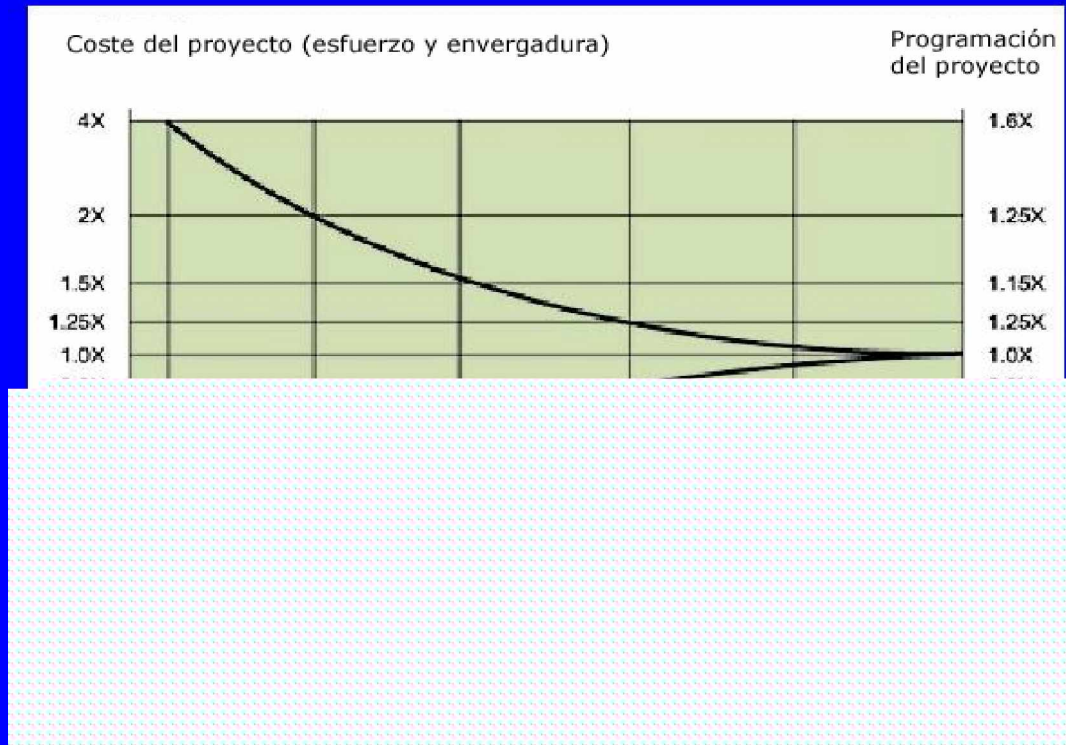


1.264 Tema 3

Cálculo de la programación del software

Cálculo



Adaptación de *Rapid Development: Taming Wild Software Schedules* (McConnell 1996).

Pasos para el cálculo

- **Calcular el tamaño del producto (puntos de función o líneas de código).**
- **Calcular el esfuerzo (personas-meses).**
- **Calcular la programación (calendario-meses)**
- **Calcular el tamaño del equipo (persona-mes calendario-meses).**
- **Ofrecer cálculos en rangos y reajustar según avance la precisión a lo largo del proyecto.**

Cálculo del tamaño del producto (punto de función)

- **Contar:**
 - **Entrada:** pantallas, formularios, cuadros de diálogo, controles, mensajes.
 - **Salida:** pantallas, informes, gráficos, mensajes.
 - **Consultas:** entrada simple, generalmente produce una salida simple.
 - **Archivos lógicos internos:** tablas o archivos de la base de datos.
 - **Archivos externos:** utilizados en otros programas.

Multiplicadores de puntos de función

Multiplicadores de puntos de función			
Característica del programa	Puntos de función		
	Complejidad baja	Complejidad media	Complejidad alta
Número de entradas	x 3	x 4	x 6
Número de salidas	x 4	x 5	x 7
Consultas	x 3	x 4	x 6
Archivos internos lógicos	x 7	x 10	x 15
Archivos internos de interfaz	x 5	x 7	x 10

Adaptación de *Rapid Development: Taming Wild Software Schedules* (McConnell 1996).

Cálculo del esfuerzo (ejemplo de punto de función)

Ejemplo del cálculo del número de puntos de función			
Característica del programa	Puntos de función		
	Complejidad baja	Complejidad media	Complejidad alta
Número de entradas	$6 \times 3 = 18$	$2 \times 4 = 8$	$3 \times 6 = 18$
Número de salidas	$7 \times 4 = 28$	$7 \times 5 = 35$	$0 \times 7 = 0$
Consultas	$0 \times 3 = 0$	$2 \times 4 = 8$	$4 \times 6 = 24$
Archivos internos lógicos	$5 \times 7 = 35$	$2 \times 10 = 20$	$3 \times 15 = 45$
Archivos internos de interfaz	$9 \times 5 = 45$	$0 \times 7 = 0$	$2 \times 10 = 20$
Total de puntos de función sin ajustar			304
Multiplicador de influencia			1.15
Total de puntos de función ajustado			350

Adaptación de Rapid Development: Taming Wild Software Schedules (McConnell 1996)

- El esfuerzo será proporcional al de proyectos anteriores
 - Si no se conoce, de 20 (peor) a 80 (mejor) puntos de función/persona-mes.

Cálculo de la programación desde puntos de función

$$SM = FP^a$$

- Donde a se define abajo.

Exponentes para el cálculo de programaciones de puntos de función			
Tipo de software	Mejor de la clase	Media	Peor de la clase
Sistemas	0.43	0.45	0.48
Negocio	0.41	0.43	0.46
Empaquetado	0.39	0.42	0.45

Adaptación de *Rapid Development: Taming Wild Software Schedules* (McConnell 1996).

- Ejemplo:
- Empaquetado: 350 puntos de función, potencia de 0,42 -> 12 meses.

Cálculo de la programación según el esfuerzo

$$SM = 3PM^{1/3}$$

- Donde SM = meses programados.
- PM = persona-meses.
- Ejemplos:

– <u>PM</u>	<u>SM</u>
– 64	12
– 125	15

Cálculo del esfuerzo: líneas de código

Niveles aproximados de lenguajes

Lenguaje	Nivel	Afirmaciones por punto de función
Assembler	1	320
Ada 83	4.5	70
AWK	15	25
C	2.5	125
C++	6.5	50
Cobol (ANSI) 85	3.5	90
dBase IV	9	35
Excel, Lotus 123, Quattro Pro otras hojas de cálculo	-50	6
Focus	8	40
Fortran 77	3	110
GW Basic	3.25	100
Lisp	5	65
Macro assembler	1.5	215
Modula 2	4	80
Oracle	8	40
Paradox	9	35
Pascal	3.5	90
Perl	15	25
Quick Basic 3	5.5	60
SAS, SPSS, otros paquetes de estadística	10	30
Small talk 80; Smalltalk/V	15	20
Sybase	8	40
Visual Basic 3	10	30

Cálculo de la programación desde tablas

- **Programación más corta posible.**
 - Talento desde el 10%, años de experiencia en el entorno.
 - Gestión ideal, todo el personal disponible desde el primer día
 - Requisitos fijados desde el primer día y sin cambios.
 - Herramientas, oficinas y métodos ideales.
- **Programación eficaz.**
 - Talento desde el 25%, baja facturación.
 - Gestión competitiva, personal disponible según necesidades
 - Necesidad mínima de cambios (5%); herramientas y oficinas disponibles y efectivas.
- **Programación nominal.**
 - Talento desde el 50%, facturación: 12% por año.
 - Cierta familiaridad con las herramientas y el entorno.

Cálculo de la programación desde tablas

Programaciones más cortas posibles

Productos de sistema			Productos de negocio		Productos de empaquetado	
Tamaño del sistema (líneas de código)	Programación (meses)	Esfuerzo (persona-meses)	Programación (meses)	Esfuerzo (persona-meses)	Programación (meses)	Esfuerzo (persona-meses)
10,000	6	25.0	3.5	5	4.2	8
15,000	7	40	4.1	8	4.9	13
20,000	8	57	4.6	11	5.6	19
25,000	9	74	5.1	15	6	24
30,000	9	110	5.5	22	7	37
35,000	10	130	5.8	26	7	44
40,000	11	170	6	34	7	57
45,000	11	195	6	39	8	66
50,000	11	230	7	46	8	79
60,000	12	285	7	57	9	98
70,000	13	350	8	71	9	120
80,000	14	410	8	83	10	140
90,000	14	480	9	96	10	170
100,000	15	540	9	110	11	190
120,000	16	690	10	140	11	240
140,000	17	820	10	160	12	280
160,000	18	960	11	190	13	335
180,000	19	1,100	11	220	13	390
200,000	20	1,250	12	250	14	440
250,000	22	1,650	13	330	15	580
300,000	24	2,100	14	420	16	725
400,000	27	2,900	15	590	19	1,000
500,000	30	3,900	17	780	20	1,400

Adaptación de Rapid Development: Taming Wild Software Schedules (McConnell 1996)

Cálculo de la programación desde tablas

Programaciones eficientes

Productos de sistema			Productos de negocio		Productos de empaquetado	
Tamaño del sistema (líneas de código)	Programación (meses)	Esfuerzo (persona-meses)	Programación (meses)	Esfuerzo (personas-meses)	Programación (meses)	Esfuerzo (personas-meses)
10,000	8	24	4.9	5	5.9	8
15,000	10	38	5.8	8	7	12
20,000	11	54	7	11	8	18
25,000	12	70	7	14	9	23
30,000	13	97	8	20	9	32
35,000	14	120	8	24	10	39
40,000	15	140	9	30	10	49
45,000	16	170	9	34	11	57
50,000	16	190	10	40	11	67
60,000	18	240	10	49	12	83
70,000	19	290	11	61	13	100
80,000	20	345	12	71	14	120
90,000	21	400	12	82	15	140
100,000	22	450	13	93	15	160
120,000	23	560	14	115	16	195
140,000	25	670	15	140	17	235
160,000	26	709	15	160	18	280
180,000	28	910	16	190	19	320
200,000	29	1,300	17	210	20	360
250,000	32	1,300	19	280	22	470
300,000	34	1,650	20	345	24	590
400,000	38	2,350	22	490	27	830
500,000	42	3,100	25	640	29	1,100

Adaptación de Rapid Development: Taming Wild Software Schedules (McConnell 1996)

Cálculo de la programación desde tablas

Programaciones nominales

Productos de sistema			Productos de negocio		Productos de empaquetado	
Tamaño del sistema (líneas de código)	Programación (meses)	Esfuerzo (persona-meses)	Programación (meses)	Esfuerzo (persona-meses)	Programación (meses)	Esfuerzo (persona-meses)
10,000	10	48	6	9	7	15
15,000	12	76	7	15	8	24
20,000	14	110	8	21	9	34
25,000	15	140	9	27	10	44
30,000	16	185	9	37	11	59
35,000	17	220	10	44	12	71
40,000	18	270	10	54	13	88
45,000	19	310	11	61	13	100
50,000	20	360	11	71	14	115
60,000	21	440	12	88	15	145
70,000	23	540	13	105	16	175
80,000	24	630	14	125	17	210
90,000	25	730	15	140	17	240
100,000	26	820	15	160	18	270
120,000	28	1,000	16	200	20	335
140,000	30	1,200	17	240	21	400
160,000	32	1,400	18	280	22	470
180,000	34	1,600	19	330	23	540
200,000	35	1,900	20	370	24	610
250,000	38	2,400	22	480	26	800
300,000	41	3,000	24	600	29	1,000
400,000	47	4,200	27	840	32	1,400
500,000	51	5,500	29	1,100	35	1,800

Adaptación de Rapid Development: Taming Wild Software Schedules (McConnell 1996)

Ajuste del cálculo

Ejemplo de historial de cálculos de punto simple

Punto del proyecto	Cálculo (persona-meses)
Concepto inicial del productor	100
Concepto aprobado del productor	100
Especificación de requisitos	135
Especificación del diseño del producto	145
Especificación del diseño detallado	160
Final	170

Ejemplo de historial de cálculos de rango

Punto del proyecto	Cálculo (persona-meses)
Cálculo (persona-meses)	25-400
Concepto inicial del productor	50-200
Concepto aprobado del productor	90-200
Especificación de requisitos	120-180
Especificación del diseño del producto	145-180
Especificación del diseño detallado	145-180
Final	170

Adaptación de Rapid Development:
Taming Wild Software Schedules (McConnell 1996)

Programación

Historial de programación para Word en Windows 1.0

Fecha del informe	Fecha estimada de envío	Días estimados para el envío	Días reales para el envío	Error relativo
Sep-84	Sep-85	365	1887 *	81%
Jun-85	Jul-86	395	1614	76%
Ene-86	Nov-86	304	1400	78%
Jun-86	May-87	334	1245	73%
Ene-87	Dic-87	334	1035	68%
Jun-87	Feb-88	245	884	72%
Ene-88	Jun-88	152	670	77%
Jun-88	Oct-88	122	518	76%
Ago-88	Ene-89	153	457	67%
Oct-88	Feb-89	123	396	69%
Ene-89	May-89	120	304	61%
Jun-89	Sep-89	92	153	40%
Jul-89	Oct-89	92	123	25%
Ago-89	Nov-89	92	92	0%
Nov-89	Nov-89	0	0	0%

* Este valor es aproximado

Estudio de caso: cálculos

- **Problemas:**
 - No hay conciencia de métodos de cálculo de recursos.
 - Suposición aproximada de duración del proyecto (muy común)
 - Negociación de duración del proyecto sin tener en cuenta la envergadura ni los recursos del proyecto.
 - No hay capacidad para supervisar o calcular el trabajo faltante de nuevo por motivos de métodos de cálculo de recursos.
 - Gestión irresponsable y elevada que obligó a Juan a comprometerse con una programación poco realista.

Estudio de caso 2: cálculos

- **Prácticas recomendadas:**
 - Uso de métodos de cálculo de recursos.
 - Negociación fundamentada, ajuste de la programación a los requisitos y recursos.
 - Supervisión cuidadosa de los requisitos y de la programación
 - Comunicación fluida y sincera con niveles altos de gestión.
 - Predisposición de escucha por parte de la alta gestión.
- **Comparación:**
 - Los estudios de caso 1 y 2 se entregaron al mismo tiempo y con el mismo número de modificaciones en la programación.
 - Uno de ellos fue un éxito, el otro fracasó.

Problemas de programación

- Los desarrolladores fallaron una media del 20-30%.
- Desfase de cálculo de proyectos pequeños: 100%.
- En el caso de proyectos grandes, peores resultados:
 - Loral y FAA Advanced Automation: más de 5 años de retraso, más de 1.000 millones de dólares sobre el presupuesto.
- Si las fechas no se cumplen se dedica más esfuerzo ofrecer explicaciones y reestructurar la programación.
- La presión afecta al ánimo y a la calidad.
 - El 40% de los errores de software se deben a la presión.
 - Suele recurrirse a "trampas" en el enfoque técnico.

Presiones en la programación

- **Causas:**

- Ingenuidad por parte de clientes y encargados.
- Ausencia de conciencia de métodos de cálculo de software.
- Mala habilidad de negociación:
 - El 75% de los desarrolladores son introvertidos, sólo 33% no lo son
 - Los encargados y comerciales suelen ser 10 años mayor y llevan toda la vida negociando.
 - Los desarrolladores desconocen los trucos de negociación (cálculos iniciales elevados, etc.).

- **Soluciones:**

- Negociación fundamentada:
 - Distintas personas en cada puesto (cooperar, explorar opciones)
 - Centrarse en intereses, no puestos (necesidades reales).
 - Buscar beneficios mutuos (fase, menos funciones, más recursos)
 - Insistir en el uso de criterios objetivos (no negociar sobre el cálculo en sí).

Control de definición de funciones

- **Primera fase del proyecto: reducción de funciones.**
 - Expectativas mínimas.
 - Reducción de requisitos.
 - Desarrollo versionado.
- **Fase media del proyecto: control del flujo de func.**
 - Cambio de análisis, cambio de estructura de control.
 - Desarrollo versionado.
 - Reducción de ciclos de desarrollo.
- **Fase final del proyecto: reducción de funciones.**
 - Eliminación de funciones no imprescindibles.
- **Y recuerde:**
 - Reducir el tamaño del proyecto en un 50% supone un 75% de reducción de recursos y un 50% de reducción en programación.

Ahorro basado en herramientas

Ejemplo del ahorro conseguido al cambiar de un 3GL a un 4 GL en el 50% de un proyecto de 32.000 LOC

Actividad	Esfuerzo nominal (persona-meses)	Ahorro esperado por la actividad	Esfuerzo final (persona-meses)	Explicación
Arquitectura (diseño de alto nivel)	6	0%	6	Misma cantidad de arquitectura necesaria. Cualquier ahorro estaría desfasado por la necesidad de combinar la arquitectura de los lenguajes 3GL y 4GL.
Diseño detallado	8	38%	5	75% de reducción para el 50% del programa codificado en 4GL.
Código/depuración	8	38%	5	75% de reducción para el 50% del programa codificado en 4GL.
Prueba de unidad	6	0%	6	Se requieren determinadas pruebas unitarias para la misma función.
Integración	6	30%	4	75% de reducción de la parte codificada en 4GL, pero la integración adicional que se necesita para mezclar 3GL y 4GL descompensa parte del ahorro.
Prueba del sistema	6	0%	6	Se requieren determinadas pruebas de sistema para la misma función.
Esfuerzo total (persona-meses)	40	20%	32	
Ahorro de la programación				
Programación prevista (meses naturales)	10.3	8%	9.5	

Adaptación de Rapid Development: *Taming Wild Software Schedules* (McConnell 1996)

Recuperación

- **Muchos proyectos pasan tiempo en modo recuperación**
 - Problema principal: no es terminar rápido sino conseguir terminar.
- **Opciones:**
 - Reducir el tamaño del software.
 - Aumentar la productividad con mejoras a corto plazo.
 - Ajustarse a la programación.
- **Plan de recuperación:**
 - Personas: ánimo, problemas serios de personal, problemas serios de liderazgo.
 - Añadir personas a un proyecto sólo consigue retrasarlo.
 - Proceso: resolver errores clásicos, pequeños hitos/seguimiento, riesgo, gestión, etc.
 - Producto: estabilizar requisitos, reducir funciones/residuos, política de trabajo, arreglar errores.