



Gestão de Projectos de Software

**Planear, organizar e controlar o
processo de
desenvolvimento de software**

Planeamento
Estimação
Análise de riscos

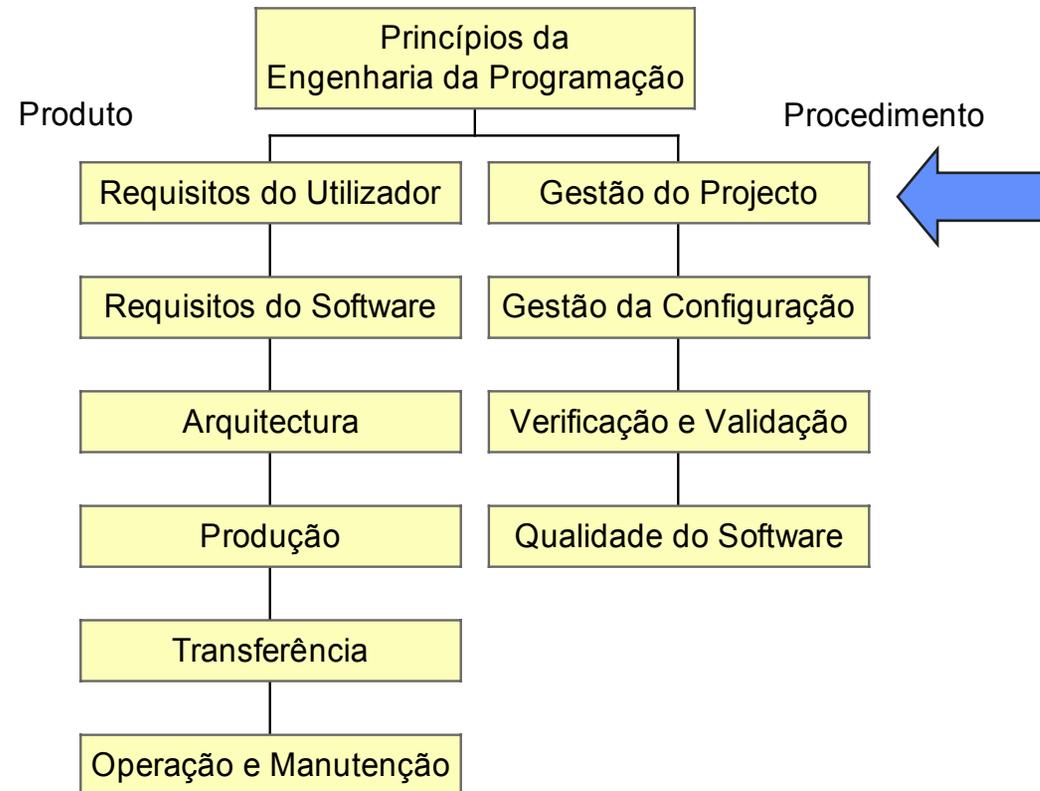


Objectivos

- **Introduzir a gestão de projectos de software e descrever as suas características distintivas**
- **Discutir o planeamento do projecto e o planeamento do processo**
- **Discutir a problemática dos custos e tempo de desenvolvimento de um projecto de software**
- **Mostrar como as representações gráficas do agendamento são utilizadas para a gestão de projectos**



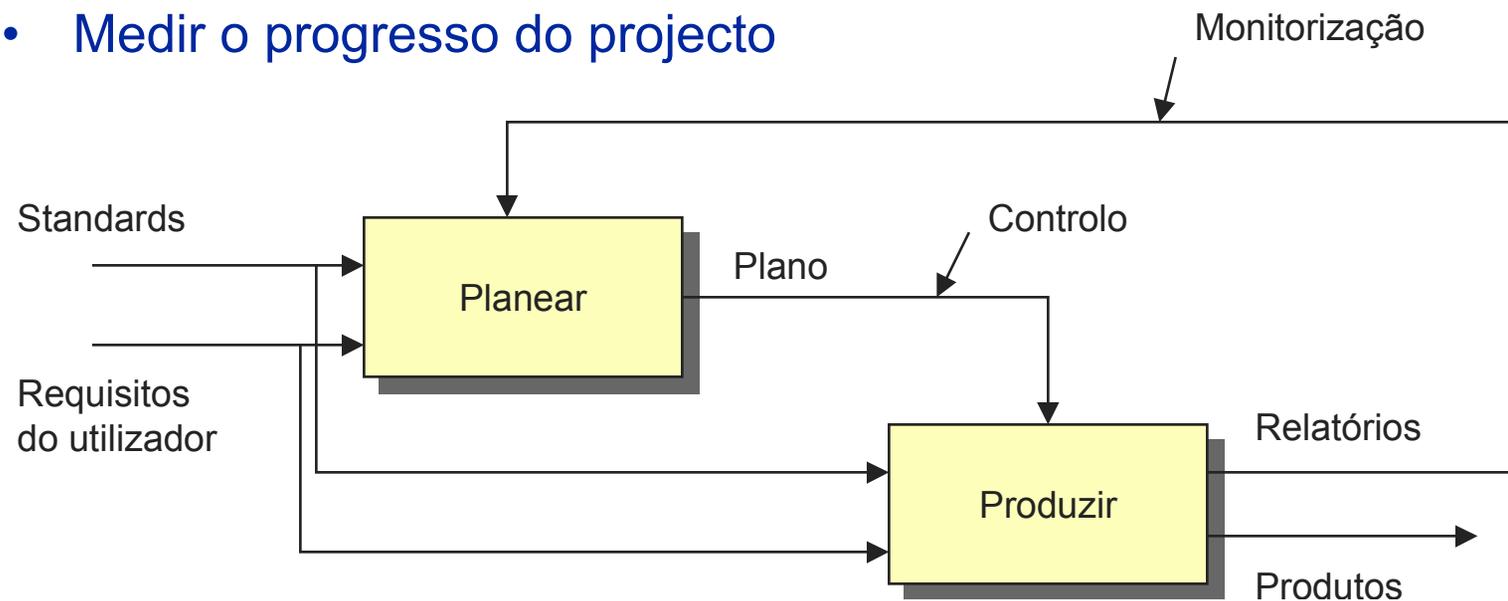
Gestão do Projecto



Gestão do Projecto

Processo de planear, organizar, definir recursos, monitorizar, controlar e liderar um projecto de software

- Definir um plano para o projecto
- Alocar pessoas e definir papeis
- Medir o progresso do projecto



Porque é Importante o Planeamento

- A engenharia de software é uma actividade económica e dessa forma está subordinada a constrangimentos económicos e não só técnicos;
- Os projectos bem geridos falham por vezes; projectos mal geridos falham inevitavelmente;
- Vamos aqui introduzir as actividades de gestão, não ensinar a ser gestores. Só se aprende a gerir, gerindo.



Gestão do Projecto

Responsabilidades do gestor:

- Entregar o produto a tempo
- Planear
- Prever
- Outras responsabilidades:
 - Interpessoais
 - Liderar a equipa, representar o projecto
 - Informacionais
 - Disseminar o plano pela equipa, monitorizar o desempenho da equipa
 - Decisionais
 - Alocar recursos, negociar alterações ao projecto, resolver falhas no plano



Decisões de Gestão

- **Têm um grande impacto nos aspectos técnicos da engenharia de software:**
 - se o desempenho for medido em termos de quantas linhas de código forem produzidas, é desencorajada a reutilização;
 - se um agendamento irrealisticamente agressivo for imposto, haverá um encorajar na tomada de atalhos que normalmente afectarão a qualidade do produto e reduzem a sua facilidade / possibilidade de manutenção;
 - uma falta de plano, encoraja no sentido de tomada de decisões grandiosas ou não prosseguir cuidadosamente, dado que se pensará haver sempre mais tempo.



Questões de Gestão de Projecto

- **Envolvem normalmente complicadas decisões de balanceamento de prós e contras:**
 - Qual será o benefício de investir em ferramentas de engenharia de software modernas?
 - Será que esse investimento possibilitará a diminuição de tempo de desenvolvimento, e se assim for, qual o valor do tempo poupado?
 - Quanto tempo de desenvolvimento adicional custará o adicionar de uma determinada característica?
 - É possível diminuir o tempo de entrega inicial do sistema a expensas de manutenção futura? E quanto custará essa manutenção futura?
 - Quais serão os custos e benefícios de uma entrega incremental, e que características deverão ser entregues no início?
 - Se uma característica do produto ainda não está devidamente testada, quais serão os benefícios de entregar o produto sem ela?
 - Modificar um produto existente ou criar um a partir do nada?

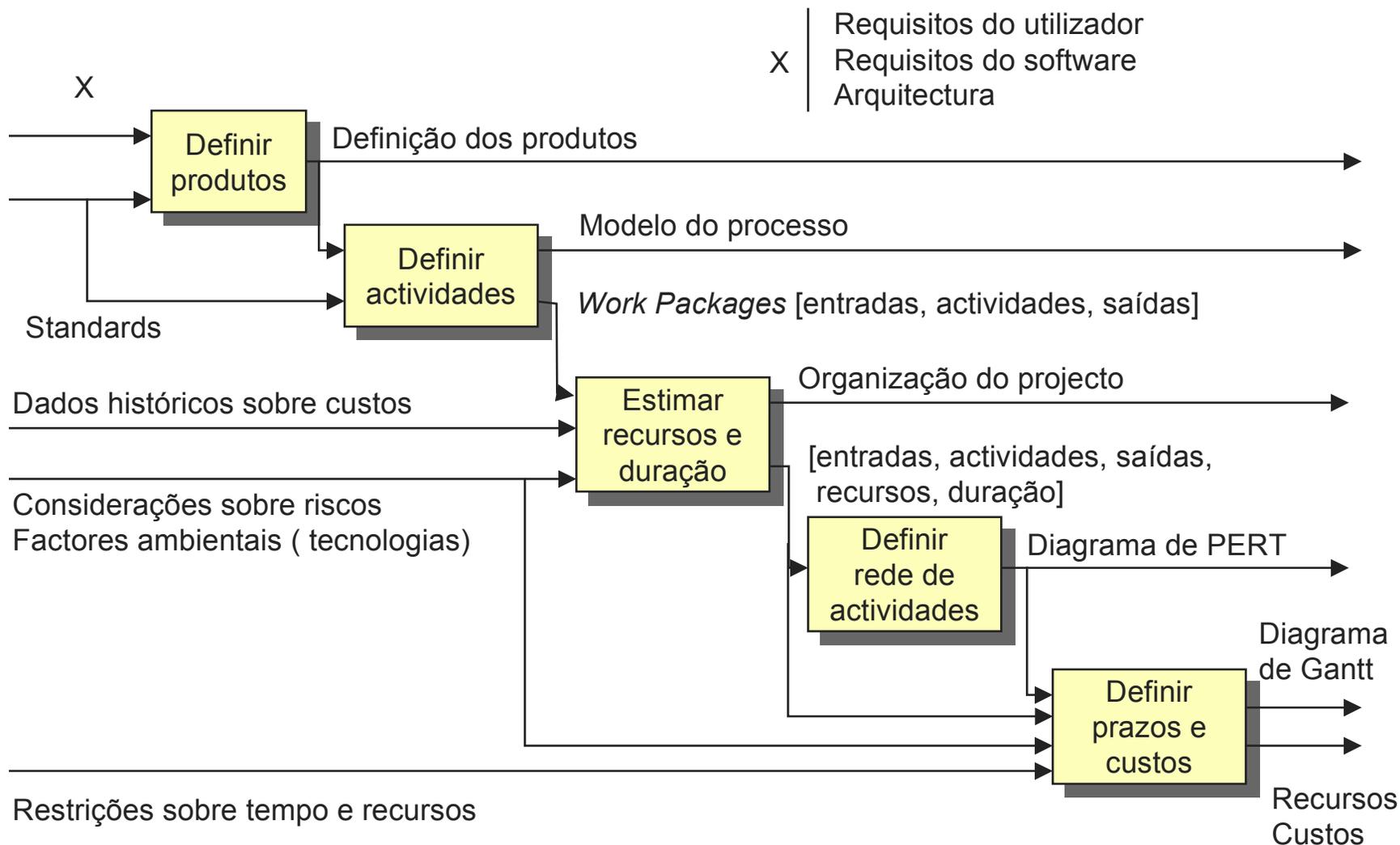


Funções de Gestão de Projecto

- **Permitir a um grupo de pessoas trabalhar em direcção a um objectivo comum.**
- **Actividades:**
 - **Planeamento** - que objectivos devem ser atingidos, que recursos são requeridos, como podem ser obtidos e como poderão ser atingidos os objectivos. Determina o fluxo de informação, pessoas e produtos dentro da organização.
 - **Organização** - Envolve o estabelecimento de linhas claras de autoridade e responsabilidade para grupos de actividades que permitam atingir os objectivos.
 - **Recursos humanos** - Contratar pessoas para as posições identificadas pela estrutura da organização.
 - **Dirigir** - Liderar e guiar o membros do grupo por forma a facilitar a compreensão e identificação da estrutura da organização, objectivos da empresa e sua cultura.
 - **Controlar** - Medir e corrigir actividades para assegurar que os objectivos serão atingidos. Controlar requer a medida de desempenho relativamente a planos e a tomada de acções, na ocorrência de desvios.



Planeamento do Projecto



Planeamento do Projecto

Definir os produtos

Fase

Requisitos do software
Arquitectura
Produção
Transferência

Entrada

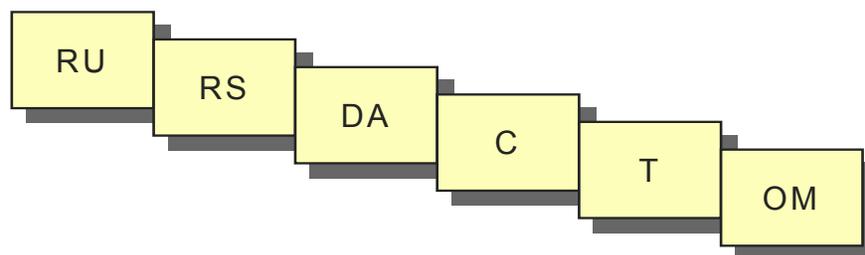
Req. utilizador
Req. software
Desenho arqu.
Código. Man. Util.

Saída

Req. software
Desenho arqu.
Código, Man. Util.
Man. Instalação

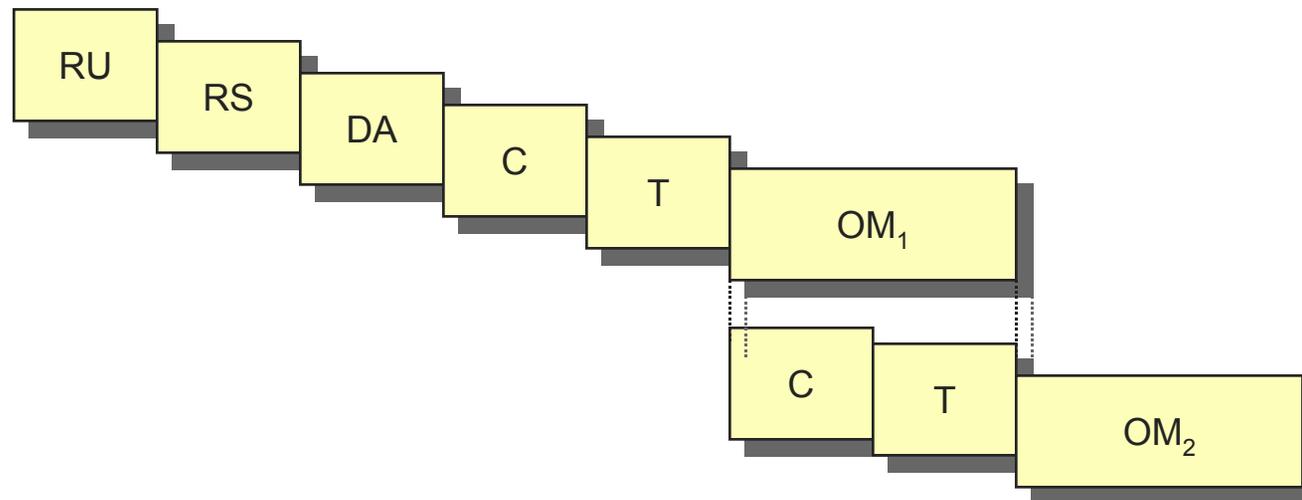
Definir as actividades

- Definir modelo do processo
 - **Cascata** - Requisitos do utilizador conhecidos e estáveis, duração do projecto reduzida (< 2 anos), produto deve ser entregue de uma vez



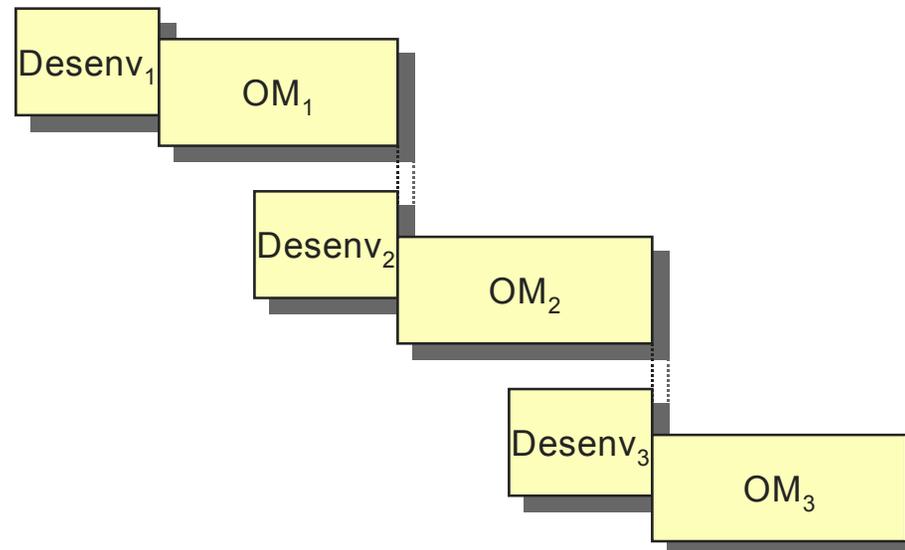
Planeamento do Projecto

- Definir modelo do processo
 - **incremental** - Entrega do produto de acordo com prioridades do utilizador, necessidade de melhorar a integração com o sistema, evidência de que o produto será aceite pelo utilizador



Planeamento do Projecto

- Definir o modelo do processo
 - **Evolucionário** - Necessária experimentação do utilizador para completar os requisitos; partes do produto dependem da disponibilidade de futuras tecnologias; requisitos do utilizador antecipados mas não conhecidos; não impedir o desenvolvimento de partes do produto devido a dificuldades na realização de outras



Planeamento do Projecto

Definir as actividades (continuação)

- Seleccionar ferramentas a utilizar em cada actividade
- Definir pacotes de actividades (*work packages*)
 - Critérios a utilizar
 - Coerência - As tarefas de um pacote devem ter o mesmo objectivo
 - Coesão - As dependências entre pacotes devem ser minimizadas
 - Continuidade - As tarefas de produção devem ser contínuas de modo a maximizar a eficiência
 - Quanto maior o nível de detalhe maior o grau de precisão na estimação de custos



Planeamento do Projecto

Recursos Humanos:

- **Pode não ser possível designar as pessoas ideais para trabalhar num projecto**
 - O orçamento do projecto pode não comportar a afectação ou contratação de pessoal muito bem pago;
 - Pessoal com experiência adequada pode não estar disponível;
 - Uma organização pode desejar desenvolver competências de empregados seus, no âmbito de um projecto de software.

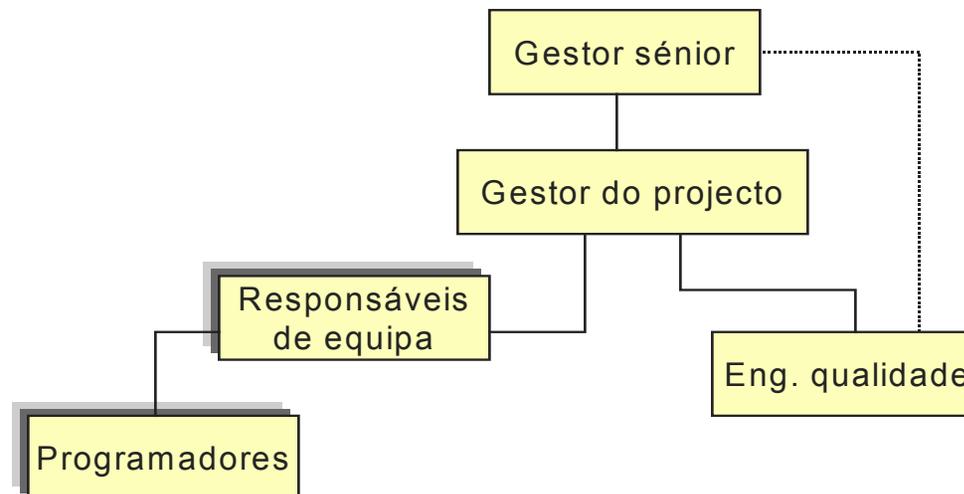


Planeamento do Projecto

Estimar recursos e duração

- Recursos humanos

- **Papeis** - Gestor do projecto, responsável pela equipa, programadores, engenheiros de teste, engenheiros de qualidade
- **Regras** - Garantir que cada membro da equipa reporta a um único indivíduo; garantir que não há mais de 7 membros a reportar a um único indivíduo
- **Organigrama**



Planeamento do Projecto

Estimar recursos e duração

- Esforço
 - Número de **pessoas-mês** necessárias à execução do projecto. Valor obtido através da aplicação de modelos predictivos (Decomposição, COCOMO, Putnam, definidos mais adiante)
 - 1 pessoa-mês = 1 pessoa a trabalhar durante um mês (22 dias) ou 22 pessoas a trabalhar durante um dia
 - 1 pessoa-ano = 11 pessoas-mês
 - 1 pessoa-mês = 22 pessoas-dia
 - 1 pessoa-dia = 8 pessoas-hora
- Custos não laborais
 - Ferramentas comerciais utilizadas para produzir o produto, serviços externos, despesas com viagens, despesas de envio do produto, seguros
- Duração - Calcula-se a partir do esforço e de valores históricos da produtividade média da equipa do projecto



Planeamento do Projecto

Definir rede de actividades

- Representar os pacotes de actividades como um conjunto de nós com ligações entre eles. Uma sequência de ligações define um caminho a seguir pelo projecto. Ligações circulares não são permitidas
- O objectivo é desenhar um percurso que considera todas as **dependências** do projecto
- **Caminho crítico** - Percurso mais longo através das ligações entre actividades, em termos de duração do projecto
- **Flutuações dos pacotes de actividades** - Diferença entre o tempo mais adiantado (*earliest*) e mais atrasado (*latest*) de início de um pacote do projecto



Planeamento do Projecto

Definir prazos e custos

- Definir datas de início e fim dos pacotes de actividades
 - Cumprindo com as restrições de tempo e recursos
 - Minimizando o custo total
 - Admitindo riscos que possam atrasar o projecto
 - Se o prazo total do projecto viola restrições de tempo, é necessário redefinir os pacotes de actividades
- O custo total mínimo do projecto corresponde à soma de todos os pacotes de actividades
 - Os custos laborais devem ser calculados a partir da soma do tempo dispendido por cada indivíduo no projecto
 - Deve-se incluir os tempos de espera pelo início de pacotes de actividade
 - Os prazos devem ser ajustados de modo a minimizar o custo total do projecto



Processo de Planeamento do Projecto

Estabelecer os constrangimentos (data de entrega, staff disponível, orçamento, etc.) que afectam o projecto

Efectuar a atribuição dos parâmetros iniciais do projecto (estrutura, tamanho e distribuição de funções)

Definir metas do projecto e entregas

while o projecto não foi completado ou cancelado **loop**

 Desenhar uma agenda do projecto

 Iniciar as actividades de acordo com o agendamento ou dada permissão para continuar

 Esperar (por um tempo, 2-3 semanas)

 Rever o progresso do projecto e discrepâncias notadas

 Rever as estimativas dos parâmetros do projecto

 Actualizar a agenda do projecto

 Renegociar os constrangimentos do projecto e entregas (se o projecto estiver atrasado)

If (surgirem problemas, não sendo possível revisão)

 Iniciar pesquisa técnica e possível revisão, para encontrar possíveis alternativas de desenvolvimento que permitam responder aos constrangimentos do processo

end if

end loop



Processo de Planeamento do Projecto

- **Observações:**

- Mostrar que o planeamento do projecto é um processo iterativo que só está completo quando o projecto está completo;
- À medida que a informação sobre o projecto fica disponível durante o projecto, o plano deve ser revisto regularmente;
- Se o projecto ficar atrasado, tentar renegociar os constrangimentos do projecto e entregas;
- Caso a renegociação falhe, tentar encontrar alternativas de desenvolvimento que permitam colocar o projecto dentro dos constrangimentos;
- Não deve assumir-se que tudo irá correr bem; as suposições e agendamento iniciais devem ser pessimistas e não optimistas;
- Deve ser incluído suficiente lugar a contingências no plano de forma a que não haja lugar a uma renegociação constante.



Estrutura do Plano de Projecto

Estabelece o recursos disponíveis, a divisão do trabalho e a agenda do projecto

- **Introdução**
 - Descreve brevemente os objectivos e estabelece os constrangimentos (orçamento, tempo, etc.) que afectam a gestão do projecto.
- **Organização do Projecto**
 - Descreve a forma como a equipa de desenvolvimento está organizada, as pessoas envolvidas e as suas funções na equipa.
- **Análise de Risco**
 - Descreve os riscos possíveis do projecto, a possibilidade desses riscos surgirem e as estratégias para a redução dos riscos propostas.



Estrutura do Plano de Projecto

- **Requisitos de recursos de hardware e software**
 - O hardware e software de suporte que é necessário para o projecto; são incluídos cálculos, estimativas e a agenda de entrega respectiva.
- **Divisão do trabalho**
 - Divisão do trabalho em actividades, identificando as metas intermédias e entregas associadas a cada actividade.
- **Gestão de projecto**
 - Descreve a dependência entre actividades, o tempo estimado para atingir cada meta intermédia e a alocação de pessoas a actividades.
- **Mecanismos de monitorização e relatórios**
 - Descreve os relatórios de gestão que devem ser produzidos, quando devem ser produzidos e mecanismos de monitorização.



Organização de Actividades

- **As actividades num projecto devem ser organizadas para produzir resultados tangíveis. Sem isto não é possível avaliar o processo e as estimativas de custo e a agenda não podem ser actualizadas.**
- **Podem ser:**
 - Metas parciais são o ponto final de uma actividade de um processo, findo o qual deve ser apresentado um relatório do progresso à gestão do projecto.
 - Entregas são resultados do projecto a ser entregues aos clientes. Podem ser metas parciais. Constituem em regra o final de uma fase do projecto. Por exemplo, especificação, concepção... As metas parciais não constituem necessariamente entregas, dado que os resultados podem ser para consumo interno.
- **O processo em cascata conduz à definição directa de metas parciais, daí a sua grande utilização.**



Agenda do Projecto

Estimar tempo e recursos necessários para completar as actividades e organizá-las numa sequência coerente.

- **Divisão do projecto em tarefas e estimar o tempo e recursos necessários para completar cada tarefa**
- **Organizar as tarefas concorrentemente para permitir uma utilização óptima da força de trabalho**
- **Minimizar as dependências entre tarefas para evitar atrasos motivados pela espera forçada de uma tarefa, dado que outra ainda não está completa**
- **Dependente da intuição dos gestores de projecto e da sua experiência**



Problemas do Agendamento

- **Estimar o grau de dificuldade dos problemas e os consequentes custos do desenvolvimento de uma solução é difícil**
- **A produtividade não é proporcional ao número de pessoas a trabalhar numa tarefa**
- **Acrescentar pessoas a um projecto atrasado, faz com que ele se atrase mais, devido a sobrecargas de comunicação**
- **O inesperado sempre acontece; deve permitir-se sempre margem para as contingências no planeamento**



Gráficos de Barras e Redes de Actividades

- **Notações gráficas para ilustrar a agenda do projecto**
- **Mostram a divisão do projecto em tarefas**
- **Os gráficos de actividades mostram as dependências entre tarefas e o caminho crítico**
- **Os gráficos de barras mostram o agendamento das actividades no tempo e mostram quem é o responsável por cada tarefa e quando tem início e fim.**



Duração das Tarefas e Dependências

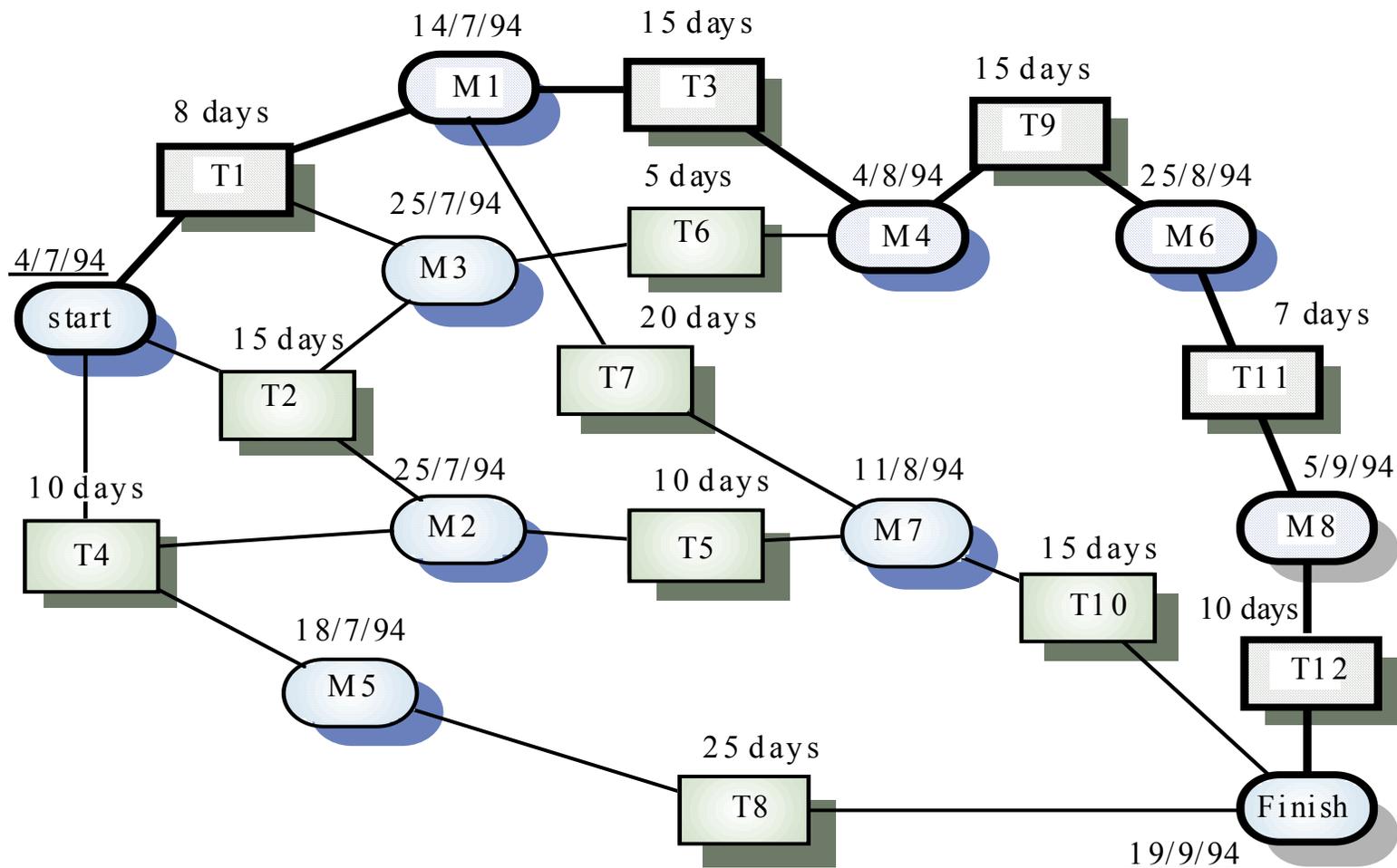
Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1
T4	10	
T5	10	T2, T4
T6	5	T1, T2
T7	20	T1
T8	25	T4
T9	15	T3, T6
T10	15	T5, T7
T11	7	T9
T12	10	T11

Tem de estar terminada antes de iniciar T3.

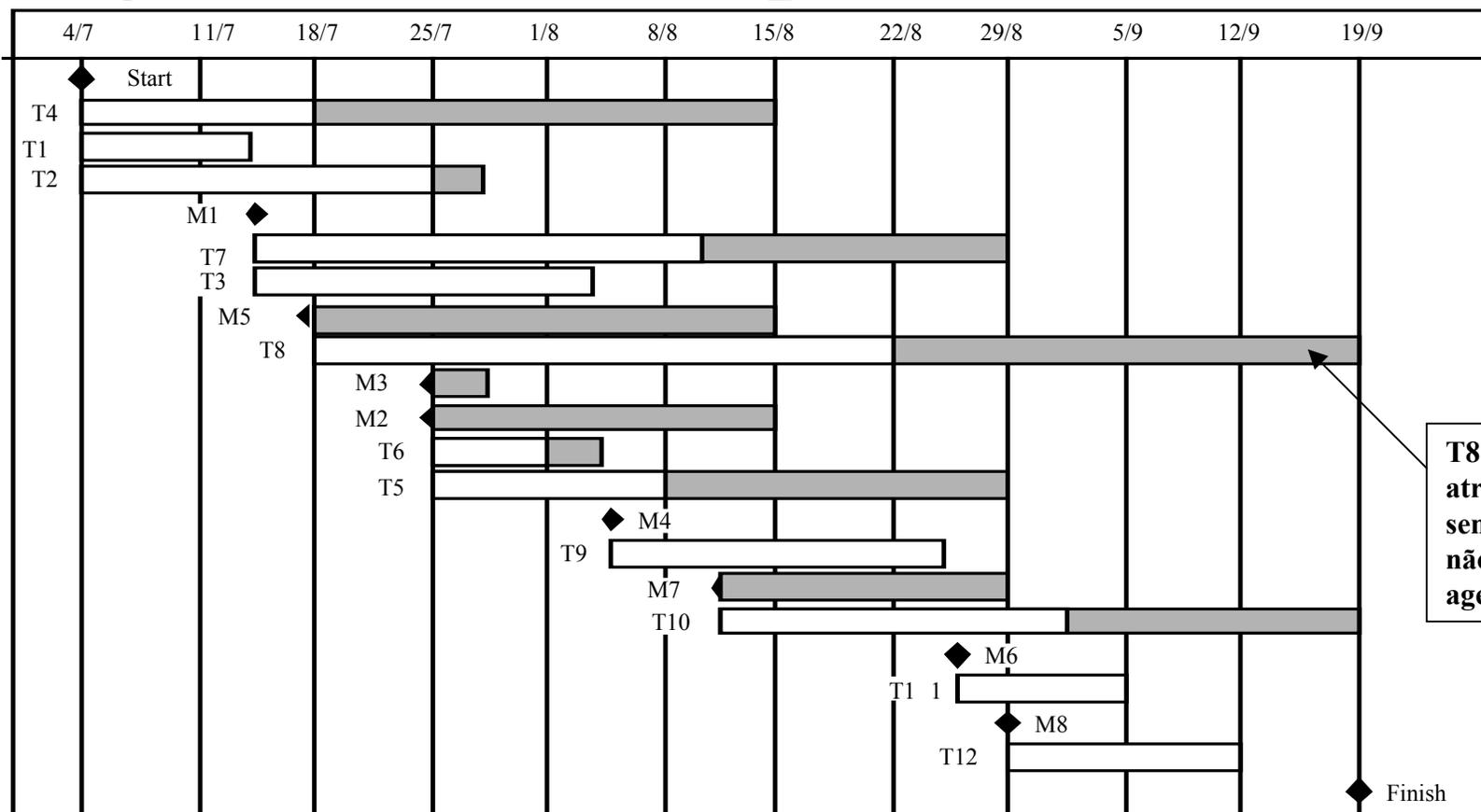
Por exemplo, T1 pode ser a concepção de um componente e T3 a sua implementação.



Rede de Actividades



Agendamento Temporal de Actividades



T8 pode atrasar-se 4 semanas que não afecta o agendamento

Barra Sombreada: Flexibilidade permitida

Barra sem Sombras: Actividades que constituem o caminho crítico, qualquer atraso implica atraso do projecto



Métodos e Técnicas

Aplicação ao planeamento do projecto

- Modelação de processos
- Organização de equipas
- Estimação de recursos e duração
- Definição de redes de actividades
- Definição de prazos



Organização de Equipas

- **Racional** - Eficiência é o seu objectivo, decisões racionais
 - A organização tem níveis hierarquicos definidos mas funciona democraticamente
 - Considera múltiplas equipas lideradas por engenheiros séniores
 - Cada equipa executa uma dada tarefa. Os engenheiros séniores reportam directamente ao seu supervisor
 - Aplica-se a projectos de grande dimensão
- **Burocrática** - Estabilidade é o seu objectivo, decisões rotineiras
 - Os membros da equipa reportam directamente ao seu supervisor que controla as suas tarefas e é responsável pela produtividade
 - Nota-se um défice de comunicação entre os membros da equipa
 - Funciona bem quando a tarefa é simples, bem conhecida, pode ser desempenhada pelos participantes e a importância de terminar o projecto ultrapassa outros factores (moral, desenvolvimento individual)



Organização de Equipas

- **Descentralizada** - Objectivos dispersos, decisões imprevisíveis
 - A responsabilidade encontra-se distribuída, não existindo níveis hierárquicos. O trabalho final provém de todo o grupo
 - Privilegia o consenso, satisfação e motivação da equipa
 - Reduz a produtividade individual
 - Mais adequado a projectos complexos e abertos, exigindo criatividade
 - Mais comunicação entre o grupo leva a projectos de longa duração. Não é apropriada a grandes equipas
- **Poder político** - Objectivos conflituosos, decisões desordenadas
 - Desaconselhável no contexto do desenvolvimento de software



Organização de Equipas

- Linhas de guia para a definição de equipas
 - Controlo descentralizado é melhor quando a comunicação entre membros da equipa é fundamental para o processo de desenvolvimento
 - Controlo centralizado é melhor quando o tempo é o factor fundamental no desenvolvimento do projecto
 - A organização deve limitar a comunicação entre os membros da equipa ao essencial, nem mais nem menos
 - A organização tem de considerar outros factores para além da produtividade: rotação de pessoal, desenvolvimento de cada individuo, disseminação de conhecimento, etc.



Métodos e Técnicas

Modelos de estimação de recursos e duração do projecto

- Comparação do projecto com outros projectos semelhantes, para os quais se possui informação sobre produtividade, duração e custos
 - **Método de Decomposição**
- Utilização de fórmulas predictivas, obtidas empiricamente, que permitem obter valores para as variáveis de estimação em função das características do software a desenvolver
 - **Método COCOMO** (1981)
 - **Método Putnam**



Método de Decomposição

- Aproximação do tipo “dividir e conquistar”
 - Decompor o projecto em funções principais
 - Calcular o esforço de cada função utilizando dados históricos sobre a produtividade
 - Utilizar métricas sobre a produtividade baseadas em:
 - **Linhas de código (LDC)** - Medida directa sobre o produto, focando na sua dimensão
 - **Pontos de função (PF)** - Medida indirecta sobre o produto. Em vez de medir directamente as características do software produzido, classifica-se este de acordo com critério que medem o seu grau de **funcionalidade e utilidade**
 - Fazer a média entre valores pessimista, esperado e optimista da produtividade



Linhas de Código

Vantagens da utilização de LDC

- LDC é um artefacto que existe em todos os processos de desenvolvimento de software
- LDC é facilmente calculável
- É fácil estabelecer honorários standard a partir desta métrica

Desvantagens

- LDC depende muito da linguagem de programação utilizada
- LDC penaliza os programadores mais organizados
- LDC penaliza a estruturação do código, assim como os programas muito pequenos
- A utilização de LDC em estimação requer um nível de detalhe difícil de obter a priori
- Falta de consensos (ex: em C, conta-se o nº de “{” ?)



Pontos de Função

- **Já sabemos que uma forma de modelar um sistema é utilizando diagramas de fluxos de dados (DFD's)**
 - mostram as várias transformações de dados
 - proporcionam uma rede das funções a serem executadas.
- **DeMarco desenvolveu um modelo de estimação, baseado em primitivas funcionais;**
- **Posteriormente foi desenvolvido um modelo de estimação de custos, baseado na contagem do número de estruturas de dados a ser utilizadas.**
- **Neste método assume-se que que o número de diferentes estruturas de dados é um bom indicador do tamanho.**
 - Trata-se dum método particularmente adaptado a projectos ligados a aplicações de negócio.
 - Menos bem adaptado em projectos onde as estruturas de dados desempenhem um papel menos preponderante, onde a ênfase seja nos algoritmos (p. ex. compiladores e aplicações em tempo real).



Pontos de Função

- Baseia-se numa pontuação da funcionalidade do software
- Deriva-se a partir de uma relação empírica entre valores mensuráveis do software e previsões sobre o seu grau de complexidade
- Valores mensuráveis:
 - **Entradas** - Cada tipo de dados destinado à aplicação e introduzido pelo utilizador no sistema
 - **Saídas** - dados/ecrãs/erros fornecidos ao utilizador pela aplicação (itens individuais não são contados separadamente)
 - **Queries** - Conjuntos indissociáveis de interacções (entradas + saídas) distintas das anteriores
 - **Ficheiros** - Agrupamentos lógicos de dados (não necessariamente ficheiros) mantidos pela aplicação
 - **Interfaces** - Trocas de informação com outros sistemas (um ficheiro vindo de outro sistema conta igualmente como interface)



Pontos de Função

Tabela de cálculo

Parâmetro	Nº	Pesos			Sub-Total
		Simples	Médio	Complexo	
Entradas		3	4	6	
Saídas		4	5	7	
Queries		3	4	6	
Ficheiros		7	10	15	
Interfaces		5	7	10	
					TOTAL

Nota: Os pesos são seleccionados empiricamente, a partir de dados históricos

Pontos de Função

$$FP = \text{total-tabela} * (0.65 + 0.01 * \text{SUM}(F_i)) \quad i=1-14$$

(F_i = ajuste de complexidade, variando entre 0 (sem influência) e 5 (essencial))

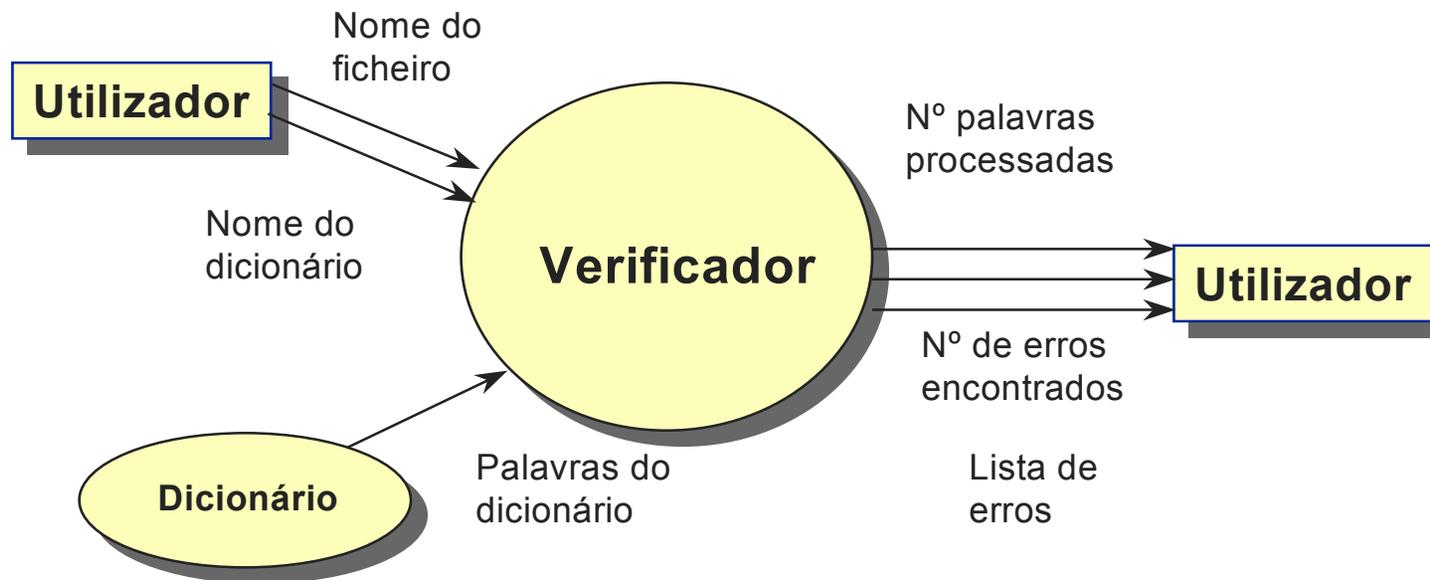
- 1 - Precisa de sistema de *backup/recovery* fiável?
- 2 - Requer comunicação de dados?
- 3 - Existem funções de processamento distribuído?
- 4 - O desempenho é crítico?
- 5 - Executa-se num ambiente pesado?
- 6 - É interactivo?
- 7 - Entrada de dados requer múltiplas janelas?
- 8 - Actualiza ficheiros de dados em tempo real?
- 9 - Entradas/saídas/queries/ficheiros complexos?
- 10 - Complexidade do processamento interno?
- 11 - Código deve ser reutilizado?
- 12 - Produto deve incluir a instalação?
- 13 - Código deve ser desenhado para múltiplas instalações?
- 14 - Código deve ser desenhado para facilitar utilização?

- **O ajuste de complexidade leva em conta características da aplicação, que influenciam o esforço de desenvolvimento.**



Exemplo

Verificador léxico



- Nº de entradas = 2 (nome do ficheiro + dicionário)
- Nº de saídas = 3 (nº palavras, nº erros, lista de erros)
- Nº de queries = 2 (nome do documento + dicionário)
- Nº de ficheiros = 1 (dicionário)
- Nº interfaces = 1 (dicionário)



Exemplo

		Pesos				
Parâmetro	Nº	Simple	Médio	Complexo	Sub-Total	
Entradas	2	3	4	6	6	
Saídas	3	4	5	7	12	
Queries	2	3	4	6	8	
Ficheiros	1	7	10	15	10	
Interfaces	1	5	7	10	0	43
						TOTAL

Exemplo

1 - Backup/recovery	5
2 - Comunicação dados	0
3 - Proces. distribuído	0
4 - Desempenho crítico	1
5 - Ambiente pesado	4
6 - Interactivo	2
7 - Janelas	0
8 - Acesso a ficheiros	1
9 - Complexidade entradas/...	5
10 - Complexidade process.	4
11 - Reutilização	2
12 - Instalação	3
13 - Múltiplas instalações	4
14 - Facilidade utilização	3

0 - Sem influência

5 - Essencial

$$\text{SUM}(F_i) = 34$$

$$\text{PF} = 43 * (0.65 + 0.01 * 34) = 42.57$$

Se 1 PF = 2 pessoas-dia

Então o programa leva 85 dias a ser desenvolvido



Pontos de Função

Versão 2 - Mais adequada a programas com complexidade algorítmica (versus manipulação de dados)

Parâmetro	Nº	Peso	Sub-Total
Entradas		4	
Saídas		5	
Queries		4	
Ficheiros		7	
Interfaces		7	
Algoritmos		3	
			TOTAL

Pontos de Função

Vantagens da utilização de PF

- Estas métricas não dependem da linguagem de programação
- Podem ser aplicadas a programas de grande complexidade
- Representam a funcionalidade ou utilidade do software
- Baseiam-se em dados que podem ser conhecidos no início do projecto

Desvantagens

- Baseiam-se em dados subjectivos
- Os dados são difíceis de obter a posteriori
- PF é apenas um número, sem qualquer realidade física



Relação entre LDC e PF

A relação entre LDC e PF depende da linguagem de programação utilizada:

	LDC/PF
– Assembly	300
– C	150
– Cobol	100
– Fortran	100
– Pascal	90
– ADA	70
– APL	32
– O-O	30
– Smalltalk	20
– Ger. de cód.	15
– Spreadsheet	6

1 LDC de ADA oferece
1.4 vezes mais
funcionalidade que 1 LDC
de Fortran



Método de Decomposição

Estimar esforço do projecto

- Examinar o contexto do software e extrair as suas funções principais
- Estimar para cada função os valores LDC e PF
- Determinar valores de produtividade adequados a cada função
- Atribuir a cada função uma estimação do esforço exigido para três situações: optimista, mais provável e pessimista
- Valor esperado do esforço = $(a + 4 * m + b) / 6$
(a - optimista, m - mais provável, b - pessimista)
- Adicionar os valores obtidos para cada função

Proje cto	Opti mista	+ prová vel	Pessi mista	Espe rado	\$/LDC	Linha/ mês	Custo	Meses
T1								
T2								
...								

Método de Decomposição

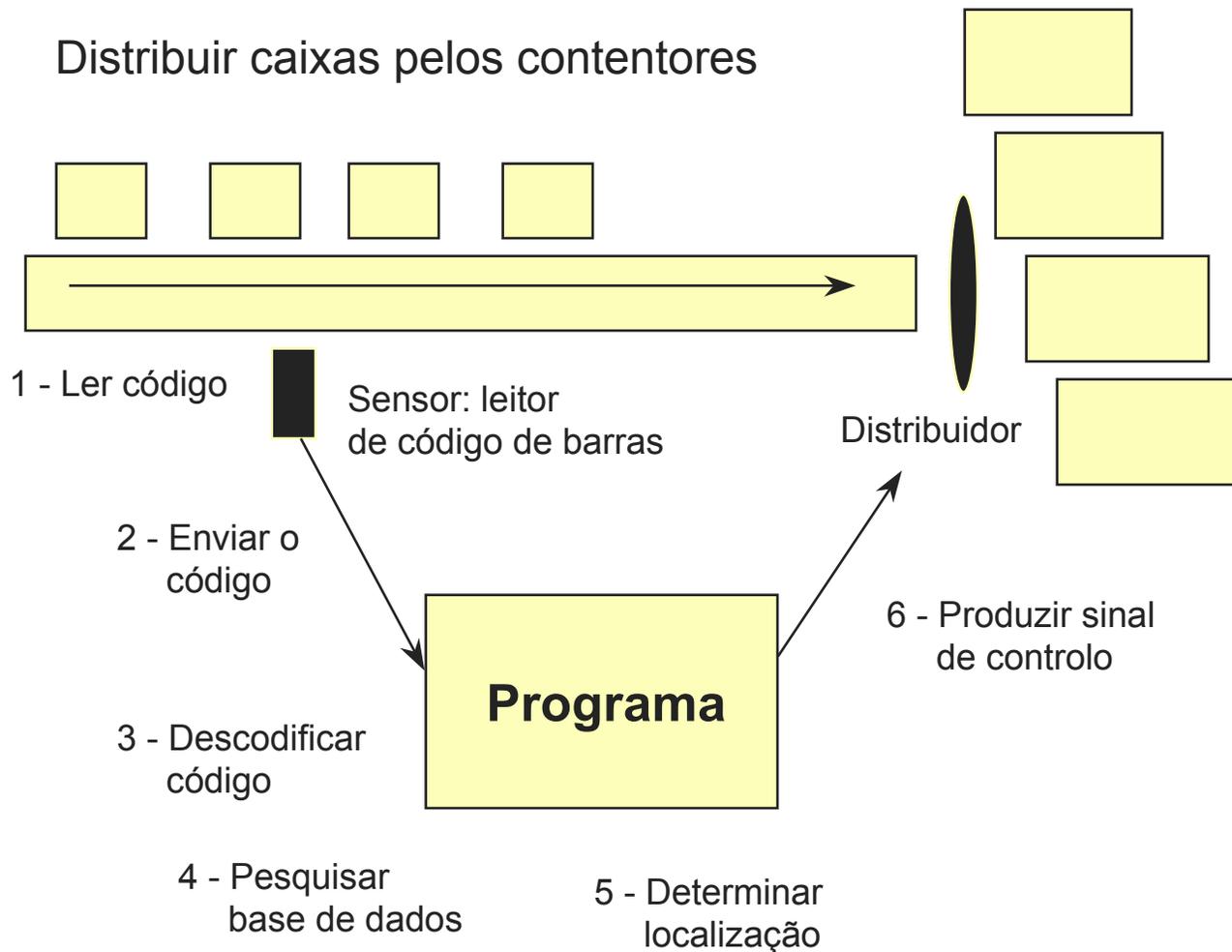
Quando não existem valores históricos da produtividade

- Examinar o contexto do software e extrair as suas funções principais
- Define-se para cada função as etapas necessárias à sua realização: análise, desenho, codificação e teste
- Estimar para cada actividade um esforço necessário (em pessoas-mês)
- Atribuir um custo de mão-de-obra a cada etapa e calcular os valores totais

Funções	Análise requisitos	Desenho	Codificar	Testar	Total
F1					
F2					
F3					
Total					

Exemplo

Distribuir caixas pelos contentores



Recursos:
Sensor
Distribuidor
Base de dados
MS Access
MS Developer Studio

Restrições:
Velocidade das caixas
Velocidade do sensor
Robustez

Interfaces:
Leitor de códigos
Distribuidor
Base de dados



Exemplo

Funções

- F1 - Descodificar código
- F2 - Pesquisar base de dados
- F3 - Determinar localização
- F4 - Produzir sinal de controlo

Estimar individualmente os valores LDC e PF

- LDC: F1=100 F2=200 F3=50 F4=500 TOT=850
- PF \approx LDC / 30 (para código C++)
- Produtividade = 25 LDC/pessoa-dia (valor histórico)
- Esforço = $850 / 25 = 34$ pessoas-dia
- 1 pessoa-hora = 10 KPTE
- Custo = $34 * 10 * 8 = 2720$ KPTE



Exemplo

Estimação pessimista

- Produtividade = 15 LDC/pessoa-dia (valor histórico)
- Esforço = $850 / 15 = 57$ pessoas-dia
- Custo = $57 * 10 * 8 = 4560$ KPTE

Estimação otimista

- Produtividade = 50 LDC/pessoa-dia (valor histórico)
- Esforço = $850 / 50 = 17$ pessoas-dia
- Custo = $17 * 10 * 8 = 1360$ KPTE

Média ponderada

- $E = (a + 4m + b) / 6 = (57 + 4 * 34 + 17) / 6 = 35$ pessoas-dia
(a - otimista, m - mais provável, b - pessimista)



Método de Decomposição

Comparação entre LDC/PF e estimação

- Se se possuírem valores históricos devem-se utilizar os dois métodos
- As diferenças de resultados permitem aferir a precisão da estimação
- Se o desvio for $< 10\%$ a estimação é boa
- Se for $> 10\%$ então:
 - O contexto e objectivos do projecto não foram bem definidos
 - Os dados utilizados na produtividade são inapropriados ou obsoletos
 - Torna-se necessário determinar as causas do problema



Método COCOMO

COCOMO - Constructive Cost Model

Utiliza uma fórmula matemática baseada nas LDC

- **Modelo 1. COCOMO Básico**
 - Determina o esforço (e o custo) em função da dimensão do software (LDC)
- **Modelo 2. COCOMO Intermédio**
 - Determina o esforço (e o custo) em função da dimensão do software (LDC) e de um conjunto de factores subjectivos (produto, hardware, pessoas, atributos do projecto)
- **Modelo 3. COCOMO Avançado**
 - COCOMO Intermédio + classificação dos factores subjectivos em função das etapas do processo (análise, desenho, etc.)



Método COCOMO

Os métodos COCOMO categorizam o software desenvolvido nos modos

- **Orgânico** - Pequenos e simples, com pequenas equipas
- **Semi-independente** - Complexidade e dimensão médias, equipas com conhecimentos diferenciados (e.g. sistema distribuído)
- **Embebido** - Projectos com requisitos muito bem definidos e restrições muito apertadas (e.g. controlo de voo), equipas multidisciplinares



Método COCOMO

Modos de desenvolvimento

Funcionalidade	Modo		
	Orgânico	Semi-destacado	Embebido
Compreensão dos objectivos do produto	Alargado	Considerável	Geral
Experiência com produtos semelhantes	Extensivo	Considerável	Moderado
Necessidade de conformidade com requisitos pré-estabelecidos	Básico	Considerável	Total
Necessidade de conformidade com interfaces externas	Básico	Considerável	Total
Desenvolvimento concorrente de hardware e software		Moderado	Extensivo
Necessidade de estruturas de dados ou algoritmos inovadores	Mínimo	Algum	Considerável
Interesse em terminar o produto cedo	Reduzido	Médio	Elevado
Dimensão do produto	< 50 KLDC	< 300 KLDC	Qualquer



COCOMO Básico

$$E = a * KLDC^b \quad (\text{em pessoas-mês})$$

$$D = c * E^d \quad (\text{em meses de desenvolvimento})$$

Projecto	a	b	c	d
Orgânico	2.4	1.05	2.5	0.38
Semi-ind.	3	1.12	2.5	0.35
Embebido	3.6	1.2	2.5	0.32



COCOMO Intermédio

$$E = a * KLDC^b * EAF$$

EAF - Factor de ajustamento obtido a partir de 15 atributos

$$EAF = Me1 \times Me2 \times Me3 \times \dots \times Me15$$

Me - Multiplicador de esforço, retirado da tabela seguinte

Projecto	a	b
Orgânico	3.2	1.05
Semi-ind.	3	1.12
Embebido	2.8	1.2



COCOMO Intermédio

Multiplicadores de esforço

	Classificação				
	Muito reduzido	Reduzido	Nominal	Elevado	Muito elevado
Atributos do produto					
Fiabilidade do software necessária	.75	.88	1.0	1.15	1.4
Dimensão da base de dados		.94	1.0	1.08	1.16
Complexidade do produto	.70	.85	1.0	1.15	1.3
Atributos do computador					
Restrições ao tempo de execução			1.0	1.11	1.3
Restrições ao armazenamento de dados			1.0	1.06	1.21
Volatilidade da máquina virtual		.87	1.0	1.15	1.3
Tempo disponível do computador		.87	1.0	1.07	1.15
Atributos dos indivíduos					
Capacidade dos analistas	1.46	1.19	1.0	.86	.71
Experiência no desenv. De aplicações	1.29	1.13	1.0	.91	.82
Capacidade de programação	1.42	1.17	1.0	.86	.70
Experiência com a máquina virtual	1.21	1.1	1.0	.90	
Experiência com a linguagem de prog.	1.14	1.07	1.0	.95	
Atributos do projecto					
Uso de práticas modernas de programação	1.24	1.1	1.0	.91	.82
Uso de ferramentas de software	1.24	1.1	1.0	.91	.83
Prazos de desenvolvimento	1.23	1.08	1.0	1.04	1.1



Método COCOMO

- O método permite analisar a sensibilidade do projecto a determinados factores (e.g. variar a experiência dos programadores)
- O método avançado apresenta resultados semelhantes ao método intermédio, pelo que este último deve ser utilizado preferencialmente
- O método básico deve ser utilizados apenas para estimativas preliminares
- Não é espectável que este tipo de modelo apresente estimativas precisas
- Resultados com variações $<20\%$ na estimativa dos custos e $<30\%$ na estimativa da duração são considerados aceitáveis



Exemplo

Distribuição de caixas pelos contentores

- COCOMO Básico
- Projecto orgânico
- 850 LDC

$$E = 2.4 * 0.85^{1.05} = 2.02 \text{ pessoas-mês}$$

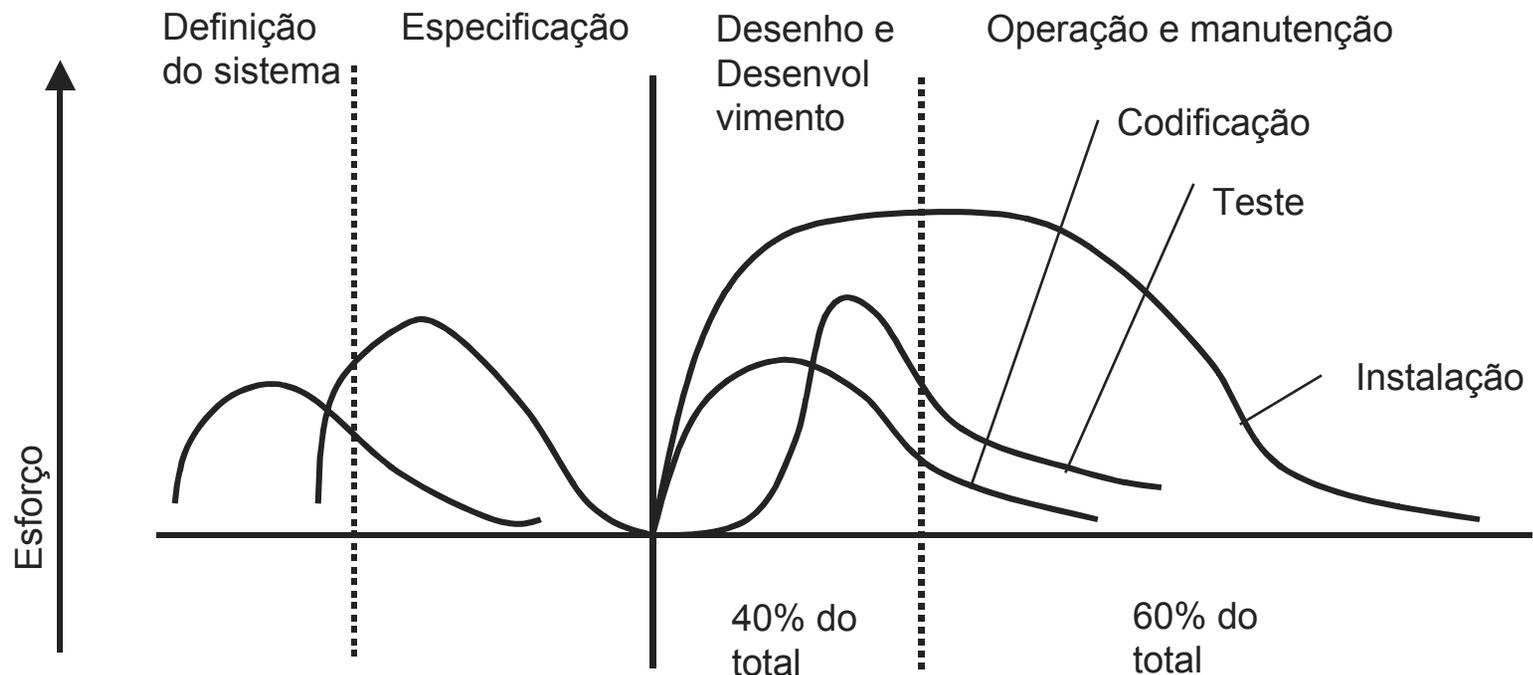
$$D = 2.5 * 2.02^{0.38} = 3.26 \text{ meses}$$

- Número de pessoas
 - $N = E / D = 0.6$ pessoas (leia-se: ~ dois terços do trabalho de uma pessoa)



Método Putnam

- Assume uma distribuição do esforço em função da duração de um projecto
- Foi derivado de projectos > 30 pessoas-ano



Método Putnam

Relaciona linhas de código com o **esforço e tempo** necessário para desenvolver o software

$$L = Ck * K^{1/3} t^{4/3}$$

- L - nº linhas de código fonte
- K - esforço (pessoas-ano)
- t - tempo de duração do projecto (em anos)
- Ck - constante que caracteriza o estado de amadurecimento do ambiente de desenvolvimento (2000 < Ck < 11000; 2000 - fraca metodologia, pouca documentação; 8000 - bom ambiente de desenvolvimento, boa documentação; 11000 - ambiente excelente, com ferramentas de geração automática de código)



Método Putnam

Esforço de desenvolvimento de código

$$K = L^3 / (Ck^3 t^4) \quad (\text{pessoas-ano})$$

- Note-se que o modelo apresenta uma relação não linear entre o esforço e a duração do projecto, pelo que um pequeno aumento do tempo de duração permite reduzir significativamente o esforço



Exemplo

Distribuição de caixas pelos contentores

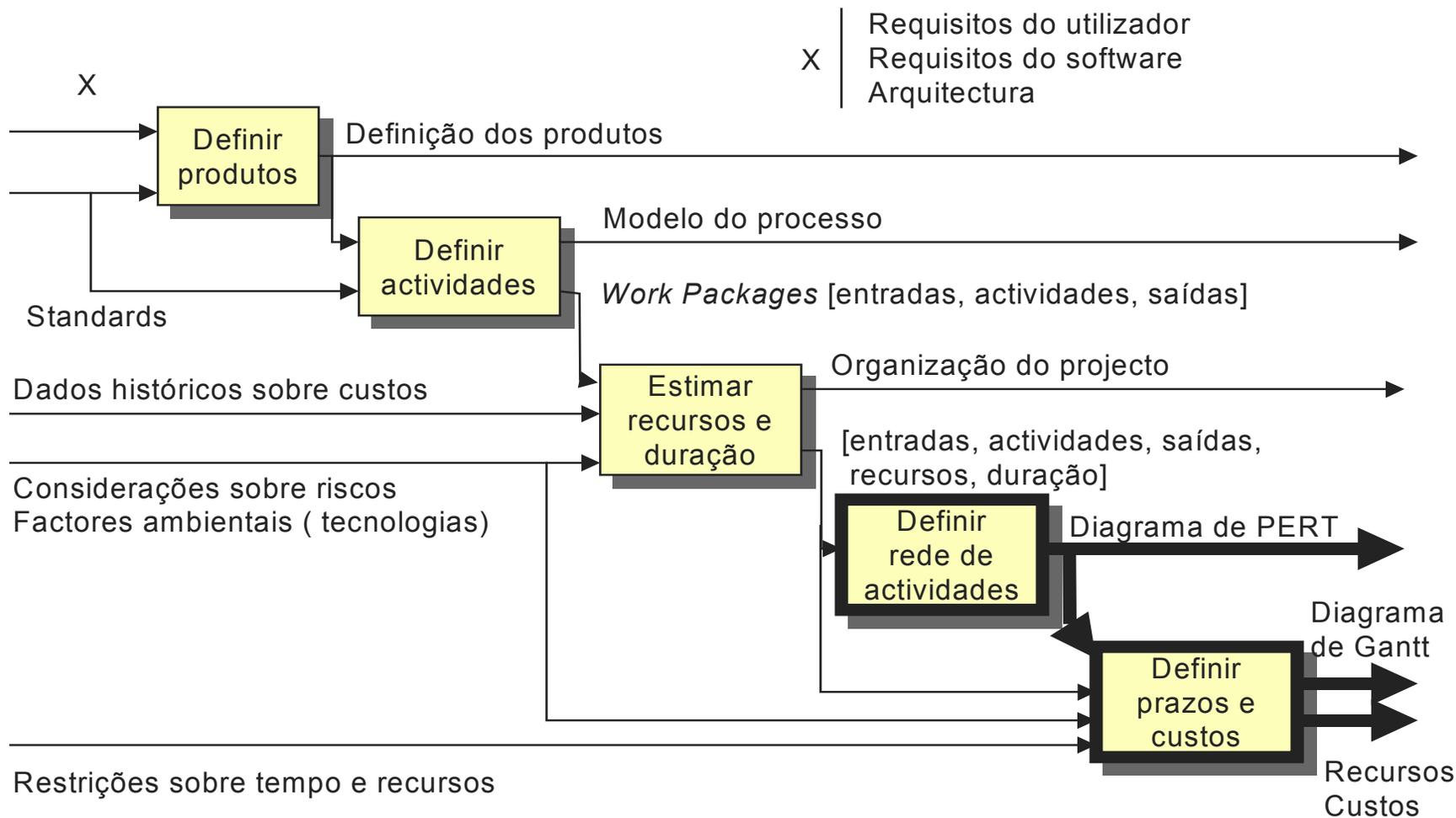
- 850 LDC
- $C_k = 8000$ (bom ambiente)
- $t = 0.25$ (3 meses)

$$K = 850^3 / (8000^3 * 0.25^4) = 0.3 \text{ pessoas-ano}$$

- 1 pessoa-ano = 11 pessoas-mês
- $K = 3.3$ pessoas-mês



Métodos e Técnicas



Métodos e Técnicas

Definição de redes de actividades

- Gráfico de **PERT** (Program Evaluation and Review Technique)
 - Rede de caixas e setas
 - Cada caixa representa uma actividade
 - Cada seta representa uma dependência entre duas actividades
 - Uma actividade dependente não pode começar antes da anterior terminar
 - **Metas importantes** (*milestones*) - Actividade cuja conclusão representa um ponto importante na vida do projecto
 - Requer
 - Lista de todas as actividades do projecto
 - Estimativa da duração de cada actividade



Exemplo

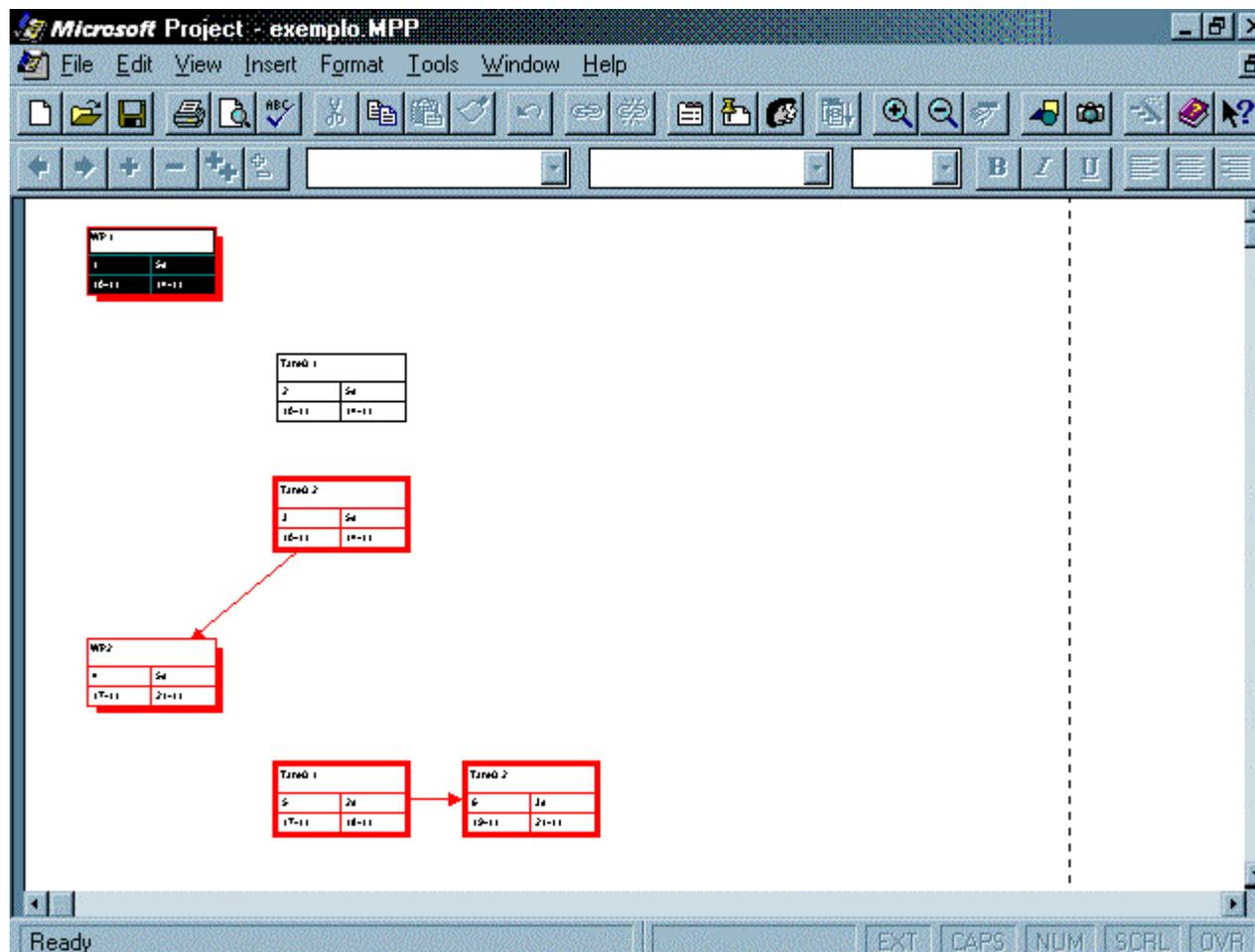


Gráfico de PERT

Vantagens

- Força o planeamento
- Mostra as interrelações entre tarefas do projecto
- Identifica o caminho crítico
- Mostra possíveis paralelismos entre actividades
- Permite simular alternativas
- Permite o control do projecto



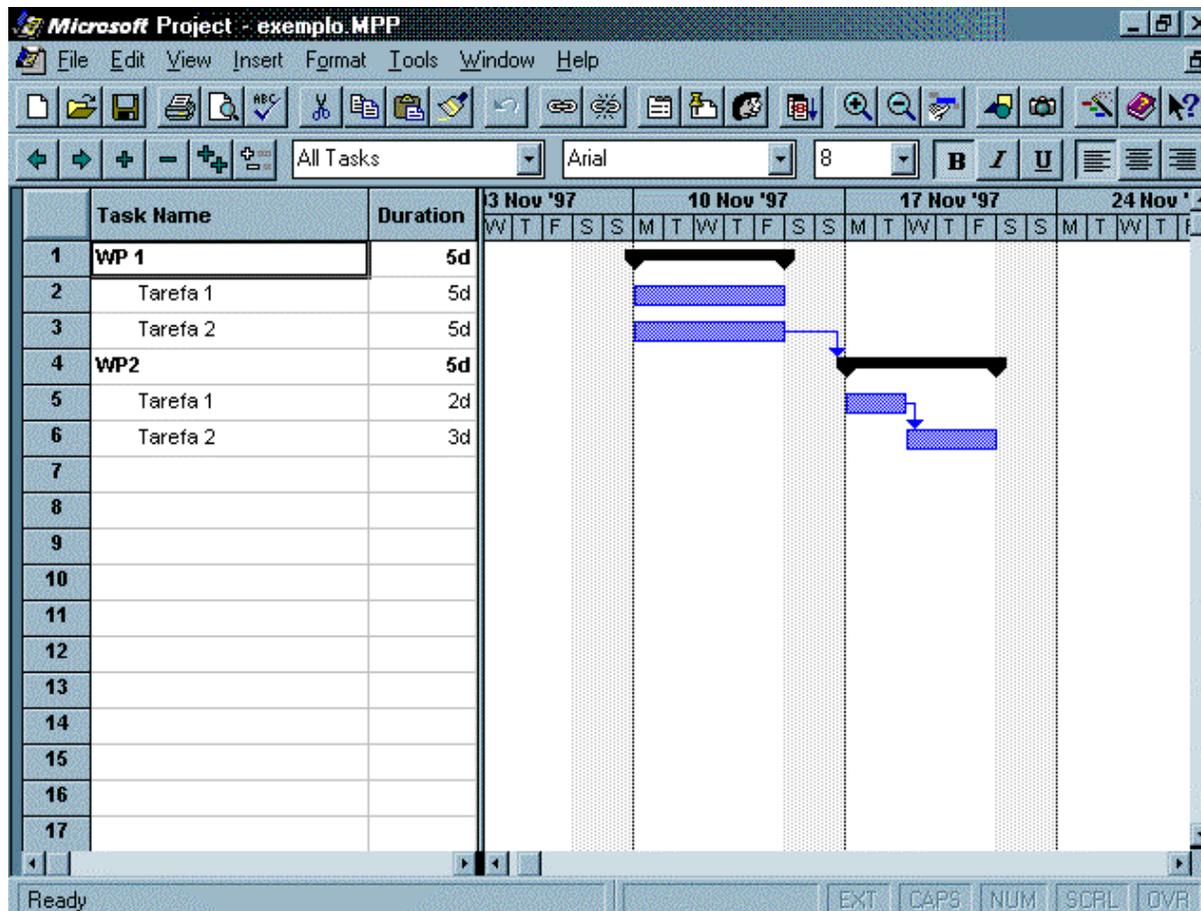
Métodos e Técnicas

Definição de prazos

- Gráfico de Gantt
 - Gráfico de barras onde cada barra representa uma actividade
 - Pode ser utilizado em múltiplas actividades:
 - Definir prazos
 - Definir custos
 - Planear utilização de recursos
 - Definir flutuações nas actividades



Exemplo



Definição de Prazos

- **Utilização do gráfico de Gantt**
 - **Graus de liberdade: custo (esforço) e duração**
 - **A duração é normalmente mais importante do que o custo**
 - **A relação entre o esforço e duração não é linear**

Formula de Putnam $K = L^3 / (C^3 * t^4)$

$L = 10\ 000$ LDC

$C = 8\ 000$

para $t = 1$ ano

- $K = 10\ 000^3 / 8\ 000^3 = 1.95$ pessoas-ano

para $t = 1.5$ anos

- $K = 10\ 000^3 / 8\ 000^3 * 1.5^4 = 0.38$ pessoas-ano

Conclusão: estender o projecto em 6 meses, reduz o esforço em 80%.



Definição de Prazos

- A relação entre a dimensão da equipa e a produtividade é não linear (*mythical man-month conflict*)
 - Existe um custo associado à comunicação entre os membros da equipa
 - 1 pessoa produz 5 000 LDC/ano individualmente
 - Custo de comunicação = 250 LDC/ano
 - Grupo de 4 pessoas = 6 canais de comunicação
 - Produtividade do grupo = LDC/ano grupo - custo comunicação
 - $P = 4 * 5\ 000 - (250 * 6) = 18\ 500$ LDC/ano (7.5% menos)
 - Adicionar 2 pessoas 2 meses antes do fim do ano:
 - Grupo de 6 pessoas = 14 canais de comunicação
 - $P = 4 * 5\ 000 - (250 * 6 / 12 * 10) + (5\ 000 / 12 * 2) - (250 * 14 / 12 * 2) = 19\ 000$ LOC/ano



Distribuição de Trabalho no Tempo

- **Acabamos de ver que um pequeno alongamento de tempo de execução, implica uma grande redução de esforço.**
- **Devido a:**
 - sobrecarga de comunicação
 - não há um crescimento linear do número de canais de comunicação e o número de pessoas do grupo:
 - 4 - 6 canais
 - 5 - 10 canais
 - 6 - 14 canais
 - ao adicionar mais pessoas a um grupo durante a execução do projecto, leva a que a produtividade decresça a princípio:
 - no início, os novos membros da equipa não são produtivos;
 - requerem atenção e tempo por parte dos outros membros, durante o processo de aprendizagem.
- **Leva à lei de Brooks: acrescentar pessoas a um projecto atrasado, só o vai atrasar mais**



Distribuição de Trabalho no Tempo

- Apesar da produtividade individual decrescer, a adição de mais elementos a uma equipa (mais manpower) permite aumentar a produtividade total;
- mas o crescimento da produtividade total vai crescendo à medida que o nº de elementos da equipa aumenta, atingindo um máximo (valor de produtividade total óptima), decrescendo seguidamente.

Supõe-se uma produtividade de 500 LDC/pessoa-mês e quebra de produtividade de 10% por canal de comunicação

Tamanho da Equipa	Produtividade Individual	Produtividade Total
1	500	500
2	450	900
3	400	1200
4	350	1400
5	300	1500
5.5	275	1512
6	250	1500
7	200	1400
8	150	1200

A produtividade total máxima atinge-se para equipa com 5.5 pessoas

Tudo leva o seu tempo:
não podemos encurtar indefinidamente substituindo tempo por pessoas



Definição de Prazos

Distribuição do esforço pelos pacotes de actividades

- Utilizar a regra 40-20-40
 - 40 - 50 % - Análise e desenho
 - 15 - 20 % - Codificação
 - 30 - 40 % - Teste

(O planeamento do projecto não deve implicar um esforço superior a 2-3 %)

