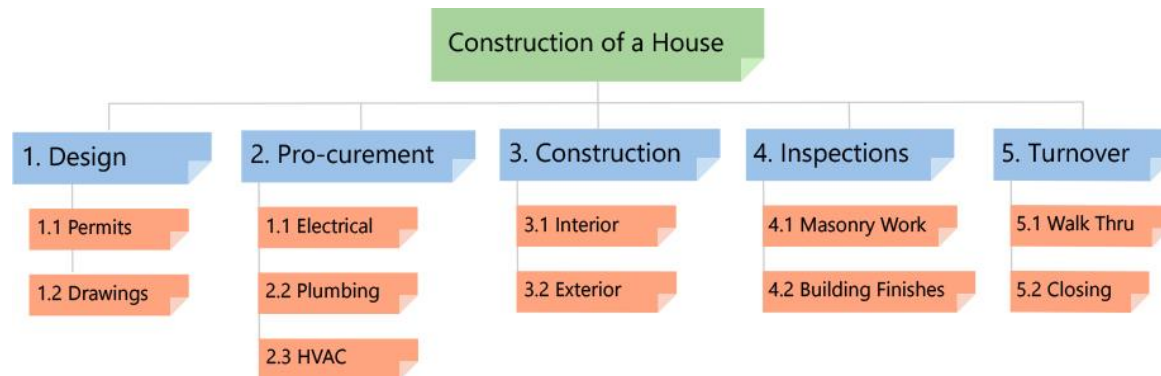


<https://www.workbreakdownstructure.com/>

Descomposición en tareas

WBS [Work Breakdown Structure]

- Orientado a las fases del proyecto



Descomposición en tareas

WBS [Work Breakdown Structure]

1.1 **Concept scoping** determines the overall scope of the project.

Task definition: Task 1.1 Concept Scoping

- 1.1.1 Identify need, benefits and potential customers;
 - 1.1.2 Define desired output/control and input events that drive the application;
 - Begin Task 1.1.2
 - 1.1.2.1 FTR [formal technical review]: Review written description of need
 - 1.1.2.2 Derive a list of customer visible outputs/inputs
 - 1.1.2.3 FTR: Review outputs/inputs with customer and revise as required;
 - endtask Task 1.1.2
 - 1.1.3 Define the functionality/behavior for each major function;
 - Begin Task 1.1.3
 - 1.1.3.1 FTR: Review output and input data objects derived in task 1.1.2;
 - 1.1.3.2 Derive a model of functions/behaviors;
 - 1.1.3.3 FTR: Review functions/behaviors with customer and revise as required;
 - endtask Task 1.1.3
 - 1.1.4 Isolate those elements of the technology to be implemented in software;
 - 1.1.5 Research availability of existing software;
 - 1.1.6 Define technical feasibility;
 - 1.1.7 Make quick estimate of size;
 - 1.1.8 Create a Scope Definition;
- endTask definition: Task 1.1

Roger S. Pressman:

"Software Engineering: A Practitioner's Approach", 7th edition, 2009

Descomposición en tareas

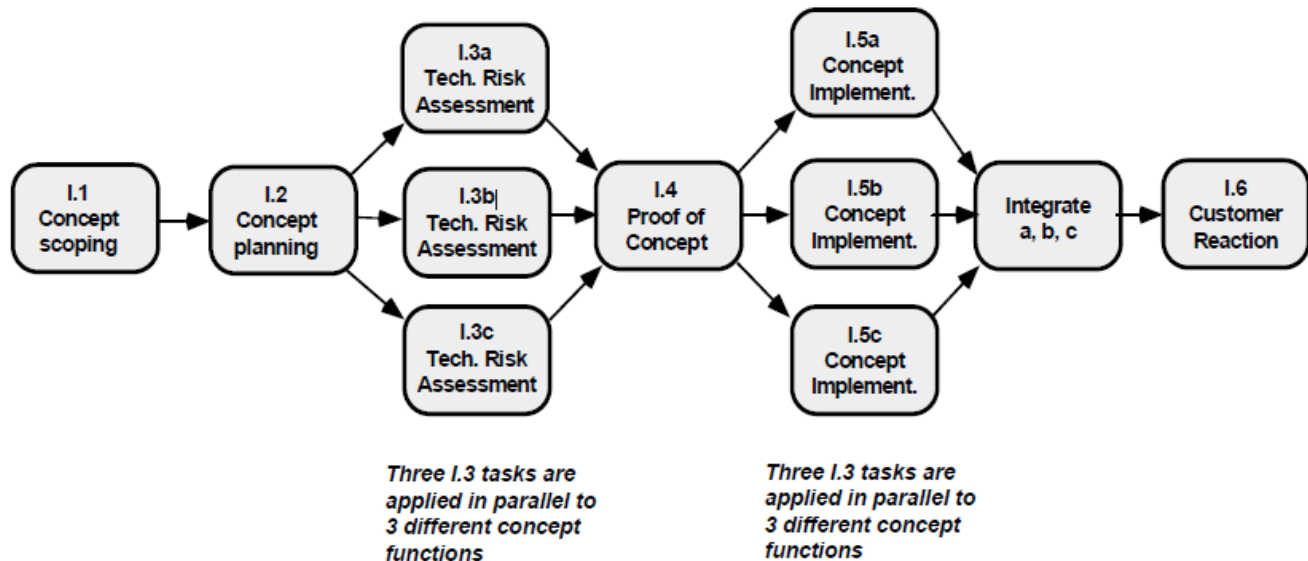
¿Cuánto debería de durar cada tarea?

Al descomponer actividades en tareas individuales, las tareas deberían ser lo suficientemente grandes como para requerir al menos dos días de trabajo, pero lo suficientemente pequeñas para no requerir más de una semana.

- Las tareas de menos de dos días tienden a “rellenarse”.
- Las tareas de más de una semana indican que no se entiende realmente qué es necesario o que es demasiado general.

Descomposición en tareas

Red de tareas o actividades



Roger S. Pressman:

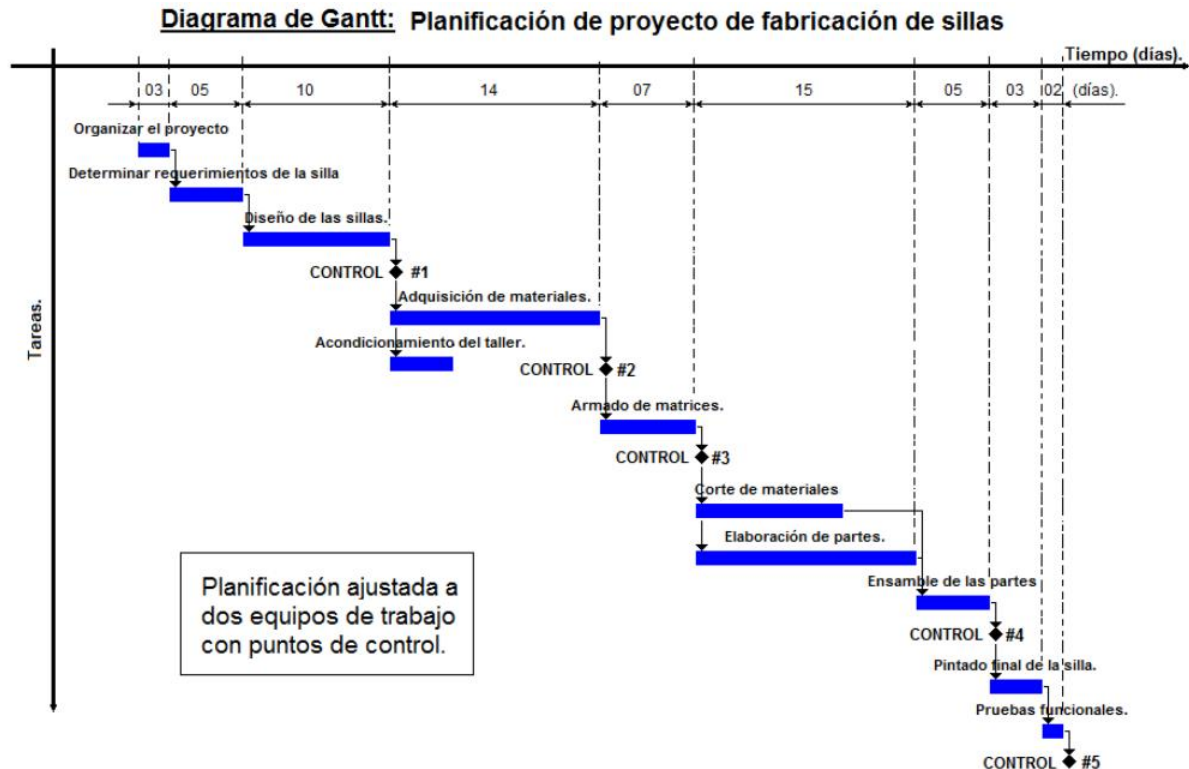
"Software Engineering: A Practitioner's Approach", 7th edition, 2009

Descomposición en tareas

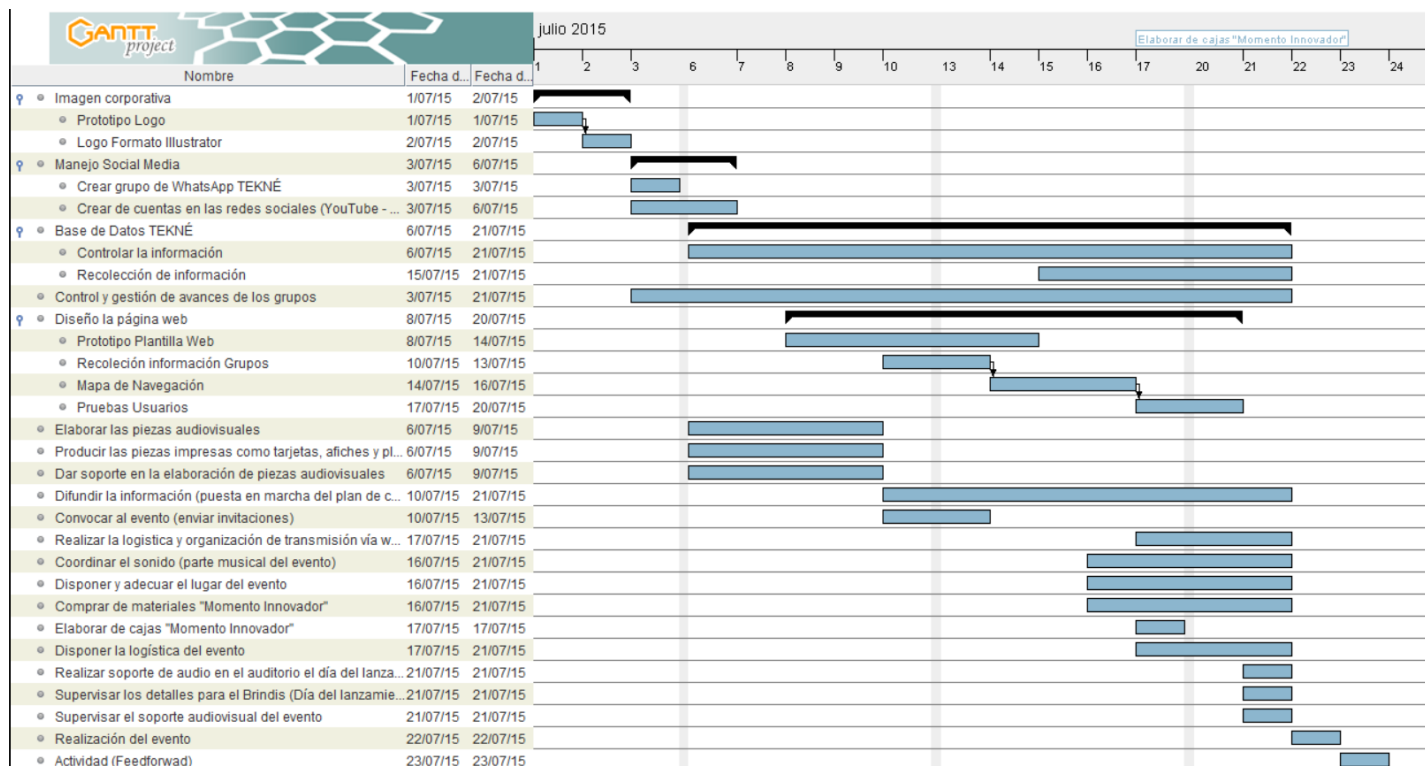
Diagrama de Gantt

- Tareas/actividades (fecha de inicio y duración)
- Dependencias entre actividades
- Hitos (rombos)
- Asignación de recursos y ajuste del calendario (con ayuda de herramientas software)

Descomposición en tareas



Descomposición en tareas



Camino crítico

- Conjunto de tareas que deben completarse de acuerdo al plan si queremos que el proyecto en su conjunto termine en la fecha establecida.
- Si no se completan el proyecto acabará más tarde de lo previsto.

Los métodos de planificación de proyectos proporcionan herramientas para determinar el camino crítico:

- **CPM:** Critical Path Method.
- **PERT:** Program Evaluation and Review Technique.

CPM

Dadas las duraciones de cada tarea
y las dependencias existentes entre ellas:

- **ES** [Earliest Start]: Comienzo más temprano
 $ES(t) = \max_{p \rightarrow t} \{ ES(p) + \text{duración}(p) \}$
siendo $ES(t)=0$ para las tareas sin predecesoras.
- **LS** [Latest Start]: Comienzo más tardío
 $LS(t) = \min_{s \leftarrow t} \{ LS(s) - \text{duración}(t) \}$
siendo $LS(t)=ES(t)$ para las tareas sin sucesoras.

James Kelley (Remington Rand) & Morgan Walker (DuPont)
"Critical-Path Planning and Scheduling"
Proceedings of the Eastern Joint Computer Conference, 1959.

CPM

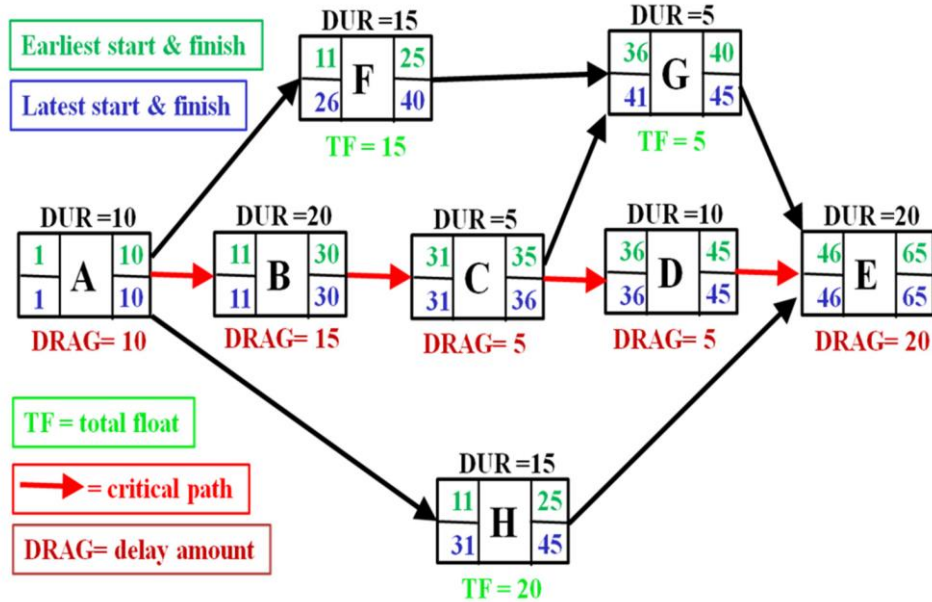
Dadas las duraciones de cada tarea
y las dependencias existentes entre ellas:

- **Holgura [slack, a.k.a. float]**
 $\text{slack}(t) = \text{LS}(t) - \text{ES}(t)$

- **Actividades críticas:**
Tareas sin holgura.

Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

CPM

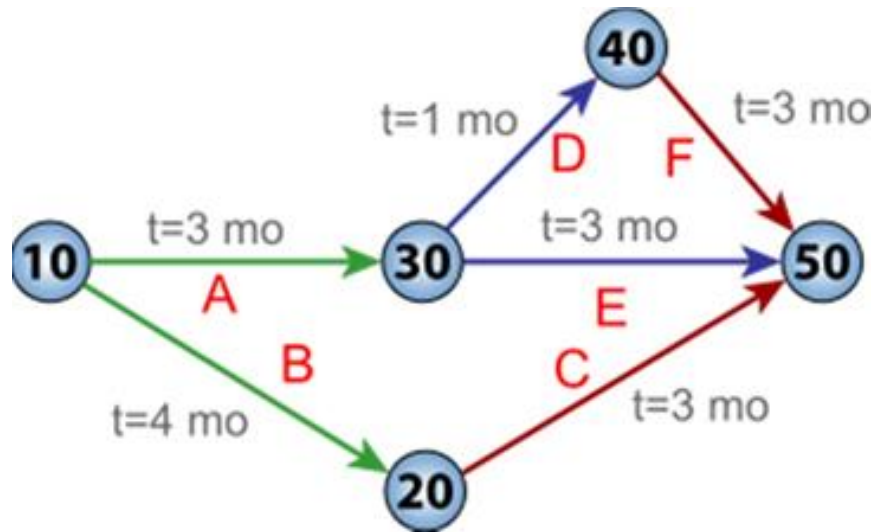


CPM

Calculo de Time Float y Drag

- **Time Float (TF):** diferencia entre latest start y earliest start.
- **Delay Amount (DRAG):** el menor valor entre la duración de la tarea y los TFS de las tareas en paralelo.

PERT



- Actividades → arcos
- Hitos → nodos.

PERT

Estimaciones

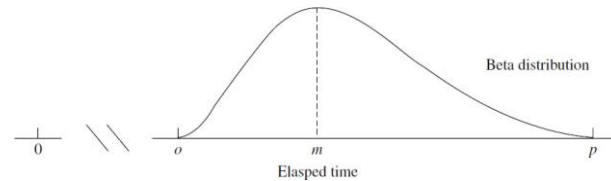
- O [optimista]: Tiempo mínimo posible requerido para realizar una tarea, asumiendo que todo va mejor de lo que normalmente se espera.
- P [pesimista]: Tiempo máximo posible requerido para realizar una tarea, asumiendo que todo va mal (salvo catástrofes que impidan la realización de la tarea).
- M [most-likely] / N [normal]:
Mejor estimación del tiempo necesario para completar la tarea, en condiciones normales.

PERT

Estimaciones

Tiempo esperado

$$T_E = (O + 4M + P) / 6$$



Tarea	Pred.	O	N	P	T_E
A	-	1	3	5	3.00
B	-	2	4	6	4.00
C	B	2	3	4	3.00
D	A	1	1	2	1.17
E	A	2	3	7	3.50
F	D	2	3	6	3.33

Camino crítico – Diagrama de Gantt

Camino crítico

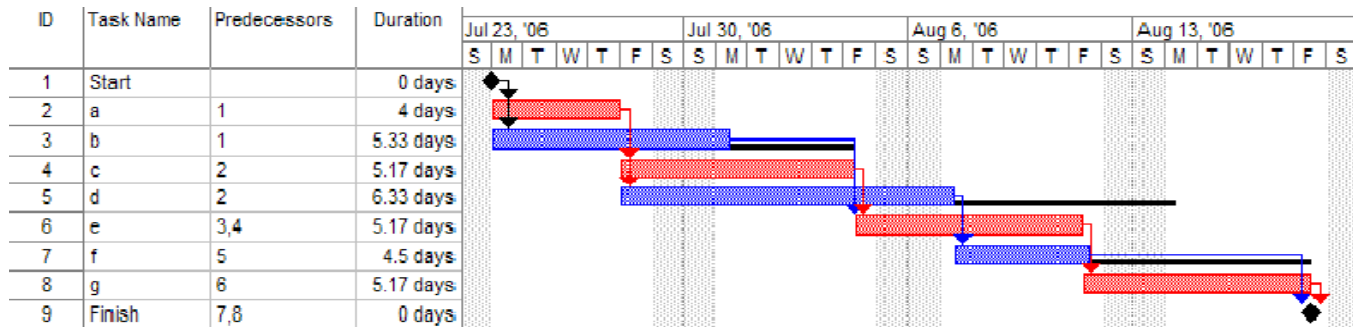


Diagrama de Gantt
Microsoft Project

Diagrama AON

Camino crítico

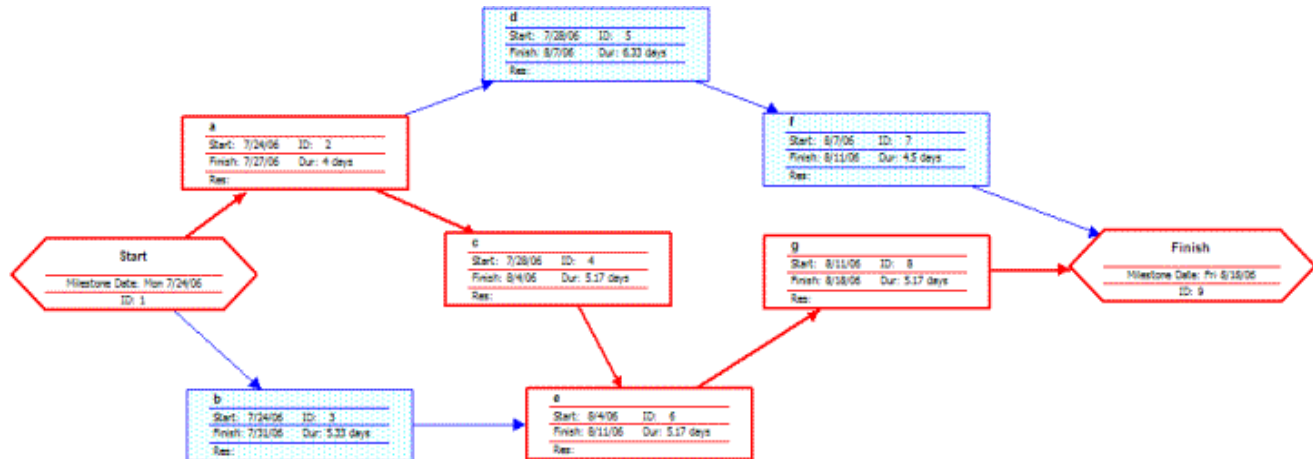


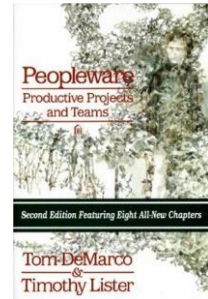
Diagrama AON [activity on node]
Microsoft Project

Slack

- Hay muchas tareas que no se pueden planificar, es imposible hacer un seguimiento de todo.
 - Por ejemplo:
 - Instalaciones de nuevas versiones de sistema operativo o de compilador.
- Para todas estas tareas dejar un tiempo, 15%-20% del proyecto.

Slack

En un trabajo creativo no se está el 100% del tiempo “haciendo” cosas (como en una cadena de montaje).



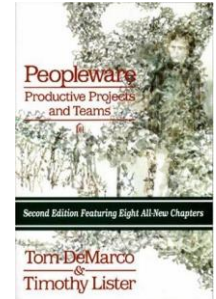
Hay que provisionar tiempo para...

- Intercambiar ideas, p.ej. brainstorming.
- Investigar nuevos métodos, técnicas y herramientas.
- Averiguar cómo evitar determinadas subtareas.
- Leer.
- Probar alternativas.
- ... y “perder el tiempo” (horas no productivas).

Slack

"Spanish Theory managers"

Aumentar niveles de productividad mediante horas extra no pagadas (con la ayuda de "juegos mentales" con sus empleados, p.ej. sentimiento de culpa, seguridad del puesto de trabajo...)



- "Teoría española del valor"
 - Existe una cantidad fija de valor sobre la Tierra.
- "Teoría inglesa del valor"
 - la ingenuidad y la tecnología permiten crear valor.
- Resultado: **Una hora de baja productividad por cada hora extra...**

Slack

Estrategias típicas para aumentar la productividad:

- Presionar a los empleados para que echen más horas.
- Mecanizar el proceso de desarrollo.
- Comprometer la calidad del producto.
- Estandarizar procedimientos
(como si las personas fuesen partes intercambiables).

Todas esas medidas pueden aumentar el estrés, reducir la satisfacción en el trabajo y destruir la motivación... además de causar bajas en el equipo de desarrollo.

People under pressure don't work better;
they just work faster -- Tom DeMarco

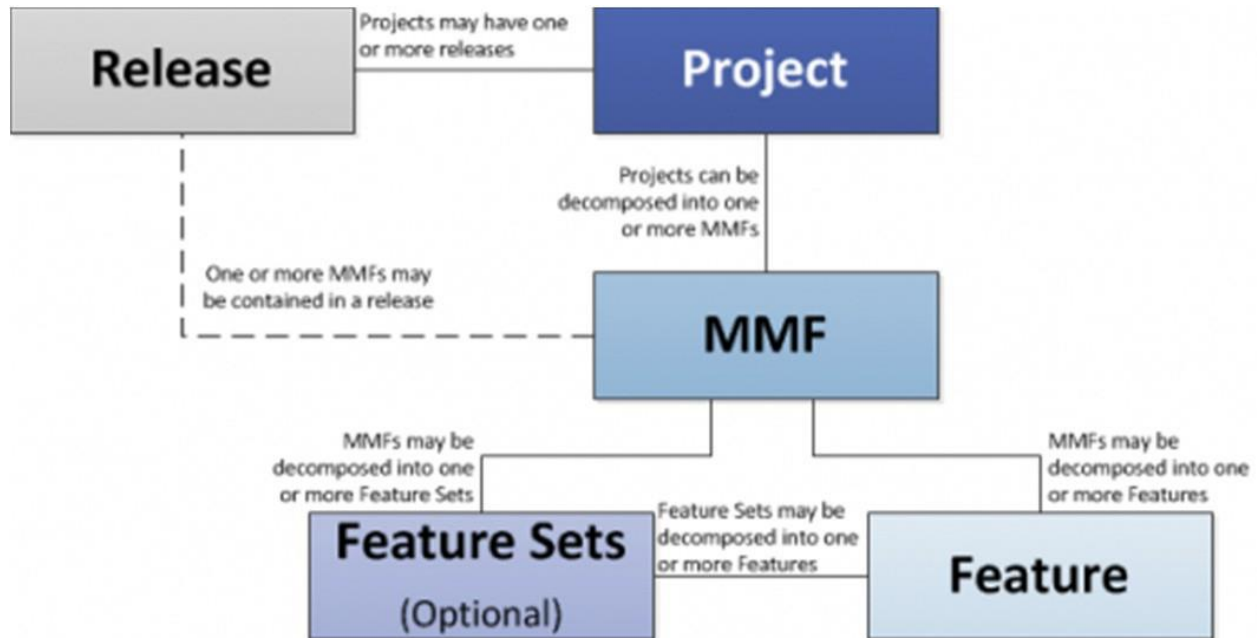


Prioridades

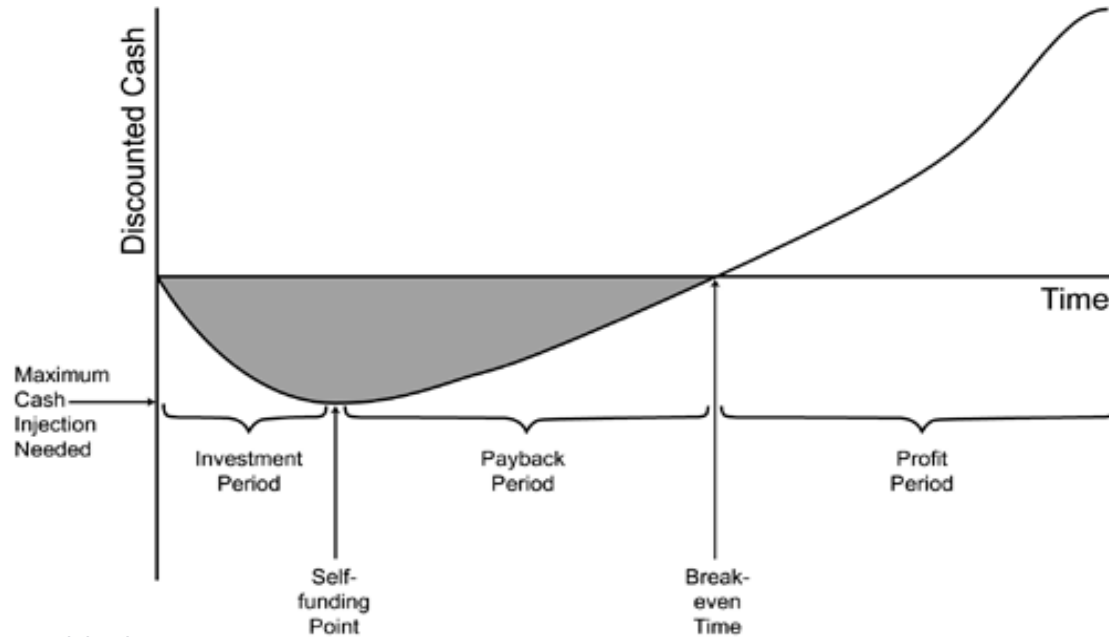
- Establecer prioridades es fundamental:
 - Distinguir entre características esenciales y características deseables.
 - EJ: MMF (Minimum Marketable Feature).
 - Unidad más pequeña de funcionalidad con valor de mercado intrínseco.
- Identificar riesgos y darles prioridad a las actividades que los minimizan o eliminan es de igual importancia: "risk-first development"
 - p.ej. Modelo en espiral de Boehm

(Veremos más de esto en el tema de Gestión de Riesgos)

Minimum Marketable Features

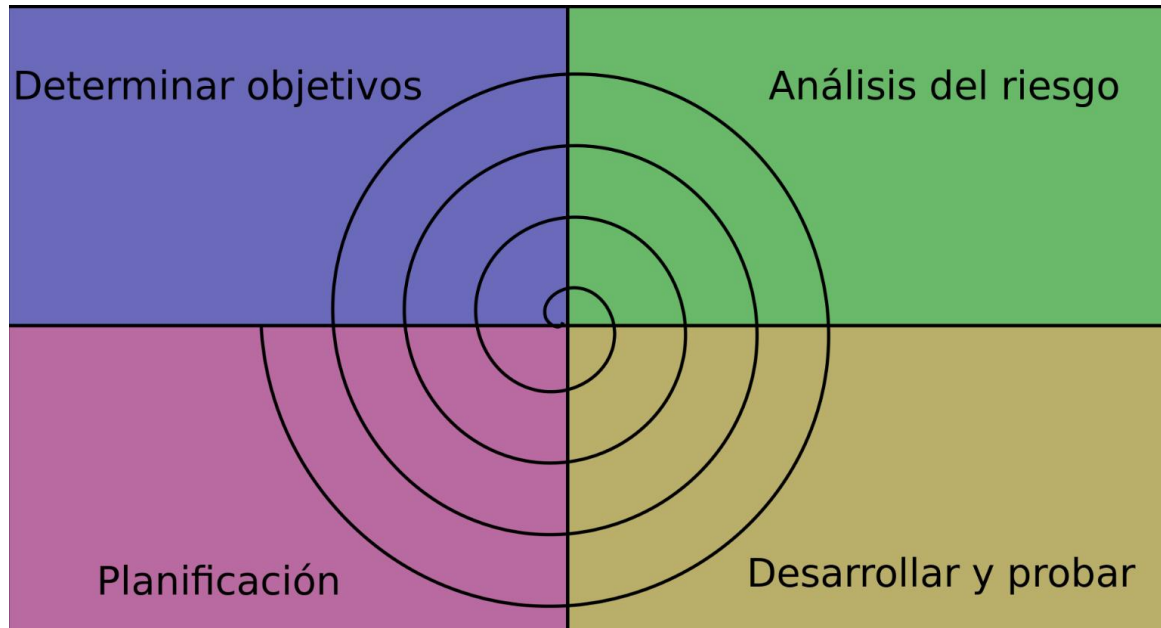


Minimum Marketable Features



Mark Denne & Jane Cleland-Huang:
 "Software by Numbers: Low-Risk, High-Return Development" Prentice Hall
 PTR, 2003. ISBN 0-13-140728-7

Modelo en espiral de Boehn



Seguimiento del proyecto

- Es importante comprobar periódicamente que la planificación se cumple.
- Si no se toman medidas cuando el estado del proyecto se desvía de la planificación puede que la situación se vuelva irrecuperable.

“How does a project get to be a year late?...
One day at a time.” -- **Frederick P. Brooks**

Seguimiento del proyecto

Periódicamente, deben realizarse reuniones de seguimiento:

- Cada miembro del equipo informa del progreso del proyecto y de los problemas que han aparecido.
- Se evalúan los resultados de las revisiones realizadas.
- Se determina si se han alcanzado los hitos del proyecto en las fechas previstas.
- Se comparan las fechas de inicio previstas con las fechas de inicio reales para cada tarea.

Seguimiento del proyecto

- El plan de proyecto debe mantenerse actualizado.
- Se actualiza, por ejemplo, en función de lo descubierto en las reuniones de seguimiento.
- Un plan no actualizado puede ser incluso peor que no tener plan ya que puede dar la sensación de que todo va bien y no hay problemas.

Seguimiento del proyecto

¿Las metodologías ágiles requieren menos planificación?

¡NO!

- Los equipos que usan metodologías ágiles hacen mucha planificación, es sólo que esta está más repartido a lo largo del proyecto.
- los equipos ágiles se enfrentan directamente al factor crítico que muchos equipos no ágiles ignoran: la incertidumbre.

Seguimiento del proyecto

- Cuando un proyecto se retrasa, difícilmente se recupera.
- Siempre que haya una desviación del plan:
 - Debe comunicarse.
 - Deben buscarse estrategias alternativas para poder continuar:
 - Recortar funcionalidad.
 - Aumentar plazos.
 - No debe:
 - No sacrificar calidad para ganar tiempo.

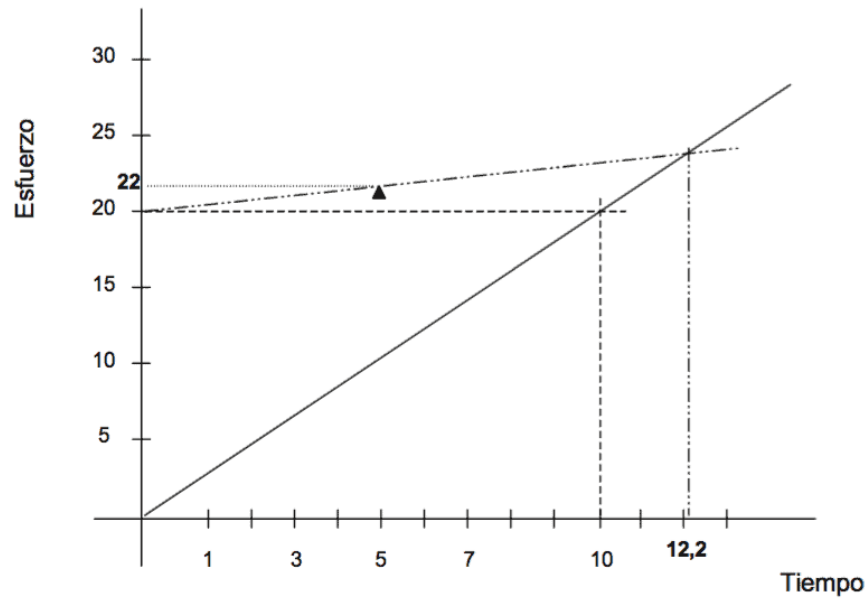
Seguimiento del proyecto

Formas de reducir tiempo en la realización de un proyecto:

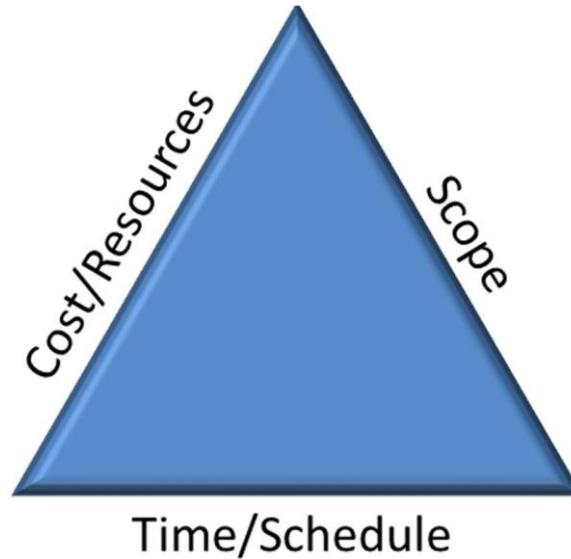
- Paralelizar determinadas tareas.
- Compromiso tiempo vs. esfuerzo [schedule vs. effort].
- Reconsiderar la inclusión de algunas características.
- Resolver conflictos para liberar a personal clave.
- Desarrollo incremental.

Seguimiento del proyecto

Diagrama de extrapolación



Objetivos y restricciones

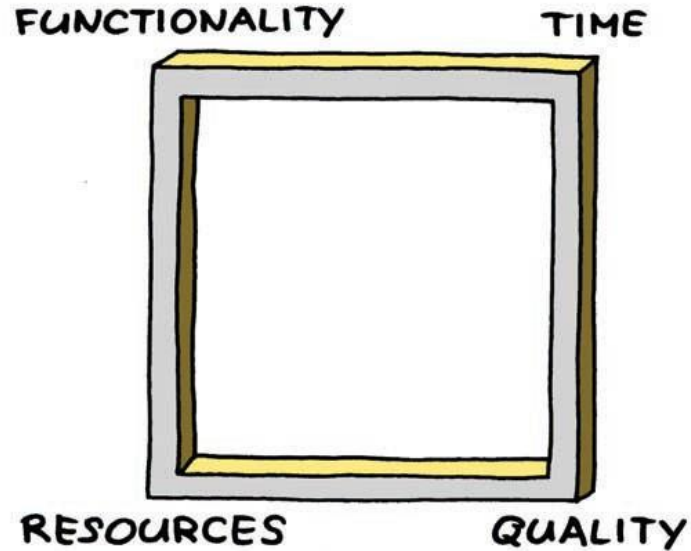
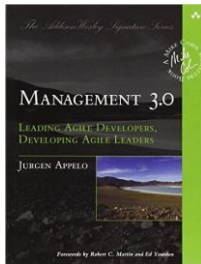


“Puedo hacerlo rápido, barato o bueno. Elige 2”

https://en.wikipedia.org/wiki/Project_management_triangle

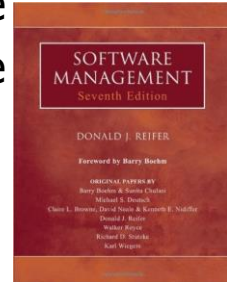
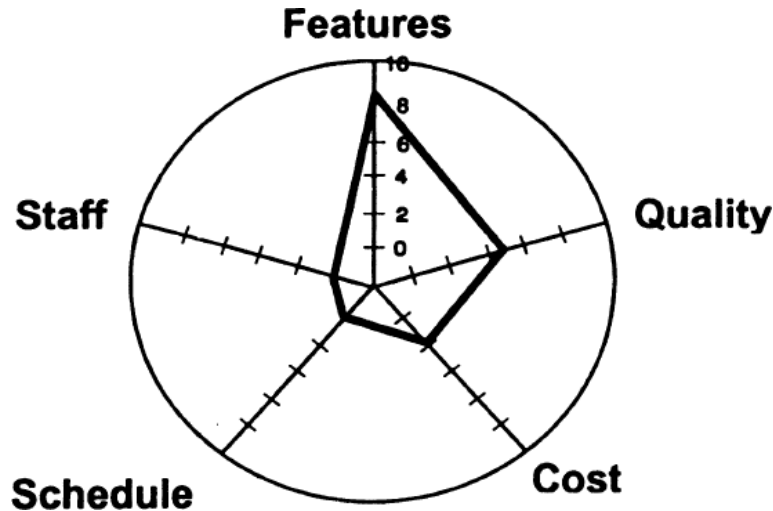
Objetivos y restricciones

La calidad también impone restricciones...



Objetivos y restricciones

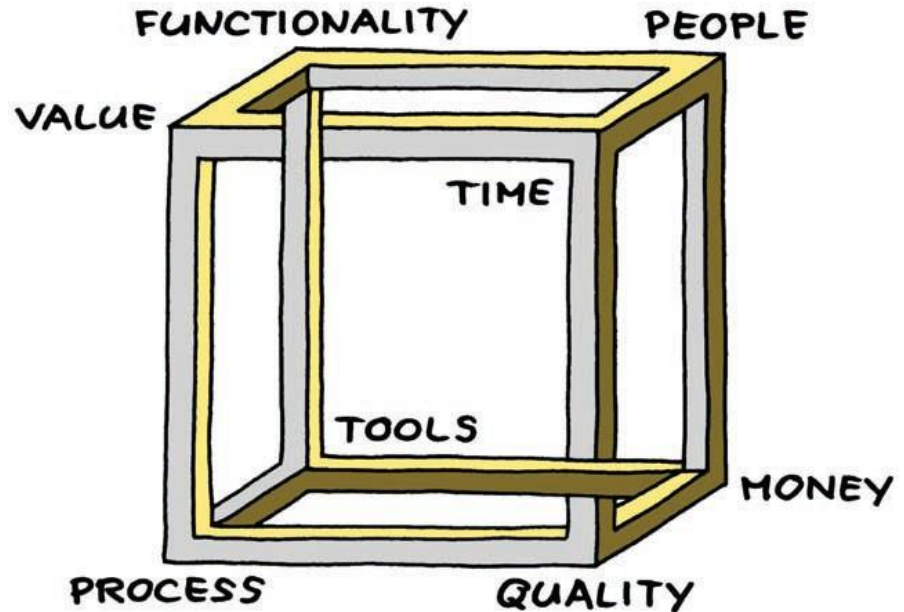
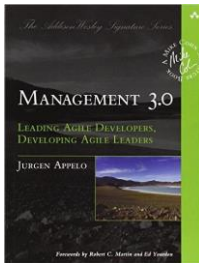
Puede dibujarse el cumplimiento de cada una de las características mediante un diagrama de Kiviat.



Karl E. Wiegers: "Creating a Software Engineering Culture", 1996

Objetivos y restricciones

Hay muchas dimensiones en juego...
Equilibrarlas todas no es sencillo.



Referencias

- Fernando Berzal:
Planificación y gestion de proyectos informáticos.
<http://elvex.ugr.es/decsai/project-management/>



Planificación
Fernando Berzal, berzal@acm.org

Bibliografía

- Roger S. Pressman:
Software Engineering: A Practitioner's Approach
McGraw-Hill, 8th edition, 2014. ISBN 0078022126
- Shari Lawrence Pfleeger & Hoanne M. Atlee: **Software Engineering: Theory and Practice** Prentice Hall, 4th edition, 2009. ISBN 0136061699
- Ian Sommerville:
Software Engineering
Pearson, 10th edition, 2015. ISBN 0133943038

Bibliografía

Lecturas recomendadas

- Dwayne Phillips:
The Software Project Manager's Handbook: Principles That Work at Work
Wiley / IEEE Computer Society, 2nd edition, 2004 ISBN 0471674206
- Donald J. Reifer (editor):
Software Management
Wiley / IEEE Computer Society, 7th edition, 2006 ISBN 0471775622
- Richard H. Thayer (editor):
Software Engineering Project Management Wiley / IEEE
Computer Society, 2nd edition, 2000 ISBN 0818680008

Bibliografía complementaria

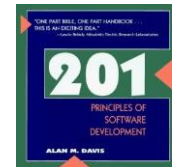
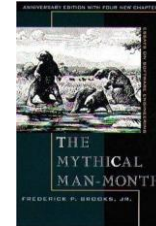
Ingeniería del Software

- Robert L. Glass:
Facts and Fallacies of Software Engineering
Addison-Wesley, 2003. ISBN 0321117425
- Barry W. Boehm & Richard Turner:
Balancing Agility and Discipline: A Guide for the Perplexed Addison-Wesley, 2003. ISBN 0321186125
- Steve Tockey: **Return on Software: Maximizing the return on your software investment**
Addison-Wesley Professional, 2004. ISBN 0321228758
- Steve McConnell:
Software Project Survival Guide
Microsoft Press, 1997. ISBN 1572316217
- Steve McConnell
Rapid Development: Taming wild software schedules
Microsoft Press, 1996. ISBN 1556159005

Bibliografía complementaria

Clásicos

- Frederick P. Brooks, Jr.:
The Mythical Man-Month: Essays on Software Engineering
Addison-Wesley, 1995. ISBN 0201835959
- Alan M. Davis:
201 Principles of Software Development
McGraw-Hill, 1995. ISBN 0070158401
- Barry W. Boehm:
Software Engineering Economics
Prentice-Hall PTR, 1991. ISBN 0138221227
- **Manager's Handbook for Software Development**
NASA Software Engineering Laboratory, SEL-84-101, rev.1, 1990.
- **Software Engineering Laboratory (SEL) Relationships, Models, and Management Rules**
NASA Software Engineering Laboratory, SEL-91-001, 1991.



Bibliografía

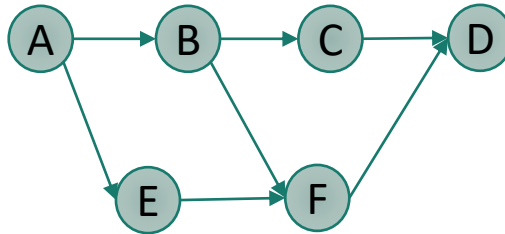
Bibliografía en castellano

- Roger S. Pressman:
Ingeniería de Software: Un enfoque práctico
McGraw-Hill, 7ª edición, 2010. ISBN 6071503140
- Ian Sommerville:
Ingeniería de Software
Pearson, 9ª edición, 2012. ISBN 6073206038

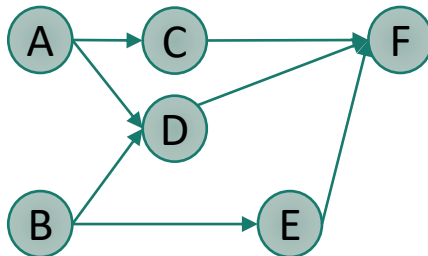


Ejercicios

Dadas las siguientes redes de tareas...



Tarea	Duración
A	6
B	4
C	5
D	3
E	2
F	7



Tarea	Duración
A	3
B	4
C	5
D	6
E	8
F	7

Ejercicios

Calcular:

- La hora de comienzo más temprana (ES) para cada tarea que nos permite completar todas las tareas en un tiempo mínimo.
- La hora de comienzo más tardía (LS) para cada tarea que nos permite completar todas las tareas en un tiempo mínimo.
- La holgura [slack] de cada tarea.
- Los caminos críticos de cada proyecto.
- La duración del plan óptimo para cada proyecto.