

Cuadernos Didácticos



**Ingeniería
Informática**

Proyectos Informáticos

Cuaderno N° 49

Aquilino Adolfo Juan Fuente
Juan Manuel Cueva Lovelle

Departamento de Informática
Universidad de Oviedo

Oviedo, Febrero 2006



Cuadernos Didácticos

Ingeniería Informática

Cuaderno N° 49

PROYECTOS INFORMÁTICOS

Autores:

Aquilino Adolfo Juan Fuente

Juan Manuel Cueva Lovelle

Universidad de Oviedo - España

Editorial:

SERVITEC

ISBN:84-689-7276-2

Deposito Legal: AS-1155-06

1ª Edición: Oviedo, Febrero 2006

Consultor Editorial

Juan Manuel Cueva Lovelle

cueva@lsi.uniovi.es

PROYECTOS DE INFORMÁTICA

Apuntes de la asignatura

INGENIERO TÉCNICO EN INFORMÁTICA
GESTIÓN Y SISTEMAS
3^{er} CURSO

2006



Universidad de Oviedo



EUITIO



Lenguajes y Sistemas Informáticos

Redactado:	Aquilino Adolfo Juan Fuente / Juan Manuel Cueva Lovelle
------------	---

Fecha:	18 de febrero de 2006
--------	-----------------------

Apuntes de la asignatura de PROYECTOS DE INFORMÁTICA	Hoja 1 de 180
--	---------------

Autores

- Aquilino Adolfo Juan Fuente (Universidad de Oviedo)
- Juan Manuel Cueva Lovelle (Universidad de Oviedo)

- Agradecimientos a Vanessa Rodríguez Álvarez por el capítulo “Resumen de la Asignatura”.

Redactado:	Aquilino Adolfo Juan Fuente / Juan Manuel Cueva Lovelle	
Fecha:	18 de febrero de 2006	
Apuntes de la asignatura de PROYECTOS DE INFORMÁTICA		Hoja 2 de 180
This document is only for academic purposes.		© 2006

ÍNDICE

PARTE I INTRODUCCIÓN.....	8
1 PRÓLOGO.....	9
1.1 CONTENIDOS DE LA ASIGNATURA	10
1.2 BIBLIOGRAFÍA.....	12
1.2.1 <i>Básica</i>	12
1.2.2 <i>Complementaria</i>	12
PARTE II CONTENIDOS DE LA ASIGNATURA.....	15
2 00. INTRODUCCIÓN Y PRESENTACIÓN DE LA ASIGNATURA	16
2.1 ÍNDICE.....	16
2.2 INTRODUCCIÓN	16
2.3 OBJETIVOS	16
2.4 CAPACIDADES	17
2.5 TEMARIO.....	18
2.6 METODOLOGÍA.....	20
2.7 EVALUACIÓN	21
2.7.1 <i>Evaluación continua</i>	21
2.7.2 <i>Examen final</i>	22
2.8 CONTACTOS Y PÁGINA WEB	22
2.9 BIBLIOGRAFÍA BÁSICA.....	23
3 01. INTRODUCCIÓN A LOS PROYECTOS DE INFORMÁTICA	24
3.1 ÍNDICE.....	24
3.2 ¿QUÉ ES UN PROYECTO?	24
3.3 CONCEPTOS ASOCIADOS AL PROYECTO.....	25
3.4 CONCEPTO DE PROYECTO	25
3.4.1 <i>Fases de los Proyectos</i>	26
3.4.2 <i>Gestión de Proyectos</i>	26
3.4.3 <i>Tipo de Proyectos Informáticos</i>	27
3.5 COMPLEJIDAD DEL DESARROLLO DE PROYECTOS DE INFORMÁTICA	27
3.5.1 <i>Modelo en cascada</i>	28
3.5.2 <i>Modelo de Prototipos</i>	29
3.5.3 <i>Modelo en espiral</i>	30
3.5.4 <i>Modelos Ágiles</i>	30
3.5.4.1 XP (eXtreme Programming).....	30
3.5.5 <i>RUP</i>	31
3.5.5.1 Definición	31
3.5.5.2 Características	31
3.5.5.3 La Iteración	32
3.5.5.4 Los Ciclos	32
3.5.5.4.1 Iteraciones y Ciclos.....	33
3.5.5.4.2 Fases dentro de un ciclo.....	33
3.5.5.5 Actividades.....	34
3.5.5.5.1 Requisitos	34
3.5.5.5.2 Análisis	34
3.5.5.5.3 Diseño	34
3.5.5.5.4 Implementación	34
3.5.5.5.5 Prueba	35
3.6 NECESIDAD DE PLANIFICACIÓN Y CONTROL	35
3.7 RESUMEN	36
4 02. MEDICIÓN DE SOFTWARE.....	37
4.1 ÍNDICE.....	37
4.2 CONCEPTO DE MÉTRICA	37
4.2.1 <i>Teoría de la Medición - Medición Cuantitativa</i>	37
4.2.1.1 Sistema Relacional Empírico	37

4.2.1.2	Sistema Relacional Formal.....	38
4.2.2	<i>Calidad de la medición, se basa en cuatro conceptos:</i>	38
4.2.2.1	Escala.....	38
4.2.2.2	Unicidad.....	38
4.2.2.3	Representatividad.....	38
4.2.2.4	Significación.....	39
4.2.3	<i>Ejemplos de Mediciones</i>	39
4.2.4	<i>Características de una buena medición</i>	39
4.2.5	<i>Medición orientada a la caracterización del objeto</i>	40
4.2.6	<i>Pseudométricas y Métricas</i>	40
4.3	MÉTRICAS DE HARDWARE.....	40
4.4	MÉTRICAS DE SERVICIOS.....	41
4.5	MÉTRICAS DE COMUNICACIONES.....	41
4.6	MÉTRICAS DE SOFTWARE.....	41
4.6.1	<i>Clasificación por el destino a que se dedica</i>	42
4.6.2	<i>Ventajas y Desventajas</i>	42
4.6.3	<i>Métricas de tamaño</i>	42
4.6.4	<i>Métricas de estructuras de datos</i>	43
4.6.5	<i>Métricas de estructuras de control</i>	43
4.6.6	<i>Métricas de esfuerzo</i>	44
4.6.7	<i>Métricas de calidad y fiabilidad</i>	44
4.6.8	<i>Métricas de diseño</i>	44
4.6.9	<i>Otras métricas de software</i>	44
5	03. MODELOS DE ESTIMACIÓN	46
5.1	ÍNDICE.....	46
5.2	INTRODUCCIÓN.....	46
5.3	MODELOS DE ESTIMACIÓN DEL SOFTWARE.....	46
5.3.1	<i>CEM Históricos</i>	47
5.3.2	<i>CEM con base estadística</i>	47
5.3.3	<i>CEM con base teórica</i>	47
5.3.4	<i>CEM compuestos</i>	47
5.3.5	<i>Problemas de los CEM</i>	47
5.3.6	<i>Generalidades de los CEM</i>	48
5.3.7	<i>Paradojas de la aplicación de los modelos de estimación</i>	48
5.3.8	<i>Factores que afectan a la estimación</i>	48
5.3.9	<i>Selección de un CEM</i>	49
5.3.10	<i>Modelos históricos o basados en la experiencia</i>	49
5.3.11	<i>TRW de Wolverton</i>	50
5.3.11.1	Desventajas.....	50
5.3.12	<i>Técnica Delphi</i>	51
5.3.13	<i>Modelos con base estadísticas</i>	51
5.3.14	<i>Curvas de Rayleig-Norden</i>	52
5.3.15	<i>Modelo de Putnam</i>	52
5.3.16	<i>COCOMO</i>	52
5.3.16.1	Factores Multiplicadores.....	53
5.3.16.2	Tipos de proyectos.....	54
5.3.16.3	Valor de los CDA.....	54
5.3.16.4	Características de COCOMO.....	55
5.3.17	<i>Puntos de Función</i>	55
5.3.17.1	Parámetros del modelo Puntos de Función.....	55
5.3.17.2	Factores a estimar.....	56
5.3.17.3	Valor ajustado.....	57
5.4	COSTE DE DESARROLLO DE UN PROYECTO.....	57
5.4.1	<i>Método PMC (Project Management and Control)</i>	57
5.4.2	<i>Método BISAD (Business Information System Analysis and Design)</i>	57
5.4.3	<i>Método compuesto</i>	58
5.4.4	<i>Metodología a usar</i>	58
6	04. ORGANIZACIÓN DE LOS RECURSOS HUMANOS	60
6.1	ÍNDICE.....	60

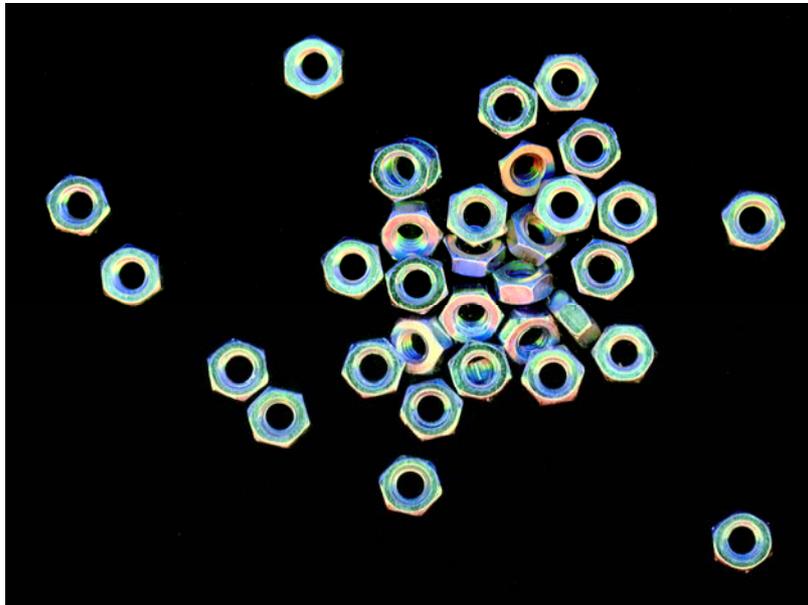
6.2	INTRODUCCIÓN	60
6.3	RECURSOS TÉCNICOS	60
6.3.1	<i>Recursos Hardware</i>	60
6.3.2	<i>Recursos Software</i>	61
6.3.3	<i>Otros recursos técnicos</i>	61
6.3.4	<i>Aportación de los recursos al proyecto</i>	61
6.3.5	<i>Establecimientos de las condiciones por contrato</i>	61
6.4	RECURSOS FINANCIEROS	62
6.5	RECURSOS HUMANOS	62
6.5.1	<i>Organización matricial</i>	63
6.5.2	<i>Determinación de los perfiles de RH en informática</i>	63
6.5.3	<i>Perfiles de RH en proyectos informáticos</i>	64
6.5.4	<i>Ámbitos de aportación de RH</i>	64
6.5.5	<i>Organización de los equipos de trabajo</i>	65
6.5.6	<i>El jefe de Proyecto</i>	66
6.5.6.1	Situación	66
6.5.6.2	Funciones	66
6.5.6.3	El perfil de un Jefe de Proyecto	67
6.5.7	<i>Organización en el Proyecto</i>	67
6.5.8	<i>Ejemplo (1)</i>	67
6.5.8.1	Recursos Técnicos	67
6.5.9	<i>Ejemplo (2)</i>	69
6.5.9.1	Recursos Financieros	69
6.5.10	<i>Ejemplo (3)</i>	69
6.5.10.1	Recursos Humanos	69
7	05. FORMALIZACIÓN DE PROYECTOS	70
7.1	ÍNDICE	70
7.2	FORMALIZACIÓN	70
7.2.1	<i>WBS – Ejemplo</i>	71
7.2.1.1	WBS del proyecto	72
7.3	NORMALIZACIÓN	83
7.3.1	<i>Normalización – UNE 157801 - AENOR</i>	86
7.3.1.1	Contenidos de la Memoria	87
7.3.1.2	Requisitos del sistema	88
7.3.1.3	ANEXOS	88
7.3.1.4	Presupuesto	88
7.3.1.5	Estudios con entidad propia	89
8	06. CALIDAD DE LOS PROYECTOS DE INFORMÁTICA	90
8.1	ÍNDICE	90
8.2	INTRODUCCIÓN Y DEFINICIÓN	90
8.2.1	<i>Adecuación al uso</i>	90
8.2.2	<i>Parámetros que determinan la adecuación al uso</i>	91
8.2.3	<i>Definiciones</i>	94
8.2.4	<i>Modelos de calidad</i>	95
8.2.4.1	ISO 9000-2000	96
8.2.4.1.1	Objetivos de ISO 9000	97
8.2.4.1.2	ISO-9000: La pirámide documental	97
8.2.4.1.3	ISO 9000: Recomendaciones	97
8.2.4.1.4	Ventajas de ISO 9000	98
8.2.4.1.5	Desventajas de ISO 9000	98
8.2.4.2	SPICE	98
8.2.4.2.1	Contenidos de SPICE	99
8.2.4.2.1.1	SPICE (Dimensión Procesos) Categoría CUS	99
8.2.4.2.1.2	SPICE (Dimensión Procesos) Categoría ENG	100
8.2.4.2.1.3	SPICE (Dimensión Procesos) Categoría SUP	100
8.2.4.2.1.4	SPICE (Dimensión Procesos) Categoría MAN	100
8.2.4.2.1.5	SPICE (Dimensión Procesos) Categoría ORG	101
8.2.4.2.1.6	SPICE (Dimensión Capacidad)	101
8.2.4.2.1.7	SPICE (Dimensión Capacidad - Prácticas)	101
8.2.4.2.1.8	SPICE (Evaluación de atributos)	101
8.2.4.3	CMM	102

8.2.4.3.1	Certificación CMM.....	103
8.3	ASEGURAMIENTO DE LA CALIDAD	103
8.4	GESTIÓN DE LA CALIDAD	104
8.5	CONCLUSIONES.....	104
9	07. GESTIÓN DE RIESGOS	105
9.1	ÍNDICE.....	105
9.2	INTRODUCCIÓN	105
9.3	MANEJO DE RIESGOS.....	105
9.4	ALGUNOS EJEMPLOS DE RIESGOS	106
9.5	EL PROCESO DE MANEJO DE RIESGOS.....	106
9.6	IDENTIFICACION DE RIESGOS	107
9.7	ANÁLISIS DE RIESGOS.....	107
9.8	PLANEACIÓN DE LOS RIESGOS	108
9.9	ESTRATEGIAS DE MANEJO DE RIESGOS	108
9.10	ESTRATEGIAS FRENTE AL RIESGO.....	109
9.11	MONITORIZACIÓN DE RIESGOS	110
9.12	FACTORES DE RIESGO.....	110
9.13	GESTIÓN DE RIESGOS	110
9.14	PRIMEROS PASOS.....	110
9.15	IDENTIFICACIÓN DE UN RIESGO	111
9.16	PLAN DE GESTIÓN DE RIESGOS	111
9.16.1	<i>Formato de Plan de Riesgos.....</i>	<i>112</i>
9.17	IDENTIFICACIÓN PERMANENTE.....	113
9.18	METODOLOGÍA.....	113
9.19	CONCLUSIONES.....	114
10	08. PLANIFICACIÓN TEMPORAL.....	115
10.1	ÍNDICE.....	115
10.2	INTRODUCCIÓN	115
10.3	PLANIFICACIÓN TEMPORAL.....	115
10.4	MÉTODOS DE PLANIFICACIÓN	117
10.5	PERT	117
10.5.1	<i>Pasos para construir un PERT</i>	<i>118</i>
10.5.2	<i>Ejemplo PERT</i>	<i>118</i>
10.6	GANTT	121
10.7	EL PROBLEMA DE LOS RECURSOS	122
11	09. SEGUIMIENTO Y CONTROL	124
11.1	ÍNDICE.....	124
11.2	INTRODUCCIÓN	124
11.2.1	<i>El Plan de Proyecto.....</i>	<i>126</i>
11.2.2	<i>Ante el cambio.....</i>	<i>126</i>
11.3	CONTROL DE LA CALIDAD.....	127
11.4	CONTROL DE PLAZOS	128
11.5	CONTROL DE COSTES	129
11.5.1	<i>Tipos de costes.....</i>	<i>130</i>
11.5.1.1	Costes externos	130
11.5.1.2	Costes internos	130
12	10. MODELOS DE DESARROLLO DE PROYECTOS DE SOFTWARE LIBRE.....	132
12.1	ÍNDICE.....	132
12.2	INTRODUCCIÓN	132
12.2.1	<i>¿Por qué hablar de software libre?.....</i>	<i>132</i>
12.2.2	<i>¿Qué es software libre?.....</i>	<i>132</i>
12.2.3	<i>Consecuencias de la “libertad” del software.....</i>	<i>133</i>
12.3	TIPOS DE LICENCIAS.....	135
12.3.1	<i>¿En qué se basan las licencias?.....</i>	<i>135</i>
12.3.2	<i>BSD (Berkeley Software Distribution).....</i>	<i>136</i>
12.3.3	<i>GPL (GNU Public License).....</i>	<i>136</i>

12.3.4	<i>LGPL (Library GPL)</i>	136
12.3.5	<i>Otras licencias</i>	136
12.3.6	<i>Licencias duales</i>	136
12.4	MODELOS DE NEGOCIO	137
12.4.1	<i>Negocios con financiación externa (Financiación pública)</i>	137
12.4.2	<i>Negocios con financiación externa (Financiación pública (motivaciones y caso))</i>	137
12.4.3	<i>Negocios con financiación externa (Financiación privada sin ánimo de lucro)</i>	138
12.4.4	<i>Negocios con financiación externa</i>	138
12.4.5	<i>Negocios con financiación externa (Financiación indirecta)</i>	138
12.4.6	<i>Negocios autofinanciados (Basados en el mejor conocimiento)</i>	139
12.4.7	<i>Negocios autofinanciados (Basados en el mejor conocimiento con limitaciones)</i>	139
12.4.8	<i>Negocios autofinanciados (Basados en ser la fuente de un programa)</i>	140
12.4.9	<i>Negocios autofinanciados (Basados en ser la fuente de un programa con limitaciones)</i> 140	
12.4.10	<i>Negocios autofinanciados (Basados en licencias especiales)</i>	140
12.4.11	<i>Negocios autofinanciados (Basados en venta de marca)</i>	140
12.4.12	<i>Desarrollos sin financiación directa</i>	141
12.4.13	<i>Desarrollos para uso interno</i>	141
12.4.14	<i>Otra clasificación (por OSI, Hecker)</i>	141
12.4.15	<i>Otros modos de financiación</i>	142
12.4.16	<i>Modelos mixtos</i>	142
12.5	CONCLUSIONES	142
PARTE III RESUMEN Y PREGUNTAS DE EXAMEN		144
13	RESUMEN	145
14	PREGUNTAS DE EXAMEN.....	151
14.1	TEMA 1	151
14.2	TEMA 2:.....	154
14.3	TEMA 3.....	157
14.4	TEMA 4.....	160
14.5	TEMA 5.....	162
14.6	TEMA 6.....	165
14.7	TEMA 7.....	168
14.8	TEMA 8.....	171
14.9	TEMA 9.....	174
14.10	TEMA 10.....	176

PARTE I

Introducción



1 Prólogo

La asignatura de PROYECTOS DE INFORMÁTICA, tiene diferentes enfoques y cubre diferentes aspectos de la vida profesional del Ingeniero Técnico en Informática.

Hasta el momento actual el alumno ha sido introducido en las técnicas de desarrollo de software, de diseño de proyectos de hardware y de comunicaciones. Ahora queda el enfoque final de ver el proyecto como un conjunto de actividades y tareas, sujeto a unos presupuestos y a una reglamentación legal o formal y en el que el responsable debe garantizar una calidad suficiente al cliente.

Este nuevo enfoque de la realización de proyectos utiliza de todas las técnicas que el alumno haya conocido a lo largo de la carrera.

Algunas competencias importantes que ofrece esta asignatura son:

1. Familiarizar al alumno en el trabajo como integrante de un grupo, enfrentándolo así con los problemas típicos de comunicación y organización, así como la cooperación para conseguir un fin común: el desarrollo de un sistema Informático, que, dada su complejidad, debe ser abordado por un grupo grande de profesionales.
2. Concienciar al alumno de la importancia de la gestión de proyectos software, hardware y de comunicaciones desde las perspectivas de personal, de problema y de proceso.
3. Conocer, analizar y potenciar los factores que influyen en el éxito de un proyecto.
4. Conocer el papel y uso de los estándares de informática y de la legislación que pueda afectar al desarrollo de sus proyectos.
5. Elegir y aplicar los modelos de desarrollo de proyectos de informática existentes y que el alumno ha adquirido en otras asignaturas.
6. Planificar y gestionar temporalmente un proyecto de informática, con asignación y gestión de recursos.
7. Establecer las medidas de seguimiento y control de un proyecto. Realizar el seguimiento de un proyecto utilizando las herramientas y técnicas adecuadas.
8. Estimar y medir proyectos software. Conocer las diferentes técnicas de estimación.
9. Identificar riesgos. Establecer planes de Prevención, Supervisión y Gestión de riesgos.
10. Establecer políticas de gestión y distribución de versiones. Desarrollar proyectos utilizando herramientas para controlar las distintas versiones de los programas y documentos.

1.1 Contenidos de la asignatura

1. Introducción a los Proyectos de Informática

- ¿Qué es un proyecto?
- Concepto de proyecto de acuerdo a diversas fuentes
- Complejidad del desarrollo de proyectos de informática
- Necesidad de planificación y control

2. Mediciones de Software

- Introducción a las métricas de software
- Tipos de Métricas
- Métricas de proceso y de producto
- Métricas para gestión de proyectos

3. Modelos de Estimación

- Definición
- Aspectos que afectan a la estimación
- Elementos sobre los que hay que efectuar estimación
- Técnicas de estimación
- Técnicas de descomposición
- Técnicas empíricas
- COCOMO
- PUTNAM
- Puntos de función
- Estimación de Costes

4. Organización de recursos humanos

- Principios de dirección de personal.
- El equipo del proyecto informático.
- Organización personal en los proyectos
 - En función del tipo de proyecto
 - En función de la metodología
- Establecimiento de responsabilidades
- Determinación de los perfiles

5. Formalización de Proyectos

- Aspectos de la formalización de proyectos

- Tipos de documentación y estándares de documentación
- Normativa aplicable

6. Calidad de los Proyectos de Informática

- Sistemas de calidad.
- Gestión de calidad.
- Aseguramiento de calidad de software.
- Aplicación de técnicas de revisión
- Pruebas de software: técnicas avanzadas
- Mejora de procesos
 - Modelos de mejora.
 - CMM.
 - SPICE
 - ISO 9001, ISO 9000-3, PSP

7. Gestión de riesgos

- Riesgos de los proyectos de informática. Identificación del riesgo. Tipos de riesgos.
- Proyección del riesgo.
- Reducción, supervisión y gestión del riesgo.
- Riesgos y peligro para la seguridad.
- El plan de Riesgos.

8. Planificación Temporal

- Elementos sobre los que hay que efectuar planificación
- Planificación de recursos
- Distribución de recursos
- Planificación temporal
- Creación del equipo de desarrollo
- Etapas por las que pasa el equipo de desarrollo
- Otras actividades de planificación
- Perfil del planificador
- Errores típicos de la planificación
- Plan de proyecto

9. Seguimiento y Control

- Desarrollo de proyectos: monitorización y control

- Herramientas de control
- Técnicas de control
- Control de plazos y tiempos
- Control económico

10. Modelos de Desarrollo de Proyectos con Software Libre

- Introducción al software libre.
- Situación actual del software libre.
- Modelos de desarrollo del software libre.
- Situación legal del software libre.
- Economía del software libre.
- Influencia del software libre en otros sectores relacionados con la información.
- El software libre en diversos sectores.

1.2 **Bibliografía**

1.2.1 Básica

- "Proyectos Informáticos". Francisco Sanchis Marco. Servicio de Publicaciones EUI/UPM, 1.998
- "Desarrollo y Gestión de Proyectos Informáticos". Steve McConnell. McGraw-Hill, 1997
- "Técnicas de Programación y Control de Proyectos". 3ª Edición. C.Romero López. Pirámide, 1988

1.2.2 Complementaria

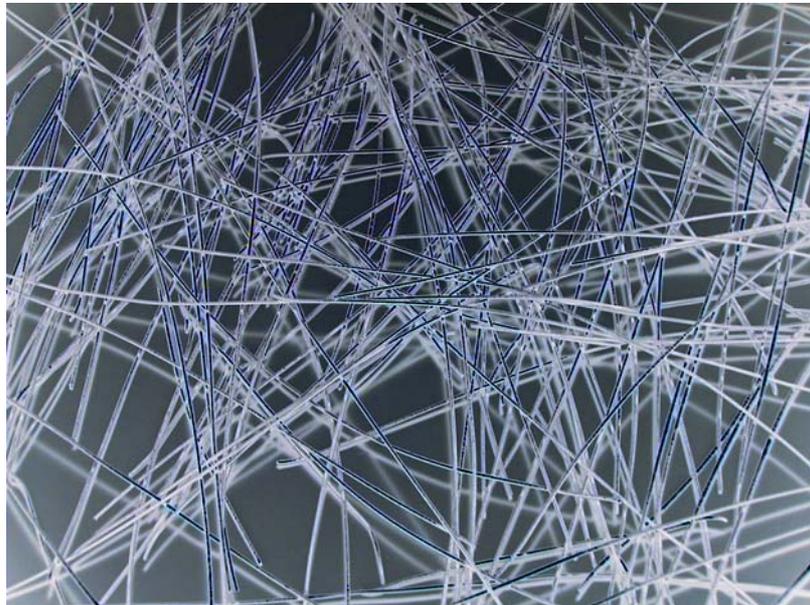
- "A Discipline for Software Engineering", W.S. Humphrey. SEI Series in Software Engineering, Addison-Wesley. 1995.
- "An ISO-9000 approach to building quality software". Oskarsson Östen, 1996.
- "Applying Software Metrics", P. Oman, S.L. Pfleeger (eds.), IEEE Computer Society Press, 1997.
- "CASE. La Automatización del Software". McCLURE, C. Ra-Ma.1992.
- "CMM in Practice: Processes for Executing Software Projects at Infosys (The SEI Series in Software Engineering)", P. Jalote, Addison-Wesley, 1999.
- "Cultivating successful software development. A practitioner's view". Scott E. Donaldson, Stanley G. Siegel, 1997. Editorial Prentice Hall
- "Desarrollo y Gestión de Proyectos Informáticos", McConnell, S., McGraw-Hill, 1997.
- "Desarrollo y gestión de proyectos informáticos", MCCONNELL, S., Madrid, McGraw-Hill / Interamericana de España, 1997.

- “Elementos y Herramientas en el Desarrollo de Sistemas de Información”. PIATTINI, M., DARYANANI, S. Ra-Ma. 1995.
- “Encyclopedia of software engineering”, J. J. MARCINIAK, Nueva York, IEEE (The Institute of Electrical and Electronics Engineers), 1995.
- “Gestión de proyectos con Microsoft Project 2000”, Colmenar Santos, Antonio, Manuel-Alonso Castro Gil, Julio Pérez Martínez, , Ra-Ma, 2001.
- “IEEE Standards Collection: Software Engineering”, IEEE, Nueva York, IEEE (The Institute of Electrical and Electronics Engineers), 1994.
- “Ingeniería del Software: un enfoque práctico (5º Edición)”. Roger S. Pressman. McGraw-Hill, 2002.
- “ISO 9000 for software developers”. Schmauch, Charles H.. ASQC Quality press.
- “ISO 9000-3 A tool for software product and process improvement”. Kehoe Raymond, 1995. Editorial Springer
- “Managing the Software Process”, W.S. Humphrey. SEI Series in Software Engineering, Addison-Wesley. 1990.
- “Medición para la gestión en la Ingeniería del Software”, Dolado Cosín, J. Javier, Luis Fernández Sanz, , Ra-Ma, 2000.
- “Object Oriented Software Metrics”. Lorenz, M., Kidd, J. Prentice Hall.
- “Software configuration management. An investment in product integrity”, BERSOFF, E. H., HENDERSON, V.D. & SIEGEL, S. G., Nueva York, Addison-Wesley, 1980
- “Software Engineering Economics”, B. W. BOEHM, Englewood Cliffs, Prentice Hall, 1981.
- “Software Engineering Project Management”, Thayer, R.H., IEEE Computer Society Press, 1998.
- “Software Engineering Risk Management”, Karolak, D.W., IEEE Computer Society Press, 1996.
- “Software Engineering Standards”, C.Mazza, JFairclough, B.Melton, D. dePablo, A.Scheffer, R.Stevens, Prentice Hall. 1994.
- “Software Engineering with student software guidance”. Barbee, Teasley, Mynatt 1990. Editorial Prentice Hall,
- “Software engineering” 6th edition. Ian Sommerville. Addison-Wesley. 2001.
- “Software Metrics. A rigorous and practical approach”, N. E. FENTON & S. L. PFLEEGER, Nueva York, PWS, 1997.
- “Software Project Dynamics. An Integrated Approach”, Abdel-Hamid, T., S.E. Madnick, Prentice-Hall, 1991.
- “Técnicas de Programación y Control de Proyectos”, Romero, C., Pirámide, 1997.

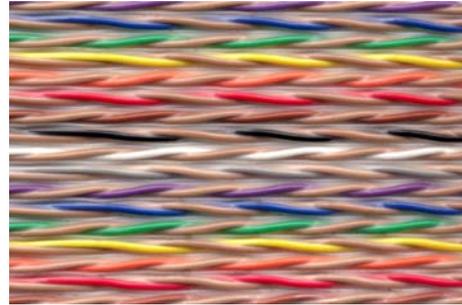
- “The capability maturity model.Guidelines for improving software process”, M. PAULK, C. V. WEBER & B. CURTIS, Reading, Addison-Wesley, 1995.
- “The Mythical Man-Month”, F.P. Brooks, Essays on Software Engineering, Addison-Wesley. 1995.
- “The Unified Software Development Process”, I. Jacobson, G. Booch, J. Rumbaugh, Addison Wesley, 1999.
- Ceballos, 1994 Enciclopedia de Visual Basic Editorial Ra-ma
- Gonzalez Alfons, 1996. Programación de bases de datos con Visual Basic. Editorial Ra-ma.
- MAP (Ministerio para las Administraciones Públicas), <http://www.map.es/csi>.
- “A.y D.de Sistemas de Información”. Senn, J.A.
- “Becoming a technical leader”. WEINBERG, Dorset House, 1986.
- “Controlling Software Projects”. De Marco
- “Desarrollo y gestión de proyectos informáticos”. Steve McConell. Mc Graw Hill.1997.
- “Ingeniería del Software: Aspectos de Gestión”. López-Cortijo y Amescua, Instituto Ibérico de la Industria del Software, 1998.
- “Introduction to the personal software process”. HUMPHREY WATTS, Addison-Wesley, 1997.
- “Manual de Informática y derecho”. PÉREZ LUÑO, Ariel, 1996.
- “Software engineering project management”. THAYER, IEEE-Computer Society Press, 1992.
- “Técnicas de programación y gestión de proyectos”. ROMERO, Pirámide, 1988.
- “The software project manager's handbook”. PHILLIPS, IEEE-Computer Society Press, 1998.
- ANSI/IEEE Std. 1058-1987. IEEE Standard for Software Project Management Plan.
- ANSI/IEEE Std. 729-1983. IEEE Standard Glosary of Software Engineering Terminology.

PARTE II

Contenidos de la asignatura



2 00. Introducción y presentación de la asignatura



2.1 Índice

1. Introducción
2. Objetivos
3. Capacidades
4. Temario
5. Organización
6. Metodología
7. Evaluación
8. Contactos y Página Web
9. Bibliografía

2.2 Introducción

- Asignatura de carácter
 - Técnico
 - Organizativo
 - Económico y Administrativo
 - Comercial
 - Legal y Normativo
 - Etc.

2.3 Objetivos

Familiarizar al alumno en el trabajo como **integrante de un grupo**

- Enfrentamiento a los problemas típicos de comunicación y organización
- Cooperación para conseguir un fin común:
 - El desarrollo de un sistema Informático, que, dada su complejidad, debe ser abordado por un grupo grande de profesionales

Concienciar al alumno de la importancia de la **gestión de proyectos** software, hardware y de comunicaciones

- Perspectivas de...
 - Personal
 - Problema
 - Proceso

Conocer, analizar y potenciar los factores que influyen en el éxito de un proyecto

Conocer el papel y uso de los **estándares** de informática y de la **legislación** que pueda afectar al desarrollo de sus proyectos

Elegir y aplicar los **modelos de desarrollo de proyectos de informática** existentes y que el alumno ha adquirido en otras asignaturas

Planificar y gestionar temporalmente un proyecto de informática, con asignación y gestión de recursos

Establecer las medidas de seguimiento y control de un proyecto

- Realizar el seguimiento de un proyecto utilizando las herramientas y técnicas adecuadas

Estimar y medir proyectos software

- Conocer las diferentes técnicas de estimación

Identificar riesgos

- Establecer planes de Prevención, Supervisión y Gestión de riesgos

Establecer políticas de gestión y distribución de versiones

- Desarrollar proyectos utilizando herramientas para controlar las distintas versiones de los programas y documentos

2.4 Capacidades

El alumno debe adquirir las siguientes capacidades:

- Interpretación de Proyectos para poder ejecutarlos a partir de una Arquitectura de Software, Hardware o de Comunicaciones
- Elaboración de Proyectos sencillos
 - Visión de los factores (legales, administrativos, etc.) que afectan a un proyecto
 - Organización y dimensionamiento de equipos de trabajo
 - Medición del esfuerzo del proyecto, estimación de Proyectos
 - Seguimiento y control del proyecto (Gestión de hitos, Riesgos, costes, información hacia los equipos y hacia el cliente, estado de avance)

- Gestión de la calidad del proyecto
- Uso de criterios adecuados para la toma de decisiones en los proyectos

2.5 Temario

1. Introducción a los Proyectos de Informática
2. Medición de software
3. Modelos de Estimación
4. Organización de los Recursos Humanos
5. Formalización de Proyectos
6. Calidad de los Proyectos de Informática
7. Gestión de Riesgos
8. Planificación Temporal
9. Seguimiento y Control
10. Modelos de Desarrollo de Proyectos de Software Libre

Organización

Carpeta del Alumno (Contenidos) en formato electrónico (RTF, WORD, otros formatos a negociar con los profesores)

- Hoja identificativa del alumno con nombre, email, DNI, etc.
- Informes personales de las discusiones en clase
- Informes que se soliciten
 - Estimación previa de un proyecto
 - Informe sobre la formación del equipo más adecuado para el desarrollo del proyecto del grupo
 - Informe sobre los sistemas de calidad. ¿Cuál es el más adecuado al proyecto en de tu grupo?
 - Definición de un Riesgo
 - Informe personal de seguimiento de la planificación prevista
 - Plantear un modelo de Proyecto de Sw. Libre
- Informe de las tareas realizadas para el proyecto

Buena presentación

- Exposiciones claras (sin rollos)
- Uso correcto del procesador de textos y de otras herramientas ofimáticas y técnicas

Carpeta del Grupo (Contenidos) en formato electrónico (RTF, WORD, otros formatos a negociar con los profesores)

- Hoja identificativa de los alumnos del grupo con nombre, email, DNI, etc.
- Informes solicitados en clase
 - Estimación del coste del proyecto y Arquitectura del sistema
 - Listado de riesgos identificados
 - Informe sobre las diferencias entre el modelo de desarrollo elegido (Software libre o comercial) y lo contrario
- Contenido íntegro del Proyecto siguiendo la norma UNE-157801 y los modelos que se propongan en clase

Buena presentación

- Exposiciones claras (sin rollos)
- Uso correcto del procesador de textos y de otras herramientas ofimáticas y técnicas

	Fecha	Varios	Teoría	Tablero	Carpeta del Alumno	Entregables Individuales	Entregables de Grupo
1	22/02/2006	Introducción y Presentación de la Asignatura					
2	23/02/2006		1. Introducción a los Proyectos de Informática				
3	01/03/2006		2. Medición de software			Estimación previa de un proyecto	
4	02/03/2006			Elección del Proyecto y formación de equipos	Definición del Proyecto y formación del equipo		
5	08/03/2006		3. Modelos de Estimación				
6	09/03/2006			Diseño de la carpeta del Alumno de la carpeta docente	Definición del índice de contenidos de la carpeta docente		
7	15/03/2006	Discusión sobre un tema de actualidad		Diseño de la Arquitectura	Plantilla de equipo de trabajo y responsabilidades	Informe de opinión personal	
8	16/03/2006			Diseño de la Arquitectura	Modelo de Arquitectura		Estimación del coste del proyecto
9	22/03/2006		4. Organización de los Recursos Humanos				Arquitectura del sistema
10	23/03/2006			Definición del equipo de trabajo en el proyecto final	Documento de definición del equipo de trabajo		
11	29/03/2006		5. Formalización de Proyectos			Informe sobre la formación del equipo más adecuado para el desarrollo del proyecto del grupo	
12	30/03/2006			Diseño de los entregables normalizados	Lista y contenidos de los entregables		
13	05/04/2006		6. Calidad de los Proyectos de Informática				
14	06/04/2006			Diseño de los entregables de Calidad	Plan de Calidad + Gestión del cambio		
15	19/04/2006		7. Gestión de Riesgos			Informe sobre los sistemas de calidad. ¿Cuál es el más adecuado al proyecto en su grupo?	
16	20/04/2006			Identificación de riesgos	Plan de Riesgos	Informe de opinión personal	
17	26/04/2006	Discusión sobre un tema de actualidad					
18	27/04/2006		8. Planificación Temporal				Listado de riesgos identificados
19	03/05/2006			Identificación de Tareas e Hitos	Plan de ejecución del Proyecto		
20	04/05/2006			Identificación de Tareas e Hitos	Plan de ejecución del Proyecto		
21	10/05/2006		9. Seguimiento y Control			Definición de un Riesgo	
22	11/05/2006			Identificación de elementos de Control	Plan de ejecución del Proyecto		
23	17/05/2006			Control Económico	Cálculo y seguimiento de los costes		
24	18/05/2006		10. Modelos de Desarrollo de Proyectos de Software Libre			Informe personal de seguimiento de la planificación prevista	
25	24/05/2006			Repaso de los trabajos realizados			
26	25/05/2006			Replanteamiento del proyecto como uno de software libre	Definición de la alternativa		
27	31/05/2006			Solución de dudas y Preparación de la presentación	Presentación terminada y Carpeta del alumno	Planear un modelo de Proyecto de Sw. Libre	Informe sobre las diferencias entre el modelo de desarrollo elegido (Software libre o comercial) y lo contrario
28	01/06/2006	Presentación de Trabajos					
29	07/06/2006	Presentación de Trabajos					
30	08/06/2006	Presentación de Trabajos					

2.6 Metodología

El alumno adquirirá en las clases de teoría los conocimientos y base suficiente para conocer y comprender tanto los riesgos como las técnicas asociadas a la gestión de proyectos de informática.

En las clases de tablero se harán ejemplos prácticos de los temas vistos en teoría, asimismo el alumno deberá ir preparando un trabajo (Carpeta del Alumno) que entregará al final del curso y que estará relacionado con la dirección y control de proyectos.

El alumno se integrará en un grupo de prácticas y realizará un proyecto que irá documentando en un entregable denominado Carpeta del Grupo

Durante el curso el alumno será requerido a leer, para discusión en clase, artículos o temas relacionados con la gestión de proyectos y con aspectos conflictivos o relacionados con ésta.

2.7 Evaluación

Dos tipos de Evaluación

- Evaluación continua
- Examen final

No se consideran “compensables” y será obligatorio superar el cinco en las pruebas y exámenes

No se guardarán notas entre convocatorias

Si el alumno no se presenta a ninguno de...

- Examen final
- Presentación del Ejercicio (Carpeta del grupo)
- Carpeta del alumno (individual)
- Test

Será evaluado como No Presentado

Si se presenta a alguno de ellos será evaluado como Suspenso

2.7.1 Evaluación continua

La evaluación de la parte de Fundamentos teórico-prácticos se realizará mediante la realización de un examen final (test).

Si se suspende el test o la documentación presentada es insuficiente para aprobar se deberá concurrir al examen final.

Durante el curso, tanto el alumno como el grupo deberán ir entregando los informes solicitados. Estos informes tienen un día de entrega y se descontará un punto por cada día de retraso. Aunque sean evaluados a cero puntos deberán ser entregados en cualquier caso para poder optar a la evaluación continua

$$\text{NotaPresentAlumno} = \text{MEDIA} (\text{NotaInformAlumno}) * 0,7 + \text{MEDIA} (\text{NotaInformGrupo}) * 0,3$$

La no presentación de alguno de los informes de alumno y de grupo será evaluada como *NoPresentado*.

Para evaluar el Ejercicio, cada grupo de trabajo entregará el informe resultante de la realización del caso que se le asigne y defenderá los resultados mediante un examen

práctico en clase para el que dispondrá de un tiempo aproximado de entre 15 y 30 minutos. A esta presentación es obligatoria la presencia de todos los miembros del grupo, los que falten serán evaluados como *NoPresentado*.

A esta presentación se sumará la calificación de la carpeta del alumno y los informes

$$\text{NotaEjercicio} = \text{NotaPresentación} * 0,4 + \text{NotaCarpetaAlumno} * 0,3 + \text{NotaPresentAlumno} * 0,3$$

La falta de la presentación o de la carpeta será evaluada como *NoPresentado*.

Asistencia individual a las clases de teoría y de prácticas de tablero:

- La asistencia no es obligatoria pero se valorará positivamente. No es condición suficiente la asistencia para aprobar ni la “no asistencia” para suspender

La calificación final de la asignatura se obtendrá de la siguiente forma:

$$\text{Nota} = \text{NotaEjercicio} * 0,4 + \text{NotaTeoría} * 0,6.$$

2.7.2 Examen final

Los alumnos que no puedan ir a clase habitualmente (< 80% de asistencia) deberán hacer el examen final en la convocatoria de Junio y presentar todos los contenidos de la **Carpeta del Alumno** y de la **Carpeta del Grupo** (Deberán ponerse en contacto con los profesores para determinar el alcance de esta última carpeta) lo mismo que los que asisten a clase.

2.8 Contactos y Página web

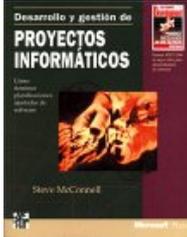
<http://www.di.uniovi.es/~aquilino/> y <http://petra.eutio.uniovi.es/~aquilino/>

Dentro del apartado Asignaturas/Proyectos de Informática.

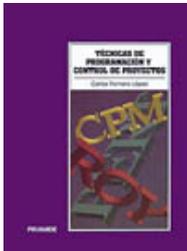
2.9 Bibliografía Básica



“Proyectos Informáticos”. Francisco Sanchis Marco. Servicio de Publicaciones EUI/UPM, 1.998



"Desarrollo y Gestión de Proyectos Informáticos". Steve McConnell. McGraw-Hill, 1997 ISBN: 84-481-1229-6



"Técnicas de Programación y Control de Proyectos". 3ª Edición. C.Romero López. Pirámide, 1988 ISBN:84-368-1151-8

3 01. Introducción a los Proyectos de Informática



3.1 Índice

1. ¿Qué es un Proyecto?
2. Conceptos asociados al Proyecto
3. Concepto de Proyecto
4. Complejidad del desarrollo de Proyectos de Informática
5. Necesidad de Planificación y Control

3.2 ¿Qué es un Proyecto?

La definición de proyecto que da el Project Management Institute (PMI) es:

“Un proyecto es un esfuerzo temporal acometido para crear un único servicio o producto”

Temporal quiere decir que todo proyecto tiene un comienzo claro y un final claro

Único significa que el producto o servicio es diferente de alguna forma clara de todos los productos o servicios similares

Norma UNE 157.801 “Criterios Generales para la elaboración de Proyectos Informatizados de Sistemas de Información”:

Conjunto de actividades coordinadas, controladas, presupuestadas, y documentadas con fechas de comienzo y finalización, que se emprende para alcanzar unos objetivos conforme a requisitos específicos, por una organización temporal adaptada a sus necesidades

Especificación de Proyecto:

- Documento o conjunto de documentos que describen el enfoque técnico y de gestión que va a ser seguido por un proyecto
- La Especificación de Proyecto describe el trabajo que se va a llevar a cabo, los recursos requeridos, los métodos que se van a usar, los procedimientos a seguir, los plazos a cumplir, y la organización que lo va a ejecutar

3.3 Conceptos asociados al Proyecto

Existe un objetivo claro

El producto final tendrá que cumplir las especificaciones

Se desea un determinado nivel de calidad en el producto

Sobre las tareas a realizar

- Se pueden identificar
- No son habituales
- Es necesario realizarlas para conseguir el objetivo
- Tienen que realizarse de forma ordenada

Es necesaria la intervención de varias personas

- Especialistas en Técnicas de Construcción
- Conocedores del Dominio del Problema

Los Recursos

- Se utilizan recursos de diversos tipos
- Existen limitaciones en los recursos

Limitaciones

- El presupuesto es limitado
- Los recursos son limitados
- El objetivo se tiene que alcanzar en plazo de tiempo limitado
- Tiene una fecha de inicio y otra de final

3.4 Concepto de Proyecto

Conceptos ambiguos

- Proyecto, Ejecución, etc.

El proyecto se caracteriza por

- Concreción
 - Objetivo definido
- Unicidad
 - Responde a una demanda o necesidad puntual
- Dominio de Aplicación
 - Versa sobre un dominio de aplicación concreto (Expertos de dominio)
- Flexibilidad

- Disponibilidad de asignación de recursos
- Duración limitada
 - Los objetivos, los plazos, los recursos y el coste están íntimamente relacionados

3.4.1 Fases de los Proyectos



Especificación de Proyecto

- Definición estándar encaminada a describir cómo se ha de ejecutar el proyecto

Ejecución del Proyecto

- Desarrollo del proyecto para conseguir el producto con los criterios acordados

3.4.2 Gestión de Proyectos

Iniciación del Proyecto

- Fija los objetivos

Calificación del Proyecto

- Estimación de la Carga de trabajo
- Estimación del Coste
- Estimación de Riesgos
- Decisión o rechazo de emprender el proyecto

Desarrollo del Proyecto

- Planificación y descomposición en fases o tareas
- Lanzamiento del Proyecto: Ejecución y desarrollo
- Seguimiento y control del proyecto
- Cierre de fases

Cierre del Proyecto

Un Proyecto Informático no es más que un proyecto cuyo objetivo se enmarca en el Plan General Informático de la empresa

La aplicación de técnicas de ingeniería al desarrollo de proyectos software constituye la Ingeniería del Software

Proyecto de Informática ≠ Proyecto Software

3.4.3 Tipo de Proyectos Informáticos

Software

- Metodologías, Ingeniería del software, etc.
- Software paquetizado

Hardware

- Velocidad de Proceso, S.O., Servicios, etc.

Comunicaciones y Redes

- Protocolos, Buses, Cableado, etc.

Instalaciones de Hardware

- Peso de los equipos, Instalación de aire acondicionado, suelo flotante, Extinción de incendios, Conectividad externa, etc.
- CPD's, Sites de Internet, etc.

Sistemas de Misión Crítica

- Industrial, Médica, Nuclear, Militar, Aeronáutica, etc.
- Tiempo real, Esquemas productivos, etc.

Auditorías

- Sistemas, Seguridad, Calidad, Legislación ...

Peritajes

- Civiles, Penales, Laborales ...

Consultoría y Asesoría

- Sobre cualquier actividad

Seguridad Informática (ISO 17799)

- Seguridad de la Información

Reingeniería de Proyectos

- De cualquiera de los tipos

3.5 Complejidad del desarrollo de Proyectos de Informática

Un Proyecto de Informática comprenderá uno o varios de los Tipos de Proyectos anteriores

Complejidad por las características de cada tipo de Proyecto y de su combinación

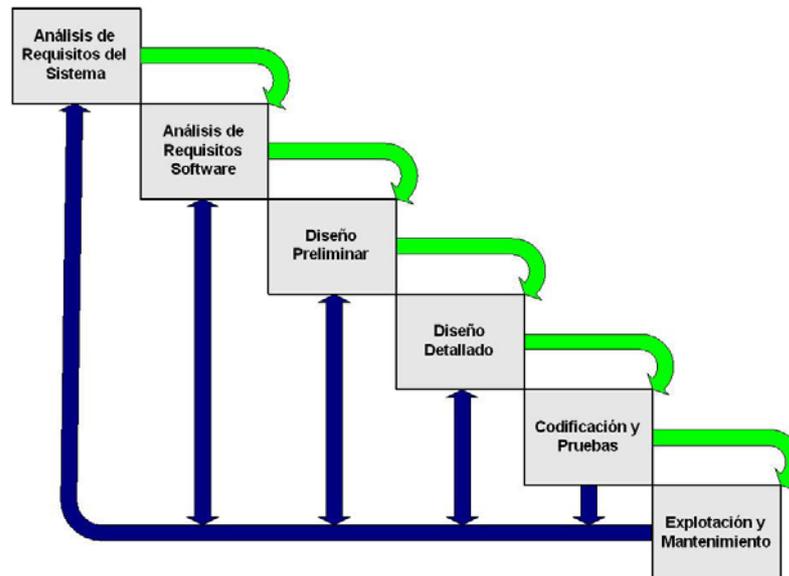
Complejidad especial de los Proyectos de Software

- Interpretación de las especificaciones del cliente
- Fabricación usando elementos de muy bajo nivel (Lenguajes de Programación)
- Complejidad de adaptación de software vertical

Proyectos de Desarrollo de Software

- Crisis del Software
- Metodologías de Desarrollo
 - RUP, Ágiles, etc.
- Paradigmas
 - Estructurado, OOP, Funcional, 4GL, MDA, etc.
- Concepto del Ciclo de Vida de las Aplicaciones software

3.5.1 Modelo en cascada



No refleja realmente el proceso de desarrollo del software

Se tarda mucho tiempo en pasar por todo el ciclo

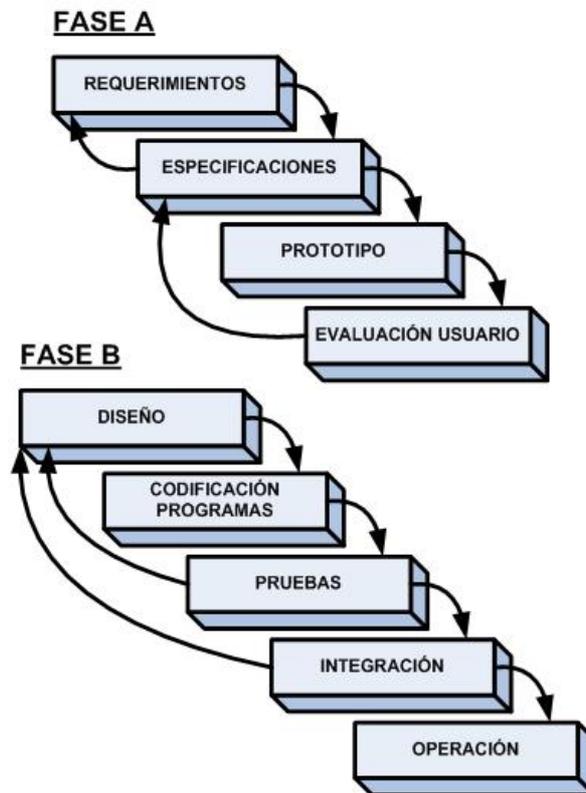
Perpetúa el fracaso de la industria del software en su comunicación con el usuario final

El mantenimiento se realiza en el código fuente

Las revisiones de proyectos de gran complejidad son muy difíciles

Impone una estructura de gestión de proyectos

3.5.2 Modelo de Prototipos



Obtención de modelos de escala reducida

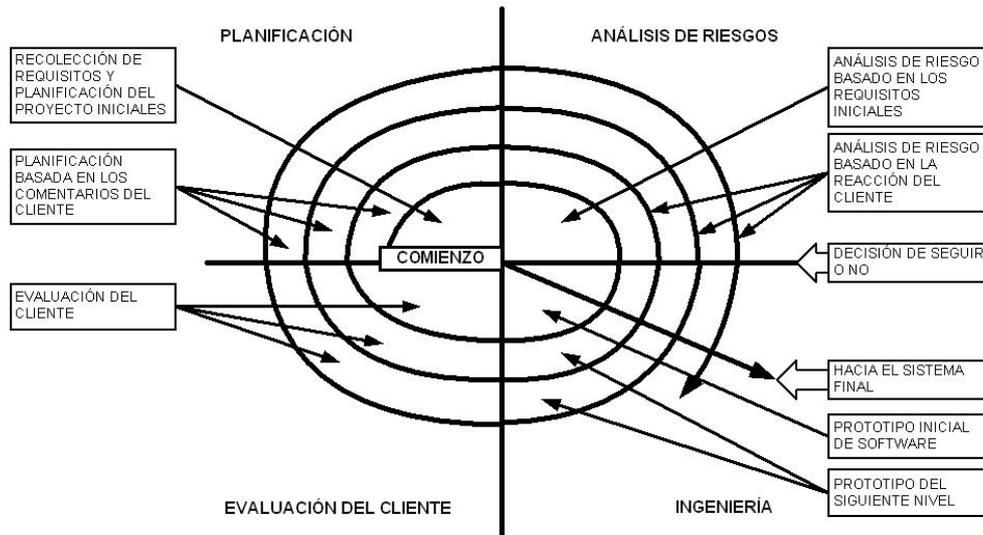
Los modelos se validan por el usuario

Desventaja: Se tiende a reutilizar el prototipo y se le colocan “parches”

Se divide en dos fases:

- Fase A: Diseño, construcción y evaluación de prototipos
- Fase B: Diseño, desarrollo, pruebas e integración

3.5.3 Modelo en espiral



Planificación: determinación de objetivos, alternativas y restricciones.

Análisis de riesgo: análisis de alternativas e identificación / resolución de riesgos.

Ingeniería: desarrollo del producto del "siguiente nivel",

Evaluación del cliente: Valorización de los resultados de la ingeniería.

3.5.4 Modelos Ágiles

Estas son basadas en la adaptabilidad, más que en el carácter predictivo

Son más orientadas a las personas que a los procesos

Para entender mejor los fundamentos de las metodologías ágiles para el desarrollo de software es conveniente revisar un poco la naturaleza de esta actividad

- El diseño especifica las piezas y como ellas se relacionan
- El diseño es la base del plan de construcción
- El plan define tareas y dependencia entre tareas. Permite definir la agenda y el presupuesto de construcción
- El diseño requiere de gente más preparada, creativa y costosa. La construcción de gente menos preparada y costosa
- Un buen diseño establece una forma directa y planificada de construir la aplicación

3.5.4.1 XP (eXtreme Programming)

XP es una Metodología Ágil, los principios generales de esta metodología son los siguientes:

1. **El proceso de planificación:** Define características que se incluirán, estima costos, hace explícitas las necesidades diferidas.
2. **Los pequeños “releases”:** Un sistema simple en producción temprana, se actualiza frecuentemente en un ciclo corto.
3. **Metáfora:** Un sistema de nombres común y una descripción común.
4. **Diseño simple:** Los programas deben ser lo más simples posibles.
5. **Prueba:** El equipo valida el software permanentemente. Las pruebas se hacen antes, durante y después que el código se escribe.
6. **Refactoring:** El diseño se mejora en todo el ciclo, manteniendo el software limpio, sin duplicación, simple y completo.
7. **Pair Programming:** La programación se hace entre dos personas.
8. **Propiedad colectiva:** Todo el código pertenece al grupo.
9. **Integración continua:** El software se integra múltiples veces al día.
10. **40 horas / semana:** Se evitan los sobretiempos excesivos y los equipos cansados.
11. **Cliente en sitio:** Acceso directo a usuarios finales (requerimientos, prioridades, respuestas a preguntas).
12. **Estándar de codificación:** Para compartir el código la codificación debe ser similar.

3.5.5 RUP

RUP es una mitología semiágil, se comporta a veces como ágil y a veces como pesada.

3.5.5.1 Definición

RUP o Rational Unified Process (Proceso unificado de Rational) es la metodología o proceso de desarrollo propuesta por Rational para el desarrollo de proyectos software orientados a objeto

El método RUP definido por un peculiar grupo conocido como The three Amigos (Rumbaugh, Jacobson y Booch) toma las mejores aportaciones de cada uno de los métodos propuestos por los autores, y los unifica en una metodología única que además emplea como lenguaje de modelado un estándar de OMG

3.5.5.2 Características

El modelo de proceso que propone RUP es sin duda la característica más interesante y diferenciadora.

Se trata de un modelo de proceso **iterativo e incremental**. El proyecto se divide en **iteraciones del ciclo de vida**, cada una de las cuales supone se trata como un miniproyecto en si mismo.

Cada **miniproyecto** o **iteración** produce un incremento en el sistema, y abarca las actividades propias de un proyecto en toda regla (análisis, diseño, implementación y prueba).

Cada iteración se construye sobre el resultado de la iteración anterior, y **se guían por el conjunto de casos de uso que se decida abarcar en dicha iteración.**

Algunas iteraciones no producen un efecto de aditivo, sino de refinamiento o corrección, sobre todo en fases tempranas del proyecto donde la labor principal del equipo es comprender el sistema a desarrollar y su contexto de negocio.

3.5.5.3 La Iteración

Para cada iteración:

- Los desarrolladores identifican y especifican los casos de uso relevantes.
- Crean un diseño utilizando la arquitectura seleccionada como guía.
- Implementan el diseño mediante componentes.
- Verifican que los componentes satisfacen los casos de uso.

Si una iteración cumple los objetivos, el desarrollo continúa con la siguiente iteración

Cuando por el contrario no se alcanzan los objetivos prefijados de antemano, se debe revisar las decisiones previas y probar a aplicar un nuevo enfoque

Este tipo de iteraciones controladas aporta según RUP [Jacobson00] las siguientes **ventajas** con respecto al modelo de proceso clásico en cascada:

- Se reduce el riesgo
 - El alcance de un error queda delimitado al tamaño de la iteración
- Se reduce el riesgo de desviaciones en la planificación total del proyecto
 - La identificación de los riesgos más graves se produce durante las fases tempranas del desarrollo
- Acelera el ritmo de esfuerzo del equipo de desarrollo
 - Permite a los desarrolladores apreciar los frutos de su trabajo a corto plazo
- La iteración controlada reconoce la realidad (a menudo ignorada) de que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio del proyecto

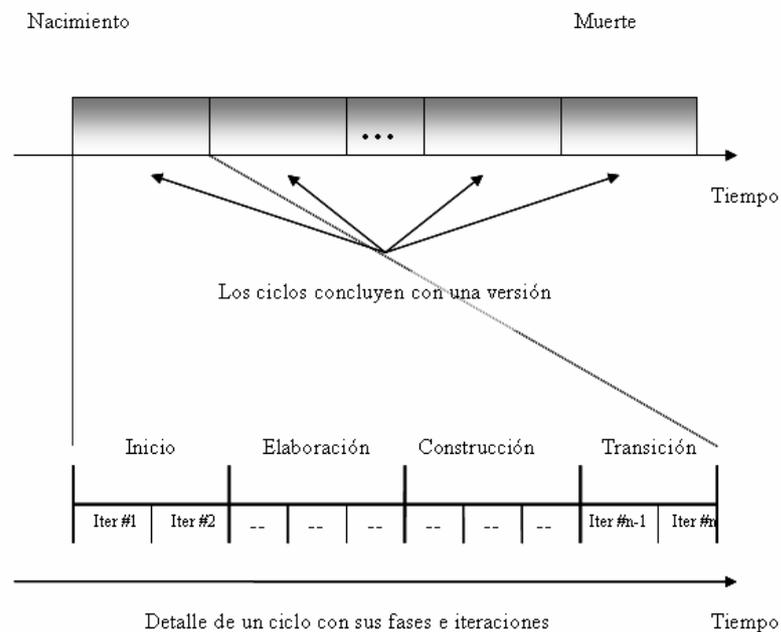
3.5.5.4 Los Ciclos

Para RUP la vida de un sistema se divide en **ciclos**

En cada ciclo se produce una nueva versión del producto para los clientes, y consta de cuatro fases: **inicio, elaboración, construcción y transición**

En cada fase se producirán una o más iteraciones o miniproyectos

3.5.5.4.1 Iteraciones y Ciclos



3.5.5.4.2 Fases dentro de un ciclo

Fase de inicio

- Durante esta fase se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto [Jacobson00]
- Durante esta fase se deben establecer las principales funciones del sistema para los usuarios más importantes, la arquitectura del sistema a grandes rasgos y el plan de proyecto con una aproximación del coste del producto

Fase de elaboración

- Durante esta fase se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema
- Al final de esta fase el responsable del proyecto estará en condiciones de planificar las actividades y estimar los recursos necesarios para llevar a cabo el proyecto

Fase de construcción

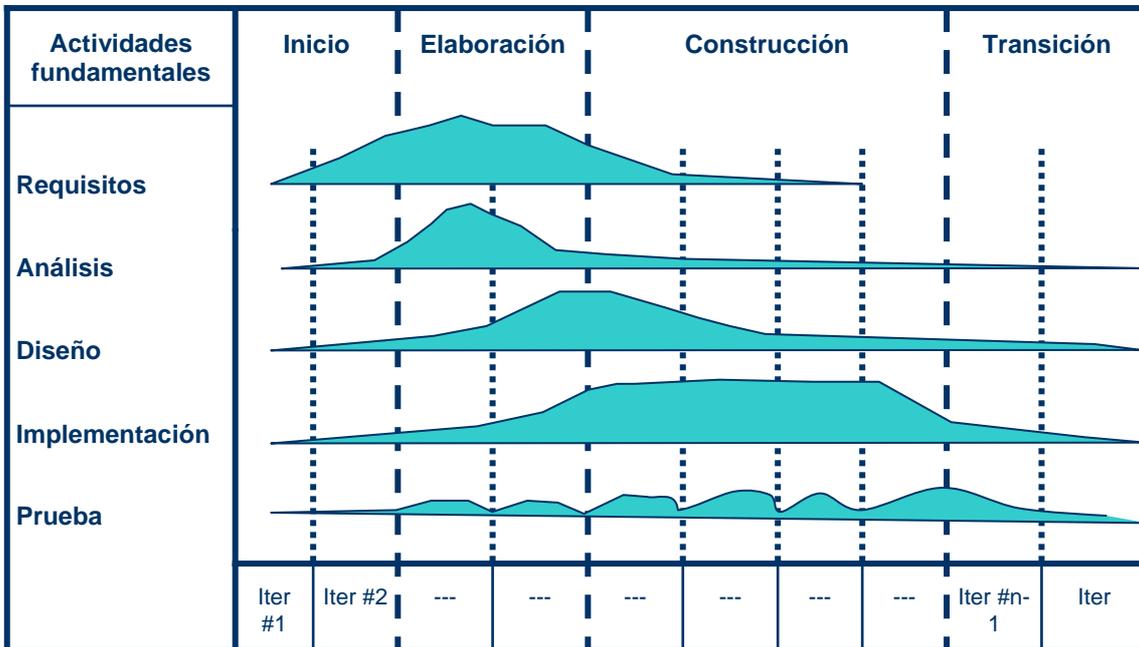
- En esta fase la línea base de la arquitectura obtenida como producto durante la fase anterior crece hasta convertirse en el sistema completo.

Fase de transición

- Cubre el período durante el cual el producto se convierte en versión beta

- La formación al cliente, la asistencia, resolución de incidencias y clasificación de aquellas que justifican una nueva versión del producto.

3.5.5.5 Actividades



3.5.5.5.1 Requisitos

- Enumerar los requisitos candidatos.
- Comprender el contexto del sistema
- Capturar los requisitos funcionales
- Capturar los requisitos no funcionales.

3.5.5.5.2 Análisis

- Análisis de la Arquitectura
- Analizar un caso de uso
- Analizar una clase
- Analizar un paquete

3.5.5.5.3 Diseño

- Diseño de arquitectura
- Diseño de casos de uso
- Diseño de una clase
- Diseño de subsistema

3.5.5.5.4 Implementación

- Implementación de la arquitectura
- Plan de integración del sistema
- Implementación de subsistemas
- Implementación de clases
- Pruebas unitarias

3.5.5.5.5 Prueba

- Planificar prueba
- Diseñar prueba
- Implementar la prueba
- Realizar pruebas de integración
- Realizar pruebas del sistema
- Evaluar prueba
- Métricas sobre las pruebas
 - **Compleción de la prueba**, obtenida a partir de la cobertura de los casos de prueba y de la cobertura de los componentes probados
 - **Fiabilidad**, la cual se basa en el análisis de las tendencias en los defectos detectados y en las tendencias en las pruebas que se ejecutan con el resultado esperado

3.6 Necesidad de Planificación y Control

Un proyecto es algo complejo que es difícil de prever

En la ejecución de un proyecto influyen demasiadas variables y no es posible identificarlas todas

Si no hay control sobre las variables o sobre sus efectos el proyecto puede no lograr sus objetivos

Es necesario utilizar técnicas y herramientas que ayuden en la Gestión de un proyecto

Plan de Proyecto:

- Se puede partir de la Especificación de Proyecto
- Se deben definir los Objetivos, las metodologías a utilizar y las técnicas para cada elemento a controlar
- Dimensionamiento de los Recursos
- Planificación Temporal y definición de hitos
- Determinación de los entregables

Control en diferentes ámbitos:

- Gestión de Recursos
 - Humanos
 - Materiales
- Gestión Económica del Proyecto
- Administración de los Requerimientos
- Control de Calidad
- Gestión de Riesgos
- Control de cambios (administración de la configuración)

3.7 Resumen

Un proyecto de informática es una actividad compleja con un objetivo determinado

Tiene carácter temporal

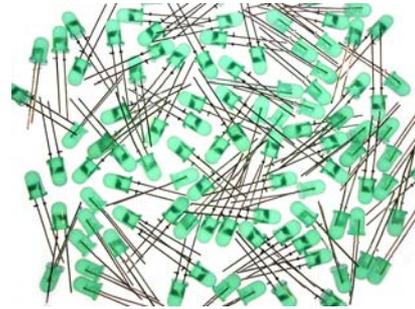
Precisa de Recursos

Los objetivos pueden ser muy diferentes y no solamente software

Debe ser gestionado para lograr que los objetivos se cumplan

El uso de estándares probados y de metodologías puede ayudar en la gestión del proyecto

4 02. Medición de software



4.1 Índice

1. Concepto de métrica
2. Métricas de Hardware
3. Métricas de Servicios
4. Métricas de Comunicaciones
5. Métricas de software
6. Otras métricas relacionadas con la informática

4.2 Concepto de métrica

Necesidad de medir...

Cuando podemos medir aquello sobre lo que hablamos, y expresarlo en números, entonces conocemos algo sobre ello; pero cuando no podemos medirlo, ni expresarlo en números, nuestro conocimiento es de un tipo pobre e insatisfactorio. (Lord Kelvin)

Medir ayuda a...

PROBLEMA	MEDIR AYUDA A
Incorrecciones	Proporcionar requerimientos verificables, expresados en términos medibles
Toma de decisiones	Proporcionar evidencia cuantificable para apoyar las decisiones.
Falta de control	Hacer más visible el desarrollo e identificar problemas anticipadamente
Exceso de gasto	Producir predicciones de coste y plazo justificables
Costes de mantenimiento	Recomendar determinadas estrategias de prueba e identificar los módulos problemáticos
Evaluación de nuevos métodos	Valorar los efectos en la productividad y calidad

4.2.1 Teoría de la Medición - Medición Cuantitativa

4.2.1.1 Sistema Relacional Empírico

Es el mundo real a medir.

Sea $A = \{a, b, \dots, z\}$ el conjunto objeto y K la propiedad elegida para realizar la medición. El sistema relacional empírico, comprende el conjunto modelo elegido y todas las relaciones y operaciones que sobre dicho modelo se desee realizar, esto es, además de A y K comprenderá:

R : conjunto de relaciones n -arias en A .

O : conjunto de operaciones binarias en A .

Entonces, definiremos el sistema relacional empírico (E) como una 3-tupla formada por los elementos anteriormente descritos:

$$E = (A, R, O) = (\{a, b, \dots, z\}, \{r_1, r_2, \dots, r_n\}, \{o_1, o_2, \dots, o_n\})$$

que describe el sistema relacional para una propiedad concreta (K).

4.2.1.2 Sistema Relacional Formal

Representación matemática de las medidas

Una vez elegido el sistema relacional empírico, definiremos el sistema relacional formal que debe ser homomorfo del anterior

Sea $F = (A', R', O')$ dicho sistema. Algunos autores llaman a éste el sistema relacional numérico, estableciendo de esta manera que las mediciones son realizadas siempre con una representación numérica. En este caso y de acuerdo a la definición que se ha hecho de medición, se ha preferido el término formal para indicar que cualquier sistema formal es válido. El sistema relacional formal F debe cumplir las siguientes condiciones:

El sistema formal debe ser capaz de expresar todas las relaciones del sistema empírico E y debe también representar cualquier conclusión significativa prevista originalmente y obtenida a partir de los datos

El paso de E a F debe ser de tal manera que represente todas las observaciones, conservando todas las relaciones y operaciones del sistema empírico relacional

4.2.2 Calidad de la medición, se basa en cuatro conceptos:

4.2.2.1 Escala

Nominales, ordinales, intervalo, ratio, absolutas

4.2.2.2 Unicidad

En cualquier escala la medida debe ser única

4.2.2.3 Representatividad

Condiciones para definir la transformación T

Constructivamente: en este caso se trabaja hacia adelante, esto es, se busca el sistema formal a partir del empírico

Retrospectivamente: en este caso se trabaja hacia atrás, el sistema empírico y formal ya existen (a veces de forma intuitiva)

4.2.2.4 Significación

Una afirmación que incluye escalas (numéricas) es significativa si y sólo si su valor (verdadero o falso) es un invariante bajo transformaciones admisibles.

4.2.3 Ejemplos de Mediciones

A modo de ejemplos de mediciones significativas y no significativas, se puede tomar las tablas realizadas por Fenton en 1988. Algunos ejemplos están listados a continuación:

- *El punto A está al doble de distancia del punto B.*- No es significativa en una escala de ratio porque no existe un punto de referencia. Sería significativa si se tomase nuestra posición actual como punto de referencia.
- *El coste de producción del programa A es el doble que el de producir el programa B.*- Es significativa si el coste está definido en una escala de ratio.
- *Cuesta tres meses escribir el programa A.*- Es significativa para un intervalo de tiempo definido en una escala de ratio.
- *La calidad del programa A es del 97%.*- No es significativa porque no hay referencia a la escala, además no hay consenso acerca de cómo se define la calidad de un programa.
- *La complejidad de este programa es 42.*- No es significativa porque no hay referencia a la escala y además la complejidad de un programa no ha sido definida.
- *La profundidad de este árbol es 5.*- Es significativa, la escala es absoluta pues la profundidad de un árbol está perfectamente definida.

A la vista de estos resultados se comprende que la mayoría de las métricas usadas en software ni siquiera son significativas.

4.2.4 Características de una buena medición

Pero para que una medición sea válida se debe asegurar su calidad, lo que obliga a ésta a tener una serie de características importantes ([FEN91] y [FEN96]):

- Debe ser válida para el propósito que se pretende y debe obtener resultados significativos con respecto a ese propósito
- El código usado en la medición no debe ser arbitrario, debe estar definido de manera que represente adecuadamente las propiedades medidas
- No debe interferir en el proceso observado, de lo contrario los resultados obtenidos estarían falseados
- Debe ser exacta, el error y la incertidumbre de la medición deben estar acotados

- Debe ser reproducible, esto es, debe ser estable y objetiva, permitiendo que en las mismas condiciones, diferentes personas y en diferentes momentos obtengan el mismo resultado
- Debe ser factible de realizar y no debe derrochar recursos ni malgastarlos en cosas irrelevantes
- Los resultados de las medidas deben tener integridad, debe ser posible, por tanto, demostrar la imparcialidad de la medición

4.2.5 Medición orientada a la caracterización del objeto

Medición Directa

- Normalmente no es suficiente

Medición Indirecta

- Composición de medidas

Medición combinada

- Varias mediciones directas e indirectas caracterizan un objeto

Comparación de objetos

- Permite la comparación de objetos en función de varias medidas

4.2.6 Pseudométricas y Métricas

Pseudométricas. Para cada x, y, z

- *identidad*: $\delta(x, y) = 0$
- *simetría*: $\delta(x, y) = \delta(y, x)$
- *desigualdad triangular*: $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$

Una métrica es una pseudométrica que satisface un axioma de identidad más estricto:

- *identidad*: $\delta(x, y) = 0 \Leftrightarrow x = y$

Por tanto una métrica es un concepto formal con base matemática.

Una métrica no es una medida.

Muchas supuestas métricas no cumplen con los conceptos más básicos exigidos, en muchos casos ni siquiera son objetivas e independientes del observador.

4.3 Métricas de Hardware

Las métricas del hardware son muy habituales y hasta incluso cotidianas:

- Velocidad del procesador (MHz, GHz,..)

- Tamaño de la memoria (Mbytes)
- Número de ciclos de máquina por segundo
- Número de máquinas en cluster
- Número de conexiones USB disponibles
- Etc.

4.4 Métricas de Servicios

Son menos conocidas

- Número de usuarios concurrentes
- Tiempo de denegación de servicio (en sistemas de alta disponibilidad y fault-tolerant)
- Número de conexiones perdidas por unidad de tiempo
- Número de sesiones concurrentes
- Porcentaje de fallos en servicio de página web
- Etc.

4.5 Métricas de Comunicaciones

Las métricas de comunicaciones, como las de hardware son muchas veces conocidas intuitivamente

- Velocidad de la línea (Baudios)
- Porcentaje de ocupación del ancho de banda
- Porcentaje de paquetes perdidos
- Tiempo de respuesta
- Relación señal/ruido
- Etc.

4.6 Métricas de Software

Clasificación de las medidas de software [FEN91] y [FEN96]

- Como entidades pueden ser:
 - **Procesos** son actividades que tienen una cierta duración, p. e. la codificación, integración, etc.
 - **Productos** son resultados tangibles de los procesos, p. e. tipos abstractos de datos, objetos codificados, etc.
 - **Recursos** son entradas a procesos, p. e. personal, hardware, otro software, etc.
- Como atributos pueden ser:

- Atributos **internos** de una entidad son atributos que pueden ser medidos en términos de la entidad misma, p. e. tamaño, funcionalidad, esfuerzo, número de errores detectados, etc.
- Atributos **externos** de una entidad son atributos que sólo pueden ser medidos en términos de otras entidades, p. e. mantenibilidad, comprensibilidad, calidad, coste, etc.

4.6.1 Clasificación por el destino a que se dedica

- Métricas de tamaño
- Métricas de estructuras de datos
- Métricas de estructuras de control
- Métricas compuestas
- Métricas de esfuerzo
- Métricas de calidad y fiabilidad
- Métricas de diseño
- Otras métricas de software

4.6.2 Ventajas y Desventajas

- (V) Hay que medir para conocer
- (D) No hay un sistema uniforme de medidas

4.6.3 Métricas de tamaño

- Ejemplo LOC (KLOC)
 - Problemas:
 - Comentarios
 - Líneas en blanco
 - Declaraciones, cabeceras, if, while, '{', '}'...
- Ejemplo “número de tokens”
- Ejemplo “Métricas Científicas del Software” de Halstead
 - n_1 , definido como el número de operadores diferentes en el programa.
 - n_2 , definido como el número de operandos diferentes en el programa.
 - N_1 , definido como el número total de operadores en el programa.
 - N_2 , definido como el número total de operandos en el programa.
 - A partir de estas cuatro métricas básicas, se obtienen otras tres:
 - $n = n_1 + n_2$, llamada el vocabulario del programa.
 - $N = N_1 + N_2$, llamada la longitud del programa.
 - $V = N \log_2 n$, llamada el volumen del programa.

- Se ha demostrado como una métrica muy robusta
- Otras Métricas: COCOMO, Núm. Clases, Num. Funciones, etc.

4.6.4 Métricas de estructuras de datos

- VARS: Número de variables
- n^2 de Halstead = VARS + constantes únicas + etiquetas
- TAYLOR: [TAY93]
 - n° de servicios
 - n° de objetos
 - n° total de mensajes
 - n° de objetos que reciben mensajes
 - n° total de parámetros
 - ratio de servicios públicos/privados
- CHIDAMBER Y KEMERER (MOOSE)
 - Servicios ponderados por clase (WMC)
 - Profundidad del árbol de herencia (DIT)
 - Número de hijos (NOC)
 - Acoplamiento entre clases y objetos
 - Respuesta para una clase (RFC)
 - Falta de cohesión en los servicios (LCOM)
- Otras: MOOD, MOSES, SOMA, etc.

4.6.5 Métricas de estructuras de control

Complejidad Ciclomática de McCabe

Esta medida se deriva del grafo de flujo de control de un programa y mide el número de caminos independientes a través de un programa

Es un indicativo de la complejidad y testabilidad de un programa

También es usado como un método de evaluación de la completud de testeo de las sentencias de un programa.

El Número Ciclomático [MCC76] se determina con la siguiente fórmula:

$$V(G) = e - n + 2$$

donde **e** es el número de arcos del grafo de control, **n** es el número de nodos del grafo de control.

4.6.6 Métricas de esfuerzo

Miden el esfuerzo necesario para llevar a cabo un proyecto

- Persona / día
- Persona / mes
- Tiempo de comprensión
- Otros factores:
 - Intensidad del trabajo
 - Interrupciones
 - Etc.

4.6.7 Métricas de calidad y fiabilidad

La mayor de las métricas de calidad se basan en contar errores o fallos

- Número de cambios (diseño, programa, etc.)
- Densidad de fallos = número de errores / S

Las métricas de fiabilidad

- Probabilidad de error / unidad de tiempo
- Tasa de riesgo de fallo
- Tiempo medio entre fallos

4.6.8 Métricas de diseño

Principios que se miden:

- Acoplamiento
- Cohesión
- Complejidad
- Modularidad
- Tamaño

4.6.9 Otras métricas de software

- Métricas de rehusabilidad
- Métricas de portabilidad
- Métricas de mantenibilidad
- Métricas de flexibilidad
- Métricas de usabilidad
- Etc.

Elección de Métricas

Un problema difícil es el de seleccionar métricas.

No hay consenso acerca de cuáles son las mejores métricas o las que una normativa obliga a aplicar.

Cada responsable de proyecto debe elegir las que le parezcan más indicadas.

Algunas reglas generales:

- No usar métricas compuestas a no ser que estén adecuadamente calibradas y validadas para un entorno particular.
- Elegir sólo aquellas métricas que sean significativas en términos de ingeniería del software y que tengan, por tanto, clara interpretación.
- Asegurarse de que se eligen métricas para cada fase, tanto para análisis y diseño como para implementación, pruebas y tiempo de ejecución.
- Cuando se eligen métricas en tiempo de ejecución, se debe hacer un estudio de los elementos que se desean medir y controlar y diseñar la métrica adecuada a cada caso.
- Usar siempre métricas estándar, comprobadas y calibradas antes que diseñar soluciones ad-hoc. Dejar estas métricas ad-hoc para los casos en que las métricas estándar son claramente inadecuadas.
- Antes de medir plantear y planificar adecuadamente los objetivos de la medición y usar las métricas que realmente tienen significado.
- Medir es una operación de elevado coste en general, por tanto se debe elegir una colección de métricas lo más reducida posible y se debe estimar el coste de realizar esa medición.

5 03. Modelos de Estimación



5.1 Índice

1. Introducción
2. Modelos de estimación del software
3. Coste de desarrollo de un Proyecto

5.2 Introducción

Estimación de proyecto \neq Estimación de software

Los proyectos pueden ser compuestos de

- Hardware
- Software
 - Desarrollado
 - Adquirido
- Comunicaciones

Las estimaciones de costes para desarrollo de software son una parte de la estimación de proyectos

5.3 Modelos de estimación del software

CEM \rightarrow Cost Estimating Model

Modelos de CEM:

- Históricos basados en la experiencia
- De base estadística
- Con base teórica
- Compuestos

Hasta la aparición de métodos empíricos, la estimación de costes y esfuerzos se basaba en apreciaciones personales de los directores de proyectos

La ingeniería del software propugna el uso de métodos científicos

No existe una linealidad entre coste y esfuerzo

- Los métodos deben corregir las desviaciones
- Muchas de estas desviaciones son debidas a características intrínsecas de las empresas

5.3.1 CEM Históricos

Se basan en la conjetura de que el comportamiento de un proyecto debe ser forzosamente similar al de otros proyectos del mismo tipo

Sus resultados sólo son fiables para proyectos similares

5.3.2 CEM con base estadística

Se basan en los CEM-Históricos

A partir de una gran base de datos de proyectos se estudia estadísticamente el comportamiento de las variables

El resultado se aplica al proyecto en curso para estimarlo

5.3.3 CEM con base teórica

Buscan obtener empíricamente las relaciones que determinan el comportamiento de un proyecto

Resultados poco fiables

5.3.4 CEM compuestos

Son los que poseen características de los CEM con base Estadística y los CEM con base teórica

Son conceptos teóricos complementados por la información estadística extraída de la información de base de datos de otros proyectos

Los más complejos son a su vez los más fiables

5.3.5 Problemas de los CEM

No tienen toda la difusión que sería de desear porque

- No son modelos completamente fiables
 - Se deben tener en cuenta las características particulares de cada caso: subjetividad
- Son en general complejos de aplicar
 - Pocas herramientas los implementan bien

- Falta de cultura de que el software es un proceso de ingeniería
 - Se sigue confiando en la experiencia personal

El desarrollo de nuevos modelos más fiables pasa por la resolución del problema de las métricas de software

- La mayoría de los modelos se basan en las KLOC
 - Problemas de cambio de
 - Equivalencia: entre instru. fuente de diferentes lenguajes
 - Esfuerzo: entre uso de diferentes lenguajes
 - Ámbito: ¿Qué es una LOC?

Para utilizar los métodos es necesario tener el diseño

- Imposibilidad según la metodología de obtener estimaciones en los primeros momentos del proyecto.

5.3.6 Generalidades de los CEM

Se basan en la no-linealidad entre esfuerzo y coste

Aplican factores de corrección

$$E = A * f(S)^B + C$$

Donde E es el esfuerzo, f(S) es una función dependiente del tamaño y A, B, C son variables dependientes del modelo.

El tamaño estimado del proyecto es un elemento medido con cierta subjetividad mediante alguna métrica de software, generalmente KLOC o Puntos-Función.

5.3.7 Paradojas de la aplicación de los modelos de estimación

La medición es un punto crítico y pone de manifiesto diversas paradojas:

- La medida de LOC penaliza a los lenguajes de alto nivel
- El coste por defecto o coste para escribir una línea de código penaliza a los programas hechos con calidad
- Los ratios establecidos para actividades como diseño, codificación, integración, etc. varían en respuesta a factores poco considerados o mal cuantificados (capacidad individual, estructura del equipo de trabajo, etc.)

5.3.8 Factores que afectan a la estimación

- **Tipo de aplicación:** Sistema Operativo, Software de gestión,...
- **Tamaño:** Lenguaje empleado: Ensamblador (muchas líneas de código) vs. lenguaje de alto nivel (menos líneas) ¿Qué programador es más productivo?

- **Métodos y herramientas de desarrollo:** Reducción del esfuerzo y del tiempo de desarrollo
- **Complejidad:** Es difícil de medir pero afecta grandemente al proyecto
- **Grado de innovación:** Lo conocido frente a lo desconocido, pero ¿Cómo medirlo?
- **Reusabilidad:** ¿Es reusable? Y si no es totalmente reusable ¿coste de la reutilización?
- **Entorno técnico:** No es lo mismo hacer software stand-alone o software de alta disponibilidad (24x7) para múltiples usuarios
- **Exigencias de documentación:** ¿Cuánto hay que documentar?, exigencias del cliente en el proyecto, etc.
- **Composición del equipo:** Nivel de los componentes, grado de integración, experiencia,...
- **Disponibilidad de medios:** Diversos factores que afectan a la buena marcha del proyecto, disponibilidad de plataforma de desarrollo, plataforma de pruebas, herramientas (CVS, compartición documental, etc...), puestos de trabajo adecuados, etc.

5.3.9 Selección de un CEM

Para seleccionar un CEM se puede valorar:

- **Precisión:** Debe ser preciso en un amplio abanico de aplicaciones (60-90%)
- **Objetividad:** Debe evitar dar la mayor parte del peso de la estimación a elementos de carácter subjetivo o mal ponderados
- **Carácter constructivo:** El papel de cada parámetro en la estimación debe ser conocido para poder ser valorado independientemente
- **Estabilidad:** El modelo no debe mostrar discontinuidades en sus límites debido al carácter binario de algunos parámetros

5.3.10 Modelos históricos o basados en la experiencia

Opiniones de cada experto

Reuniones para acordar el coste final

Desventajas

- Es un modelo enormemente subjetivo
- Problemas con la intencionalidad de cada experto
- Tendencia a la excesiva globalización de la estimación sin considerar los detalles importantes

Puede aplicarse a sistemas top-down o bottom-up

5.3.11 TRW de Wolverton

Método de base histórica

Coste del sistema: matriz de coste del sistema (SCM)

- Coste por línea de código
- Coste por tipo de software
 - Control, E/S, Pre/Postprocesador, Algoritmo, Gestion de datos y Tiempo real
- Coste por dificultad del software
 - Viejo-Fácil (OE, old-easy)
 - Viejo-Medio (OM, old-medium)
 - Viejo-Difícil (OH, old-hard)
 - Nuevo-Fácil (NE, new-easy)
 - Nuevo-Medio (NM, new-medium)
 - Nuevo-Difícil (NH, new-hard)

Dificultad	OE	OM	OH	NE	NM	NH
Tipo-Módulos	1	2	3	4	5	6
1-Control	21	27	32	22	40	49
2-E/S	17	24	27	28	35	43
3-Pre/Post-proc	16	23	26	28	34	42
4-Algoritmo	15	20	22	25	30	35
5-Gest.-Datos	24	31	35	37	46	57
6-Tiempo-Real	75	75	75	75	75	75

$$C(k) = S_s(k) * C_{i(k)j(k)}$$

$$\text{COSTE SISTEMA} = \sum C(k)$$

$C(k) \rightarrow$ Coste del módulo

$S_s(k) \rightarrow$ Tamaño LOC

$C_{i(k)j(k)} \rightarrow$ Entrada en SCM

5.3.11.1 Desventajas

- Mantenimiento al día de la SCM
- Introducir factores como nuevas herramientas o lenguajes
- Introducir la capacidad del personal
- Etc.

5.3.12 Técnica Delphi

Esta técnica es similar al Juicio Experto, excepto que se usan múltiples expertos para tratar de alcanzar un consenso acerca de la estimación del trabajo

- Identificación de los expertos en el tipo de proyecto que se requiere estimar
- Estudio de la información relevante
- Los expertos devuelven su estimación del esfuerzo, así como cualquier suposición, riesgos, etc. que hayan identificado
- Si el número resultante de cada experto está relativamente cerca, se puede usar un promedio de las respuestas como estimación final sin mayor problema
- Si éste no es el caso, se deben regresar las estimaciones y toda la información asociada para que sea revisada
- Se debe pedir a los expertos que consideren la información proporcionada por sus colegas
- Entonces se debe solicitar a cada participante que proporcione una segunda estimación del trabajo
- El proceso puede repetirse varias veces hasta alcanzar un consenso (riesgos)
- Se debe tomar en consideración un factor de riesgo derivado de las observaciones de aquellos expertos que no pudieron llegar a consenso. Por ejemplo, si tres expertos estiman el trabajo alrededor de 1000 horas, pero uno de ellos sostiene la creencia de que el esfuerzo es de 2000 horas, se tiene que ir con el consenso de 1000 horas, asentando un riesgo de que el esfuerzo puede incrementarse hasta por el doble del monto estimado, basándose en la opinión de al menos un experto.

5.3.13 Modelos con base estadísticas

La mayor parte usan el esquema

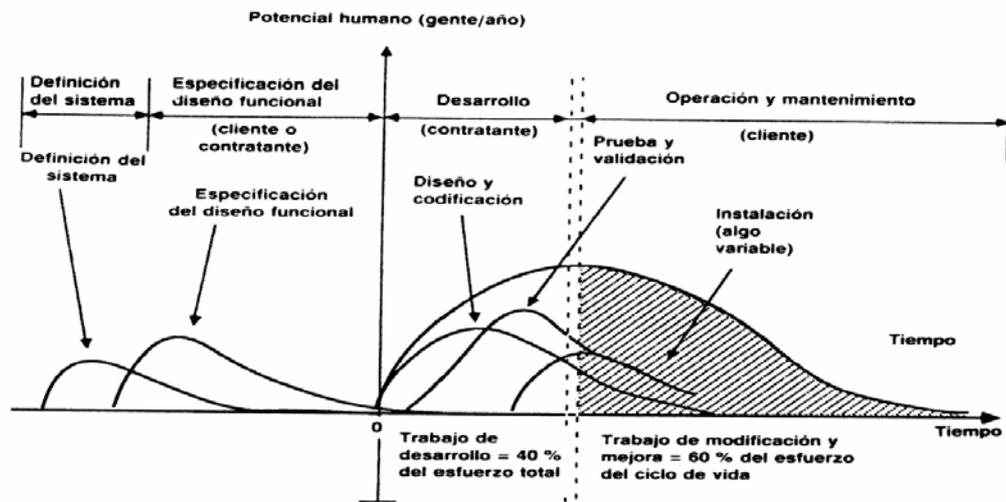
- $E = (a + b S^c) * m(X)$
- S es el tamaño, a, b y c son constantes y m(X) es un elemento ajustador

El principal factor considerado son las LOC y por tanto sus predicciones no son buenas (¿cómo estimar en los inicios del proyecto?)

Principales modelos:

- $E=5.2*(KLOC)^{0.91}$ Modelo de Walston-Felix
- $E=5.5+0.73*(KLOC)^{1.16}$ Modelo de Bailey-Basili
- $E=3.2*(KLOC)^{1.05}$ Modelo simple de Boehm
- $E=5.288*(KLOC)^{1.047}$ Modelo Doty KLOC >10

5.3.14 Curvas de Rayleigh-Norden



Es un modelo con base teórica.

5.3.15 Modelo de Putnam

Modelo con base teórica basado en las Curvas de Rayleigh – Norden

- $Y = 2 * K * a * t * e^{-at^2}$
 - Y = esfuerzo
 - t = tiempo
 - a = parámetro (superficie de la curva)
 - K = área (en intervalo $(0, -\infty)$)
- $Y(t) = K * (1 - e^{-at^2})$
 - $Y(t)$ = esfuerzo acumulado en el periodo t
- dificultad $D = K / T^2$
- productividad $L = S / E$
- tamaño $S = C * K^{1/3} * T^{4/3}$
- C : factor estado de la tecnología
 - Proyectos grandes. Tamaño y tiempo. Factor de nivel de la tecnología (C).
 - Pequeñas prolongaciones de plazos suponen reducciones grandes de esfuerzo
 - Relación no lineal entre tiempo y esfuerzo

5.3.16 COCOMO

CONstructive COSt MOdel. Modelo compuesto

- Ecuaciones analíticas (Teórico)

- Estadísticas de datos propios (Estadístico)
- Juicio de expertos (Histórico)

COCOMO de Boehm:

- Básico
 - Modelo estático válido para estimación rápida al principio del proyecto.
- Intermedio
 - Calcula el esfuerzo de desarrollo como función del tamaño y de una **guías de coste** que incluye una evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- Detallado
 - Incorpora las estimaciones del nivel intermedio teniendo en cuenta la jerarquía de los módulos del sistema y la influencia de varios factores de coste (CDA) sobre las fases individuales del proyecto.
- Expresión:
 - $E = a \cdot S^b \cdot m(X)$
 - Donde a y b dependen del tipo de proyecto y m(X) es función de los CDA del modelo.

5.3.16.1 Factores Multiplicadores

Quince CDA (*Cost Driver Attributes* - Factores multiplicadores):

- Software (3)
 - RELY --- Required system reliability → Fiabilidad requerida
 - DATA --- Size of database used → Tamaño de la base de datos
 - CPLX --- Complexity of system modules → Complejidad del sw
- Hardware (4)
 - TIME --- Execution Time Constraint → Restricciones de tiempo de ejecución
 - STOR --- Main Storage Constraint → Restricciones de memoria
 - VIRT --- Virtual Machine Volatility → Volatilidad del hw
 - TURN --- Computer Turnaround Time → Restricciones de tiempo de respuesta
- Personal (5)
 - ACAP --- Analyst Capability → Calidad de los analistas
 - AEXP --- Applications Experience → Experiencia con el tipo de aplicación
 - PCAP --- Programmer Capability → Experiencia con el hw

- VEXP --- Virtual Machine Experience → Experiencia con el lenguaje de progr.
- LEXP --- Programming Language Experience → Calidad de los programadores
- Proyecto (3)
 - MODP --- Modern Programming Practices → Técnicas modernas de programación
 - TOOL --- Use of Software Tools → Empleo de herramientas
 - SCED --- Require → d Development Schedule → Restricciones a la duración del proy.

5.3.16.2 Tipos de proyectos

- Orgánicos (Organic)
- Semi-acoplados (Semi-detached)
- Integrados (Embeded)

- $E = a \cdot S^b \cdot m(X)$

ESFUERZO	Básico		Intermedio		TIEMPO DE DURACIÓN	
	a	b	a	b	TD=a*Eb	
Orgánico	2,4	1,05	3,2	1,05	2.5	0.38
Semi-acoplado	3,0	1,12	3,0	1,12	2.5	0.35
Empotrado	3,6	1,2	2,8	1,2	2.5	0.32

5.3.16.3 Valor de los CDA

FACTORES (COST DRIVERS)	VALOR DE LOS FACTORES (COST DRIVERS)					
	MUY BAJO	BAJO	MEDIO	ALTO	MUY ALTO	EXTRA
Fiabilidad requerida (producto)	0.75	0.88	1.00	1.15	1.40	
Tamaño de la base de datos (producto)	0.94	1.00	1.08	1.16		
Complejidad del sw (producto)	0.70	0.85	1.00	1.15	1.30	1.65
Restricciones de tiempo de ejecución (hw)			1.00	1.11	1.30	1.66
Restricciones de memoria (hw)			1.00	1.06	1.21	1.56
Volatilidad del hw (hw)		0.87	1.00	1.15	1.30	
Restricciones de tiempo de respuesta (hw)		0.87	1.00	1.07	1.15	
Calidad de los analistas (personal)	1.46	1.19	1.00	0.86	0.71	
Experiencia con el tipo de aplicación (personal)		1.29	1.13	1.00	0.91	0.82
Experiencia con el hw (personal)	1.21	1.10	1.00	0.90		
Experiencia con el lenguaje de progr. (personal)		1.14	1.07	1.00	0.95	
Calidad de los programadores (personal)	1.42	1.17	1.00	0.86	0.70	
Técnicas modernas de programación (proyecto)		1.24	1.10	1.00	0.91	0.82
Empleo de herramientas (proyecto)	1.24	1.10	1.00	0.91	0.83	
Restricciones a la duración del proy. (proyecto)		1.23	1.08	1.00	1.04	1.10

5.3.16.4 Características de COCOMO

COCOMO es un modelo complejo pero efectivo

Los valores de $m(X)$ oscilan entre 0,088 y 72,0

- Dependiendo de los CDA la estimación de tiempos puede tener, para el mismo proyectos estimaciones de 1:800, demasiado margen y demasiado peso de los CDA

5.3.17 Puntos de Función

Es un Modelo compuesto. Estima en base a las 'funciones' que el sistema proporciona al usuario

Adecuado para proyectos 'tipo gestión'

Peculiaridades:

- Independiente del lenguaje de programación
- Estimación a partir de los requerimientos
- 'proximidad' al usuario y su problema

Cinco funciones básicas

- Entradas
- Salidas
- Ficheros maestros
- Consultas
- Interfaces

A tres niveles:

- elemental
- medio
- complejo

5.3.17.1 Parámetros del modelo Puntos de Función

CF → Contador de Función

Parámetro de medida	Factor de peso		
	Elemental	Medio	Complejo
Nº Entradas de usuario (IN)	3	4	6
Nº Salidas de usuario (OUT)	4	5	7
Nº de ficheros maestros (FM)	7	10	15
Nº Consultas usuario (Q)	3	4	6
Nº de interfaces externos	5	7	10

$$CF = \sum_{i=1}^5 \sum_{j=1}^3 w_{ij} x_{ij}$$

5.3.17.2 Factores a estimar

Factores: Evaluar cada factor (lista de 14) en una escala de 0 a 5 :

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Se requieren comunicaciones de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?
6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones?
8. ¿Se actualizan los archivos maestros de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
10. Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidas en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el código para soportar múltiples instalaciones en diferentes organizaciones?

14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

$$c_k = 0,65 + 0,01 \cdot \sum \text{factores de ajuste}$$

5.3.17.3 Valor ajustado

$$PF = c_k \sum_{i=1}^5 \sum_{j=1}^3 w_{ij} x_{ij} \quad 0,65 \leq c_k \leq 1,35$$

El valor calculado PF sirve como parámetro para estimar el esfuerzo (coste) de desarrollo del proyecto

- $E = f(PF)$
- $E = 585.7 + 15.12 * PF$ caso real del 'Transaction on Software Engineering' Abril-94 para 104 proyectos diferentes

5.4 Coste de desarrollo de un Proyecto

Se entiende por desarrollo de un proyecto de software a las tareas de codificación, pruebas e integración

- Método PMC
- Método BISAD
- Método compuesto

5.4.1 Método PMC (Project Management and Control)

Parte de que se tiene un buen conocimiento de las funciones a desarrollar.

Lo hace poco adecuado cuando hay mucha incertidumbre.

Muy orientado a aplicaciones de gestión y entornos estructurados.

Poco orientado a entornos OO.

5.4.2 Método BISAD (Business Information System Analysis and Design)

Se contempla el análisis y el desarrollo.

Tiene en cuenta los aspectos subjetivos del problema.

Contempla las fases de análisis y desarrollo: 60% - 40%

Exige conocimiento general de la aplicación

- $RES = (A+B)*C*(D+E+F)$
- RES se expresa en días
 - A: experiencia general de analistas y programadores (de 1 a 3)

- B: experiencia en la aplicación (1 ó 2)
- C: grado de dificultad aplicación (1.5, 1.75, 2)
- D: grado dificultad E/S (de 1 a 6)
- E: lenguaje de programación
 - 1: Cobol, Pascal, C, C++, Java
 - 2: Ensamblador
 - 3: Adaptación software ya existente
- F: disponibilidad equipo (0, 0.25,1)

5.4.3 Método compuesto

Punto de partida: modelo BISAD

- $RES = (A+B+C+D)*6*E/F$
 - A: grado de conocimiento de aplicación y S.I. (de 0 a 2)
 - B: experiencia en programación (1, 1.5, 2)
 - C: tipo de aplicación (1, 1.5, 2)
 - D: disponibilidad de S.I. (0, 0.25, 1)
 - E: complejidad del desarrollo (1, 1.5, 2, 3)
 - F: disponibilidad de herramienta CASE (1, n)
 - n: factor de productividad
 - $n = 1/GP$
 - GP: ganancia de productividad

Ampliación del método para estimar TIEMPO DE ANÁLISIS Y DISEÑO

- $RES = (A+B)*6*C$
 - A: Experiencia del analista en la aplicación (1, 1.5, 2)
 - B: disponibilidad de interlocutores (0, 0.5, 1)
 - C: dificultad del análisis (de 1 a 6)

5.4.4 Metodología a usar

Partir el proyecto en ítems y distribuirlos en categorías (Identificación de categorías)

- Software
 - A desarrollar
 - A adquirir
- Hardware
 - A desarrollar
 - A adquirir

- Servicios adicionales a prestar
 - Instalación
 - Consultorías
- Coste de las instalaciones
- Coste de amortización de licencias y equipos
- Coste de las integraciones
- Coste de la gestión
 - Coste del personal asimilado al proyecto
 - Costes de representación
 - Otros costes de gestión
- Costes de servicios y materiales contratados para el desarrollo del proyecto
 - Costes de la asesoría tecnológica
 - Costes del material
 - Costes de las licencias
- Otros elementos

Valorar cada ítem utilizando el sistema más adecuado.

Obtener el total del proyecto como sumatorio de los parciales.

Considerar en cada módulo el coste de su integración.

Añadir el porcentaje de beneficio.

Obtener el precio total y aplicarle los impuestos correspondientes.

6 04. Organización de los Recursos Humanos



6.1 Índice

1. Introducción
2. Recursos Técnicos
3. Recursos Financieros
4. Recursos Humanos

6.2 Introducción

Proyecto en marcha

- Planificación del Proyecto
- Planificación de los recursos
 - Técnicos
 - Financieros
 - Humanos
- Naturaleza del recurso x Cantidad x Tiempo

6.3 Recursos Técnicos

6.3.1 Recursos Hardware

Entorno de Desarrollo

- Equipos y Sistemas
- Dispositivos (PDA, Comunicaciones, etc.)
- Lugares de trabajo

Entorno de Pruebas

- Equipos, Sistemas, Dispositivos
- Planificación de las pruebas
- Pruebas de carga y pruebas en explotación

- Lugares de realización de las pruebas

Entorno de Explotación

- Equipos, Sistemas, Dispositivos
- Pruebas de Aceptación del proyecto
- Puesta en marcha del proyecto

6.3.2 Recursos Software

Entorno de Desarrollo

- Compiladores
- Debbugers
- Profilers
- CVS
- Documentación
- Herramientas CASE

Entorno de Pruebas

- Sistemas de Gestión de Prueba unitaria
- Sistemas de Gestión de Pruebas funcionales

Entorno de Explotación

- Bibliotecas (Desarrollo, ejecución)
- Otro software
- Integración de Aplicaciones empresariales (EAI)
 - Middleware

6.3.3 Otros recursos técnicos

Apoyo Técnico

Disponibilidad de documentación

6.3.4 Aportación de los recursos al proyecto

Cliente

Contratista

6.3.5 Establecimientos de las condiciones por contrato

Todas estas condiciones deben establecerse por contrato con el cliente.

6.4 Recursos Financieros

Establecimiento del coste

- Coste del proyecto
- Margen de beneficio
- Concepto de retorno de la inversión

Financiación del Proyecto

- Financiación pública
 - Subvenciones
- Financiación privada
 - Por el cliente
 - Por el contratista
- Financiación mixta

Plan de gastos del proyecto

- Recursos Materiales
- Recursos Humanos
- Gastos financieros (El problema del líquido)
- Otros gastos (Representación, imprevistos, varios,...)

Riesgo económico

- Tamaño de la empresa

Plan de cobros del Proyecto

- Cuadrar costes y gastos
 - Mantener el proyecto en positivo

Desviaciones de gastos

Final del proyecto

- ¡¡¡El proyecto ya no es viable!!!
- Obtención de beneficios

6.5 Recursos Humanos

Disponibilidad de Recursos humanos

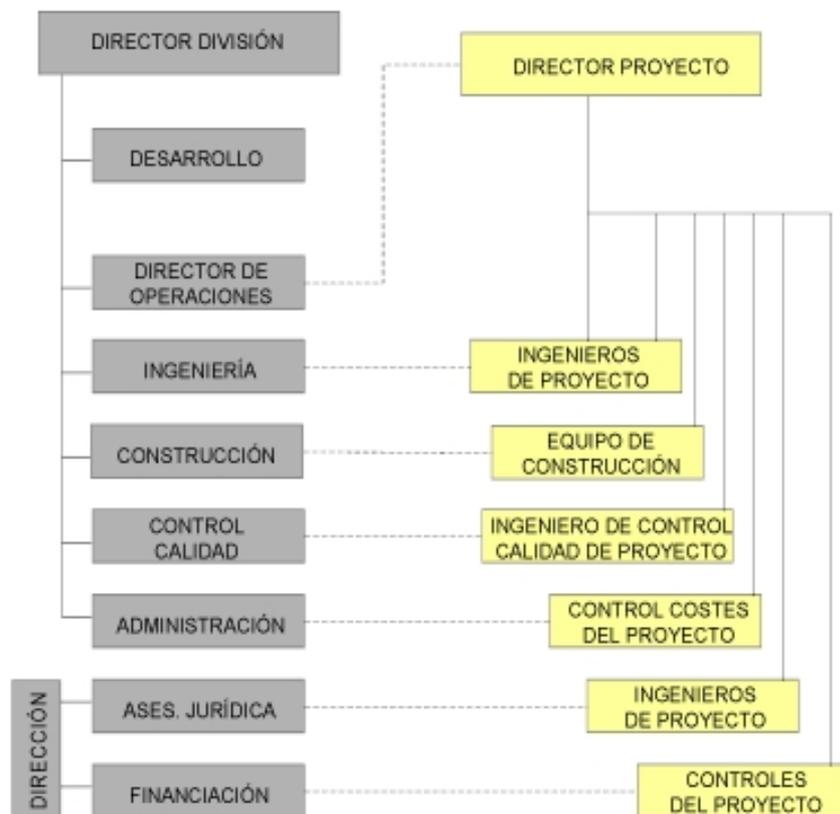
- El valor de los RH
- Disponibilidad just-in-time
- Costes de los recursos

- Formación, Sueldos, Otros

- Organización empresarial de los recursos
 - Organización matricial
 - Organización ad-hoc

6.5.1 Organización matricial

- Unidades funcionales vs Recursos de Proyecto



6.5.2 Determinación de los perfiles de RH en informática

Determinación de un perfil:

- Conocimientos generales requeridos
- Conocimientos técnicos especializados requeridos
- Habilidades de comunicación requeridas
- Actitudes requeridas en el trabajo
- Relación con otros perfiles
- Recursos materiales asociados al perfil
- Características temporales

6.5.3 Perfiles de RH en proyectos informáticos

En un proyecto de Informática se pueden encontrar (sin intención de exhaustividad) los siguientes perfiles:

- Documentalistas
- Diseñadores
- Analistas
- Probadores
- Implementadores
- Comerciales
- Director de Proyecto
- Psicólogos
- Controladores de tiempos
- Administrativos
- Filólogos
- Otros

6.5.4 Ámbitos de aportación de RH

Asignación permanente

- Grupo de personas con unos conocimientos específicos sobre un tema
- Generalmente, esta situación tiene reflejo en la estructura organizativa de la compañía

Asignación temporal

- Personas que se incorporan de la misma unidad organizativa para la ejecución de ese proyecto, pero que al finalizar éste continúan a disposición del Jefe de la Unidad (no necesariamente el Director del Proyecto)

Reclutamiento de nuevas personas

- El proyecto requiere más mano de obra de la disponible, o con conocimientos no disponibles
- Hay que tener en cuenta que, en función de la duración del proyecto, esta situación puede ser inviable puesto que el tiempo requerido para seleccionar y contratar a una nueva persona puede ser muy alto

Transferencia de personas de otros departamentos

- Disponibilidad desde otros departamentos
- Tendencia a aportar el “peor” personal

Consultores

- Personas externas a la organización que poseen conocimientos muy específicos de los que no se dispone internamente
- Están ligados a las tecnologías que se van a utilizar en el proyecto

Subcontratadas

- Personas que van a ejecutar una determinada actividad que se subcontrata
- Body-shopping

6.5.5 Organización de los equipos de trabajo

Dependencia de los miembros del equipo:

- Aislamiento
 - La relación entre los componentes es mínima
 - Las tareas se descomponen en subunidades independientes y el control se basa en relaciones jerárquicas
- Interdependencia
 - Las relaciones se maximizan
 - Las tareas se hacen muy dependientes
- Cooperación
 - Realización de tareas conjuntas
 - Existe un apoyo mutuo entre subunidades

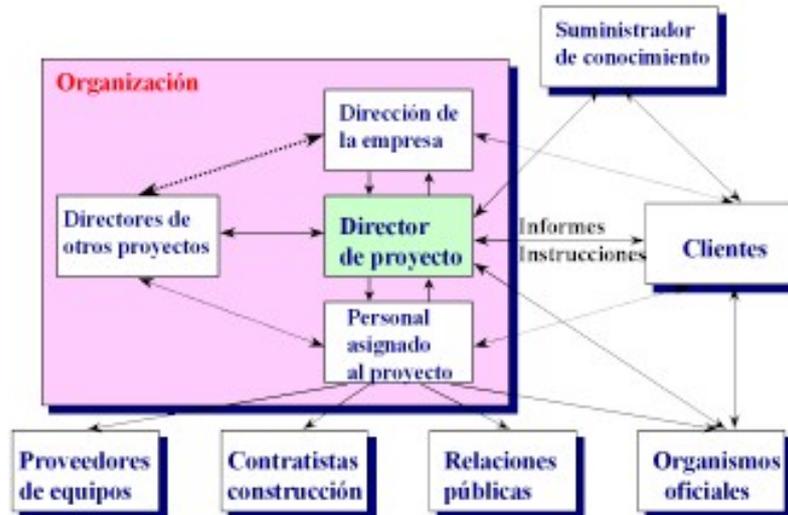
La organización interna de un equipo de trabajo depende fuertemente de dos factores:

- Tamaño del equipo de trabajo
 - Grande
 - Los costes y esfuerzo para la comunicación dentro del equipo de trabajo son altos, requiriéndose la existencia de mecanismos formalizados para ello. Se requieren inversiones tecnológicas para promocionar el trabajo en equipo
 - Se requiere un Director de Proyecto con más experiencia
 - Pequeño. Se caracteriza por:
 - Pueden requerirse generalistas
 - Puede ser adecuado un Director de Proyecto con menos experiencia
- Duración del proyecto
 - Corto. Se caracteriza por:
 - Contribuciones de persona a tiempo parcial
 - Dificultades para justificar la recolocación física del personal
 - Mantenimiento del Director de Proyecto en todas las fases

- Largo. Se caracteriza por:
 - Contribuciones de persona a dedicación plena
 - El Director de Proyecto puede variar con las fases
 - Posible recolocación física del equipo de trabajo

6.5.6 El jefe de Proyecto

6.5.6.1 Situación



6.5.6.2 Funciones

Colaboración con el cliente en la definición y concreción de los objetivos del proyecto.

Planificación del proyecto en todos sus aspectos, identificando las actividades a realizar, los recursos a poner en juego, los plazos y los costes previstos.

Dirección y coordinación de todos los recursos empleados en el proyecto.

Mantenimiento permanente de las relaciones externas del proyecto: clientes, proveedores, subcontratistas, otras direcciones, etc.

Toma de decisiones necesarias para conocer en todo momento la situación en relación con los objetivos establecidos.

Adopción de las medidas correctoras pertinentes para poner remedio a las desviaciones que se hubieran detectado.

Responder ante clientes y superiores de la consecución de los objetivos del proyecto.

Proponer, en su caso, modificaciones a los límites u objetivos básicos del proyecto cuando concurran circunstancias que así lo aconsejen.

6.5.6.3 El perfil de un Jefe de Proyecto

Técnico

- Dominio de la Tecnología
- Capacidad para adaptarse y buscar soluciones

Gestor

- Objetivos financieros
- Coste
- Plazo

Relaciones personales

- Entre empresa y cliente
- Entre equipo y empresa
- Entre equipo y cliente

6.5.7 Organización en el Proyecto

Director del Proyecto

- Cliente

Comité de Dirección

- Director del Proyecto, Jefe de proyecto, otros

Comité de Coordinación Técnica de Proyecto

- Anteriores, más consultores, jefes de equipo, etc

Grupos de Trabajo

- Definición Funcional
- Análisis y Diseño
- Desarrollo
- Pruebas
- Implantación
- Etc.

6.5.8 Ejemplo (1)

6.5.8.1 Recursos Técnicos

Grupo de herramientas enfocadas a la gestión del proyecto (documentación, organización, seguimiento)

- Herramientas ofimáticas:
 - Procesador de texto: Word 2000
 - Hoja de cálculo: Excel 2000
 - Presentaciones: PowerPoint 2000
 - Documentación digitalizada: Adobe Acrobat
- Herramientas planificación:
 - Seguimiento del proyecto: Microsoft Project 2000
- Herramientas de diseño:
 - Microsoft Visio 2000
 - Autocad
- Herramientas de organización de grupos de trabajo:
 - SourceSafe
- Herramientas de gestión documental:
 - Microsoft Share Point Portal Server

Grupo de herramientas enfocadas a la construcción del proyecto (desarrollo, explotación, transporte,..)

- Herramientas de desarrollo
 - JAVA: Jbuilder 9 o Eclipse
 - XML: XMLSpy
 - Macromedia Dreamweaver
 - Tibco Adapter SDK 5
 - Terminal Server o herramienta FTP
- Servidor de Aplicaciones J2EE
 - Bea Weblogic 8.1
- Middleware orientado a mensajes
 - Tibco Rendez Vous
- Sistemas operativos
 - Microsoft Windows 2000
 - Unix
- Bases de datos
 - Oracle 8.1.7
- Presentación
 - Microsoft Internet Explorer

6.5.9 Ejemplo (2)

6.5.9.1 Recursos Financieros

Proyecto con financiación propia (Pagos de Cliente)

Periodo	Fecha	Concepto	Gastos	Acumulado	Ingresos	Acumulado	Total
1		Certificación (Firma)			150.000,00 €	150.000,00 €	150.000,00 €
2		Sueldos	58.000,00 €	58.000,00 €			94.000,00 €
3		Arranque	6.000,00 €	62.000,00 €			88.000,00 €
4		Adquisición Material	23.000,00 €	85.000,00 €			85.000,00 €
5		Sueldos	58.000,00 €	141.000,00 €			9.000,00 €
6		Pago licencias	12.000,00 €	153.000,00 €			-3.000,00 €
7		Certificación (Hito)			100.000,00 €	250.000,00 €	97.000,00 €
8		Sueldos	58.000,00 €	209.000,00 €			41.000,00 €
9		Pago extras	2.500,00 €	211.500,00 €			38.500,00 €
10		Certificación (Entrega)			50.000,00 €	300.000,00 €	88.500,00 €

6.5.10 Ejemplo (3)

6.5.10.1 Recursos Humanos

- Director de Proyecto
 - AAA
- Comité de Dirección
 - AAA, BBB, CCC
- Comité de Coordinación Técnica de Proyecto
 - AAA, BBB, CCC, DDD, EEE, FFF, GGG, TTT
- Grupos de Trabajo
 - Análisis
 - EEE, MMM, NNN
 - Desarrollo
 - FFF, OOO, PPP, QQQ
 - Pruebas
 - GGG, RRR, SSS
 - Implantación
 - HHH, TTT, UUU, VVV

Referencias

<http://www.getec.etsit.upm.es/docencia/gproyectos/gproyectos.htm>

7 05. Formalización de Proyectos



7.1 Índice

1. Introducción
2. Formalización de Proyectos Informáticos
3. Normalización de Proyectos Informáticos

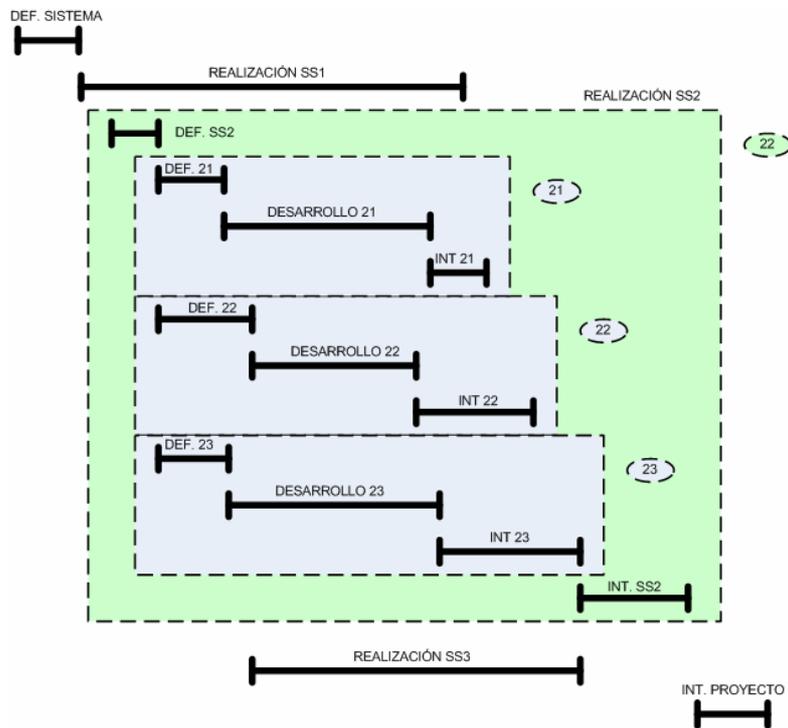
7.2 Formalización

Formalismos:

- WBS: (Work Breakdown Structure)
 - Estructura de descomposición del trabajo total en subtrabajos.
- PBS: (Product Breakdown Structure)
 - Describe el objetivo del proyecto.
- OBS: (Organization Breakdown Structure)
 - Describe quién hace qué.
- RED LÓGICA
 - Tareas, relaciones de dependencias y recursos

7.2.1 WBS – Ejemplo

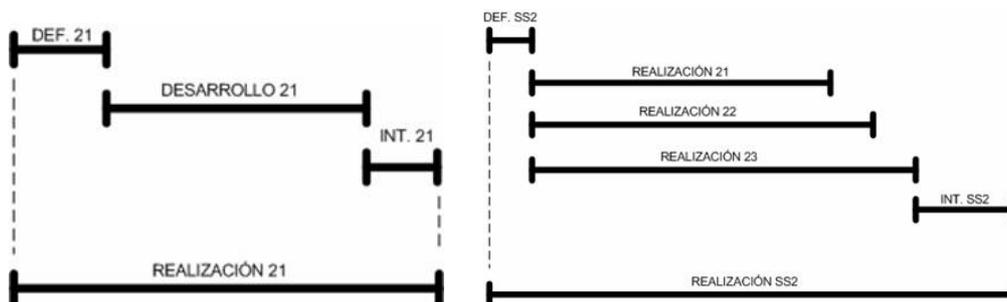
Proyecto: tres subsistemas. El 1 y el 3 subcontratados a terceros y el subsistema 2 descompuesto en tres módulos



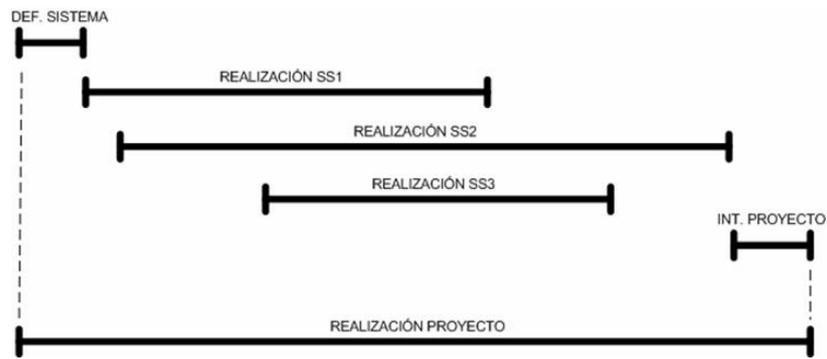
Reagrupamiento de tareas: **tarea hamaca**

- **Definida por**: un conjunto de tareas detalladas
- **Carga**: suma de cargas de sus componentes
- **Comienzo**: la menor de las fechas de comienzo
- **Fin**: la más tardía de sus componentes

No supone ninguna economía de realización tan solo de representación



Planning de síntesis del proyecto que permite ver las articulaciones del proyecto



El reagrupamiento (tarea hamaca) tiene una operación inversa: el 'zoom' que, realizado sobre las componentes del proyecto permite ver de modo detallado una parte del mismo.

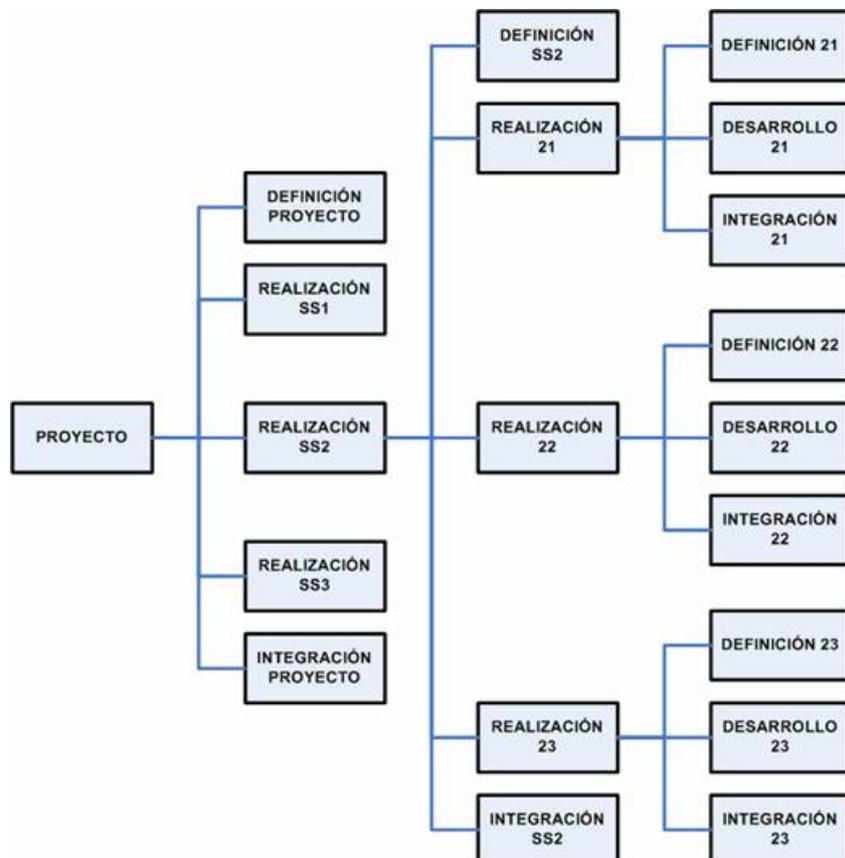
Reagrupamientos y zooms siguen una estructura

En árbol en la que:

- Nodos terminales: tareas
- Otros nodos: tareas hamacas

A esta estructura en árbol se le denomina Work breakdown structure (WBS)

7.2.1.1 WBS del proyecto



Navegación:

- ‘ESTÁ COMPUESTO DE’ Cuando viene de la raíz del árbol
- ‘FORMA PARTE DE’ Cuando se va hacia la raíz
- En el WBS no tiene importancia la disposición de los rectángulos, y no representa ninguna cronología ni encadenamiento (solo enlaces): por ello el WBS de un proyecto no es único

EL WBS permite diferentes tipos de agrupamiento:

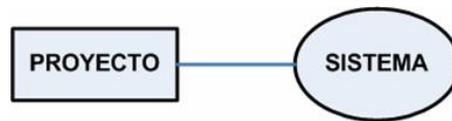
- Por productos
- Por tipos de trabajos
- Por fases de realización
- Por responsables
- etc...

Un WBS completo es, a la vez, simple:

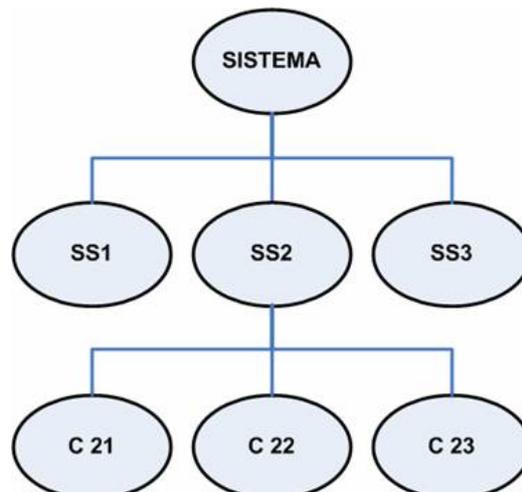
- Es la ‘lista estructurada’ de todos los trabajos de un proyecto

El WBS se refiere a acciones a realizar si lo que se pretende es representar los ‘resultados o productos’ a obtener hay que acudir a otra representación: el PBS.

Relación WBS/PBS:



PBS del proyecto-ejemplo:



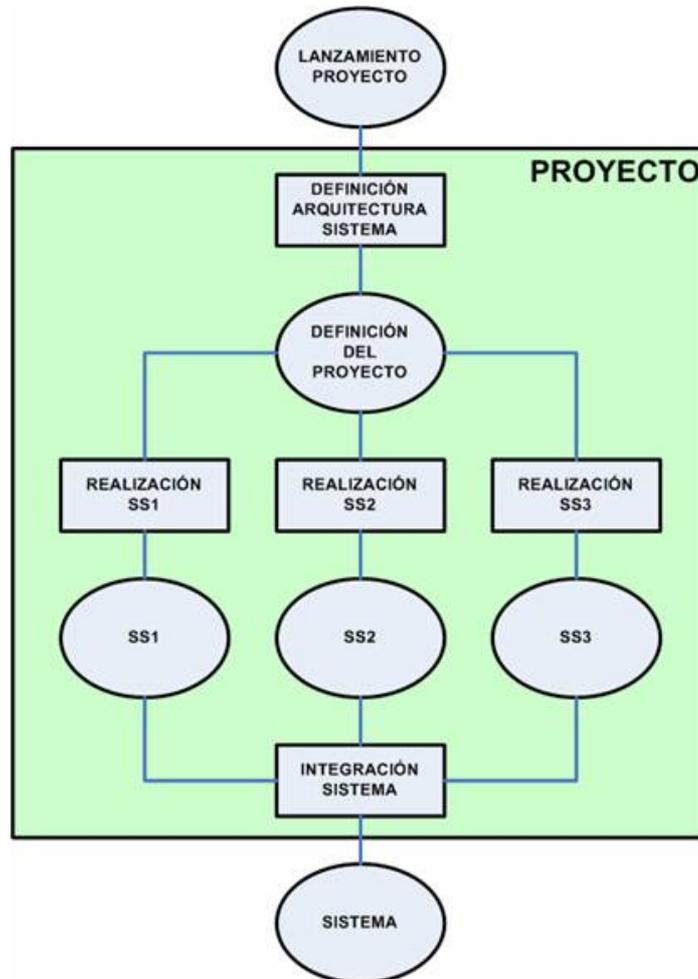
Los rectángulos del WBS son acciones las elipses del PBS son productos (o estados)



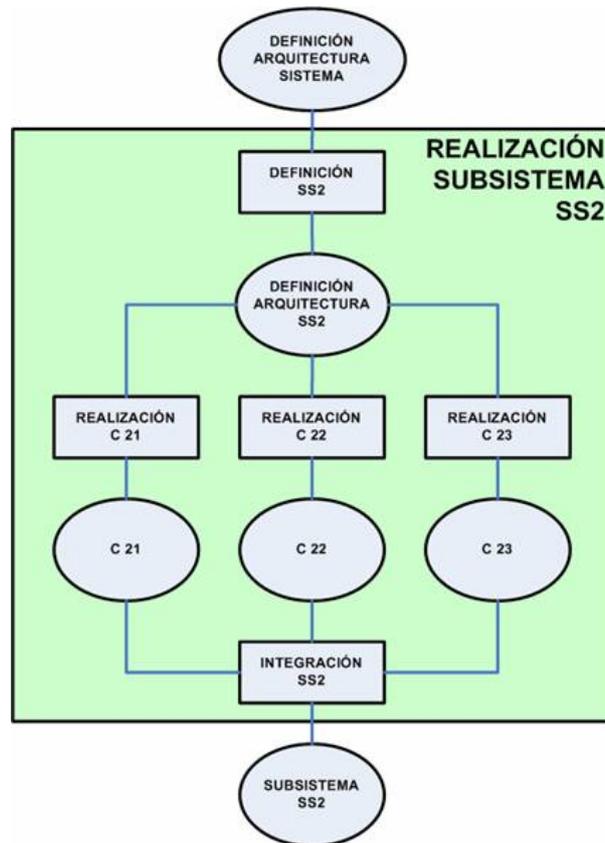
Una acción transforma un estado en otro



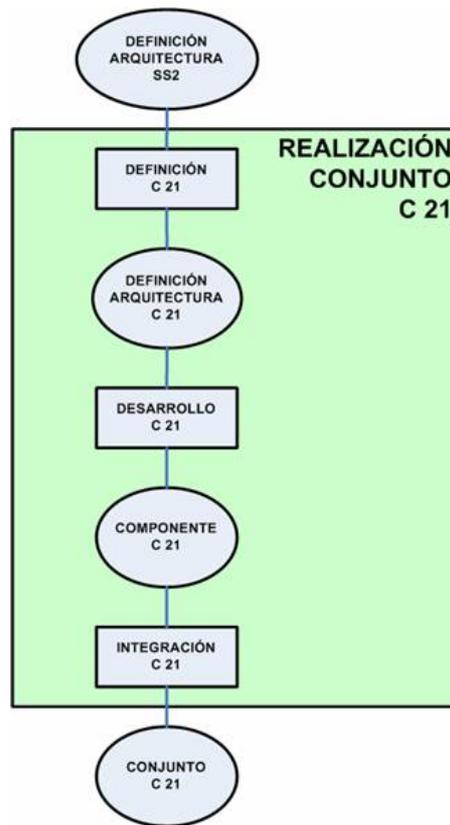
WBS/PBS del proyecto ejemplo (nivel de detalle). La representación elipses/rectángulos hace aparecer un producto nuevo: el cuaderno de cargas del sistema



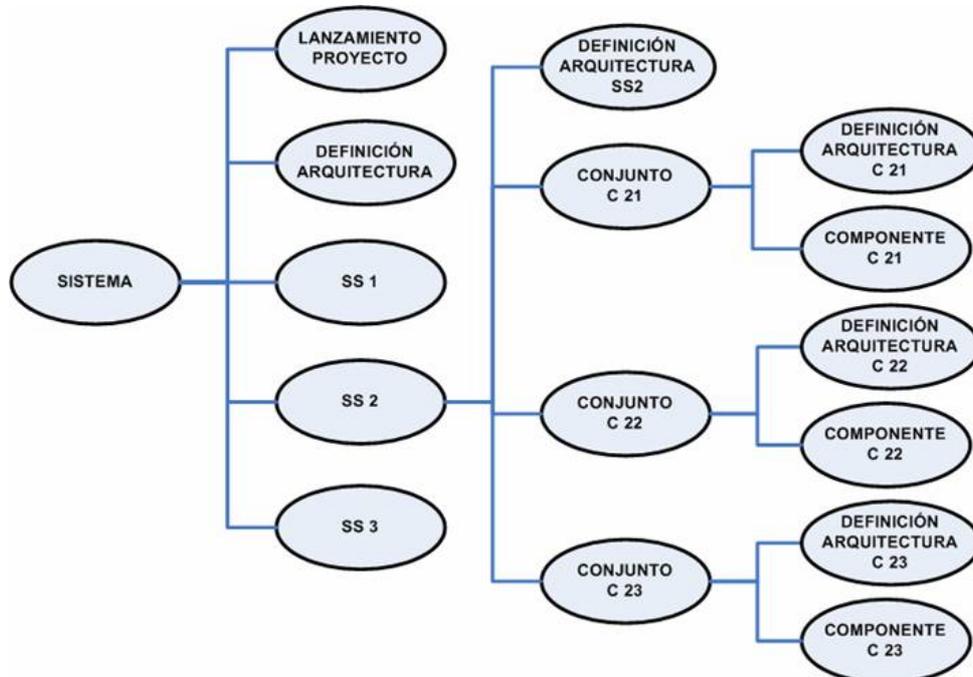
WBS/PBS detallado del subsistema 2 aparece un nuevo producto suplementario: cuaderno de cargas del SS2



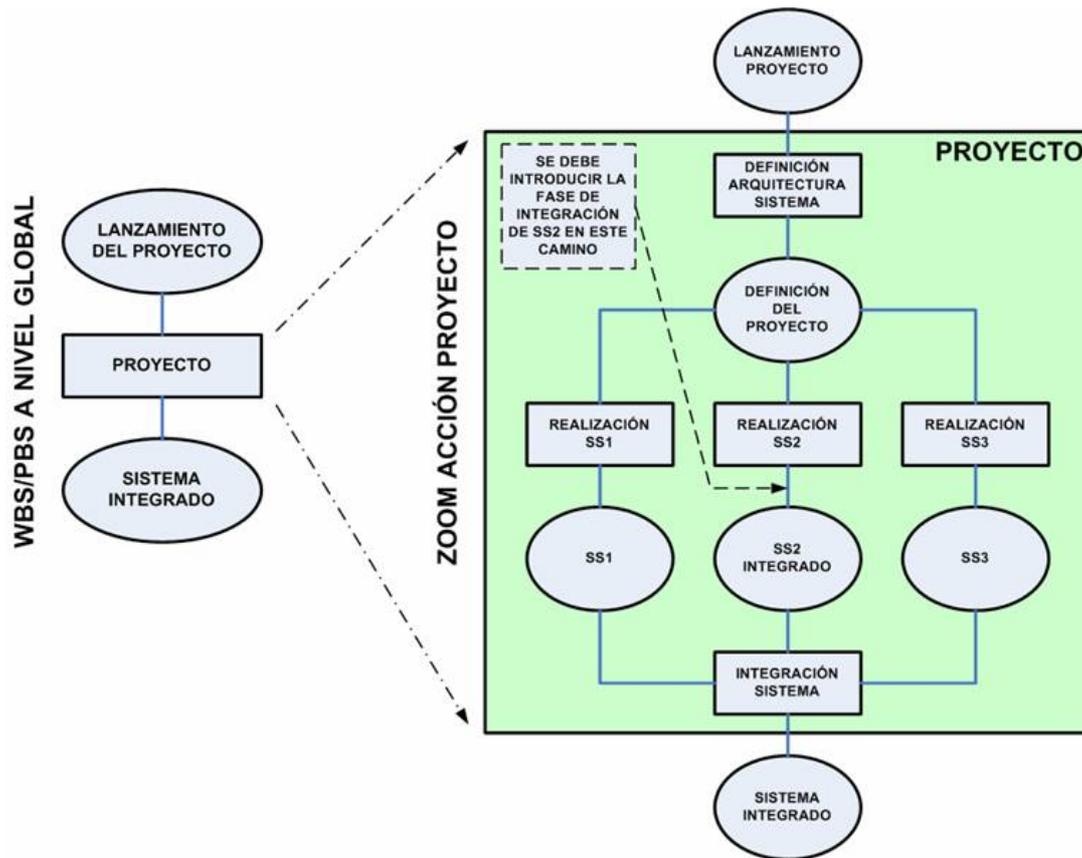
WBS/PBS del Conjunto 21

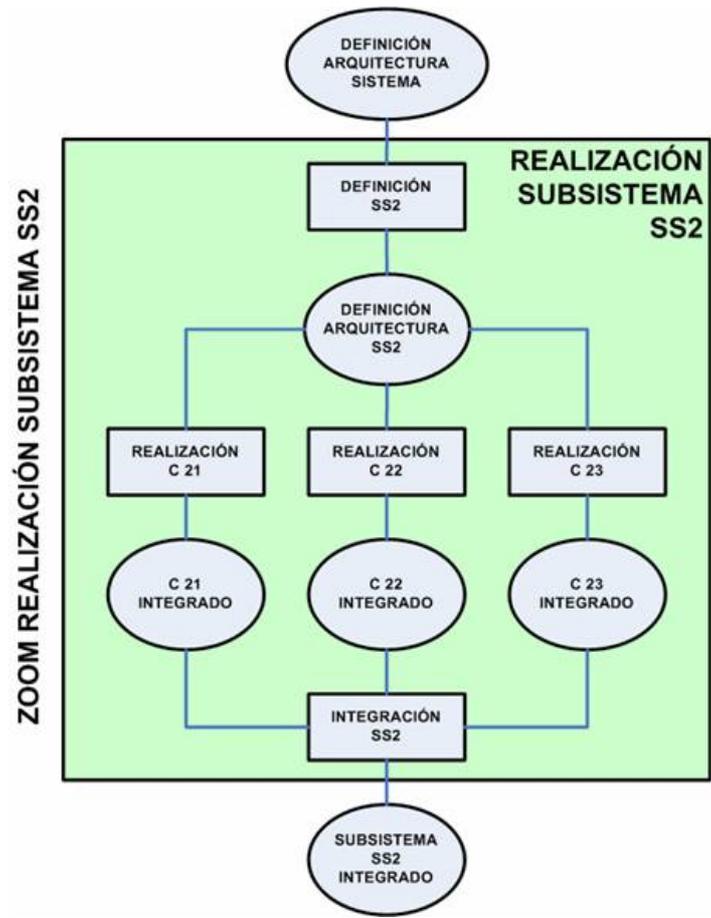


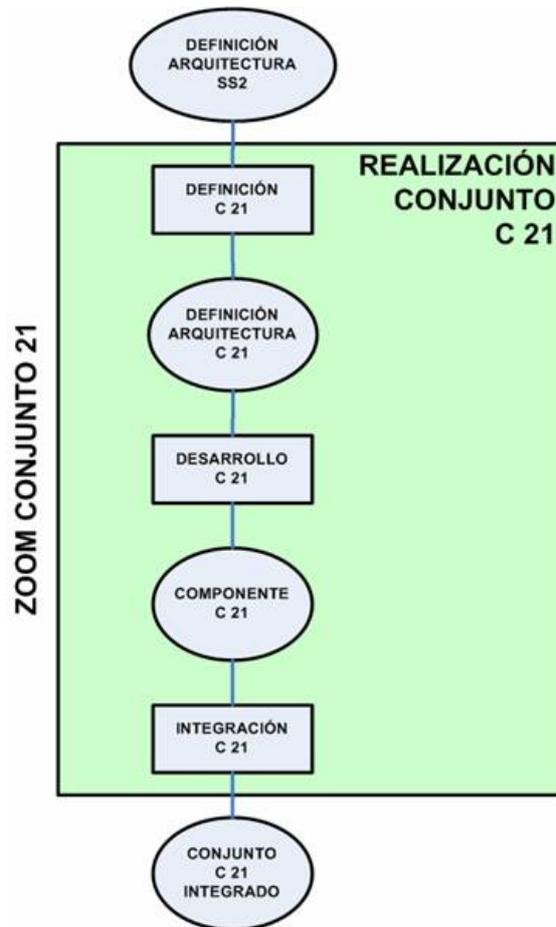
PBS del Proyecto ejemplo



Falta la integración en el subproyecto SS 2.

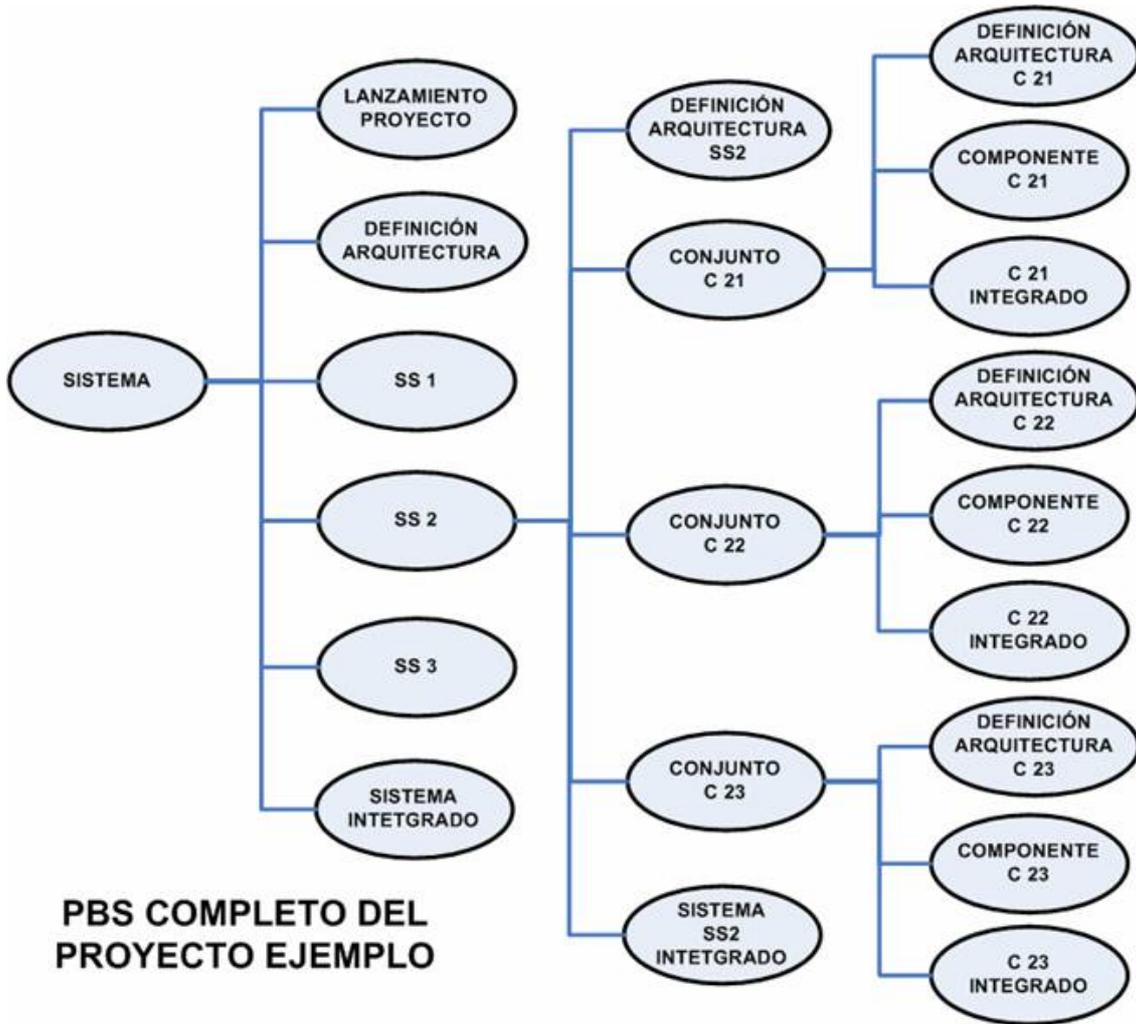






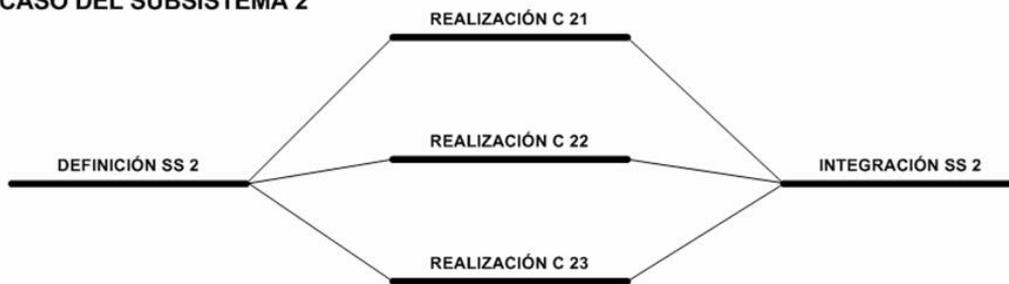
CONJUNTO 21:

- CUADERNO DE CARGAS C21
- COMPONENTES C21
- INTEGRACIÓN C21



La estructura en árbol del WBS induce un mecanismo de herencia de las relaciones de dependencia:

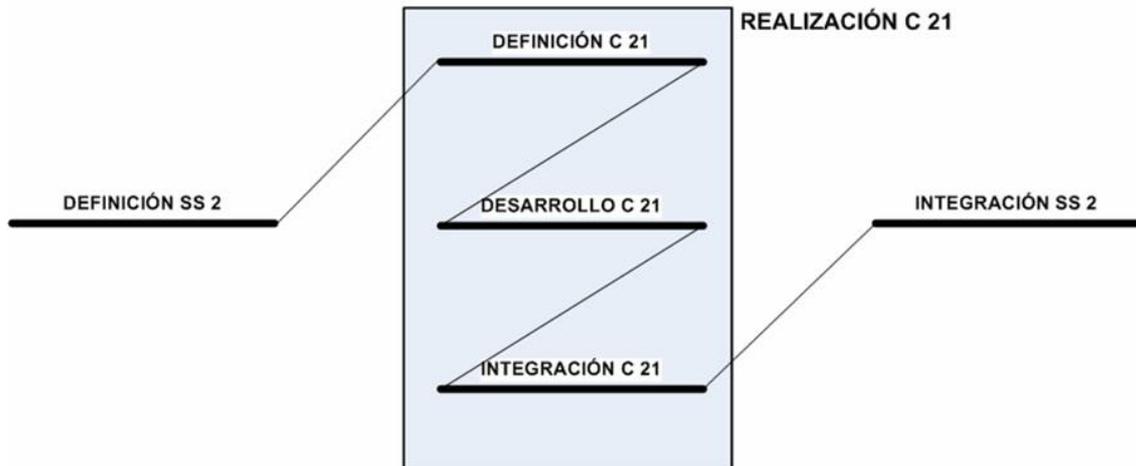
CASO DEL SUBSISTEMA 2



CASO DEL CONJUNTO 21



RELACIONES DE DEPENDENCIA HEREDADAS



RED LÓGICA: recoge el conjunto de los trabajos definidos en el **WBS**, identifica las relaciones y los recursos: ‘**quién hace qué**’ su formalización es el **OBS: organization breakdown structure**

Es una tabla cuyas filas son las tareas del **WBS** y cuyas columnas representan los responsables de su realización

OBS DEL PROYECTO EJEMPLO

	INTERVINIENTES			
	A	B	C	CIA XYZ
DEF. SISTEMA	X			
REAL. SS1				X
DEF. SS2	X	X		
DEF. C21		X	X	
.				
INT. SS2	X		X	
REAL. SS3				X
INT. SISTEMA	X	X	X	X

OBS DEL PROYECTO EJEMPLO CON CARGAS DE TRABAJO

	INTERVINIENTES				
	A	B	C	CIA XYZ	TOTALES
DEF. SISTEMA	100 h*d				100 h*d
REAL. SS1				300 h*d	300 h*d
DEF. SS2	50 h*d	150 h*d			200 h*d
DEF. C21		100 h*d	75 h*d		175 h*d
.					
INT. SS2	25 h*d		125 h*d		150 h*d
REAL. SS3				100 h*d	100 h*d
INT. SISTEMA	200 h*d	100 h*d	100 h*d	250 h*d	650 h*d
TOTALES	TA	TB	TC	TXYZ	TP

OBS DEL PROYECTO EJEMPLO CON ROLES

	INTERVINIENTES			
	A	B	C	CIA XYZ
DEF. SISTEMA	R y P			
REAL. SS1				P
DEF. SS2	A	R		
DEF. C21		R y P	S	
.				
INT. SS2	C		P	
REAL. SS3				A
INT. SISTEMA	A y P	P	P	P

- P – Producción
- R – Responsabilidad
- A – Aprobación
- C – Certificación
- S - Soporte

7.3 Normalización

Una norma es un documento establecido por consenso y aprobado por un organismo reconocido, que provee, para el uso común y repetitivo, reglas, directrices o características para actividades o sus resultados dirigido a alcanzar el nivel óptimo de orden en un concepto dado (ISO/IEC 2:1991).

Los beneficios de la normalización son múltiples, y apuntan, básicamente, a crear criterios mínimos para la aceptación de un producto, proceso o servicio.

La normalización promueve la creación de un idioma técnico común a todas las organizaciones y es una contribución importante para la libre circulación de los productos.

Además, tanto en el mercado local como a nivel global, fomenta la competitividad empresarial, principalmente en el ámbito de las nuevas tecnologías.

Tipos de normas

- Por el tipo de certificación:
 - Certificables
 - No certificables
- Por la obligatoriedad de su cumplimiento
 - Orientativas
 - De uso obligatorio
- Por el ámbito
 - Internacionales, nacionales, regionales, sectoriales, empresariales, etc.

Entidades de normalización:

- Generales
 - ISO
 - IEEE
 - ANSI
 - AENOR
 - Etc.
- Técnicas
 - W3C
 - OMG
 - ECMA
 - DoD
 - Etc.

IEEE - Process Standards:

- IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans
- IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning
- IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans
- IEEE Std 1008-1987 (Reaff 1993), IEEE Standard for Software Unit Testing
- IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation
- IEEE Std 1012a-1998, Supplement to IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997
- IEEE Std 1028-1997, IEEE Standard for Software Reviews
- IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management
- IEEE Std 1045-1992, IEEE Standard for Software Productivity Metrics
- IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans
- IEEE Std 1059-1993, IEEE Guide for Software Verification and Validation Plans
- IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes
- IEEE Std 1219-1998, IEEE Standard for Software Maintenance
- IEEE Std 1490-1998, IEEE Guide--Adoption of PMI Standard--A Guide to the Project Management Body of Knowledge

IEEE - Product Standards:

- IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software
- IEEE Std 982.2-1988, IEEE Guide for the Use of Standard Dictionary of Measures to Produce Reliable Software
- IEEE Std 1061-1998, IEEE Standard for Software Quality Metrics Methodology
- IEEE Std 1063-1987 (Reaff 1993), IEEE Standard for Software User Documentation
- IEEE Std 1465-1998, IEEE Standard Adoption of International Standard ISO/IEC 12199:1994 (E) --Information Technology--Software packages--Quality requirements and testing

IEEE - Resource and Technique Standards

- IEEE Std 829-1998, IEEE Standard for Software Test Documentation
- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Documentation
- IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions

- IEEE Std 1044-1993, IEEE Standard Classification for Software Anomalies
- IEEE Std 1044.1-1995, IEEE Guide to Classification for Software Anomalies
- IEEE Std 1320.1-1998, IEEE Standard for Functional Modeling Language--Syntax and Semantics for IDEF0
- IEEE Std 1320.2-1998, IEEE Standard for Conceptual Modeling Language--Syntax and Semantics for IDEF1X97(IDEFobject)
- IEEE Std 1348-1995, IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools
- IEEE Std 1420.1-1995, IEEE Standard for Information Technology--Software Reuse--Data Model for Reuse Library Interoperability: Basic Interoperability Data Model (BIDM)
- IEEE Std 1420.1a-1996, Supplement to the IEEE Standard for Information Technology--Software Reuse--Data Model for Reuse Library Interoperability: Asset Certification Framework
- IEEE Std 1430-1996, IEEE Guide for Information Technology--Software Reuse--Concept of Operations for Interoperating Reuse Libraries
- IEEE Std 1462-1998, IEEE Standard Adoption of ISO/IEC 14102:1995--Information Technology--Guidelines for the Evaluation and Selection of CASE Tools

ISO Standards:

- ISO/IEC FCD 9127 Software engineering -- User documentation and cover information for consumer software packages
- ISO/IEC CD 12119 Software Engineering -- Software product evaluation -- Requirements for quality of Commercial Off The Shelf software product (COTS) and instructions for testing
- ISO/IEC AWI 14143-1 Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts
- ISO/IEC FCD 14143-6 Information technology -- Software measurement -- Functional size measurement -- Part 6: Guide for use of ISO/IEC 14143 series and related international standards
- ISO/IEC FDIS 14764 Software Engineering -- Software Life Cycle Processes -- Maintenance
- ISO/IEC 15414:2002/FPDAmD 1 Substantive Changes and Additional Text Clarifications and Correction of Editorial Errors
- ISO/IEC FCD 15476-4 Information Technology -- CDIF Semantic Metamodel - - Part 4: Data Models
- ISO/IEC FCD 15476-6 Information Technology -- CDIF Semantic Metamodel - - Part 6: State/Event Model
- ISO/IEC FCD 15504-5 Information technology -- Process Assessment -- Part 5: An exemplar Process Assessment Model

- ISO/IEC FCD 15940.2 Information Technology -- Software Engineering -- Environment Services
- ISO/IEC FCD 16085 Information technology -- Software life cycle processes -- Risk management
- ISO/IEC 19501 Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2
- ISO/IEC DTR 19759 Software Engineering -- Guide to the Software Engineering Body of Knowledge -- SWEBOK
- ISO/IEC CD 19770 Information technology -- Software Asset Management Process
- ISO/IEC WD 19793 Information technology -- Open Distributed Processing -- Use of UML for ODP system specifications
- ISO/IEC DIS 20000-1 IT service management -- Part 1: Specification for service management
- ISO/IEC DIS 20000-2 IT service management -- Part 2: Code of practice for service management
- ISO/IEC DIS 23025 Common Industry Format for Usability Test Reports
- ISO/IEC DIS 23026 Software Engineering -- Recommended Practice for the Internet -- Web Site Engineering, Web Site Management, and Web Site Life Cycle
- ISO/IEC NP 24744 Standard Metamodel for Software Development Methodologies
- ISO/IEC AWI 24748 Systems and Software engineering -- Life cycle management
- ISO/IEC FDIS 25000 Software Engineering -- Software Product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE
- ISO/IEC CD 25020 Software and System Engineering -- Software quality requirements and evaluation (SQuaRE) -- Quality measurement -- Measurement reference model and guide
- ISO/IEC DTR 25021 Software and System Engineering -- Software Product Quality Requirements and Evaluation (SQuaRE) -- Measurement Primitives
- ISO/IEC CD 25030 Software engineering -- Software quality requirements and evaluation (SQuaRE) -- Quality requirements

7.3.1 Normalización – UNE 157801 - AENOR

Propuesta de Norma UNE 157801: “Criterios Generales para la elaboración de Proyectos Informatizados de Sistemas de Información”

Esta norma tiene por objeto establecer las características generales que deben ser cubiertas en los proyectos de los Sistemas de Información a realizar, para que satisfagan los fines a los que están destinados.

El Proyecto debe constar de los siguientes documentos básicos:

- Índice General
- Memoria,
- Requisitos del Sistema
- Anexos
 - Diseño del Sistema
 - Estimación de Tamaño y Esfuerzos
 - Planes de Gestión de la Ejecución del Proyecto
 - Gestión del alcance del proyecto.
 - Gestión de plazos del proyecto.
 - Gestión de costes del proyecto.
 - Gestión de la calidad del proyecto.
 - Gestión de recursos humanos del proyecto.
 - Gestión de comunicaciones del proyecto.
 - Gestión de riesgos del proyecto.
 - Gestión de adquisiciones del proyecto.
 - Seguridad
- Presupuesto
- Estudios con Entidad Propia, cuando proceda

7.3.1.1 Contenidos de la Memoria

- M1 Introducción
- M2 Objeto
- M3 Antecedentes
- M4 Descripción de la situación actual
- M5 Normas y referencias
 - M5.1 Disposiciones legales y normas aplicadas
 - M5.2 Bibliografía
 - M5.3 Métodos, Herramientas, Modelos, Métricas y Prototipos
 - M5.4 Plan de gestión de la calidad aplicado durante la redacción del Proyecto
 - M5.5 Otras referencias
- M6 Definiciones y abreviaturas
- M7 Requisitos iniciales
- M8 Alcance

- M9 Hipótesis y restricciones
- M10 Estudio de alternativas y viabilidad
- M11 Descripción del sistema propuesto
- M12 Análisis de Riesgos
- M13 Organización y gestión del proyecto
- M14 Planificación temporal
- M15 Presupuesto

7.3.1.2 Requisitos del sistema

- Funcionalidad y transformación de entradas en salidas
- Rendimiento. Especificación de cada requisito de rendimiento de forma comprobable y cuantitativamente. Aspectos que deberían reflejarse.
- Atributos de calidad. Usabilidad, eficiencia, factibilidad, mantenibilidad, y portabilidad.
- Restricciones de diseño e implementación. Restricciones impuestas externamente (estándares, departamento de marketing,..), limitaciones de hardware (compiladores, memoria,..), y requisitos de proyecto (características de seguridad).
- Requisitos externos.
- Requisitos de seguridad física.
- Requisitos de seguridad lógica. Requisitos prioritarios. Pueden ser identificados por análisis de riesgos.
- Documentación de usuario. Descripción de requisitos referentes a contenidos.
- Especificación de la interacción hombre-máquina. Descripción de requisitos para una apropiada interacción, sistemas automáticos.

7.3.1.3 ANEXOS

- Documentación de entrada
- Catálogos de los elementos constitutivos del objeto del Proyecto
- Listados
- Información en soportes lógicos, magnéticos, ópticos u otros
- Otros documentos que se juzguen necesarios

7.3.1.4 Presupuesto

- Determinación del coste de cada item del proyecto
- Determinación de los costes acumulados de cada parte del proyecto
- Determinación del coste total del proyecto

7.3.1.5 Estudios con entidad propia

- Aplicación de las Medidas de Seguridad impuestas por el Reglamento de Medidas de Seguridad (Real Decreto 994/1999, de 11 de junio) de desarrollo de la LOPD y por la propia LOPD
- Aplicación de las restricciones impuestas por la LEY 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (LSSI)
- Prevención de Riesgos Laborales
- Impacto Ambiental

8 06. Calidad de los Proyectos de Informática



8.1 Índice

1. Introducción y Definción
2. Modelos de Calidad
 - CMM, SPICE, ISO 9001, ISO 9000-3, PSP
3. Aseguramiento de calidad
4. Gestión de la calidad
5. Conclusiones

8.2 Introducción y Definición

ISO 8402

- La Calidad es el conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas

AENOR

- La calidad es la aptitud de un producto o un servicio para satisfacer las necesidades de los usuarios

UNE-EN ISO 9000:2000 Apartado 3.1.1.

- Grado en el que un conjunto de características inherentes cumple con los requisitos

8.2.1 Adecuación al uso

Implica todas aquellas características de un producto que el usuario reconoce que benefician, y siempre será determinada por el usuario, no por el productor, vendedor o persona que repara el producto

8.2.2 Parámetros que determinan la adecuación al uso

Calidad de diseño

- Es la parte de la calidad que asegura que el producto diseñado satisfaga las necesidades del usuario, y en su diseño se contemple el uso que se le va a dar

Calidad de conformidad

- Se define como el proceso de elaboración de un producto o servicio que tiene que ver con el grado de que el producto o servicio se acerca a las características diseñadas y cumple con las especificaciones de proceso y diseño

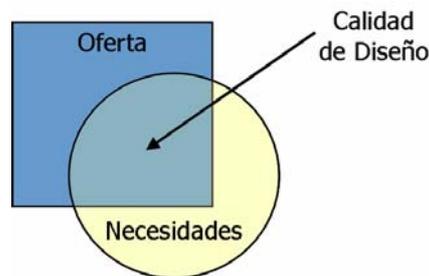
Disponibilidad

- Se define durante el uso del producto y tiene que ver con su desempeño y vida útil

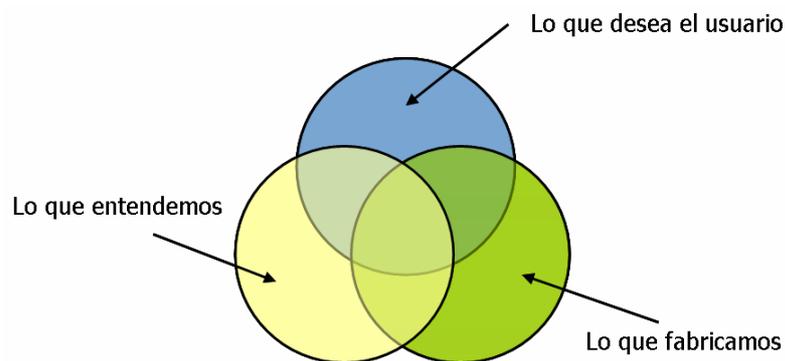
Servicio técnico

- Tiene que ver con el factor humano, debe tener una velocidad de respuesta óptima, competencia y ser íntegro y competente

La Calidad del Diseño se corresponde con la proporción de necesidades del usuario que son cubiertas por el diseño realizado.



Y aún más... ¿Cuántos factores más influyen?



De manera análoga hay otros conceptos importantes

- **Satisfacción percibida:** Proporción de necesidades satisfechas
- **Sobrecalidad:** Características que reúne el producto y que no han sido solicitadas por el usuario

- **Defectos:** Mal funcionamiento del diseño
- **Cero defectos:** Ausencia de defectos (no necesariamente calidad)
- **Oferta percibida:** Oferta prometida, expectativas y decepción

QCD (Quality, Cost, Delivery)

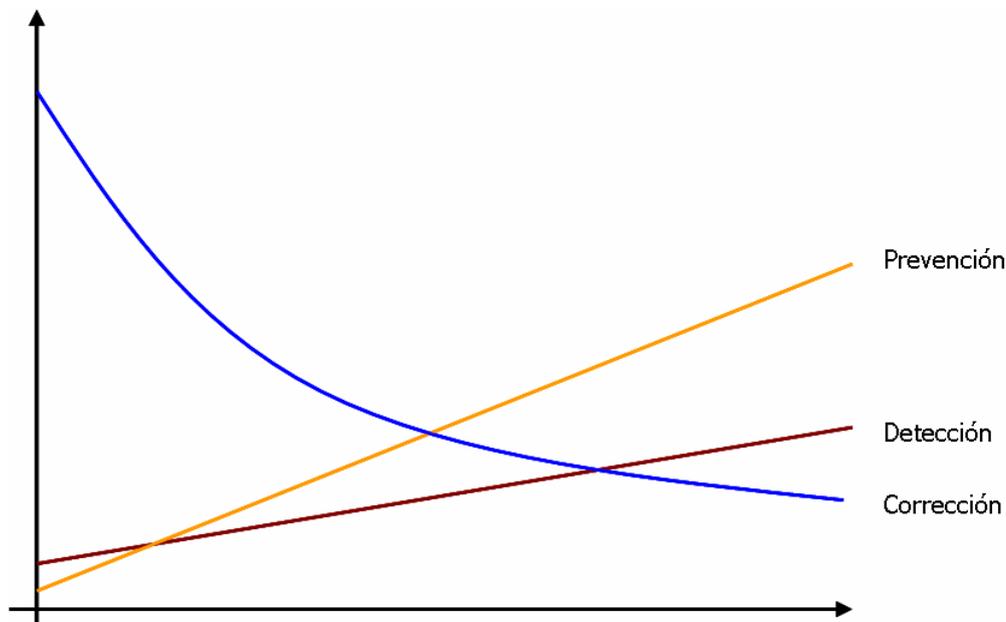
- Factores asociados a la calidad: Calidad, Coste y Plazo

En la ingeniería del software:

- Adecuadas prácticas de ingeniería
 - Prevención de defectos desde el principio
 - Detección y corrección cuanto antes
 - Establecer y eliminar las causas que los producen

En resumen:

- Prevención
 - Planificación de la mejora de la calidad, Auditoría de calidad, Revisión de Proyectos, Formación en calidad
- Detección
 - Calificación de nuevos programas, Control de programas desarrollados, Controles y pruebas en fase de desarrollo, Control final de integración
- Corrección
 - Defectos, Paradas de Actividad, Malfuncionamientos, Rechazos



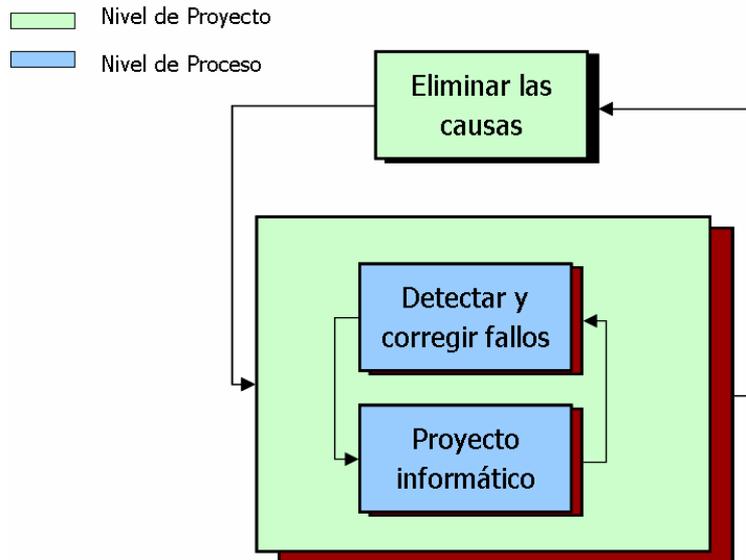
Doble actuación

- Nivel de Proyecto

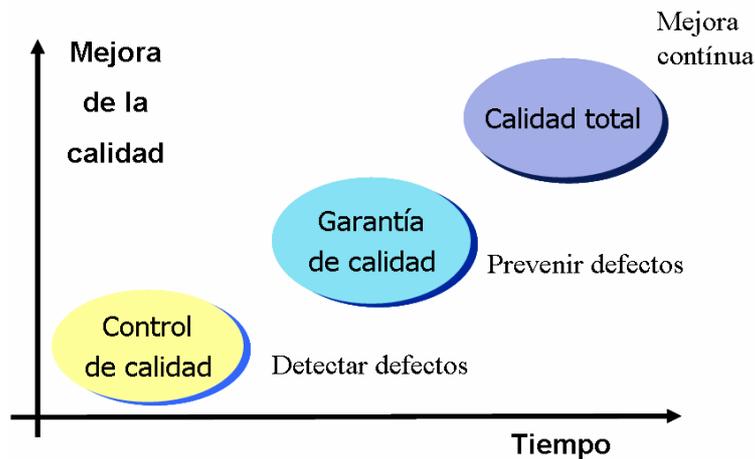
- Calidad gestionada y controlada desde el propio proyecto
 - Validación externa
 - Determinación del uso de estándares
 - Determinación de la adecuación de los estándares al proyecto
- Nivel de Proceso
 - Procedimientos definidos y adecuados
 - Recolección de datos para ampliar la experiencia (Medición)
 - Además es necesaria la comunicación
 - Entre usuario (cliente) y desarrollador
 - Entre los miembros del equipo de desarrollo

Por tanto es imprescindible asegurar la calidad del proceso

- Definición clara de la estructura, funciones y responsabilidades del equipo de proyecto software
- Comunicación fácil entre los componentes del equipo y con el cliente
- Planificación cuidadosa de todos los aspectos relevantes para el proyecto (restricciones de calidad, tiempo y coste)
- Seguimiento continuado de la realización (Definición de puntos de revisión)



Evolución Histórica



8.2.3 Definiciones

Calidad (ISO-8402)

- Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas

Control de calidad (ISO-8402)

- Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio

Garantía de calidad (ISO-8402)

- Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad

Gestión de la calidad (ISO-8402)

- Aspecto de la función de gestión que determina y aplica la política de la calidad, los objetivos y las responsabilidades y que lo realiza con medios tales como la planificación de la calidad, el control de la calidad, la garantía de calidad y la mejora de la calidad
- La gestión de la calidad es responsabilidad de todos los niveles ejecutivos, pero debe estar guiada por la alta dirección. Su realización involucra a todos los miembros de la organización
- En la gestión de la calidad, se tienen en cuenta también criterios de rentabilidad

Sistema de gestión de la calidad (ISO-8402)

- Conjunto de la estructura de la organización, de responsabilidades, procedimientos, procesos y recursos que se establecen para llevar a término la gestión de calidad
- El QS debe tener el volumen y alcance suficiente para conseguir los objetivos de calidad

- El QS de una organización está fundamentalmente previsto para satisfacer las necesidades internas de la organización. Es más amplio que los requerimientos de un cliente concreto que únicamente valora el QS que le interesa (directamente)
- Para finalidades contractuales o vinculantes en la valoración de la calidad, se puede exigir que se ponga de manifiesto la realización de ciertos elementos del QS

8.2.4 Modelos de calidad

Principales iniciativas

- ISO 9000 (9001 y 9000-3)
- Capability Maturity Model (CMM)
- Bootstrap
- SPICE
- Otras como SPR, Malcolm Balridge, etc.

ISO 9000 (9001 y 9000-3)

- Útil en compañías que fabrican software y hardware
- Procesos seriados que incluyan fase de diseño
- En Europa se requiere con bastante frecuencia

Capability Maturity Model (CMM)

- Pertenece al Instituto de Ingeniería del Software
- Es el modelo más empleado y maduro
- Diseñado para valorar sistemas de gran complejidad
- Facilita una visión del proceso de madurez organizacional
- Incluye mecanismos para la mejora continua de procesos

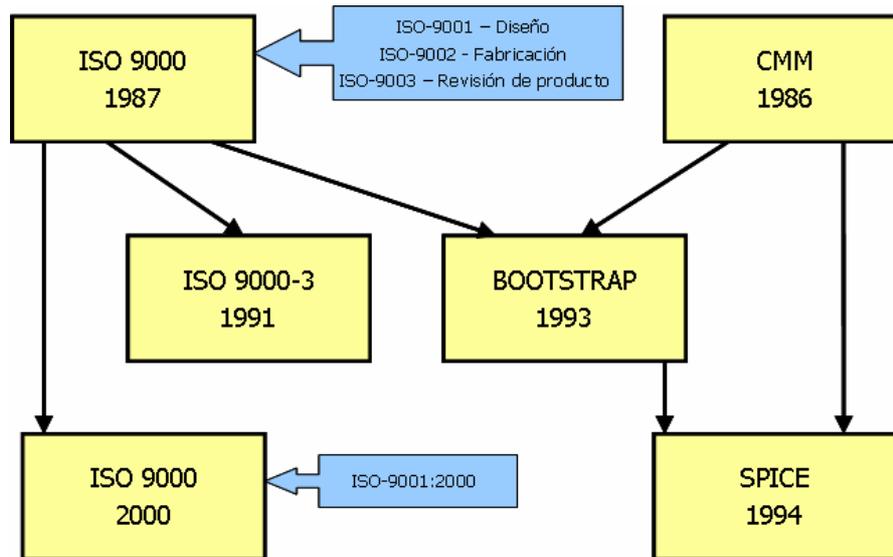
Bootstrap

- Enfocado a pequeñas y medianas empresas
- Valoración de la madurez global
- También se emplea para examinar procesos individuales
- Valora la conveniencia de determinadas tecnologías y el impacto de estas en los modelos de desarrollo

SPICE

- Combina elementos de las tres anteriores
- Estudia el nivel de madurez de los procesos individuales
- Define un marco común de referencia para los demás modelos enunciados

8.2.4.1 ISO 9000-2000



ISO 9000-2000 está formada por las siguientes normas:

- 9000 : 2000 Sistemas de gestión de la calidad - Fundamentos y vocabulario.
- 9000-3 : 1997 Guía para la aplicación de la Norma ISO 9001 al Desarrollo, Suministro y Mantenimiento de Software.
- 9000-4 : 1993 Gestión de la seguridad de funcionamiento.
- 9001 : 2000 Sistemas de gestión de la calidad - Requisitos.
- 9004 : 2000 Sistemas de gestión de la calidad - Directrices para la mejora del desempeño.
- 9004-4 : 1993 Gestión de la Calidad y elementos del Sistema de la Calidad - Guía para la Mejora de la Calidad.
- 10005 : 1996 Guía para Planes de la Calidad.
- 10006 : 1997 Guía para la Gestión de Proyectos.
- 10007 : 1995 Guía para la Gestión de la Configuración.
- 10011-1 : 1992 Lineamientos para las Auditorías de los Sistemas de Calidad. Auditoría.
- 10011-2 : 1992 Lineamientos para las Auditorías de los Sistemas de Calidad. Calificación de auditores.
- 10011-3 : 1992 Lineamientos para las Auditorías de los Sistemas de Calidad. Gestión de los Programas de Auditorías.
- 10012-1 : 1996 Requisitos de aseguramiento de la calidad para equipamiento de medición - Sistema de confirmación metrológica para el equipamiento de medición.
- 10012-2 : 1995 Requisitos de aseguramiento de la calidad para equipamiento de medición - Aseguramiento de la medición.

- 10013 : 2001 Guía para la documentación de los sistemas de gestión de la Calidad.
- 10014 : 1998 Guía para la gestión de los efectos económicos de la Calidad.
- 10015 : 1999 Gestión de la Calidad - Lineamientos para el entrenamiento.
- 10016 : 1998 Registros de Inspección y Ensayo. Guía para la presentación de resultados.
- 10017 : 1999 Guía para la aplicación de Técnicas Estadísticas en la Normas ISO 9000.

8.2.4.1.1 Objetivos de ISO 9000

Proporcionar una guía para la gestión de la calidad: diseño e implantación de sistemas de calidad. (ISO 9000 no normaliza el sistema de gestión de calidad, ya que esto depende del tipo de sector, tamaño de la empresa, organización interna, etc, sino que normaliza las verificaciones que se han de realizar sobre el sistema de calidad)

Describir los requerimientos generales para garantizar la calidad (demostrar la idoneidad del sistema de calidad).

8.2.4.1.2 ISO-9000: La pirámide documental



- El Manual de Calidad
- Los Procedimientos
- Instrucciones Técnicas, Registros y Normativa externa

8.2.4.1.3 ISO 9000: Recomendaciones

Comenzar con ISO 9004-1(1994). Gestión de calidad y elementos del sistema de calidad, para diseñar y implementar el sistema de gestión de calidad.

- ISO 9004 es la directriz para el establecimiento o ampliación de un sistema de calidad. Esta norma amplía partes de ISO 9000 que no se pueden verificar o que una parte contratante no desea dar a conocer, como por ejemplo los gastos asociados a la gestión de calidad.

Una vez implantado el sistema de calidad, utilizar los modelos de garantía de calidad ISO 9001-2000 para demostrar su idoneidad

8.2.4.1.4 Ventajas de ISO 9000

Es un factor competitivo para las empresas

Proporciona confianza a los clientes

Ahorra tiempo y dinero, evitando recertificar la calidad según los estándares locales o particulares de una empresa

Se ha adaptado a más de 90 países e implantado a todo tipo de organizaciones industriales y de servicios, tanto sector privado como público

Proporciona una cierta garantía de que las cosas se hacen tal como se han dicho que se han de hacer

8.2.4.1.5 Desventajas de ISO 9000

Es costoso

Muchas veces se hace por obligación

Es cuestión de tiempo que deje de ser un factor competitivo

Hay diferencias de interpretación de las cláusulas del estándar

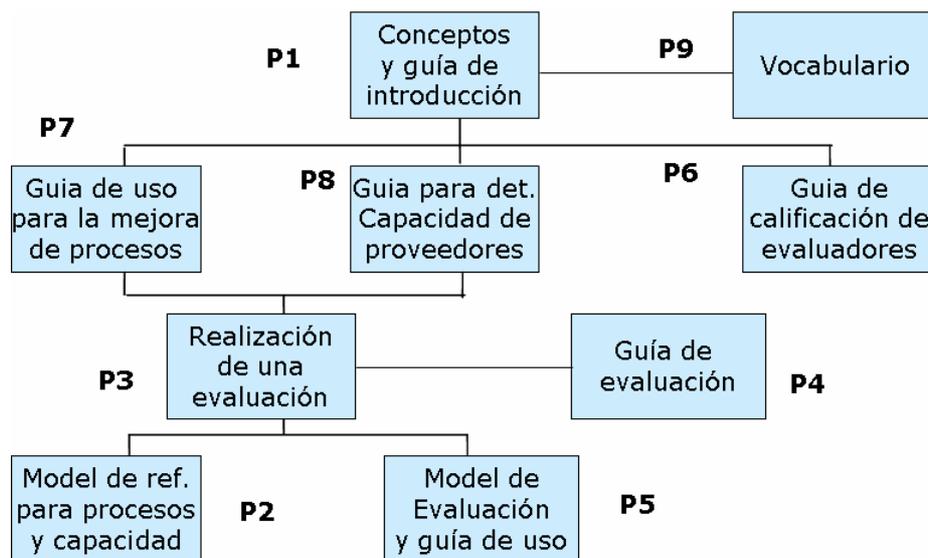
No es indicativa de la calidad de los productos, procesos o servicio

8.2.4.2 SPICE

Software Process Improvement Capability dEtermination

- Evaluación y mejora de procesos software
- Inicio del proyecto 1.993
- Se halla en fase de Informe Técnico
- Es aplicable a cualquier organización o empresa que quiera mejorar la capacidad de cualquiera de sus procesos de software
- Se puede utilizar como herramienta de evaluación del estado de los procesos de software de la empresa
- Es independiente de la organización, modelo del ciclo de vida, metodología y tecnología
- Marco para métodos de evaluación, no un método o modelo en sí
- Abarca:
 - Evaluación de procesos
 - Mejora de procesos
 - Determinación de capacidad
- Alineado con el ISO/IEC 12207

- Intenta proporcionar un marco en el que armonizar los enfoques existentes



8.2.4.2.1 Contenidos de SPICE

El modelo de referencia de SPICE describe los procesos que una organización puede realizar para comprar, suministrar, desarrollar, operar, mantener y soportar el software, así como los atributos que caracterizan la capacidad de estos procesos

Proporciona una base para medir la capacidad de los procesos, en función de grado de consecución de sus atributos.

Tiene dos dimensiones: Procesos y Capacidad

Contiene los procesos que se han de evaluar. Se corresponden con los procesos del ciclo de vida del software, definidos al estándar ISO 12207:1995

Se agrupan en categorías, en función del tipo de actividad al cual se aplican:

- CUS: Cliente-Proveedor
- ENG: Ingeniería
- SUP: Soporte
- MAN: Gestión
- ORG: Organización

8.2.4.2.1.1 SPICE (Dimensión Procesos) Categoría CUS

La categoría CUS está formada por procesos que afecta directamente al cliente, soportan el desarrollo y la transición del software al cliente y permiten la correcta operación y uso del producto y/o servicio software.

- CUS.1 Adquisición de productos software y/o servicios
- CUS.2 Establecimiento de contratos

- CUS.3 Identificar las necesidades del cliente
- CUS.4 Realizar auditorías y revisiones conjuntas
- CUS.5 Entrega e instalación del software
- CUS.6 Mantenimiento del software
- CUS.7 Proporcionar servicio al cliente
- CUS.8 Valorar la satisfacción del cliente

8.2.4.2.1.2 SPICE (Dimensión Procesos) Categoría ENG

La categoría ENG está formada por procesos que directamente especifican, implementan o mantienen el producto software, su relación con el sistema y su documentación.

- ENG.1 Análisis y diseño de requerimientos del sistema
- ENG.2 Análisis de requerimientos del software
- ENG.3 Diseño del software
- ENG.4 Construcción del software
- ENG.5 Integración y pruebas del software
- ENG.6 Integración y pruebas del sistema
- ENG.7 Mantenimiento del software y del sistema

8.2.4.2.1.3 SPICE (Dimensión Procesos) Categoría SUP

Está formada por procesos que dan soporte a cualquiera del resto de procesos (incluidos los SUP), en distintos puntos del ciclo de vida del software.

- SUP.1 Documentación
- SUP.2 Gestión de la configuración del software
- SUP.3 Garantía de calidad
- SUP.4 Resolución de problemas
- SUP.5 Realizar revisiones conjuntas

8.2.4.2.1.4 SPICE (Dimensión Procesos) Categoría MAN

Formada por procesos utilizados en la gestión de cualquier tipo de proyecto o proceso en el ciclo de vida del software

- MAN.1 Gestionar el proceso
- MAN.2 Gestionar el proyecto
- MAN.3 Gestionar la calidad
- MAN.4 Gestionar los riesgos

8.2.4.2.1.5 *SPICE (Dimensión Procesos) Categoría ORG*

Formada por procesos que establecen los objetivos de negocio de la organización

- ORG.1 Alineamiento de la organización
- ORG.2 Establecimiento del proceso
- ORG.3 Evaluación del proceso
- ORG.4 Mejora del proceso
- ORG.5 Gestión de recursos humanos
- ORG.6 Infraestructura
- ORG.7 Reutilización

8.2.4.2.1.6 *SPICE (Dimensión Capacidad)*

Define una escala de medida para determinar la capacidad de cualquier proceso

Consta de seis niveles de capacidad y nueve atributos de procesos

- 0 Incompleto
- 1 Realizado (Realización del proceso)
- 2 Gestionado (Gestión de realización, Gestión de productos)
- 3 Establecido (Definición de procesos, Recursos de procesos)
- 4 Predecible (Medición de procesos, Control de procesos)
- 5 En optimización (Cambio de procesos, Mejora continua)

8.2.4.2.1.7 *SPICE (Dimensión Capacidad - Prácticas)*

Prácticas Base

- Cada proceso tiene un conjunto de prácticas base asociadas
- Las prácticas base describen las actividades esenciales de un proceso específico
- La realización de las prácticas base indica el grado de alcance de la finalidad del proceso

Prácticas de Gestión

- Cada atributo de proceso tiene un conjunto de prácticas de gestión asociadas
- Las prácticas de gestión son las que implementan o institucionalizan un proceso de una manera general
- La realización de las prácticas de gestión indica la consecución del atributo en esa instancia del proceso

8.2.4.2.1.8 *SPICE (Evaluación de atributos)*

Los atributos de un proceso se evalúan con N (Not), P (Partially), L (Largely) y F (Fully), siendo:

- N No alcanzado (0% a 15%)
 - Poca o ninguna evidencia de la consecución del atributo
- P Parcialmente alcanzado (16% a 50%)
 - Evidencia de un enfoque sistemático y de la consecución del atributo. aunque algunos aspectos de la consecución pueden ser impredecibles
- L Ampliamente alcanzado (51% a 85%)
 - Evidencia de un enfoque sistemático y de una consecución significativa del atributo. La realización del proceso puede variar en algunas áreas
- F Totalmente alcanzado (86% a 100%)
 - Evidencia de un enfoque completo y sistemático y de la consecución plena del atributo

8.2.4.3 CMM

CMM es una aplicación de sentido común de los conceptos de gestión de procesos y mejora de la calidad al desarrollo y mantenimiento del software (Mark C. Paulk)

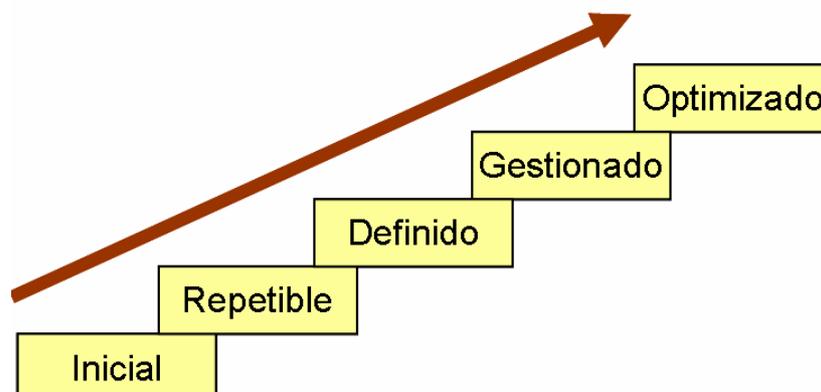
Estudia los procesos de desarrollo de software de una organización y produce una evaluación de la madurez de la organización según una escala de cinco niveles

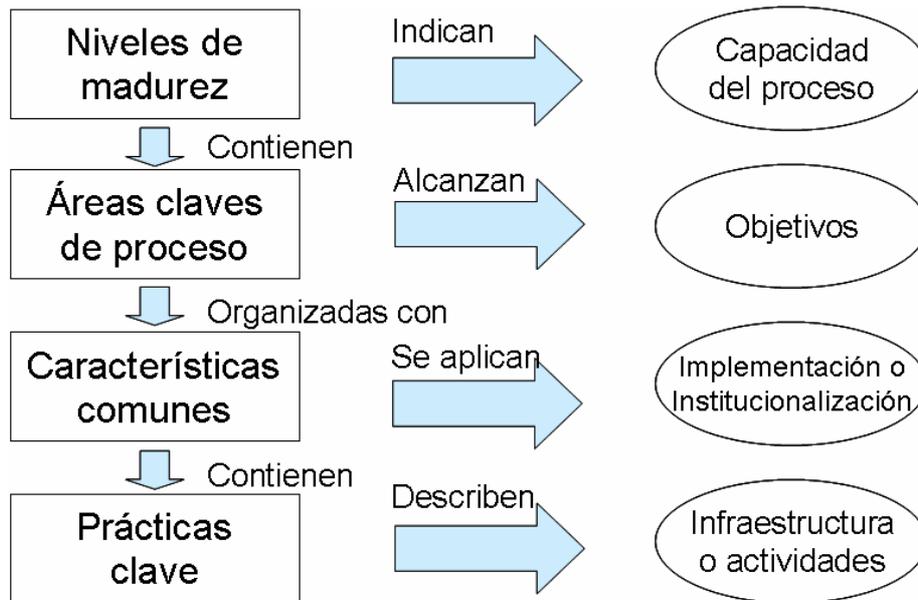
La madurez de un proceso es un indicador de la capacidad para construir un software de calidad

Es un modelo para la mejora de las organizaciones

Obliga a una revisión constante

Niveles CMM:





8.2.4.3.1 Certificación CMM

Proceso habitual de certificación

- Motivación
- Selección de la norma aplicable
- Subcontratación a empresa externa
- Auditoría de certificación
- Informe de acciones correctoras
- Certificado
- Imposición de seguimiento
- Incumplimiento

8.3 Aseguramiento de la Calidad

El aseguramiento de la calidad, se puede definir como el esfuerzo total para plantear , organizar, dirigir y controlar la calidad en un sistema de producción con el objetivo de dar al cliente productos con la calidad adecuada

Es simplemente asegurar que la calidad sea lo que debe ser

- Planteamiento de
- Sistemas de Gestión
- Métricas
- Procedimientos
- Organización
- etc.

8.4 Gestión de la Calidad

Elaboración y seguimiento de los Planes de Gestión que aseguren la calidad

- Gestión del alcance del proyecto
- Gestión de plazos del proyecto
- Gestión de costes del proyecto
- Gestión de la calidad del proyecto
- Gestión de recursos humanos del proyecto
- Gestión de comunicaciones del proyecto
- Gestión de riesgos del proyecto
- Gestión de adquisiciones del proyecto
- Gestión de la configuración

8.5 Conclusiones

Se basa siempre en la combinación de dos disciplinas:

- La Técnica de Informática
- La propia de gestión de la calidad

La calidad ayuda a mejorar los sistemas de producción

La calidad tiene un coste de implantación

- Coste de la calidad
- Costes de la “no calidad”

Para implantar un sistema de calidad deben estar implicados todos los estamentos (desde la dirección a los operarios)

9 07. Gestión de Riesgos



9.1 Índice

1. Introducción y Definición de Riesgo
2. Identificación
3. Gestión de Riesgos
4. Factores de Riesgo
5. Metodología de gestión de riesgos

9.2 Introducción y definición

Un riesgo es cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro

El riesgo se haya asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas

El riesgo, por tanto, acompaña a todo cambio y está presente en cada decisión

El riesgo implica elección e incertidumbre

9.3 Manejo de Riesgos

Manejo de riesgos consiste en la identificación de riesgos y la escritura de planes para minimizar el efecto de estos en el proyecto

Un riesgo se relaciona con la probabilidad de que ocurra alguna circunstancia adversa al proyecto

Hay tres clases principales de riesgos:

- Los **riesgos de un proyecto** afectan a la planificación o a los recursos
- Los **riesgos del producto** afectan a la calidad o al desempeño del software por desarrollarse
- Los **riesgos del negocio** son aquellos que afectan a la organización que desarrolla el software

9.4 Algunos ejemplos de riesgos

RIESGO	TIPO	DESCRIPCIÓN
Cambio de equipo de trabajo	Proyecto	Las personas con más experiencia en el equipo dejan su trabajo antes de la terminación
Cambio de Gestión	Proyecto	Hay un cambio de la dirección del proyecto con otras prioridades diferentes
Indisponibilidad del Hardware	Proyecto	El hardware necesario para el proyecto no se entrega a tiempo
Cambio de los requerimientos	Proyecto y Producto	Hay un gran número de cambios en las especificaciones iniciales que no estaba previsto
Retrasos en las especificaciones	Proyecto y Producto	Algunas especificaciones de los interfaces principales entre módulos no se terminan de acuerdo a la planificación
Subestimación del tamaño	Proyecto y Producto	El tamaño del sistema está subestimado
Insuficiencia de las herramientas	Producto	Las herramientas elegidas para el desarrollo son insuficientes y no tienen las facilidades con las que se contaba en principio
Cambios de la tecnología	Negocio	La tecnología en la que se ha basado el proyecto es sustituida por una tecnología más moderna
Competencia de productos	Negocio	Un producto de la competencia es puesto en el mercado antes de terminar el proyecto

9.5 El proceso de manejo de riesgos

Identificación de riesgos

- Identifica riesgos en el proyecto, en el producto y en el negocio

Análisis de Riesgos

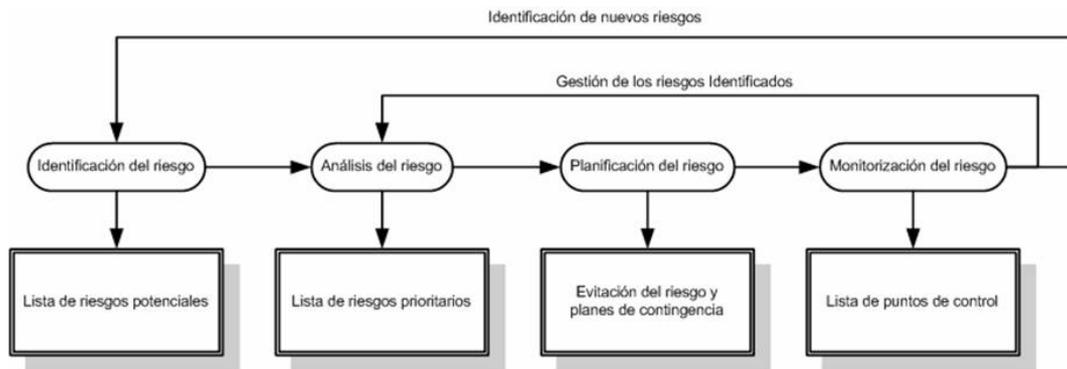
- Cálculo de la posibilidad de que ocurran estos riesgos y de sus consecuencias

Planeación de Riesgos

- Trazar planes para evitar o minimizar el efecto de los riesgos

Monitorización de Riesgos

- Monitorizar los riesgos durante el proyecto



9.6 Identificación de Riesgos

- Riesgos en la tecnología
- Riesgos humanos
- Riesgos organizacionales
- Riesgos en las herramientas
- Riesgos en los Requerimientos
- Riesgos de estimación
- Riesgos políticos
- Riesgos económicos

RIESGO	TIPO	CLASE
Cambio de equipo de trabajo	Proyecto	Riesgo humanos
Cambio de Gestión	Proyecto	Riesgo organizacional
Indisponibilidad del Hardware	Proyecto	Riesgo de las herramientas
Cambio de los requerimientos	Proyecto y Producto	Riesgo de los requerimientos
Retrasos en las especificaciones	Proyecto y Producto	Riesgo de estimación
Subestimación del tamaño	Proyecto y Producto	Riesgo de estimación
Insuficiencia de las herramientas	Producto	Riesgo de las herramientas
Cambios de la tecnología	Negocio	Riesgo en la tecnología
Competencia de productos	Negocio	Riesgo político Riesgo económico

9.7 Análisis de riesgos

Determina la probabilidad y la seriedad de cada riesgo

Las probabilidades pueden variar entre muy alta, alta, moderada, baja o muy baja

Los efectos de los riesgos pueden ser:

- Catastróficos
- Serios
- Tolerables
- Insignificantes

RIESGO	PROBABILIDAD	EFFECTOS
Cambio de equipo de trabajo	ALTA	TOLERABLE
Cambio de Gestión	BAJA	SERIO
Indisponibilidad del Hardware	MODERADA	SERIO
Cambio de los requerimientos	ALTA	TOLERABLE
Retrasos en las especificaciones	MODERADA	SERIO
Subestimación del tamaño	ALTA	TOLERABLE
Insuficiencia de las herramientas	BAJA	SERIO
Cambios de la tecnología	MODERADA	CATASTRÓFICO
Competencia de productos	BAJA	SERIO

9.8 Planeación de los riesgos

Considera cada riesgo y desarrolla una estrategia para manejarlo

- Estrategias de evitación
 - Se trata de minimizar la probabilidad de que el riesgo se presente
- Estrategias de minimización
 - Se trata de reducir el impacto del riesgo en el producto o en el proyecto
- Planes de contingencia
 - Si el riesgo se presenta, el plan de contingencia se encargará de tratar este riesgo
 - Palia los efectos negativos del riesgo pero éste ya se ha producido

9.9 Estrategias de manejo de riesgos

RIESGO	ESTRATEGIA
Problemas financieros organizacionales	Preparación de un documento para la dirección de la empresa en la que se muestre la importante contribución del proyecto a los objetivos del negocio

RIESGO	ESTRATEGIA
Retrasos en las entregas	Comunicación al cliente de posibles dificultades y de la posibilidad de que se retrase en proyecto. Investigación de los componentes que provocan el retraso
Enfermedad de algún componente	Reorganización del trabajo de manera que no haya solapamientos ni huecos en los trabajos de manera que todos puedan saber sus responsabilidades y conozcan el trabajo de los demás
Componentes defectuosos	Reemplazar los componentes defectuosos con otros comprados (si existen) y que tengan una efectividad reconocida
Cambios en los requerimientos	Comprobar la trazabilidad de los requerimientos modificados o eliminados para detectar el impacto en el proyecto ya desarrollado Introducir los nuevos requerimientos y estudiar la trazabilidad de su impacto en el software desarrollado Rediseñar los componentes afectados
Reestructuración organizacional	Preparación de un documento que demuestre a la dirección la importancia del proyecto para los objetivos empresariales
Fallos en las facilidades de la BD	Investigar la posibilidad de comprar una nueva base de datos con mejores características
Subestimación del tiempo de desarrollo	Investigar la posibilidad de comprar componentes y de usar generadores de código

9.10 Estrategias frente al riesgo

Estrategias reactivas

- Las estrategias reactivas son aquellas que se dan cuando se deja que los riesgos produzcan sus efectos (en este momento ya no es un riesgo, es una realidad) y entonces se actúa en consecuencia.
- En estas condiciones lo único que cabe es tomar medidas correctoras (apagar incendios), lo que produce muchos tiempos perdidos, retrasos en el proyecto, gabinetes de crisis, etc.
- Las estrategias reactivas no son aconsejables porque ponen en grave peligro el proyecto.

Estrategias proactivas

- Las estrategias proactivas pasan por la evaluación previa y sistemática de todos los riesgos inherentes al proyecto, evaluando sus consecuencias.
- Esto produce la creación de un Plan de Gestión de Riesgos, con sus planes de evitación, minimización de consecuencias, planes de contingencia, etc.
- En estas condiciones el objetivo es la evasión del riesgo, con menor tiempo de reacción frente a los efectos negativos y una mejor gestión del proyecto en su conjunto: menor tiempo y menor coste.

9.11 Monitorización de riesgos

Determina regularmente cada riesgo identificado y decide si es probable o no que se presente.

Determina si los efectos de que produciría el riesgo, han cambiado.

Cada riesgo clave debe discutirse en las reuniones de avance del proyecto.

9.12 Factores de riesgo

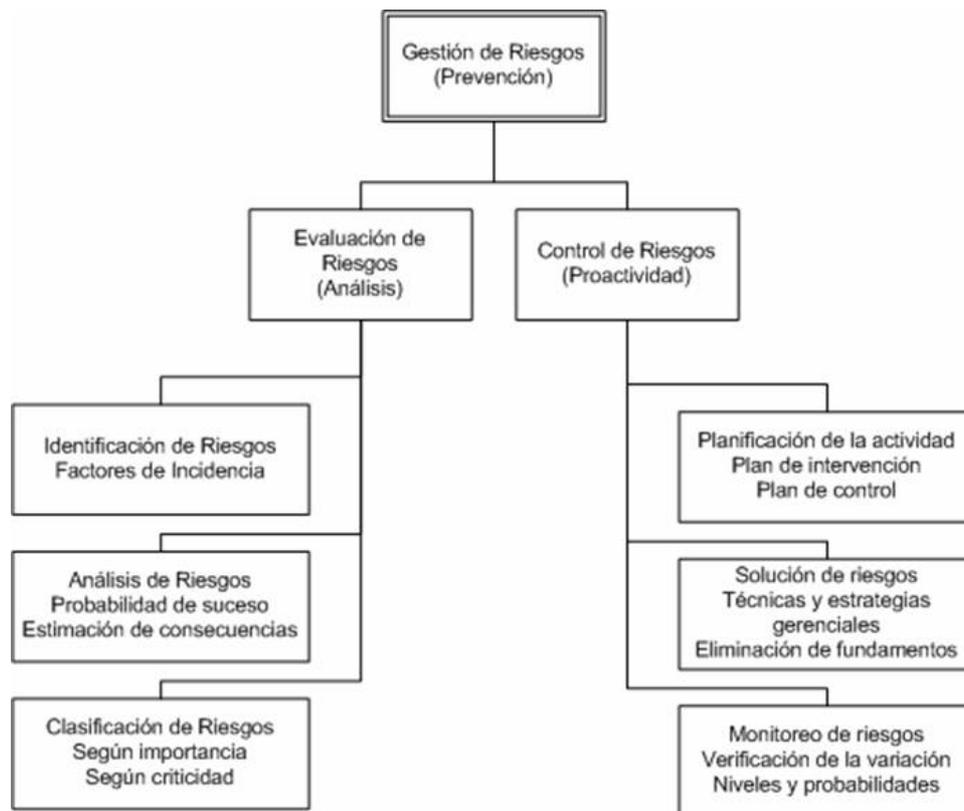
Los factores de riesgo son los elementos que determinan el riesgo.

Su medición y monitorización determina el grado de acercamiento al riesgo.

Por cada riesgo es necesario identificar los factores que lo determinan.

Un riesgo puede ser, a su vez, factor de riesgo en otros riesgos.

9.13 Gestión de Riesgos



9.14 Primeros pasos

EL mayor de todos los riesgos:

CREENCIA DE QUE TODOS LOS RIESGOS CONFORMAN UN CONJUNTO CERRADO Y ESTÁN INCLUIDOS EN EL PRIMER PLAN DE RIESGOS REALIZADO

Identificación y almacenamiento de riesgos

Planes de Riesgos

Identificación permanente

9.15 Identificación de un riesgo

Nombre del riesgo. Un nombre que lo identifique de manera rápida

Descripción del riesgo. Nos permite tener una descripción más completa del alcance del riesgo y de los perjuicios potenciales para el proyecto

Escala afectada por el riesgo, esto es, si es un riesgo que afecte a una persona, a un grupo de trabajo, a una parte del proyecto, al conjunto, etc.

Excepciones que afectan al riesgo, situaciones donde el riesgo es inevitable.

Variaciones del mismo riesgo que se puedan identificar para evitar tener el mismo riesgo con distintos nombres o tener riesgos identificados que se solapen

Personas o entidades implicadas en el riesgo y que en caso de cualquier cambio o decisión deben ser consultadas previamente

Evitación del riesgo

- **Estrategia general** descripción general de la estrategia prevista para la evitación del riesgo
- **Medidas correctoras** o soluciones que se deben tomar con respecto al riesgo para evitar o mitigar el riesgo, quien debe tomarlas y en qué momento

Monitorización

- **Indicadores** de que se pueda producir el riesgo. La evaluación de éstos indicadores definirá cuán cerca estamos en cada momento de que se produzca el riesgo
- **Modos de evaluación de los indicadores**, previsión de la frecuencia con que debe reevaluarse el riesgo, personas o entidades implicadas en la reevaluación, etc.

Gestión

- **Plan de contingencias** donde se especifiquen las acciones correctoras en caso de fallo en la gestión del riesgo, las personas implicadas y las que tienen la responsabilidad de la toma de decisiones
- **Consideraciones especiales** que concurran en la gestión del riesgo

9.16 Plan de Gestión de Riesgos

El objetivo de este plan debe ser triple, de un lado la identificación de riesgos (de la que este informe en una parte), de otro la monitorización de los factores que afectan al

riesgo y en tercer lugar la previsión de planes de contingencia si la evitación del riesgo ha resultado imposible

Identificación y Evitación del riesgo

- Se deben definir las estrategias necesarias (las soluciones) para que el riesgo no se produzca o, en caso de que se produzca, tomar las medidas encaminadas a que se minimicen sus efectos perniciosos

Monitorización del riesgo

- Para cada riesgo se deben definir indicadores que influyan en la posibilidad de que el riesgo se produzca
- Estos deben ser comprobados con cierta frecuencia para obtener referencias de qué riesgos son más probables y cuan cerca estamos de un riesgo se convierta en una problema
- También deben revisarse periódicamente la efectividad de las acciones previstas para evitar el riesgo y los planes de contingencia en caso de que el riesgo ya fuera inevitable
 - Hay que pensar que en el transcurso del proyecto las condiciones pueden variar y las soluciones pueden quedarse obsoletas

Gestión del riesgo y plan de contingencia

- Se asume que ha fallado la evitación y la monitorización de los riesgos y su efecto ya es inevitable
- Es necesario para estos casos tener estrategias definidas que permitan unos tiempos de respuesta cortos y poder hacer, de este modo, que los riesgos se minimicen

9.16.1 Formato de Plan de Riesgos

Introducción

- Ámbito y propósito del documento
- Descripción de los riesgos principales

Responsabilidades

- Gestores de riesgos
- Personal técnico asignado a la gestión de riesgos

Tabla de evaluación de riesgos

- Identificación de todos los riesgos considerados
- Factores que influyen en la probabilidad (Monitorización)
- Impacto previsto

Descripción de los riesgos

- (Contenidos del apartado 9.15)

Planificación

- Planes de gestión, programación de reuniones, etc.
- Planes de contingencia generales, constitución de los gabinetes de crisis, etc.

9.17 Identificación permanente

Durante todo el proyecto los riesgos evolucionan

- Aparecen nuevos riesgos
- Desaparecen algunos de los identificados anteriormente
- Cambia la importancia o la gravedad
- Aparecen y desaparecen nuevos factores

En definitiva los riesgos son un elemento vivo y la monitorización debe prever este nuevo “*riesgo*”

9.18 Metodología

Se va a comenzar un proyecto, en esta fase se está estudiando la viabilidad del proyecto y se determina la “Arquitectura Software” del sistema.

Es necesario también identificar los riesgos que desde este momento afectan al proyecto.

- Estudio del Proyecto e Identificación de los Riesgos Generales existentes
 - Riesgos Económicos, Tecnológicos, Humanos, otros Riesgos identificables, Uso de experiencias anteriores, etc.
- Brainstormings Generales
 - De muy alto nivel, identificando principalmente riesgos muy generales
- Elaboración del Plan de Riesgos inicial
 - Acompañará a la Documentación del Proyecto según la norma

*En este momento el proyecto debe estar en fase inicial.
Es necesario completar el Plan de Riesgos.*

Por cada riesgo identificar los factores y determinar su influencia en el riesgo.

- Brainstormings:
 - **Alto nivel.**- Normalmente reuniendo a ejecutivos del proyecto, identificando fricciones y riesgos propios de la administración y gestión del proyecto y de las propias políticas organizativas de las empresas implicadas en el proyecto

- Medio nivel.- Reuniones con la Dirección Técnica del Proyecto, identificando riesgos inherentes a las tecnologías, a la organización y comunicaciones entre grupos, a las comunicaciones con proveedores, etc.
- **Bajo nivel.**- Reunión con los analistas, diseñadores, programadores, etc. identificando riesgos relacionado con la falta de medios, problemas organizativos, falta de dirección, etc.
- **Verticales.**- En los que se reunirán personas de los distintos niveles implicados para identificar riesgos de comunicación vertical
- **Uso de experiencias anteriores y de bibliografía** al efecto para identificar algunos riesgos ocultos, pero que se repiten como antipatronos en diferentes proyectos

En este momento el proyecto debe estar en fase de lanzamiento y se comienzan a lanzar las tareas iniciales. A partir de aquí es necesaria una identificación on-line de nuevos riesgos que puedan ser detectados en cualquier nivel de la jerarquía del proyecto

- Identificación continua o permanente de riesgos.
 - Durante el desarrollo del proyecto hay que prever la aparición de más riesgos
 - Brainstormings regulares
 - Captura de riesgos identificados por los propios participantes en el proyecto
- De manera permanente el Sistema de Gestión de Riesgos será alimentado con la información de los factores identificados como desencadenantes del riesgo y recalculará éstos manteniendo informados a los responsables de la gestión

9.19 Conclusiones

Los riesgos son un factor más a tener en cuenta en la gestión de proyectos

Están determinados por todos los elementos (factores) que afectan al proyecto y que pueden provocar situaciones adversas

Para gestionar bien los riesgos hay que establecer un Plan de Riesgos

Los riesgos son vivos y se debe definir el modo de adaptar permanentemente el Plan de riesgos

Para gestionar los riesgos es importante definir una metodología de trabajo y contar con herramientas de gestión de riesgos

10 08. Planificación Temporal



10.1 Índice

1. Introducción
2. Planificación temporal
3. Métodos de Planificación
4. PERT
5. GANTT
6. Ejemplo

10.2 Introducción

La planificación temporal de un proyecto consiste en considerar los recursos y estudiar las restricciones de un proyecto para distribuir el proyecto en tareas que permitan terminarlo en un plazo convenido.

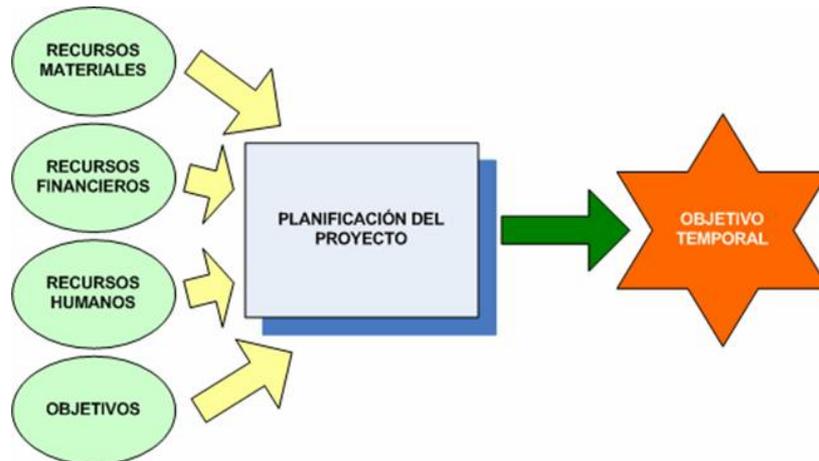
Los plazos pueden ser:

- Desde una fecha inicial
- Antes de una fecha concreta

10.3 Planificación Temporal

Cualquier cambio en las variables de entrada (recursos(), objetivos(), etc.) repercute en la planificación.

Si se cambia el plazo final del proyecto también cambia la planificación y en secuencia puede haber que cambiar algunas variables de entrada.



Aumento de personal

- No implica una reducción de tiempos proporcional
 - Problemas de comunicación
 - Formación
- Tiene un impacto directo sobre los costes

Reducción de la productividad

- Si un programador hace 5000 LDC/año
- Si el coste de comunicación es de 250 LDC/año por cada camino posible
- 4 programadores (6 vías) $4 \times 5000 - 500 \times 6 = 17.000$
- 8 programadores (28 vías) $8 \times 5000 - 500 \times 28 = 26.000$

Aunque es una simplificación indica claramente que el aumento de personal no lleva el mismo aumento de productividad y en el absurdo puede disminuirla

- 1 prog. 1 año... 365 programadores?
 - La productividad es de -31.390.000 LDC / año

Paralelismo

- Las tareas muchas veces pueden hacerse en paralelo
 - Independencia de recursos

Deslizamiento

- Muchas tareas no son críticas y admiten deslizamientos
 - Retrasos de finalización o de comienzo

Milestones

- Marcas que se ponen en el proyecto para hacer el seguimiento

Camino crítico

- Secuencia de tareas con dependencias de inicio y fin que condicionan el logro de un milestone o del final del proyecto

Tarea crítica

- Tarea que está en el camino crítico

10.4 Métodos de Planificación

Sistemas que permiten planificar los recursos de un proyecto para lograr los objetivos

PERT

- Definición en base a una red (grafo dirigido) de dependencias de tareas que llevan al final del proyecto

GANTT

- Definición en forma de barras que dan un aspecto visual temporal del Plan de Proyecto

10.5 PERT

Los proyectos complejos requieren de una serie de actividades algunas de las cuales pueden hacerse en secuencia y otras en paralelo

Historia

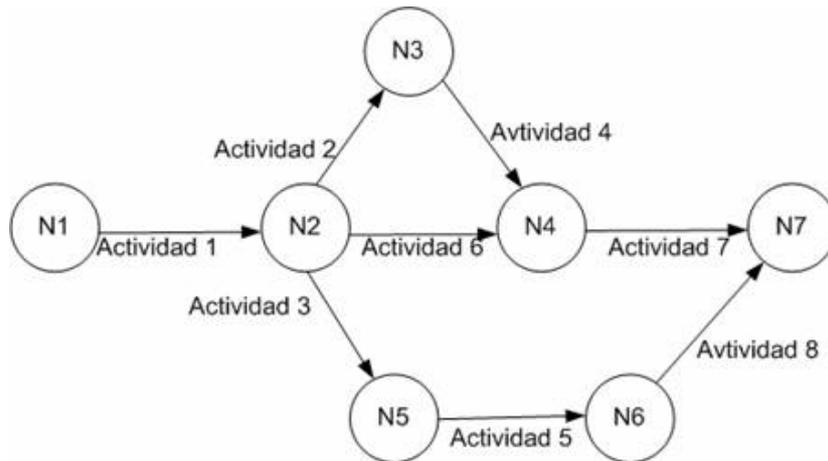
- 1957 CPM (Critical Path Method)
 - Usa un tiempo específico y estático para cada tarea
 - Es fácil de entender
 - No modela bien las variaciones de tiempos
- 1959 PERT (Program Evaluation and Review technique)
 - Desarrollado para el proyecto SOLARIS de la Marina de USA
 - Usa las holguras para buscar el tiempo mínimo para realizar un proyecto

El grafo PERT está formado por actividades y nodos.

Las actividades son tareas que permiten avanzar de un nodo a otro en el grafo.

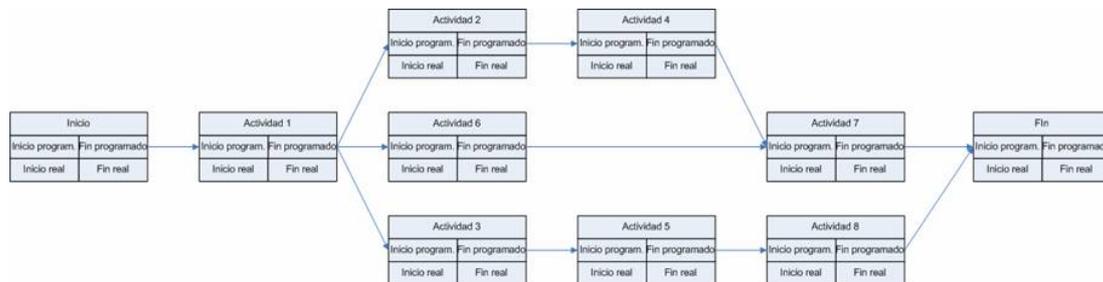
Para que una tarea pueda comenzar deben estar terminadas todas las predecesoras (las que terminan en el nodo).

En las representaciones actuales la tarea se representa como un nodo y los arcos representan prioridades.



El Diagrama de PERT puede tener el aspecto del grafo de arriba donde las tareas las representan los arcos

También pueden representarse las tareas como nodos y los arcos simplemente especifican precedencia



En la representación como arcos(), los nodos representan hitos

10.5.1 Pasos para construir un PERT

- Identificar las actividades y los hitos
- Determinar la secuencia correcta de las actividades
- Construir la red
- Estimar el tiempo requerido por cada actividad
- Determinar el camino crítico
- Actualizar el PERT a medida que el proyecto avanza

10.5.2 Ejemplo PERT

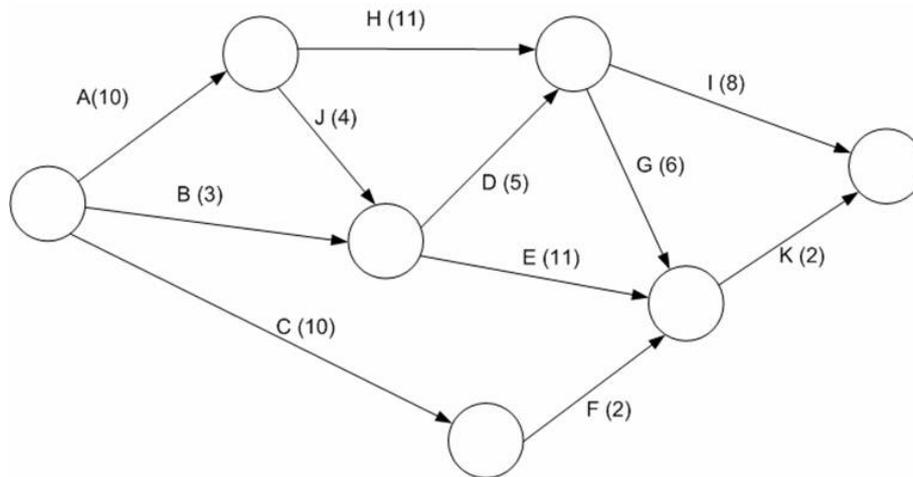
UN proyecto está compuesto por 11 actividades:

- A (10), B(3), C(10), D(5), E(11), F(2), G(6), H(11), I(8), J(4), K (2)

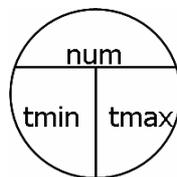
Tabla de precedencias:

Tareas Precedentes	-	-	-	B, J	B, J	C	D, H	A	D, H	A	E, F, G
Tareas	A	B	C	D	E	F	G	H	I	J	K
Tareas Siguientes	J, H	D, E	F	G, I	K	K	K	I, G	-	D, E	-

Grafo:

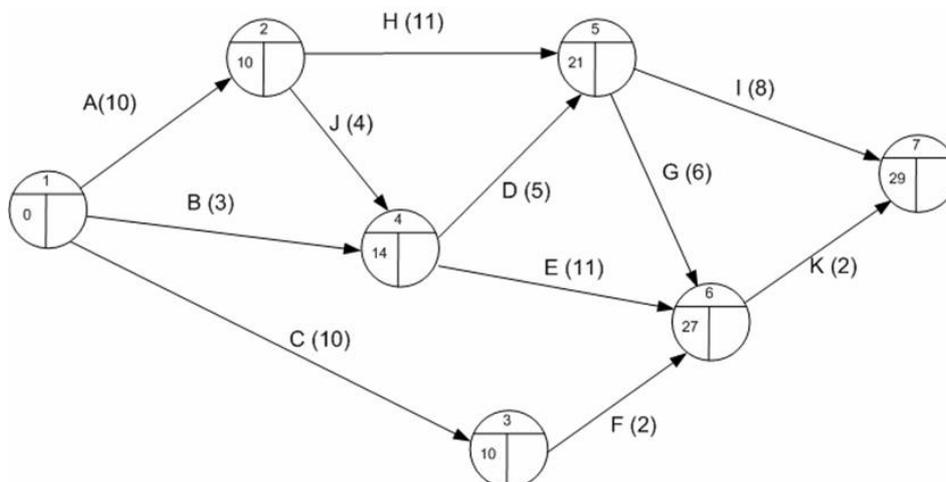


Numeración y determinación de tiempos mínimos para alcanzar el hito.



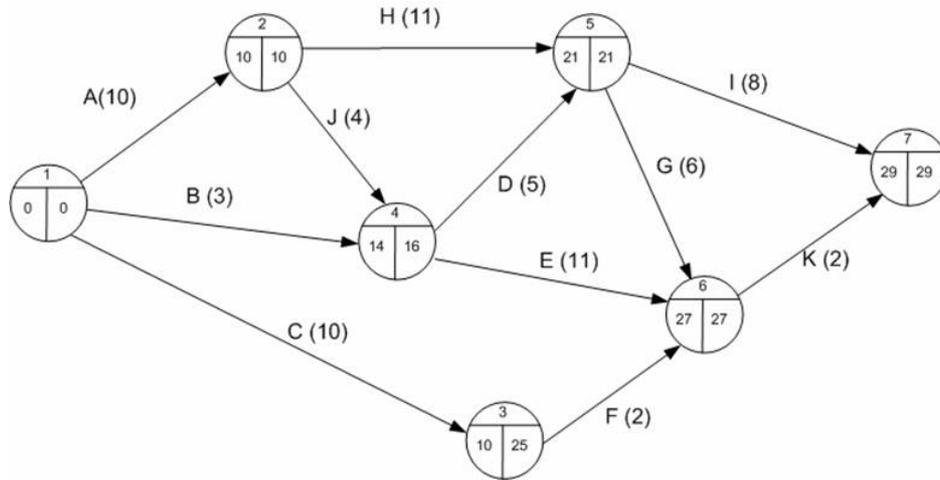
El tiempo mínimo para llegar a un hito es máximo de los tiempos de los caminos para llegar a ese hito.

Cálculo de los tiempos mínimos para llegar a un hito.



Cálculo de los tiempos máximos.

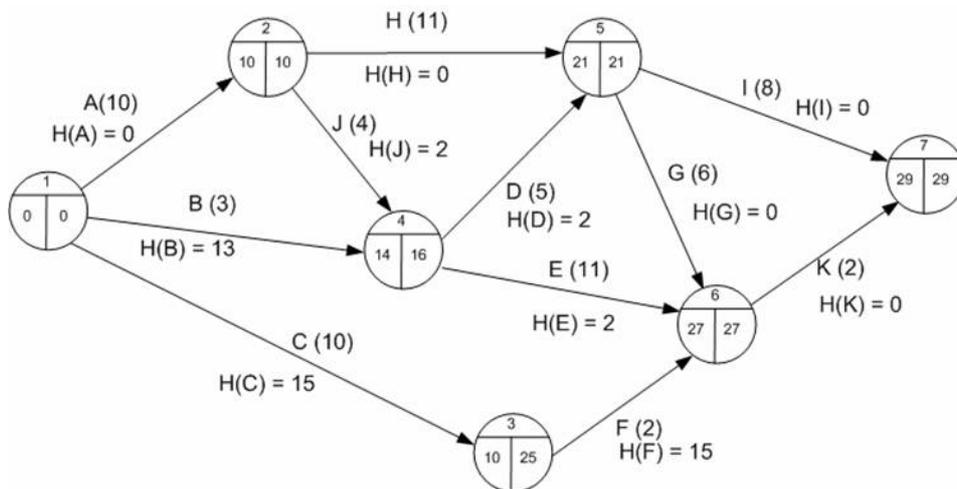
A partir del final del grafo se calculan los mínimos de las diferencias entre el nodo y su antecesor.



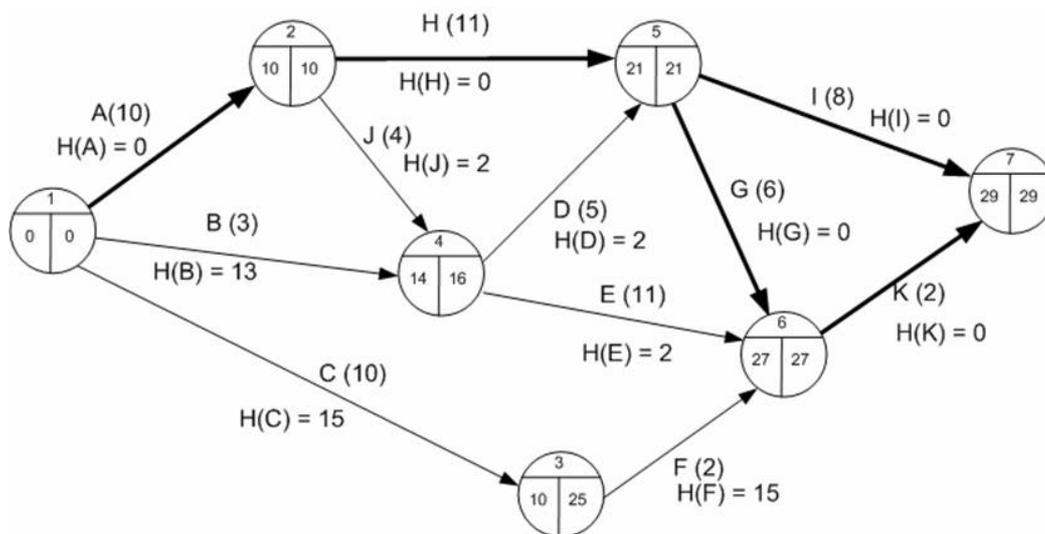
Cálculo de las holguras

Son las diferencias entre la duración de la tarea y el tiempo máximo previsto para alcanzar el hito

$$H(i..j) = T_j - t_i - d_{ij}$$



Camino crítico



10.6 GANTT

Los cronogramas de barras o “gráficos de Gantt” fueron concebidos por el ingeniero norteamericano Henry L. Gantt

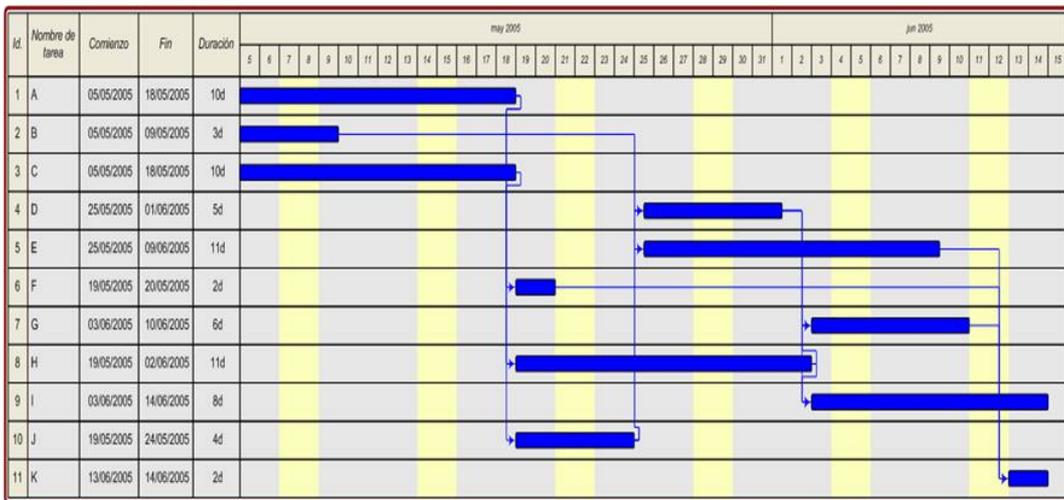
Gantt procura resolver el problema de la programación de actividades, es decir, su distribución conforme a un calendario, de manera tal que se pudiese visualizar el periodo de duración de cada actividad, sus fechas de iniciación y terminación e igualmente el tiempo total requerido para la ejecución de un trabajo

El instrumento que desarrolló permite también que se siga el curso de cada actividad, al proporcionar información del porcentaje ejecutado de cada una de ellas, así como el grado de adelanto o atraso con respecto al plazo previsto

Este gráfico consiste simplemente en un sistema de coordenadas en que se indica:

- En el eje Horizontal: un calendario, o escala de tiempo definido en términos de la unidad más adecuada al trabajo que se va a ejecutar: hora, día, semana, mes, etc.
- En el eje Vertical: Las actividades que constituyen el trabajo a ejecutar
 - A cada actividad se hace corresponder una línea horizontal cuya longitud es proporcional a su duración

Diagrama del proyecto anterior:



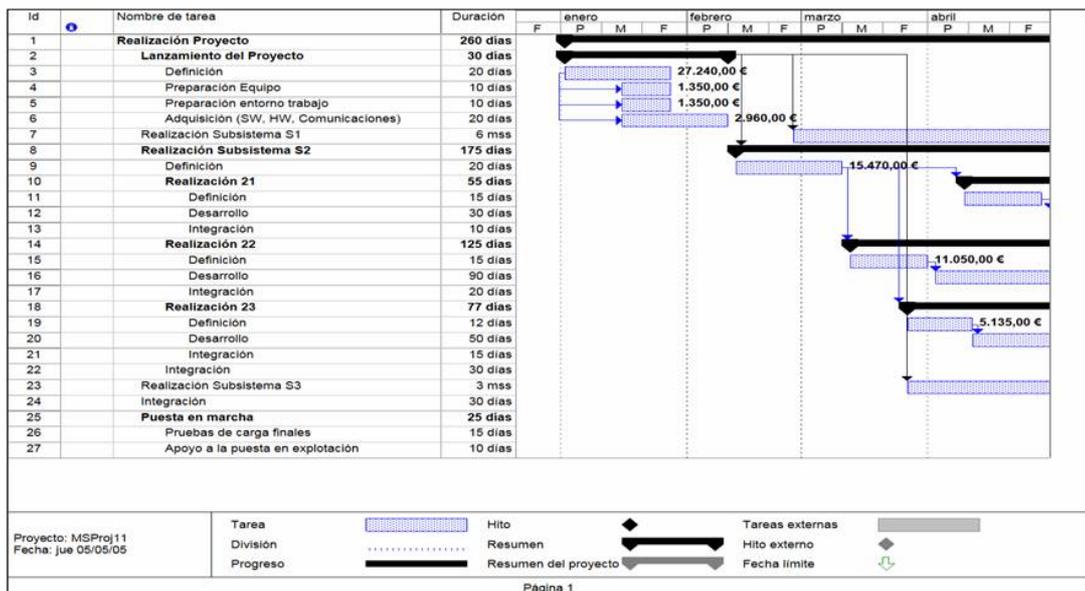
10.7 El problema de los recursos

En los ejemplos anteriores sólo se han tenido en cuenta los tiempos

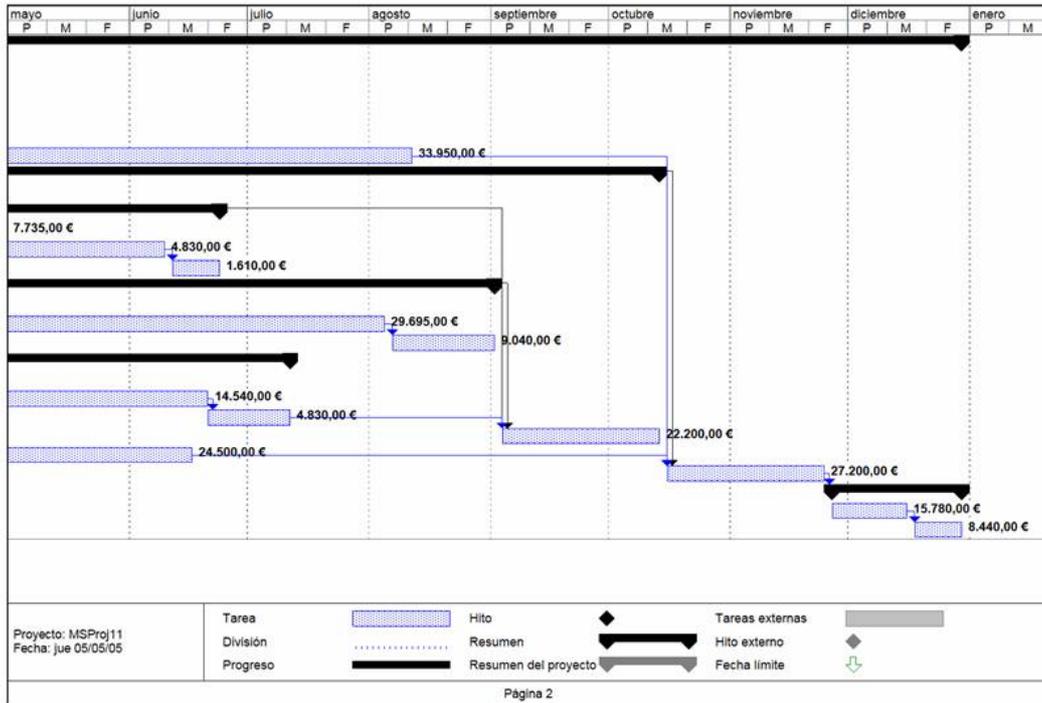
Cuando hay que utilizar recursos puede ocurrir

- Que sean limitados
- Que dos tareas necesiten el recurso al mismo tiempo
- Aunque se podrían hacer en paralelo la necesidad del recurso provoca que se realicen en secuencia
- Esto retrasa el proyecto

Ejemplo (1°)



Ejemplo (2º)



11 09. Seguimiento y Control



11.1 Índice

1. Introducción
2. Control de la calidad
3. Control de Coste
4. Control de Plazo
5. Conclusiones

11.2 Introducción

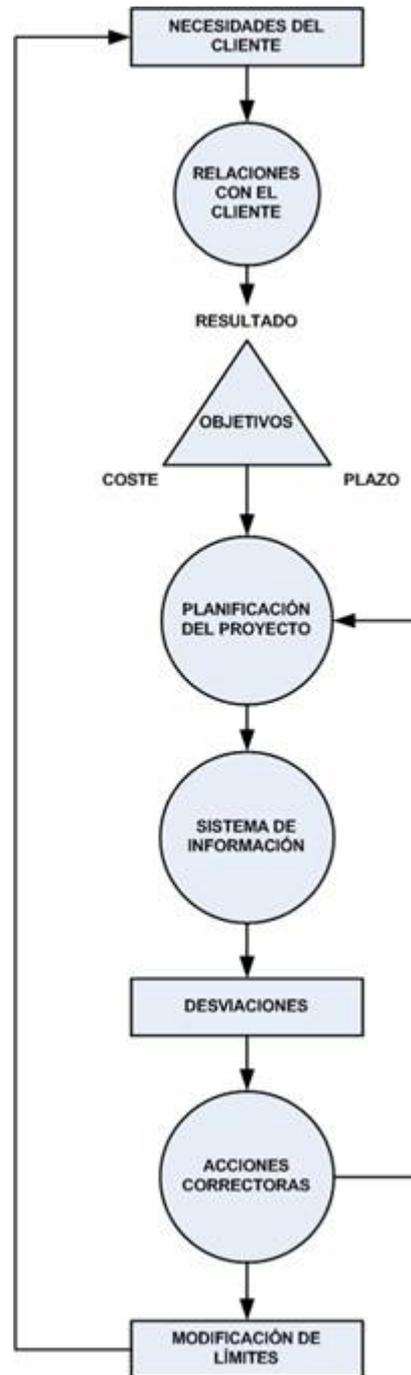
El seguimiento y control de un proyecto es un conjunto de actividades que pretenden conseguir que el proyecto se ajuste a los planes establecidos a priori y que consiga los objetivos que para él se hayan determinado.

El seguimiento de un Proyecto consiste, por tanto, en la supervisión continua ó periódica de la ejecución de un proyecto para asegurarse de que se desarrolla de acuerdo al plan previsto.

Durante la ejecución del proyecto el Jefe de proyecto es el responsable de:

- Dirección del proyecto
- Impulso al desarrollo del proyecto
- Toma de decisiones

Necesita de un sistema de información (Cuadro de Mando) que indique en todo momento la situación del proyecto en función de la planificación aprobada.



El Sistema de Información (CM) permite

- Ver la situación de los tres elementos fundamentales
 - Objetivos
 - Coste
 - Plazo
- Comparar lo previsto con lo real
 - Ver las desviaciones

- Ayudar en la toma de decisiones
 - Concentrar la atención del Jefe de Proyectos en los aspectos más relevantes

11.2.1 El Plan de Proyecto

No se realiza con la intención de acertar completamente.

No es un “corsé” sino que define el marco de referencia.

Es un elemento “vivo” que evolucionará en función de los eventos que acontezcan durante el proyecto.

Se puede comparar con:

- Los objetivos no son un horario de ferrocarril. Se pueden comparar a la aguja de la brújula que guía un barco. La aguja en sí está firme señalando en línea recta el puerto deseado. Pero durante la navegación, el barco se apartará muchas millas su curso para evitar una tormenta... Pero aún así, cuatro quintos de los viajes terminan en el puerto elegido y a la hora que se ha determinado (Peter Drucker. La gerencia de empresas. Ed. Sudamericana, Buenos Aires, 1970)

Cuanto mejor sea la planificación y más cortos y precisos sean los plazos, más fácil será la gestión del proyecto.

A medida que avanza el proyecto surgen nuevas circunstancias que hacen que los objetivos se revisen y se adapten o se incluso se cambien.

Esta es una de las principales causas de insatisfacción de clientes y equipos de trabajo.

También se usa para justificar la no consecución de los objetivos del proyecto.

Se deben evitar estas situaciones estableciendo claramente un Plan de Gestión de Cambio que determine el alcance, coste y conveniencia de cada cambio.

11.2.2 Ante el cambio...

El cliente

- Tiene completo derecho a solicitar los cambios
- Debe aceptar los procedimientos establecidos para la Gestión de Cambio
- Debe estar dispuesto a la modificación de costes y plazos

El Jefe de Proyecto

- Debe aceptar los cambios
- Estudiar éstos y replanificar el proyecto
- Presentar estos cambios al cliente

Ambos

- Deben negociar las nuevas condiciones de costes y plazos

El control de un Proyecto pasa pues por el control de

- La calidad
- El coste
- Los plazos

11.3 Control de la Calidad

El control de la calidad determina el control de los objetivos de satisfacción del cliente

- El producto sirve a las necesidades del cliente
- El coste y el plazo son importantes, pero si el producto no es el deseado los objetivos no se habrán cumplido

Para conseguir los objetivos de calidad

- Las especificaciones deben estar claras
- Conocidas y compartidos por todas las partes (cliente, jefe de proyecto, equipos de desarrollo, etc.)

El control de la calidad consiste básicamente en comparar el producto con las especificaciones y ver las desviaciones

El control de la calidad incumbe

- Al cliente
 - Debe hacer un seguimiento de las especificaciones a todo lo largo del proyecto para evitar tener que tomar decisiones al final cuando ya es demasiado tarde y costoso
- Al Jefe de Proyecto
 - Es una de sus obligaciones fundamentales

Desde el punto de vista del proyecto

- Interesa la calidad funcional, esto es, la consecución de los objetivos marcados en las especificaciones
- No interesa la calidad intrínseca, ya que no aporta necesariamente más satisfacción al cliente y puede provocar un sobrecoste importante

Ejemplo: ¿Un coche de oro o uno con altas prestaciones?

Además la calidad es un criterio subjetivo

Control preventivo:

- Adelantarse al problema *Versus* Control correctivo (Corrección del problema una vez creado)

Control y Gestión de riesgos

Control a lo largo del todo el proyecto

Se debe actuar sobre el equipo en tres direcciones

- Selección
 - Cantidad, capacidad y calidad
 - Selección de subcontratistas y colaboradores
- Formación
 - Organizar y dirigir adecuadamente la formación
 - Imponer metodologías adecuadas
 - Formación continua a cargo de los componentes de mayor experiencia
- Motivación
 - Motivar al equipo
 - Premiar el trabajo bien hecho y castigar la “chapuza”
 - No a las arengas y si a las decisiones

11.4 Control de Plazos

El control de los plazos es un proceso dinámico que se basa en la planificación temporal del proyecto

Al día siguiente de su aprobación el Plan de Proyecto puede considerarse obsoleto porque las variables que afectan al proyecto varían y los plazos varían con ellas

Se debe tener un control constante de las holguras, hitos, etc. que permita gestionar adecuadamente los recursos para adaptarse a los cambios y mejorar los plazos de ejecución del proyecto

- Aumento de los recursos, subcontratación, etc.

El control de plazos no puede hacerse sin la ayuda de los sistemas informáticos

Un PERT colgado de la pared para un proyecto largo no indica nada, quedará obsoleto y “adornará” mal la oficina

La existencia de hitos mejora la gestión

- Permite control sobre el avance

También mejora si las tareas no son excesivamente extensas y generan resultados visibles

Sistemas de control

- Informes de avance del proyecto
- Entrega de elementos “visibles”
- Realización de pruebas y de auditorías
- Reuniones de avance de proyecto
- Hitos de control

11.5 Control de Costes

Se parte de la estimación de costes inicial y se realiza un seguimiento de los costes reales para detectar las desviaciones y tomar las acciones correctoras que permitan corregir dichas desviaciones

El desglose en tareas sencillas con su coste asociado determina un mejor control del proyecto

Las tareas muy largas y con asignación de muchos recursos normalmente no se pueden controlar bien y sólo al final se saben las desviaciones

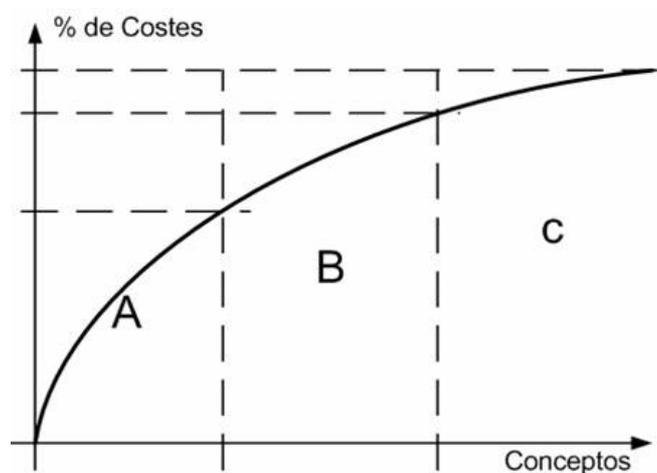
- Cuando ya es tarde para reaccionar

El uso de una contabilidad analítica o de un sistema de gestión de costes de proyectos es fundamental

- El rigor y la frialdad de una contabilidad convencional no refleja bien la gestión de costes
 - La contabilidad convencional refleja lo que ha ocurrido
 - La contabilidad analítica o un sistema de gestión permiten predecir qué pasará (Normalmente mediante escenarios de ejecución)

Habitualmente la mayor parte de los costes lo representan muy pocas partidas

- Ley 20/80 : El 20% de las partidas representa el 80% del coste y viceversa
- Curvas ABC



Estos principios permiten:

- Centrar la atención en muy pocas tareas con una efectividad muy grande sobre el coste del proyecto

Costes importantes habituales son:

- Mano de obra propia
- Maquinas
- Subcontratación
- Compras

11.5.1 Tipos de costes

- Costes externos
 - Son generados fuera de la propia empresa
- Costes internos
 - Son inherentes a la propia empresa

11.5.1.1 Costes externos

Se reflejan en factura

La gestión debe ser rápida para que se disponga de esa información cuanto antes

La factura debe ser aprobada por el equipos de proyecto (Jefe de proyecto, Jefe de compras del proyecto, etc.)

Las facturas deben repercutirse sobre las tareas que las han generado (para tener control del coste de cada tarea)

Se debe tener control de

- Presupuesto inicial (y nuevas modificaciones actualizadas)
- Gastos comprometidos hasta la fecha (y vencimientos)
- Facturas contabilizadas (pagadas)
- Parte del proyecto realizado (comparación de coste y avance para estimar los costes finales)
- Coste final previsto

11.5.1.2 Costes internos

Son peor controlados porque no existen elementos tan llamativos como lo es una factura

Ejemplos: Nomina, Máquinas, Licencias, etc.

Lo mejor es que estos recursos “pertenezcan” a la empresa y se los “subcontrate” al jefe de proyecto

- De este modo se pueden calcular los costes de recursos afectos al proyecto y a las tareas
- Suele haber resistencia a hacer esto porque evidencia la falta de productividad (si se produce)
- No deben confundirse las horas facturables al proyecto con las horas que el empleado dedica a la burocracia propia de la empresa (esto ultimo son gastos generales)

12 10. Modelos de Desarrollo de Proyectos de Software Libre



Agradecimientos a Jesús M. González Barahona por todos los trabajos publicados en la red que han servido para la creación de este capítulo de los apuntes (<http://gsyc.escet.urjc.es/personal/>)

12.1 Índice

1. Introducción ¿Qué es el software libre?
2. Historia del Software Libre
3. Tipos de Licencias de Software Libre
4. Modelos de negocio
5. Conclusiones

12.2 Introducción

12.2.1 ¿Por qué hablar de software libre?

Desde hace 30 años, nos hemos acostumbrado a que:

- Quien me vende un programa puede decirme en qué condiciones puedo usarlo
- Es normal que una empresa mantenga monopolios casi absolutos en un tipo de programas
- Si un programa tiene errores sólo su fabricante puede arreglarlos
- No tiene sentido que quiera adaptar un programa a mis necesidades
- Copiar programas es “malo”

12.2.2 ¿Qué es software libre?

Quien lo recibe puede:

- usarlo como mejor le parezca, donde mejor le parezca
- redistribuirlo a quien quiera, por los medios que quiera

- modificarlo (y mejorarlo o adaptarlo)
- redistribuir las modificaciones
 - Imprescindible: disponibilidad del código fuente.
- software libre ≠ software gratis

12.2.3 Consecuencias de la “libertad” del software

Coste: modelo de costes radicalmente distinto al del software propietario

Apertura: puede modificarse, puede inspeccionarse, puede estudiarse

Distribución: nuevos canales, nuevos métodos

Desarrollo: modelos de desarrollo “sorprendentes”

Mantenimiento y soporte: Verdadera competencia

Combinación de dos poderosos mecanismos:

- Competencia (pudiendo usar el mismo programa base)
- Cooperación (incluso involuntaria)

Historia del Software Libre

Los años 1970, los primeros 1980

- Richard Stallman, GNU, FSF
 - Fundamentos “legales” (GPL) y filosóficos
 - Infraestructura básica: editor (Emacs), compilador (GCC), depurador (GDB), etc.
 - Meta: construir un sistema completo alternativo a Unix
 - Trabajo muy estructurado y con metas claras
- Esfuerzos “aislados”: TeX, Spice, etc.
- CSRG de Berkeley:
 - Importancia de compartir fuentes (cultura Unix “original”)
 - Coartado por la licencia ATT (poco desde el punto de vista práctico, todos la tenían)
 - Énfasis en el sistema operativo (kernel, utilidades, etc.)
 - Utilizado por mucho software propietario (SunOS, Ultrix, etc.)
- Primera Internet:
 - Implementaciones de referencia, disponibles para todos
 - La Red como herramienta de cooperación (News, ftp, correo-e)
 - La comunidad de usuarios proporciona el mejor soporte

Final de los 1980, primeros 1990

- Entornos completos sobre Unix (SunOS, Solaris, etc.):
 - Muchas aplicaciones son las mejores en su campo (utilidades Unix, compiladores, etc.)
 - Especialmente interesante: XWindow
 - Sólo falta el kernel...
- 386BSD, NetBSD, FreeBSD, OpenBSD:
 - Bill Jolitz hace lo que le faltaba al kernel
 - Rápidamente: sistemas completos, similares a SunOS en funcionalidad.
 - Licencia BSD (puede usarse para distribución como software propietario)
- GNU/Linux:
 - Linus Torvalds hace su “Minix libre”
 - Cientos de desarrolladores se vuelcan sobre él, e integran con él todo el software de GNU
 - Adaptaciones de aplicaciones, nuevas aplicaciones, etc.
 - Casi todo bajo GPL: sólo puede redistribuirse con fuentes
 - Un kernel, muchas distribuciones (Slackware, Debian, RedHat, Suse, etc.)
 - Gran éxito “popular”

Finales de los 1990

- Anuncio de Netscape
- GNU/Linux y FreeBSD compiten con Windows NT
- Más y más cerca del usuario “normal”: KDE, GNOME
- GNU/Linux penetra en Universidades (y en casa de los estudiantes)
- La mejor opción es libre en muchos ámbitos (Apache, infraestructura de Internet, XFree, GCC, Gnat)
- Empresas como RedHat consiguen capital-riesgo
- La prensa comienza a atender al software libre
- Grandes empresas (Corel, Apple, IBM) estudian cómo tratar con el software libre

Principios de los 2000

- El software libre empieza a estar listo para el escritorio (GNOME
- 2.0, KDE 3.0, OpenOffice), y es simple de instalar por el usuario
- final

- El software libre se incorpora a la estrategia de grandes empresas (IBM, HP, Sun)
- Otras (como Microsoft) prefieren una estrategia de enfrentamiento parcial
- Dificultades financieras como resultado de la crisis de las puntocom
- Comienza la penetración en administraciones públicas y grandes empresas
- Aumento grande del número de desarrolladores, de la cantidad de software libre disponible, etc.

El futuro: ¿una carrera de obstáculos?

La evolución futura del software libre puede encontrarse varios obstáculos:

- Técnicas FUD (miedo, desconocimiento, duda): hasta ahora han mostrado no ser muy problemáticas
- “Disolución” (sistemas que pueden confundirse con el software libre): división de la comunidad, pérdida de las ventajas del modelo
- Desconocimiento (pérdida de visión): ¿por qué es interesante el software libre?
- Impedimentos legales: patentes de software, mecanismos de control de acceso a la información, etc.
- Abandono del Software Libre por algunas grandes corporaciones

Y habrá más...

12.3 Tipos de Licencias

Términos relacionados

- Open Source Software: término acuñado por Debian, para definir lo que incluyen en su distribución “principal”
- Public Domain Software (software del común): el autor cede sus derechos a la comunidad
- Copylefted software: software que al ser redistribuido varias veces mantiene las garantías de libertad para quien lo recibe
- Shareware: forma “diferente” de licenciar software propietario

12.3.1 ¿En qué se basan las licencias?

La legislación sobre copyright asegura que “por defecto” no se puede hacer casi nada con una obra (programa) que se recibe o se compra

Las licencias de software libre dan ciertos permisos explícitos

Cuando recibes un programa libre puedes redistribuirlo o no, pero si lo redistribuyes, sólo puedes hacerlo porque la licencia te lo permite

Para ello es preciso cumplir con la licencia...

12.3.2 BSD (Berkeley Software Distribution)

Versiones de Unix BSD

Obliga a dar crédito a los autores

Permite la redistribución binaria

Permite la redistribución fuente (pero no obliga a ella en ningún caso)

Permite las modificaciones y la integración casi sin restricciones

Licencias similares: XWindow, Tcl/Tk.

12.3.3 GPL (GNU Public License)

Software de la FSF (y mucho más, como Linux)

Interesante uso de la legislación sobre copyright (copyleft)

Permite la redistribución binaria

Permite la redistribución fuente (obliga a ella en caso de redistribución binaria)

Permite las modificaciones sin restricciones

Integración completa sólo con software cubierto por la GPL

12.3.4 LGPL (Library GPL)

Bibliotecas de la FSF (y mucho más)

Pensada para permitir el uso de bibliotecas libres con software propietario (por ejemplo, en el caso de un compilador)

Como la GPL cuando se redistribuye la biblioteca como tal

Permite la integración con cualquier otro software. En este caso, no hay prácticamente limitaciones

12.3.5 Otras licencias

Artistic (Perl, similar a la de BSD)

Aladdin (Aladdin Ghostscript, en caso de redistribución con ánimo de lucro, las condiciones son similares a las del software propietario)

NPL (Netscape, incluye ciertos privilegios para el “primer autor”)

12.3.6 Licencias duales

Un autor puede distribuir diversas obras con distintas licencias

Una nueva liberación (release) es una nueva obra

En algunos casos, se distribuyen releases que sólo difieren en la licencia

Estrategia empleada por algunas empresas, en mayor o menor medida (ej: StarOffice/OpenOffice, Mozilla/Netscape Communicator)

12.4 Modelos de Negocio

Tipos de modelo de negocio

Con financiación externa

- Quien financia decide normalmente cómo y en qué se gastan los recursos
- Normalmente, orientación a producir software
- En cierto sentido, suele ser un tipo de patrocinio

Autofinanciados

- Los ingresos provienen de actividades de la empresa
- Puede haber o no desarrollo de software nuevo

Desarrollos sin financiación directa

Desarrollos para uso interno

Modelos mixtos

12.4.1 Negocios con financiación externa (Financiación pública)

Financiación similar a la de los proyectos de I+D

La financiación suele venir de entidades promotoras de I+D

La entidad financiadora no suele buscar recuperar la inversión de forma directa

Sólo en casos muy particulares se hace de forma explícita, pero en muchos más casos es un "subproducto" de un contrato

12.4.2 Negocios con financiación externa (Financiación pública (motivaciones y caso))

Motivación "científica": que el software necesario para producir unos resultados esté disponible, para poder reproducirlos

Motivación "precompetitiva": que todo el tejido industrial se pueda beneficiar de los resultados precompetitivos

Motivación “promoción de estándares”: que se implementen versiones de referencia de un estándar

Motivación “social”: que se financie la creación de infraestructura básica común para la sociedad de la información

Caso de estudio: Gnat (compilador de Ada), alrededor de 1 M USD asignado por el Gobierno de EE.UU. a NYU para su desarrollo

12.4.3 Negocios con financiación externa (Financiación privada sin ánimo de lucro)

Normalmente realizada por fundaciones o ONGs

Motivación “directa”: producir software libre

Motivación “indirecta”: contribuir a resolver un problema mediante la producción de software libre

En general, mecanismos similares a los de la financiación pública

Casos de estudio:

- Free Software Foundation (<http://fsf.org>)
- Open Bioinformatics Foundation (<http://open-bio.org>)

12.4.4 Negocios con financiación externa

Financiación por quien necesita mejoras

Alguien necesita mejoras en un producto libre

Se financia el desarrollo por un grupo o una empresa

Caso de estudio:

- Corel quería portar sus productos a Linux
- Wine le podía permitir muchos ahorros si era mejorado
- Corel financió contribuciones de Macadamian a Wine, que incluían las mejoras que necesitaba Corel
- <http://www.macadamian.com/news/wine.html>

12.4.5 Negocios con financiación externa (Financiación indirecta)

Financiación de creación de software libre buscando beneficios en productos relacionados con él

Normalmente las ventajas (incluso para el tipo de productos que se venden) no son exclusivas

Las ventajas son mayores cuanto mayor es la cuota de mercado en el nicho en cuestión

Ejemplo: ventas de libros, hardware, distribuciones

Casos de estudio

- O'Reilly financia desarrollo en Perl, y es el principal editor sobre él (<http://ora.com>)
- VA Software (en sus comienzos como VA Research, VA Linux) colabora con el desarrollo del kernel Linux para asegurar su continuidad, vende equipos con Linux preinstalado (<http://vasoftware.com>)
- RedHat financia desarrollo de Gnome para tener un entorno de escritorio para su distribución (<http://redhat.com>)

12.4.6 Negocios autofinanciados (Basados en el mejor conocimiento)

Se venden servicios basados en el conocimiento de un producto

El conocimiento se consigue fundamentalmente trabajando en el producto y colaborando a su desarrollo

Desarrollar el producto ayuda también a la imagen

Pero no es imprescindible participar en el desarrollo

Normalmente se venden servicios de consultoría, adaptación, integración, etc.

Casos de estudio

- LinuxCare: consultoría y soporte para GNU/Linux y software libre en EE.UU (<http://linuxcare.com>)
- Alcove: consultoría y consultoría estratégica para software libre en Europa (fundamentalmente Francia. <http://www.alcove.com>)

12.4.7 Negocios autofinanciados (Basados en el mejor conocimiento con limitaciones)

Estos modelos tratan de limitar el efecto de la competencia

Típicamente se usan patentes o licencias propietarias

Normalmente se usan estos mecanismos en una parte pequeña (pero fundamental) del producto desarrollado

En muchos casos, la comunidad del software libre desarrolla su propia versión para ese componente

Caso de estudio:

- Componentes propietarios en Caldera OpenLinux (<http://www.caldera.com/products/volutionmanager/>)

12.4.8 Negocios autofinanciados (Basados en ser la fuente de un programa)

Similar a los basados en el conocimiento

La ventaja competitiva se aumenta al ser los desarrolladores del producto, y tenerlo antes que la competencia

Muy interesante en términos de imagen

Casos de estudio:

- Abiword (<http://abiword.com/>)
- Evolution, RedCarpet (Ximian - <http://ximian.com/>)
- Zope (Zope Corporation, antes Digital Creations - <http://www.zope.com/>)

12.4.9 Negocios autofinanciados (Basados en ser la fuente de un programa con limitaciones)

Se toman medidas para limitar la competencia

Ejemplo: distribución propietaria durante un tiempo, luego libre

Ejemplo: distribución limitada durante un tiempo

Estos modelos se benefician menos de las ventajas del software libre para desarrolladores (modelos menos cooperativos)

Casos de estudio:

- Alladin Ghostscript (un año entre versiones AFPL y GPL - <http://www.ghostscript.com/> / <http://www.aladdin.com>)
- Gnat (ACT) (versiones de desarrollo sólo para clientes - <http://act-europe.com>)

12.4.10 Negocios autofinanciados (Basados en licencias especiales)

Habitualmente, basados en distribuciones bajo dos o más licencias

Normalmente, complementados con consultoría sobre el producto

Ejemplo: distribución libre bajo GPL, distribución propietaria para quien quiere otras condiciones

Caso de estudio:

- dbm (SleepyCat), distribución libre obliga a incluir el fuente de cualquier aplicación que use dbm, si se redistribuye (<http://sleepycat.com/>)

12.4.11 Negocios autofinanciados (Basados en venta de marca)

Si el nombre es suficientemente reconocido, se puede vender casi cualquier cosa

Ejemplo: fabricantes de distribuciones GNU/Linux que se dedican a otras cosas

Caso de estudio:

- RedHat: comenzó dedicándose fundamentalmente a su distribución, hoy hace fundamentalmente consultoría, formación, certificación, etc. (<http://www.redhat.com>)

12.4.12 Desarrollos sin financiación directa

La mayor parte de los proyectos se están desarrollando fundamentalmente de esta forma

En muchos casos hay financiación indirecta:

- Empresas que permiten que sus empleados colaboren a tiempo parcial con proyectos
- Contribuciones por organizaciones que quieren cierta funcionalidad
- Contribuciones con infraestructura para el proyecto
- Donaciones

Casos de estudio

- Muchos grandes proyectos han establecido fundaciones que les den cobertura legal y (en parte) económica
 - Apache Software Foundation (<http://apache.org>)
 - Gnome Foundation (<http://foundation.gnome.org/>)
 - KDE League (<http://www.kdeleague.org>)

12.4.13 Desarrollos para uso interno

Al menos al comienzo, se desarrolla un producto sólo para uso interno

El desarrollador se beneficia de las ventajas del desarrollo libre (contribuciones, informes de error, parches, etc.)

La empresa se beneficia también: continuidad de soporte, inspección por terceras partes, documentación, etc.

Si hay aceptación de mercado, puede decidirse buscar ingresos relacionados con él

Caso de estudio

- Cisco Enterprise Print System (CEPS - <http://ceps.sourceforge.net>)

12.4.14 Otra clasificación (por OSI, Hecker)

“Support sellers”, venta de servicios relacionados con el producto

“Loss leader”, venta de otros productos propietarios

“Widget frosting”, venta de hardware

“Accessorizing”, venta de productos físicos (libros, etc.)

“Service enabler”, venta de servicios en línea proporcionados por el programa

“Sell it, free it”, como “loss leader” cíclico

“Brand licensing”, venta de derechos de marca

“Software franchising”, franquicia de marca

12.4.15 Otros modos de financiación

Sitios donde se ponen en contacto desarrolladores con clientes (SoureXchange)

Venta de bonos para financiar un proyecto

Cooperativas de desarrolladores (Free Developers)

Sitios:

- http://www.collab.net/sites/sxc_redirect/
- <http://freedevelopers.net/>

12.4.16 Modelos mixtos

Casi todas las empresas usan en realidad modelos mixtos

Medidas que fortalezcan la marca

Pueden contribuir o no a la creación de software

En pocos casos se dedican exclusivamente a software libre

Muchas empresas “tradicionales” están probando nuevas líneas relacionadas con software libre

12.5 Conclusiones

El software libre es un concepto económico, técnico y filosófico

Los modelos de software son muy diversos

El software libre basa su principio fundamental en que “el hombre es bueno”, “no debe aprovecharse del trabajo de otros” y “debe compartir el conocimiento”

Se ha colado con fuerza en muchos sectores, sobre todo en la Administración

Se ha convertido en un serio competidor de otros modelos

Pero...

- No se adapta a todos los modelos de producción

- No es una “panacea”
- Algunos aspectos no están aún maduros

Como conclusión

- UN JEFE DE PROYECTO DEBE SABER USAR LAS HERRAMIENTAS QUE, ESTANDO A SU ALCANCE, DEN MEJOR SATISFACCIÓN A SU CLIENTE PRODUCIENDO EL PROYECTO CON LA MEJOR CALIDAD POSIBLE...
- Si eso se consigue con Software Libre pues usémoslo y si se consigue con otros productos pues usémoslos

PARTE III

Resumen y Preguntas de Examen



13 Resumen



Este resumen ha sido realizado por Vanessa Rodríguez Álvarez y revisado y corregido por Aquilino A. Juan y Juan Manuel Cueva

La asignatura de proyectos de informática nos da los conocimientos necesarios para abordar la gestión de un proyecto de informática, teniendo en cuenta factores importantes como son el control de la calidad, el control de los costes y el control de los plazos.

Primeramente, debemos definir el concepto de proyecto: Un proyecto es un esfuerzo temporal acometido para crear un único servicio o producto. Un proyecto se caracteriza por:

- **Concreción:** Un objetivo definido.
- **Unicidad:** Responde a una demanda o necesidad puntual.
- **Dominio de aplicación:** Versa sobre un dominio de aplicación concreto.
- **Flexibilidad:** Disponibilidad de asignación de recursos.
- **Duración limitada:** Objetivos, plazos, recursos y costes íntimamente relacionados.

Es importante realizar una especificación del proyecto, que es un documento o conjunto de documentos que describen el enfoque técnico y de gestión que va a ser seguido por el proyecto, es decir, describe el trabajo que se va a llevar a cabo, los recursos requeridos, los métodos que se van a usar, los procedimientos a seguir, los plazos a cumplir y la organización que lo va a ejecutar.

Debe quedar claro que un proyecto de informática no es lo mismo que un proyecto software, pues un proyecto de informática comprende uno o varios proyectos de diversos tipos (software, hardware, comunicaciones, seguridad,...).

Para desarrollar el proyecto, se pueden seguir diversos modelos de desarrollo: modelo de cascada, modelo de prototipos, modelo en espiral o modelos ágiles. Se deberá decidir sobre la utilización de uno u otro dependiendo de las características del proyecto a ejecutar.

Cabe destacar la metodología RUP (Racional Unified Process), que es una metodología o proceso de desarrollo de proyectos software orientados a objeto. Este método toma las mejores aportaciones de cada uno de los métodos propuestos por los autores y los unifica en una metodología única que usa como lenguaje de modelado un estándar de OMG (Object Management Group). Se caracteriza por ser un modelo iterativo e incremental. La vida del sistema se divide en ciclos y en cada ciclo se producen varias iteraciones que se tratan como miniproyectos. Cada iteración se construye sobre el resultado de la anterior.

El objetivo más importante es la calidad del proyecto, entendida ésta como la aptitud del proyecto para satisfacer las necesidades de los usuarios. El sistema de gestión de la calidad (QS) es el conjunto de la estructura de la organización de responsabilidades, procedimiento, procesos y recursos que se establecen para llevar a término la gestión de calidad. Es más amplio que los requerimientos de un cliente.

Hay varios modelos de calidad, pero el más empleado y maduro es el CMM (Capability Maturity Model), diseñado para valorar sistemas muy complejos. CMM es una aplicación de sentido común de los conceptos de gestión de procesos y mejora de la calidad al desarrollo y mantenimiento del software. Estudia los procesos de desarrollo del software y produce una evaluación de la madurez de la organización según una escala de cinco niveles. Obliga a una revisión constante.

La calidad tiene un coste de implantación y para implantar un sistema de calidad, deben estar implicados todos los estamentos, desde la dirección a los operarios.

El máximo responsable de la correcta finalización de un proyecto es el jefe de proyecto, el cual se relaciona con la dirección de la empresa, con el personal asignado al proyecto, con los clientes y con otros directores de proyectos. El jefe de proyecto debe saber usar las herramientas que den mejor satisfacción al cliente, produciendo el proyecto con la mejor calidad posible, usando software libre o no. Sus funciones son:

- Colaborar con el cliente en la definición y concreción de los objetivos del proyecto.
- Planificar el proyecto: actividades, recursos, plazos y costes.
- Dirección y coordinación de los recursos empleados en el proyecto.
- Mantenimiento de las relaciones externas del proyecto: clientes, proveedores, ...
- Toma de decisiones para conocer la situación con los objetivos establecidos.
- Adopción de las medidas correctoras pertinentes.
- Responder ante clientes y superiores de la consecución de objetivos.
- Proponer, si es necesario, modificaciones a los límites u objetivos.

Un tema importante a tener en cuenta es la medición del software. Medir ayuda a proporcionar requerimientos verificables expresados en términos medibles, a proporcionar evidencias cuantificables para apoyar las decisiones, a identificar problemas anticipadamente, a producir predicciones de coste y plazo justificables, a recomendar estrategias de prueba e identificar módulos problemáticos y a valorar los efectos en la productividad y calidad.

La calidad de la medición se basa en cuatro conceptos:

- **Escala:** Nominales, ordinales, intervalo, ratio, absolutas.
- **Unicidad:** En cualquier escala la medida debe ser única.
- **Representatividad:** Condiciones para definir la transformación.
- **Significación:** Una afirmación que incluye escalas es significativa si y sólo si su valor es invariante bajo transformaciones admisibles.

Para que una medición sea válida, debe ser válida para el propósito que se pretende y debe obtener resultados significativos, el código usado debe representar adecuadamente las propiedades medidas, no debe interferir en el proceso observado, debe ser exacta y el error y la incertidumbre deben estar acotadas, debe ser reproducible, no debe derrochar recursos y los resultados deben ser imparciales.

El responsable del proyecto debe elegir las métricas que le parezcan más indicadas, siguiendo algunas reglas generales: elegir métricas estándar adecuadamente calibradas y validadas, elegir métricas para cada fase del desarrollo (análisis, diseño,...), planificar los objetivos de la medición antes de medir, elegir una colección de métricas lo más reducida posible y estimar el coste de la medición.

Es importante realizar una estimación de costes del proyecto. Para ello disponemos de varios modelos de estimación de costes (CEM) como son:

- **CEM Históricos:** Se basan en que el comportamiento de un proyecto debe ser forzosamente similar al de otros del mismo tipo. Resultados fiables sólo para proyectos similares.
- **CEM con base estadística:** Se basan en los históricos. A partir de una gran base de datos de proyectos, se estudia el comportamiento de las variables.
- **CEM con base teórica:** Buscan obtener empíricamente las relaciones que determinan el comportamiento de un proyecto. Son poco fiables.
- **CEM compuestos:** Poseen características de los CEM históricos y de los CEM con base teórica. Son los más complejos pero también los más fiables.

Para seleccionar un CEM hay que valorar la precisión, la objetividad, el carácter constructivo y la estabilidad. Los principales problemas de los CEM son que no tienen la difusión que se desearía porque no son completamente fiables y que para utilizar los métodos es necesario tener el diseño, por lo que es imposible obtener estimaciones en los primeros momentos del proyecto.

Se deben tener en cuenta los recursos utilizados en el proyecto, que se distribuyen en recursos técnicos (hardware, software,...), recursos financieros (establecimiento del coste, financiación del proyecto, plan de gastos, ...) y recursos humanos. Referente a los recursos humanos, la organización de un equipo de trabajo depende de:

- **Tamaño del equipo:** Si el equipo es grande, los costes u el esfuerzo para la comunicación entre los miembros son altos, y el director del proyecto debe tener experiencia. Si el equipo es pequeño, pueden requerirse generalistas y el director del proyecto no necesita experiencia.

- **Duración del proyecto:** Si el proyecto es corto, se tienen contribuciones de personal a tiempo parcial, y el mismo director de proyecto en todas las fases. Si el proyecto es largo, se tienen contribuciones de personal a dedicación plena y el director de proyecto puede variar en las fases.

Uno de los temas más importantes a tener en cuenta al abordar un proyecto es la gestión de riesgos. Un riesgo es cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro. El riesgo está asociado a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas. El manejo de riesgos consiste en la identificación de riesgos y a escritura de planes para minimizar el efecto de éstos en el proyecto. Hay tres clases de riesgos:

- **Riesgos de un proyecto:** Afectan a la planificación o a los recursos.
- **Riesgos del producto:** Afectan a la calidad o al desempeño del software por desarrolladores.
- **Riesgos del negocio:** Afectan a la organización que desarrolla el software.

El proceso de manejo de riesgos consiste en:

- **Identificación de riesgos:** Identifica riesgos en el proyecto, el producto y el negocio.
- **Análisis de riesgos:** Cálculo de la posibilidad de que ocurran estos riesgos y sus consecuencias.
- **Planificación de riesgos:** Trazar planes para evitar o minimizar el efecto de los riesgos.
- **Monitorización de riesgos:** Monitoriza los riesgos durante el proyecto. Determina regularmente cada riesgo identificado y decide si es probable o no que se presente.

Hay dos estrategias para enfrentarse a los riesgos:

- **Estrategias reactivas:** Aquellas que se dan cuando se deja que los riesgos produzcan sus efectos y, entonces, se actúa en consecuencia. No son aconsejables porque ponen en grave peligro el proyecto.
- **Estrategias proactivas:** Evaluación previa y sistemática de todos los riesgos, evaluando sus consecuencias. Creación de un plan de gestión de riesgos. El objetivo es la evasión del riesgo, con menor tiempo de reacción, mejor gestión del proyecto y menor coste

Los elementos que determinan el riesgo se llaman factores de riesgo y su medición y monitorización determinan el grado de acercamiento al riesgo. Un riesgo puede ser a su vez, factor de riesgo de otro. El mayor de todos los riesgos es la creencia de que todos los riesgos conforman un conjunto cerrado y están incluidos en el primer plan de riesgos realizado.

El objetivo del plan de gestión de riesgos debe ser triple, por un lado la identificación de riesgos, por otro la monitorización de los factores que afectan al riesgo y en tercer lugar la previsión de planes de contingencia si la evitación del riesgo ha resultado imposible.

Durante todo el proyecto los riesgos evolucionan, aparecen nuevos riesgos, desaparecen otros, cambia la gravedad, aparecen y desaparecen factores. Los riesgos son como un elemento vivo.

Para gestionar los riesgos es importante definir una metodología de trabajo y contar con herramientas de gestión de riesgos.

La planificación temporal de un proyecto es otro punto esencial a tener en cuenta, y consiste en considerar los recursos y estudiar las restricciones de un proyecto para distribuir el proyecto en tareas que permitan terminarlo en un plazo convenido. Cualquier cambio en las variables de entrada repercute en la planificación. Si se cambia el plazo final del proyecto también cambia la planificación y puede haber que cambiar algunas variables de entrada. Se suelen poner unas marcas en el proyecto, llamadas "milestones", que sirven para hacer el seguimiento.

Los métodos de planificación son sistemas que permiten planificar los recursos de un proyecto para lograr los objetivos. Algunos son:

- **PERT:** Definición en base a una red (grafo dirigido) de dependencias de tareas que llevan al final del proyecto. El grafo PERT está formado por actividades y nodos. Las actividades permiten avanzar de un nodo a otro. Para que una tarea pueda comenzar, deben estar terminadas todas las predecesoras. Para construir un PERT se siguen los siguientes pasos: identificar las actividades y los milestones, determinar la secuencia correcta de las actividades, construir la red, estimar el tiempo requerido por cada actividad, determinar el camino crítico y actualizar el PERT a medida que el proyecto avanza.
- **GRANTT:** Definición en forma de barras que dan un aspecto visual temporal del plan del proyecto. La distribución se hace conforme a un calendario, de manera que se pueda visualizar el período de duración de cada actividad, sus fechas de iniciación y terminación e igualmente el tiempo total requerido para la ejecución de un trabajo. Este sistema permite que se siga el curso de cada actividad al proporcionar información del porcentaje ejecutado de cada una de ellas, así como el grado de adelanto o atraso con el plazo previsto.

Para realizar la formalización del proyecto, tenemos varias estructuras:

- **WBS:** Estructura de descomposición del trabajo total en subtrabajos. Es una estructura en árbol en la que los nudos terminales son las tareas, y los nodos intermedios son tareas hamaca (reagrupación de tareas). El WBS de un proyecto no es único y permite diferentes tipos de agrupamiento (por productos, por tipos de trabajos, por responsables,...). Un WBS completo es una lista estructurada de todos los trabajos de un proyecto.
- **PBS:** Describe el objetivo del proyecto. Tiene una estructura similar al WBS pero, en vez de representar acciones (rectángulos), representa productos o estados (elipses).
- **QBS:** Describe quién hace qué. Es una tabla cuyas filas son las tareas del WBS y cuyas columnas representan los responsables de su realización.

Se pueden combinar las estructuras WBS y PBS, obteniendo el llamado "cuaderno de cargas del proyecto", donde las acciones del WBS transforman unos estados del PBS en otros.

La normalización de proyectos promueve la creación de un idioma técnico común a todas las organizaciones y es una contribución importante para la libre circulación de los productos. Además, fomenta la competitividad empresarial, sobre todo en el ámbito de las nuevas tecnologías.

Una norma es un documento establecido por consenso y aprobado por un organismo reconocido que provee, para el uso común y repetitivo, reglas, directrices o características para actividades o sus resultados dirigidos a alcanzar el nivel óptimo de orden en un concepto dado.

Los beneficios de la normalización son múltiples y apuntan a crear criterios mínimos para la aceptación de un producto, proceso o servicio. Hay entidades de normalización generales (ISO, IEEE, ANSÍ, AENOR) y técnicas (W3C, OMG, ECMA, DoD).

AENOR es una norma que trata de establecer las características generales que deben ser cubiertas en los proyectos de los sistemas de información a realizar, para que satisfagan los fines a los que están destinados. Según esta norma, los documentos básicos de los que debe constar el proyecto son:

- Índice general
- Memoria
- Requisitos del sistema
- Anexos: Diseño del sistema, estimación de tamaños y esfuerzo, planes de ejecución del proyecto, seguridad.
- Presupuesto: Determinación del coste de cada ítem del proyecto, de los costes acumulados de cada parte del proyecto y del coste total.
- Estudios con entidad propia: Aplicación de las medidas de seguridad y de las restricciones impuestas por la ley de servicio de la sociedad de la información y del comercio electrónico (LSSI). Prevención de riesgos laborales. Impacto ambiental.

El seguimiento y control de un proyecto es un conjunto de actividades que pretenden conseguir que el proyecto se ajuste a los planes establecidos a priori y que consiga los objetivos que para él se hayan determinado. Durante la ejecución del proyecto, el jefe de proyecto necesita un sistema de información (cuadro de mando) que indique en todo momento la situación del proyecto en función de la planificación aprobada. Este sistema de información permite ver la situación de los objetivos, los costes y los plazos, comprobar lo previsto con lo real y ver las desviaciones, y ayudar en la toma de decisiones.

Cuanto mejor sea la planificación y más cortos y precisos sean los plazos, más fácil será la gestión del proyecto.

14 Preguntas de Examen

14.1 TEMA 1

1. Indica cual de las siguientes afirmaciones es correcta:
 - a) **Un proyecto es un esfuerzo temporal acometido para crear un único servicio o producto**
 - b) Un proyecto informático es un esfuerzo vinculado a la informática con recursos limitados
 - c) Un proyecto es un esfuerzo humano y temporal que pretende un estándar informático

2. Señale cual de las siguientes afirmaciones es correcta:
 - a) **La RUP es la metodología o proceso de desarrollo propuesta por Racional para el desarrollo de proyectos software orientados a objeto**
 - b) Un proyecto de informática equivale a un proyecto software
 - c) La estimación del coste de un proyecto no forma parte de las responsabilidades de gestión del proyecto del Jefe de proyecto

3. RUP es:
 - a) **Una metodología para el desarrollo de proyectos software orientados a objetos que propone un modelo de procesos iterativo e incremental**
 - b) Una metodología para el desarrollo de proyectos software estructurados que propone un modelo de procesos iterativo e incremental
 - c) Una metodología para el desarrollo de procesos software orientados a objetos y que divide el desarrollo de cada mini-proyecto en cinco etapas

4. Si el proyecto se divide en iteraciones del ciclo de vida, ¿qué tipo de metodología estamos utilizando?
 - a) **RUP**
 - b) cascada
 - c) Espiral

La principal desventaja del modelo de prototipos es...

- a) **Se tiende a reutilizar el prototipo y se le colocan "parches"**
- b) El modelo de prototipos no tiene desventaja alguna
- c) Se obtienen modelos a una escala demasiado grandes

El modelo en espiral se caracteriza por:

- a) **Tener 3 fases llamadas planificación, ingeniería y evaluación y además otra fase más llamada análisis de riesgo**
- b) Tener 3 fases llamadas planificación, ingeniería y evaluación
- c) Tener 3 fases llamadas planificación, ingeniería y evaluación y también por no existir riesgo debido a que la determinación de objetivos, alternativas y restricciones claramente quedan definidos en la fase de planificación

Una de las fases del modelo en espiral es:

- a) **Análisis del riesgo**
- b) Creación de prototipos
- c) Evaluación de presupuestos

En el modelo de procesos en espiral...

- a) **La evaluación se lleva a cabo en cada iteración a lo largo de todo el proceso**
- b) La evaluación del cliente se lleva a cabo antes de entregar el producto
- c) La evaluación por parte del cliente se lleva a cabo después de entregar el producto

Cual de las siguientes afirmaciones sobre modelos ágiles es correcta:

- a) **Se basan en la adaptabilidad del proyecto**
- b) El modelo en espiral es un claro ejemplo de modelo ágil
- c) Son de carácter predictivo

Señale la respuesta correcta:

- a) **En el modelo de prototipos los modelos se validan por el usuario**

- b) El modelo en cascada refleja fielmente el proceso de desarrollo del software
- c) Los modelos ágiles son más orientados a los procesos que a las personas

¿Qué es la especificación de un Proyecto?

- a) **Documento o conjunto de documentos que describen el enfoque técnico y de gestión que va a ser seguido por un proyecto**
- b) Documento o conjunto de documentos que describen el enfoque que va a ser seguido por un proyecto
- c) Documento o conjunto de documentos que describen el enfoque técnico que va a ser seguido por un proyecto

¿Cuál de las siguientes definiciones no es cierta?

- a) **Un proyecto de informática es un proyecto SW**
- b) La Ingeniería de Software constituye la aplicación de las técnicas de Ingeniería al desarrollo de proyectos SW
- c) Los proyectos informáticos pueden ser de tipo software, hardware, comunicaciones, etc.

¿Cuál de las siguientes no es una metodología de desarrollo?

- a) **OO**
- b) RUP
- c) Ágiles

Un proyecto se caracteriza entre otras cosas por:

- a) **Objetivo definido, responde a una demanda puntual, duración limitada**
- b) Objetivo difuso, duración ilimitada, comprende distintos dominios de aplicación
- c) Unicidad, duración limitada, poco flexible a la hora de asignar más recursos tanto materiales como personales

¿Cuáles de los siguientes aspectos caracterizan a un proyecto?

- a) **Concreción, unicidad y flexibilidad, entre otras**
- b) Concreción, flexibilidad y duración no limitada

- c) Dominio de Aplicación, unicidad y duración no limitada

Hablando de métricas, el número de usuarios concurrentes, la velocidad del procesador y el tiempo de respuesta...

- a) **Son métricas de servicios, hardware y comunicaciones, respectivamente**
- b) Son métricas de software, hardware y servicios, respectivamente
- c) Son métricas de servicios, hardware y software, respectivamente

¿Cual de los siguientes modelos de desarrollo es orientado a las personas y esta basado en la adaptabilidad más que en el carácter predictivo?

- a) **Modelos Ágiles**
- b) Modelo en Espiral
- c) Modelo en Cascada

Cuál de las siguientes afirmaciones es correcta:

- a) **Un Proyecto Informático no es más que un proyecto cuyo objetivo se enmarca en el Plan General Informático de la empresa**
- b) Un Proyecto Informático no es más que un proyecto cuyo objetivo se enmarca en el Plan General Contable de la empresa
- c) Un Proyecto Informático no es más que un proyecto cuyos resultados son los que forman el Plan General Informático de la empresa

14.2 TEMA 2:

1. Cuando nos encontremos en un proyecto con situaciones no deseadas, utilizar una métrica sirve para:

- a) **Ayudar a proporcionar requerimientos verificables, expresados en términos medibles**
- b) Resolver dichas situaciones erróneas proporcionando unos requerimientos verificables
- c) Proporcionar requerimientos verificables que condicionan los objetivos del proyecto

2. Las medidas de software clasificándolas como entidades pueden ser:

- a) **De proceso, de productos y de recursos**

- b) De Atributos internos y externos
 - c) De medición directa e indirecta
3. ¿Que afirmación de las siguientes es incorrecta?
- a) **Es recomendable utilizar siempre métricas compuestas**
 - b) Hay que elegir solo aquellas métricas que sean significativas en términos de la Ingeniería del Software
 - c) Hay que elegir métricas para cada etapa tanto en análisis como en diseño, implementación, etcétera
4. Una métrica...
- a) **... debe ser válida para el propósito que se pretende y debe obtener resultados significativos con respecto a ese propósito**
 - b) ... es una medida con una base matemática, que satisface el axioma de identidad $d(x,y) = 0 \Leftrightarrow x = y$
 - c) ...sirve para tomar medidas de todas las fases de construcción del software
5. ¿Cual de las siguientes afirmaciones es correcta?
- a) **Medir ayuda a hacer más visible el desarrollo e identificar problemas anticipadamente**
 - b) Las métricas del hardware son poco habituales
 - c) No hace falta asegurar la calidad de una medición para que ésta sea válida
6. ¿Cuáles son los conceptos sobre los que se basará la calidad de una medición?
- a) **Escala, unicidad, representatividad, significación**
 - b) Escala, unicidad, representatividad, cantidad
 - c) Escala, unicidad, significación
7. Una medición válida
- a) **Debe ser exacta y su margen de error debe estar acotado**
 - b) Debe modificar el proceso observado para adaptarlo al propósito final
 - c) Es subjetiva y parcial

Un ejemplo de métrica de comunicaciones es:

- a) **Porcentaje de ocupación del ancho de banda**
- b) Número de conexiones USB disponibles
- c) Porcentaje de fallos cambio de página virtual

¿Cual de los siguientes ejemplos de mediciones, según Fenton, es significativa?

- a) **Se tarda 1 minuto en ejecutar el programa A**
- b) La complejidad del programa A es el doble que la de B
- c) La dificultad de diseño del proyecto es de 78

Para que una medición sea válida:

- a) **No debe derrochar recursos**
- b) Debe de utilizar todos los recursos disponibles
- c) No tiene importancia los recursos que consuma

¿Qué atributo debe asegurar una medición para que sea válida?

- a) **La calidad**
- b) La brevedad
- c) La redundancia

Las métricas de calidad miden, entre otras cosas,....:

- a) **...fallos o errores en el software**
- b) ...el consumo de memoria del software
- c) ...el uso adecuado de las estructuras de datos

Una métrica...

- a) **Es un mero concepto formal con cierta base matemática**
- b) Se corresponde con las medidas de ciertas variables del Proyecto
- c) No tienen aplicación en el mundo del Hardware, solo en el Software

Cual de las siguientes mediciones no es significativa

- a) **La calidad del programa A es del 97%**
- b) La profundidad de ese árbol es de 5
- c) Cuesta 3 meses escribir el programa A

14.3 TEMA 3

1. Que afirmación es VERDADERA:

- a) **Los CEM con base histórica se basan en la conjetura de que el comportamiento de un proyecto debe ser forzosamente similar al de otros proyectos del mismo tipo**
- b) Los CEM con base histórica se basan en los CEM teóricos
- c) Los CEM con base estadística buscan obtener empíricamente las relaciones que determinan el comportamiento de un proyecto

2. Los métodos CEM más fiables son:

- a) **CEM compuestos**
- b) CEM históricos para cualquier tipo de proyecto
- c) CEM con base teórica

El TRW de Wolverton:

- a) **Es un método de base histórica**
- b) Es un modelo complejo y efectivo, independiente del lenguaje de programación
- c) Contempla el análisis y el desarrollo, por tanto es un método muy orientado a entornos OO

3. ¿Cuál de las que se citan a continuación es una de las características del modelo COCOMO?

- a) **Es un modelo compuesto**
- b) Es un modelo con bases teóricas, basado en las Curvas de Taylor
- c) Se utiliza para estimar el coste de los equipos de un proyecto

4. Para la estimación del coste de un proyecto hay que tener en cuenta:

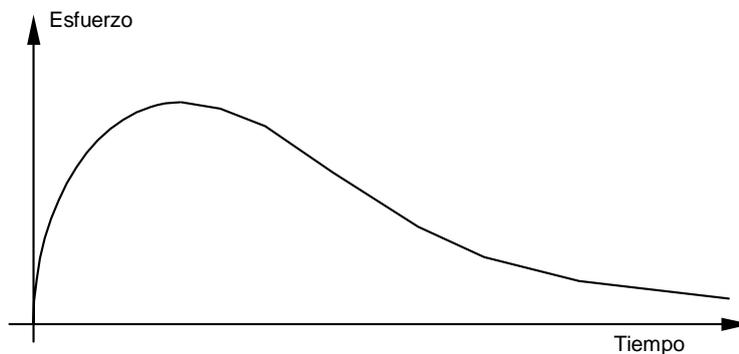
- a) **Entre otros la del Software, el Hardware, las Comunicaciones, los recursos necesarios tanto materiales como humanos**
 - b) Solo la estimación del coste del Software
 - c) La del Software y la del Hardware como principales factores y más caros (Método ABC)
5. ¿Cuál de los siguientes métodos no pertenece a los costes de desarrollo de proyecto?
- a) **Método PUTNAM**
 - b) Método PMC
 - c) Método BISAD
6. Los CEM Compuestos poseen características de otros dos modelos que son:
- a) **CEM con base Estadística y los CEM con base Teórica.**
 - b) CEM con base Histórica y los CEM con base Estadística.
 - c) CEM con base Teórica y los CEM con base Histórica.
7. La estimación de proyectos:
- a) **Se basa en modelos de estimación de costes que pueden ser, entre otros, de base estadística con base teórica**
 - b) Es equivalente a la estimación de software
 - c) Se basa en la linealidad entre coste y esfuerzo
8. ¿Qué tres componentes se dan en el modelo de estimación de Software COCOMO?
- a) **Teórico, estadístico e histórico**
 - b) Practico, analítico e histórico
 - c) Teórico, dinámico y poca dificultad
- Señale cuál de estas afirmaciones es verdadera, sobre la medición con LOCs
- a) **La medida en LOC penaliza, en general, a los lenguajes de alto nivel**
 - b) La medida en LOC premia los programas bien escritos

- c) La medida en LOC ha quedado obsoleta, y se ha sustituido por la medida en KLOC, mucho más precisa

¿Cual de las siguientes afirmaciones es correcta?

- a) **Los CEM (CostEstimatingModel) se basan en la no-linealidad entre esfuerzo y coste.**
- b) Los CEM (CostEstimatingModel) nunca aplican factores de corrección
- c) Un CEM (CostEstimatingModel) debe mostrar discontinuidades en sus límites debido al carácter binario de algunos parámetros

9. ¿Con qué modelo de estimación de software asociarías esta curva?



- a) **Rayleig-Norden**
- b) Putnam
- c) Técnica Delphi

Los modelos de CEM (Cost Estimating Model) ...

- a) **Compuestos poseen características de los CEM estadísticos de los CEM con base teórica**
- b) Históricos dan resultados fiables para todo tipo de proyectos
- c) Con base teórica buscan obtener empíricamente las relaciones que determinan el comportamiento de un proyecto y son fiables

Señala la afirmación INCORRECTA. Un buen modelo de estimación de software:

- a) **Debe basarse en la medición de las líneas de código escritas**
- b) Debe ser preciso en un amplio abanico de aplicaciones
- c) Debe evitar que la mayor parte de la estimación sea de carácter subjetivo

Según el modelo de estimación “Puntos de Función” ¿Cuál de las siguientes afirmaciones es correcta?

- a) **Realiza la estimación a partir de los requerimientos**
- b) No es un modelo adecuado para proyectos “tipo gestión”
- c) Es dependiente del lenguaje de programación

¿Cuál de las siguientes afirmaciones referentes a la estimación de proyectos y los modelos de estimación es correcta?

- a) **La estimación de costes para desarrollo de software es una parte de la estimación de proyectos**
- b) La estimación de un proyecto es equivalente a la estimación de software
- c) La ingeniería del software está en contra del uso de métodos de estimación del software científicos

De todos los modelos de estimación de coste (CEM):

- a) **Ninguno es completamente fiable**
- b) Algunos son completamente fiables
- c) Todos son completamente fiables

14.4 TEMA 4

1. Cual de las siguientes afirmaciones es correcta:

- a) **La organización interna de un equipo de trabajo depende del tamaño del equipo de trabajo y de la duración del proyecto**
- b) El entorno de desarrollo y el entorno de pruebas forman parte de los recursos financieros
- c) El cliente y el contratista forman parte del entorno de explotación

2. La organización interna de un equipo de trabajo depende fuertemente de 2 factores:

- a) **Tamaño del equipo de trabajo y duración del proyecto**
- b) Tamaño del equipo de trabajo y material necesario para el proyecto
- c) Tamaño del equipo de trabajo y recursos financieros para el mismo

3. En cuanto a la contribución de personas en un proyecto, un proyecto largo se caracteriza por:
 - a) **Contribución de personas a dedicación plena**
 - b) Contribución de personas a tiempo parcial
 - c) No contribuyen personas

4. La financiación pública de un proyecto depende:
 - a) **de subvenciones**
 - b) del cliente
 - c) del contratista

5. El entorno de pruebas se engloba dentro de:
 - a) **Los recursos Técnicos**
 - b) Los recursos Financieros
 - c) Los recursos Humanos

6. Cuando en un equipo de trabajo existe interdependencia
 - a) **Las relaciones se maximizan**
 - b) Las relaciones se minimizan
 - c) El control se basa en relaciones jerárquicas

De que tipo de dependencia estaríamos hablando si entre los miembros de un equipo de trabajo se realizan las tareas de manera conjunta...

- a) **Cooperación**
- b) Aislamiento
- c) Interdependencia

Dentro de la categoría de recursos software, los Concurrent Versions System's (CVS) pertenecen principalmente al ...

- a) **Entorno de desarrollo**
- b) Entorno de pruebas

- c) No son recursos necesarios

En los recursos financieros de un proyecto, dentro del plan de gastos del mismo, hay que tener en cuenta...

- a) **Los recursos materiales, recursos humanos, gastos financieros y otros gastos.**
- b) El tamaño de la empresa
- c) Mantener siempre el proyecto en positivo

Refiriéndose a la dependencia de los miembros de un equipo de desarrollo, lo más recomendable es:

- a) **La cooperación**
- b) El aislamiento
- c) La interdependencia

¿Cuál de las siguientes afirmaciones es correcta?

- a) **En un equipo de trabajo grande se requieren herramientas de gestión para facilitar el trabajo en equipo**
- b) En un equipo de trabajo grande los procedimientos manuales son preferibles a los automáticos
- c) En un equipo de trabajo pequeño se requiere un jefe de proyecto con más experiencia que en un equipo de trabajo grande

Indica cual de las siguientes frases es verdadera:

- a) **Los recursos técnicos, tanto de hardware como de software, deben definirse para el entorno de desarrollo, de pruebas y de explotación**
- b) El jefe de proyecto no tiene que preocuparse de los objetivos financieros del mismo, dejando esto para la empresa en la que se trabaje
- c) En el desarrollo de un proyecto no se tiene que tener en cuenta el reclutamiento de nuevas personas ya que esta situación siempre es inviable

14.5 TEMA 5

1. En la formalización de un proyecto la red lógica recoge...:

- a) **El conjunto de los trabajos definidos en el WBS e identifica las relaciones y los recursos**
 - b) El conjunto de los trabajos definidos en el PBS, identifica 'quién hace qué' y su formalización es el WBS
 - c) El conjunto de los trabajos definidos en el OBS, identificando 'quien hace que'
2. AENOR es....
- a) **Una entidad que realiza normas y estándares para la industria**
 - b) Una norma que tiene por objeto controlar el WBS que describe el objetivo del proyecto y el PBS que representa la estructura de descomposición del trabajo total en subtrabajos
 - c) Una entidad de normalización que tiene por objetivo establecer la red lógica que describe las tareas, relaciones de dependencias y recursos y las tareas hamaca que se basa en el reagrupamiento de tareas detalladas
3. Una norma:
- a) **Es un documento establecido por consenso y aprobado por un organismo reconocido**
 - b) Es un conjunto de buenas prácticas que son siempre de uso obligatorio
 - c) En un documento que describe la calidad de un software
4. En la normalización de los proyectos, según la futura norma UNE 157801, un presupuesto...
- a) **...determina el coste del proyecto desglosado por ítems y por acumulados con indicación del coste final y los beneficios esperados**
 - b) ...no determina los costes acumulados de cada parte del proyecto
 - c) ...determina solamente el coste total del proyecto
5. ¿Cuál de las siguientes afirmaciones es falsa?
- a) **La normalización perjudica la competitividad, especialmente en las nuevas tecnologías**
 - b) Los beneficios de la normalización son múltiples, y apuntan, básicamente, a crear criterios mínimos para la aceptación de un producto, proceso o servicio

- c) La normalización promueve la creación de un idioma técnico común a todas las organizaciones y es una contribución importante para la libre circulación de productos
6. ¿Como se denomina la estructura en árbol que define los trabajos a realizar en un proyecto?
- a) **Work breakdown structure (WBS)**
 - b) Organization Breakdown Structure (OBS)
 - c) Adelson Velskii Tree (AVL)
7. La IEEE Standards Association:
- a) **Es una entidad de normalización general**
 - b) Es una entidad de normalización informática
 - c) Es una entidad de formalización específica
8. La Work Breakdown Structure y la Product Breakdown Structure unidas permiten relacionar
- a) **Tareas y productos**
 - b) Personas y tareas
 - c) Personas y productos

La normalización ...

- a) **Promueve la creación de un idioma técnico común a todas las organizaciones y es una contribución importante para la libre circulación de los productos**
- b) No es un punto importante en la realización de proyectos
- c) No fomenta la competitividad empresarial, ni en el ámbito local ni en el ámbito global

Una tarea hamaca esta compuesta por:

- a) **Un conjunto de tareas agrupadas**
- b) Carga, participantes y coste total de la tarea
- c) Fecha de comienzo, coste y fecha de finalización

¿Cuál de los siguientes formalismos describe el objetivo del proyecto?

- a) **PBS: (Product Breakdown Structure)**
- b) WBS: (Work Breakdown Structure)
- c) OBS: (Organization Breakdown Structure)

Un Proyecto (según la futura UNE 157801) debe constar de los siguientes documentos básicos:

- a) **Índice General, Memoria, Requisitos del Sistema, Anexos, Presupuesto y Estudios con entidad propia**
- b) Requisitos del Sistema, Plan de Gestión de Riesgos, Plan de pruebas
- c) Memoria, Plan de Gestión de Costes, Documentación de usuario

En lo que respecta a la tarea “hamaca”. ¿Cuál de las siguientes afirmaciones es FALSA?

- a) **El comienzo lo marca la más tardía de sus tareas componentes**
- b) Está definida por un conjunto de tareas detalladas
- c) La carga es la suma de las cargas de sus componentes

La Normalización promueve:

- a) **La creación de un idioma técnico común a todas las organizaciones**
- b) El uso exclusivo de UML por parte de todas las organizaciones, para facilitar las comunicaciones entre ellas
- c) El uso exclusivo de RUP como metodología de desarrollo por parte de las empresas para una mejor interconexión entre proyectos

Indicar tres entidades que se dedican a la normalización:

- a) **W3C, IEEE, OMG**
- b) WBS, PBS, OBS
- c) HTTP, ANSI, ISO

14.6 TEMA 6

1. El modelo de calidad Bootstrap:

- a) **Está enfocado a pequeñas y medianas empresas**
 - b) Es el modelo más empleado y maduro
 - c) Define un marco común de referencia para el resto de los modelos de calidad
2. Indicar cuales de los siguientes modelos son modelos de calidad:
- a) **Malcom Balridge, SPR, ISO9000**
 - b) RUP, UML
 - c) OBS, PBS, WBS
3. El modelo de calidad 'BOOTSTRAP'...
- a) **Valora la conveniencia de las tecnologías y su impacto en los modelos de desarrollo**
 - b) Está enfocado a empresas grandes
 - c) No permite el examen de procesos individuales
4. ¿Cuáles de los siguientes son modelos de calidad?
- a) **ISO 9000, CMM, Bootstrap, Spice...**
 - b) Modelo en cascada, Modelo en espiral....
 - c) Gestión del alcance del proyecto, gestión de control de calidad del proyecto..
5. En los modelos de calidad existen varias iniciativas. ¿Qué iniciativa combina elementos de las iniciativas ISO 9000, CMN y BOOTSTRAP?
- a) **SPICE**
 - b) SPR
 - c) MALCOM BALRIDGE
6. Del modelo CMM (Capability Maturity model) podemos decir que :
- a) **Esta diseñado para valorar sistemas de gran complejidad**
 - b) Estudia el nivel de madurez de los procesos individuales
 - c) Combina elementos de otros modelos

7. La garantía de calidad esta orientada a

- a) **Proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos dados sobre calidad**
- b) Verificar los requerimientos relativos a la necesidad de un producto o servicio
- c) Determinar y aplicar la política de la calidad, los objetivos y las responsabilidades

¿A qué nos referimos cuando hablamos de calidad de diseño en un producto?

- a) **A la proporción de necesidades del usuario que son cubiertas por el diseño**
- b) A la proporción de adecuación al modelo de calidad elegido para el proyecto
- c) A la correcta aplicación de la metodología y técnicas elegidas para el desarrollo del diseño en el proyecto

Cuál de las siguientes afirmaciones sobre el modelo de calidad CMM (Capability Maturity Model) es correcta:

- a) **Produce una evaluación de la madurez de la organización según una escala de cinco niveles**
- b) Es un modelo poco empleado ya que se haya en una fase de informe técnico
- c) Los atributos de un proceso se evalúan como N(Not), P(Partially),L(Largely), y F(Fully)

¿Cuántos niveles de madurez establece el CMM?

- a) **cinco**
- b) tres
- c) cuatro

¿Cuál de los siguientes no es un modelo de calidad?

- a) **PERT**
- b) SPICE
- c) CMM

¿Cual de las siguientes afirmaciones es FALSA?

- a) **La ausencia de defectos implica calidad**
- b) La calidad tiene un coste de implantación
- c) La calidad ayuda a mejorar los sistemas de producción

El modelo de calidad CMM (Capability Maturity Model)...

- a) **Incluye mecanismos para la mejora continua de procesos**
- b) Es el menos empleado, aunque sea el más maduro
- c) Diseñado para sistemas sencillos y no de gran complejidad

14.7 TEMA 7

En el análisis de riesgos:

- a) **Se determina la probabilidad y seriedad de cada riesgo**
- b) Se considera cada riesgo y se desarrolla una estrategia y se hace un presupuesto de gestión del riesgo independiente
- c) Se determina regularmente cada riesgo identificado y se decide si es probable o no que se presente, olvidándose de él en este último caso

Cual es la definición de un Plan de contingencia:

- a) **Sirve para paliar los efectos negativos de un riesgo cuando este ya se ha producido, ofertando planes de actuación, para mitigar lo más posible los efectos del riesgo**
- b) Un plan para minimizar el impacto del riesgo y que éste no se produzca
- c) Adquirir todo el hardware necesario antes de que sea necesario su uso por las personas implicadas en el proyecto

La estrategia que pasa por la evaluación previa y sistemática de todos los riesgos inherentes al proyecto, evaluando sus consecuencias se denomina:

- a) **Estrategia proactiva**
- b) Estrategia de minimización
- c) Estrategia reactiva

¿Qué son las estrategias reactivas?

- a) **Son estrategias de gestión de riesgos consistentes en la toma de decisiones para minimizar el impacto de un riesgo cuando este ya se ha producido**
- b) Son estrategias de gestión de riesgos consistentes en minimizar los riesgos que pueda llegar a tener el proyecto que se esta desarrollando
- c) Son un conjunto de estrategias que no forman parte de los Planes de Gestión de riesgos

Los riesgos del negocio afectan a...

- a) **La organización que desarrolla el software**
- b) La calidad o al desempeño del software por desarrollarse
- c) La planificación o a los recursos

Tipos de riesgos:

- a) **Riesgos de un proyecto, riesgos del producto, riesgos del negocio**
- b) Riesgos del producto, riesgos de capital, riesgos externos a la empresa
- c) Riesgos de capital, riesgos del producto, riesgos del negocio

: Sobre los riesgos,

- a) **El análisis de riesgos determina la probabilidad y la seriedad de cada riesgo**
- b) Conforman un conjunto cerrado y están incluidos en el primer plan de riesgos
- c) Es muy importante identificar todos los riesgos al principio del proyecto, ya que si surge alguno imprevisto, no habrá posibilidad de tratarlo

El riesgo que afecta a la planificación o a los recursos es un riesgo

- a) **Riego de un proyecto**
- b) Riego del producto
- c) Riego del negocio

Dentro de la planeación de riesgos debemos hacer, entre otras cosas:

- a) **Estrategias de evitación, estrategias de minimización y planes de contingencia**

- b) Identificar los riesgos
- c) Eliminar los riesgos

¿Cuál de las siguientes afirmaciones sobre riesgos es cierta?

- a) **Un riesgo puede convertirse, a su vez, en factor de riesgo en otros riesgos**
- b) Los encargados de identificar los riesgos deben ser sólo las personas pertenecientes a la Dirección Técnica del Proyecto
- c) No pueden aparecer nuevos riesgos durante el desarrollo del proyecto, pues todos los riesgos posibles ya se han identificado al comienzo del proyecto

Las estrategias de evitación en la gestión de riesgos de un proyecto...

- a) **Tratan de minimizar la probabilidad de que el riesgo se presente**
- b) Tratan de reducir el impacto del riesgo en el producto o en el proyecto
- c) Tratan el riesgo una vez producido

Las estrategias de minimización de riesgos...

- a) **Sirven para reducir el impacto del riesgo en caso de que se produjese**
- b) Sirven para reducir el impacto del riesgo una vez que ya se han corregido sus efectos
- c) No se modifican conforme avanza el proyecto

Dentro de la planificación de riesgos, ¿qué función desempeña el plan de contingencia?

- a) **Tratar el riesgo, pero cuando este ya se ha producido**
- b) Trata de minimizar la probabilidad de que el riesgo se presente
- c) Trata de reducir el impacto del riesgo en el producto o el proyecto

Dado un proyecto, al comercializar el producto final – fruto de dicho proyecto – nos encontramos con que ya existe otro producto similar en el mercado. ¿De qué tipo es el riesgo que no hemos prevenido con eficacia?

- a) **Riesgo del negocio**
- b) Riesgo del proyecto
- c) Riesgo del producto

14.8 TEMA 8

Indique cual de las siguientes afirmaciones es correcta:

- a) **PERT y GANTT, son sistemas que permiten planificar los recursos de un proyecto para lograr los objetivos**
- b) Un aumento de personal implica una reducción de tiempos proporcional
- c) En los grafos Pert una tarea puede comenzar sin que estén terminadas las predecesoras

1. Señalar cuales de las siguientes afirmaciones sobre el PERT son ciertas:

- a) **El diagrama de PERT tiene aspecto de grafo**
- b) El PERT es la definición en forma de barras que dan un aspecto visual temporal del Plan de Proyecto
- c) El primer paso para construir un PERT es determinar el camino crítico

2. Indique cuál de las siguientes afirmaciones es correcta:

- a) **El aumento de personal de un proyecto no implica una reducción de tiempos proporcional**
- b) La planificación temporal consiste en considerar los recursos para distribuir el proyecto en tareas
- c) En el grafo de Pert para que comience una tarea no tiene importancia el estado en el que estén las predecesoras

3. El método de planificación PERT es:

- a) **Una definición en base a una red (grafo dirigido) de dependencias de tareas que llevan al final del proyecto**
- b) Una definición en forma de barras que dan un aspecto visual temporal del Plan de Proyecto
- c) Una definición en forma de grafo no dirigido de tareas que ayuda a visualizar el Plan de Proyecto

¿Cuales son los dos sistemas más usados para planificar los recursos y las tareas de un proyecto?

- a) **Pert y Gannt**

- b) Matriz de recursos y costes
- c) Gannt y Piattini

El aumento del personal en un proyecto

- a) **Cambia la productividad, pero teniendo en cuenta los costes de la comunicación**
- b) Provoca una disminución proporcional del tiempo
- c) Hace que el proyecto se retrase siempre

Las tareas del camino crítico

- a) **determinan el tiempo mínimo que va a tomar un proyecto**
- b) son las únicas que determinan la fecha de finalización del proyecto
- c) son las únicas tareas importantes del proyecto

Los milestones se deben colocar

- a) **al final de los eventos importantes del proyecto**
- b) en plazos aleatorios del proyecto
- c) siguiendo una sucesión temporal con espacios equidistantes

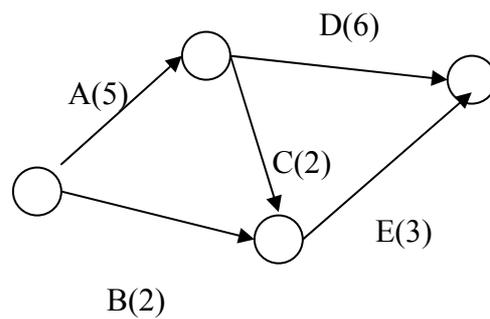
¿Que es un camino critico en un grafo Pert?

- a) **Es todo camino donde tengamos una holgura total de 0**
- b) Es el camino mas largo
- c) Es el camino mas corto

¿Qué es un Milestone?

- a) **Una marca que se pone en el proyecto para hacer el seguimiento**
- b) Una tarea que puede ejecutarse en paralelo
- c) Una tarea no crítica que admite deslizamiento

Sea el grafo de precedencia:



¿Cuál de las siguientes afirmaciones es FALSA?

- a) **No hay caminos críticos**
- b) El proyecto tiene 5 actividades las tareas precedentes de E son B y C
- c) La holgura de la actividad B es 6 y la de la actividad E es 2

¿Qué es son los milestones?

- a) **Marcas que se ponen en un proyecto para llevar un seguimiento controlado del mismo**
- b) Tareas que pueden realizarse en paralelo, que son independientes
- c) Definición en forma de barras que proporciona un aspecto visual del Plan de Proyecto

Con respecto a los Diagramas de PERT afirmar lo siguiente:

- a) **Pueden representarse las tareas como nodos y los arcos simplemente especifican precedencia**
- b) A la hora de su construcción, no es necesario estimar el tiempo de duración de las actividades
- c) El tiempo mínimo para llegar a un hito es el mínimo de los tiempos de los caminos para llegar a ese hito

Cual de las siguientes afirmaciones sobre la planificación temporal es FALSA

- a) **El aumento del personal implica una reducción de tiempos proporcional**
- b) Si se cambia el plazo final del proyecto se cambia la planificación
- c) Los Milestones son marcas que ponen en el proyecto para hacer el seguimiento

14.9 TEMA 9

1. ¿Tipos de control?

- a) **Control de calidad, Control de costes, Control de plazo**
- b) Control de calidad, Control de plazo, Control preventivo
- c) Control de riesgos, Control correctivo, Control preventivo

2. Seleccione la respuesta correcta:

- a) **La mayor parte de los costes del proyecto se concentran en unas pocas partidas (Técnicas ABC, 80%-20%)**
- b) Cuanto más largos sean los plazos, más fácil será la gestión del proyecto, puesto que habrá que controlar menos hechos
- c) La calidad del producto no se ve afectada por la claridad de las especificaciones, puesto que no busca la satisfacción del cliente, sino que el producto esté bien construido

3. Desde el punto de vista del cliente, en el proyecto interesa:

- a) **La calidad funcional**
- b) Más la calidad intrínseca que la calidad funcional
- c) La calidad intrínseca

Respecto al control de plazos indica cual de las siguientes afirmaciones es FALSA:

- a) **El control de plazos debe hacerse sin ayuda de sistemas informáticos**
- b) El control de los plazos es un proceso dinámico que se basa en la planificación temporal del proyecto
- c) La existencia de hitos mejorará la gestión

4. Los tipos de costes externos:

- a) **Son costes de servicios y productos provenientes de fuera de la propia empresa**
- b) Son inherentes a la propia empresa
- c) Son peor controlados porque no existen elementos tan llamativos como lo es una factura

5. Durante la ejecución del proyecto, el jefe de proyecto es el responsable de:
- a) **Dirección del proyecto, impulso al desarrollo del proyecto y toma de decisiones**
 - b) Definición de los requisitos del proyecto
 - c) Solicitar los cambios en el proyecto
6. En cuanto al seguimiento y control del proyecto, podemos decir que
- a) **Las variables básicas en el control de un proyecto son la calidad, el coste y los plazos**
 - b) El Plan de Proyecto debe ser realizado con el máximo nivel de acierto, ya que no se puede modificar durante el desarrollo del mismo
 - c) Es preferible la calidad intrínseca que la calidad funcional

¿A quién incumbe el control de calidad?:

- a) **Al cliente y al jefe del proyecto**
- b) Solo al cliente
- c) Solo al jefe de proyecto

Sobre el control de la calidad...

- a) **Es importante que las especificaciones estén claras**
- b) Desde el punto de vista del proyecto siempre interesa la calidad intrínseca
- c) Es algo que no incumbe al cliente

¿Qué tipo de control de calidad permite adelantarse al problema?

- a) **Preventivo**
- b) Correctivo
- c) Estructural

¿Cuál de las siguientes afirmaciones es correcta?

- a) **El control de los plazos es un proceso dinámico que se basa en la planificación temporal del proyecto**

- b) A la hora de controlar la calidad, la más importante es la calidad intrínseca, ya que aporta más satisfacción al cliente
- c) El Plan de proyecto una vez hecho, no evolucionará a lo largo de la realización del proyecto

El control de calidad....

- a) **Determina el control de los objetivos de satisfacción del cliente**
- b) Es un proceso que se basa exclusivamente en la planificación temporal del proyecto
- c) Realiza un seguimiento de los costes reales del proyecto

¿Un aumento de personal para trabajar en un proyecto trae consigo una aceleración en el desarrollo del mismo?

- a) **No, no implica una reducción de tiempos proporcional ya que puede haber problemas de comunicación y formación**
- b) Sí, puesto que a más trabajadores mayor velocidad de trabajo
- c) Sí, en todos los casos

El Control de la Calidad de un Proyecto incumbe a:

- a) **A ambos, tanto al jefe de proyecto como al cliente**
- b) Única y exclusivamente al jefe de proyecto
- c) Solo al cliente

14.10 TEMA 10

1. ¿De las siguientes afirmaciones cual es la cierta?:
 - a) **Las licencias de Software libre dan al usuario, en general, permiso para modificar el código fuente**
 - b) Los modelos de Software libre son pocos y concisos
 - c) Pocas empresas usan modelos mixtos
2. Eres director de informática de banco y tienes que encargar un producto software. ¿En cual de estos casos sería posible que el producto fuese software libre?

- a) **Si el conocimiento del código fuente no implica una ventaja a la competencia**
 - b) Que fuera Software de gestión de recursos humanos
 - c) Si es el Sistema Operativo de los escritorios
3. Indique cual de las siguientes respuestas no siempre se cumple en relación con el software libre:
- a) **Software gratis**
 - b) Disponibilidad de código fuente
 - c) Libre distribución
4. La licencia GPL(GNU Public License)
- a) **Permite la redistribución binaria y en este caso obliga la redistribución fuente**
 - b) Permite la redistribución binaria, las modificaciones, pero no permite la redistribución fuente
 - c) Permite la redistribución binaria, la redistribución fuente pero no permite realizar modificaciones
5. El software libre implica que:
- a) **Puedo modificar, adaptar y mejorar el software**
 - b) Si el software tiene errores solo su fabricante puede repararlos
 - c) El software es gratis
6. Un caso de desarrollo de software libre con financiación indirecta es:
- a) **Mediante una donación**
 - b) La financiación pública por parte de un ministerio
 - c) Financiación por parte de nuestra empresa
7. ¿En que se basan los modelos de negocio autofinanciados?
- a) **Los ingresos provienen de actividades de la empresa**
 - b) Quien financia normalmente decide cómo y en qué se gastan los recursos

- c) Suele ser un tipo de patrocinio
8. El software libre te permite
- a) **Modificarlo y redistribuirlo a quien y como tu quieras**
 - b) Usarlo de forma gratuita, pero no te permite modificarlo
 - c) Modificarlo, pero no distribuirlo libremente
9. ¿Que tipo de licencia es la que el autor cede sus derechos a la comunidad?
- a) **Public domain Software**
 - b) Open Source Software
 - c) ShareWare
10. Cuales de los siguientes elementos son licencias de software:
- a) **GPL, LGPL, BSD**
 - b) GPL, SS, DOOM
 - c) DOM, GPL, BSD

Para que podamos hablar de software libre, ¿debe ser gratuito?

- a) **No, el software libre no es lo mismo que el software gratuito.**
- b) Sí, sino estaríamos hablando de software comercial.
- c) Sí, las fuentes de ingresos no provienen de la venta, sino de otros modelos de negocio.

Los negocios con Financiación externa indirecta se basan en...:

- a) **...que el software produzca estos beneficios**
- b) ...que el estado aporte dinero para su financiación
- c) ...que se reciba dinero de organizaciones no gubernamentales

Elige cuál de estos NO es un Modelo de Negocio del Software Libre, según Hecker

- a) **Loss sellers**
- b) Widget Frosting

- c) Service enabler

La principal característica del software libre es que...

- a) **Se puede ver y modificar el código.**
- b) Es gratuito.
- c) Se puede ver y modificar el código, y además es gratuito.

Cual de estas no es una de las principales características del sw libre:

- a) **Es totalmente gratis.**
- b) Fomenta la verdadera competencia.
- c) Se usan modelos de desarrollo totalmente novedosos.

Cuál de las siguientes no es un tipo de licencia?

- a) **LDPG.**
- b) BSD.
- c) GPL.

¿Qué permite la licencia GPL?

- a) **Permite modificaciones sin restricciones.**
- b) No permite la modificación del código.
- c) Permite la modificación código con restricciones.

¿Cuál de las siguientes afirmaciones es cierta?

- a) **El software libre no siempre es software gratis**
- b) El término “shareware” es un sinónimo de “software libre”.
- c) Sólo existe un modelo de negocio basado en software libre: cobrar por servicios basados en el conocimiento de un producto

Con respecto al software libre. ¿Cuál de las siguientes afirmaciones es cierta?

- a) **Copylefted software, se trata de un tipo de software que al ser redistribuido varias veces mantiene las garantías de 'software libre' para quien lo recibe.**

- b) El modelo de software libre se adapta perfectamente a todos los modelos de producción.
- c) Las licencias de software libre no dan ningún tipo de permiso.

¿Es lo mismo decir que el software es libre y decir que es gratis?

- a) **No, aunque sea Opensource**
- b) No, a no ser que sea Opensource
- c) Sí, pero sólo si es Opensource

Si nos fijamos en los tipos de licencias software

- a) **La licencia BSD obliga a dar crédito a los autores**
- b) GPL no obliga a la redistribución del fuente en caso de distribuir el binario
- c) GPL permite la modificación con alguna restricción

Sobre la financiación externa que respuesta es FALSA

- a) **Realizada por empresas para su financiación**
- b) Las ventajas son mayores cuanto mayor es la cuota de mercado
- c) Mecanismos similares a la financiación pública



Títulos de la Colección

Ingeniería Informática

Nº CUADERNO	TÍTULO	ISBN
1	ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS	84-8497-802-8
2	DISEÑO DE PÁGINAS WEB USANDO HTML	84-8497-803-6
3	FORMATOS GRÁFICOS	84-8497-804-4
4	PROGRAMACIÓN EN LENGUAJE C	84-8497-805-2
5	ADMINISTRACIÓN DE WINDOWS N 4.0	84-8497-899-0
6	GUÍA DE USUARIO DE DERIVE PARA WUIDOWS	84-8497-919-9
7	GUÍA DE REFERENCIA DE MULTIBASE COSMOS	84-8416-001-7
8	GESTIÓN DE UN TALLER MULTIBASE CÓSOS	84-8416-034-3
9	EJERCICIOS DE LÓGICA INFOMÁTICA	84-8416-357-1
10	CONCEPTOS BÁSICOS DE PROCESADORES DE LENGUAJE	54-8416-889-1
11	INTRODUCCIÓN A LA ADMINISTRACIÓN DE UNIX	84-8416-570-1
12	LÓGICA PROPOSICIONAL PARA LA INFOMÁTICA	84-8416-613-9
13	PROGRAMACIÓN PRÁCTICA EN PROLOG	84-8416-612-0
14	LA HERRAMIENTA CASE META COSMOS	84-699-0054-4
15	GUIA DE LENGUAJE COOL DE MULTIBASE COSMOS	84-699-0053-6
16	GUÍA DE REFERENCIA PROGRESS	84-699-2083-9
17	GESTIÓN DE DEPOSITO DENTAL CON PROGRESS	84-699-2082-0
18	GUIA DE DISEÑO Y CONS. REPOSICIÓN MUL. COSMOS	84-699-2081-2
19	GESTIÓN Y ADMIN. DE UN COLEGIO MAYOR MUL-COSMOS	84-699-2080-4
20	MODELADO DE SOFTWARE EN UML	84-699-2079-0
21	EL LENGUAJE DE PROGRAMACIÓN JAVA	84-699-3880-0
22	PRINCIPIOS DE ALGORITMIA	84-699-6537-9
23	LISTAS, PILAS Y COLAS	84-699-6536-0
24	RECURSIVIDAD	84-699-6535-2
25	ARBOLES	84-699-6849-1
26	CONJUNTOS Y TABLAS DE DISPERSIÓN	84-699-7398-3
27	ALGORITMOS DE ORDENACIÓN	84-699-7397-5
28	TEORÍA DE GRAFOS	84-699-8362-8
29	INTR. AL PROCES. EFEC. DE TEXTOS CON MICROSOFT WORD	84-699-9618-5
30	ORACLE SQL	84-688-0420-7
31	TÉCNICAS DE DISEÑO DE ALGORITMOS	84-688-1764-3
32	LA PLATAFORMA . NET	84-688-3449-1
33	EJERC. DE HOJAS DE CÁL. EXCEL APLICADOS A LA GES.EMPRES.	84-688-3450-5
34	TIPOS ABSTRACTOS DE DATOS	84-688-4209-5
35	INTERPRETES T DISEÑO DE LENGUAJES DE PROGRAMACIÓN	84-688-4210-9
36	LENGUAJES Y AUTOMATAS EN PROCESADORES DE LENGUAJE	84-688-4211-7
37	METODOLOGIA DE LA PROGRAMACIÓN: GUIA DEL ALUMNO	84-688-5901-4
38	ANÁLISIS SEMÁNTICO EN PROCESADORES DE LENGUAJE	84-688-6208-8
39	ARQUITECTURA WEB EN APLICACIONES JAVA/J2EE	84-688-6580-9
40	ALGORITMICA CON FORTRAN 90	84-688-7250-4
41	TABLAS DE SÍMBOLOS EN PROCESADORES DE LENGUAJES	84-688-7631-3
42	INTRODUCCIÓN A LA COMUNICACIÓN PERSONA MÁQUINA	84-688-8362-X
43	INTRODUC. A LA PROG. ORIENTADA A OBJETOS CON JAVA	84-688-8826-6
44	DESARR. APLIC. EN SISTEM. DISTRIBUIDOS E INTERNET	84-689-3379-1
45	LÓGICA DE PREDICADOS	84-689-3380-5
46	LENGUAJE C#	84-689-5893-X
47	INTEGRACIÓN DE APLICACIONES OFIMÁTICAS	84-689-5892-1
48	INFORMÁTICA GENERAL	84-689-7033-6
49	PROYECTOS INFORMÁTICOS	

IMPRIME Y DISTRIBUYE



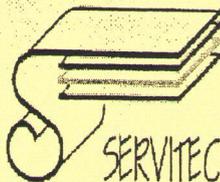
C/DOCTOR FLEMING Nº3
33005 OVIEDO
TLF-FAX 985 250581
E-mail:copisteriaservitac@fade.es

Consultor Editorial

Juan Manuel Cueva Lovelle

cueva@lsi.uniovi.es

IMPRIME Y DISTRIBUYE



C/DOCTOR FLEMING N°3
33005 OVIEDO
TLF-FAX 985 250581
www.fade.es/coplsteriaservitec
E-mail: coplsteriaservitec@fade.es