



UFOP

Universidade Federal  
de Ouro Preto

Universidade Federal de Ouro Preto (UFOP)  
CSI 403 - Engenharia de Software

# PLANEJAMENTO DE PROJETOS

Discentes: Carlos Vitor, Luccas Vinicius, João Guilherme,  
Mateus Henrique, Emanuelle Ferraz, Jonas Mota



- Precificação de Software
- Desenvolvimento dirigido por plano
- Definição do cronograma de projeto
- Planejamento ágil
- Técnicas de estimativa
- Modelagem de custos COCOMO

# Introdução

---



- O Planejamento de projeto é um dos trabalhos mais importantes de um gerente de projeto de software.
- O gerente deve dividir o trabalho em partes e designa-las a membros do time de projeto, antecipar os problemas que poderiam surgir e preparar soluções possíveis para esses problemas.
- O plano de projeto é atualizado à medida que o projeto avança é utilizado para definir como o trabalho será feito e avaliar o progresso do projeto



## Contrato

- Estágio de proposta, onde se disputa um contrato para desenvolver ou fornecer um sistema de software.

## Início do Projeto

- Planejamento dos recursos que serão alocados naquele projeto. Aqui existem mais informações do que no estágio da proposta e possibilita um refinamento das estimativas de esforço.

## Novos Requisitos

- Conforme os requisitos do software mudarem, a divisão de trabalho tem de ser alterada, e o cronograma precisa ser estendido.



- Planejamento inevitavelmente especulativo, pois não existe um conjunto completo de requisitos do software a ser desenvolvido.
- Deve se produzir um plano viável para a realização do trabalho e um plano de projeto é uma parte necessária de uma proposta bem sucedida.
- Ao ganhar o contrato, deve se então planejar o projeto novamente, levando em consideração as mudanças e as novas informações sobre o sistema.
- Ao disputar um contrato é preciso definir o preço que será proposto ao cliente, que pode ser resumido como uma estimativa dos custos de esforço necessário para cada atividade.



01

# Precificação de Software





- Quando se estima os custos do esforço de um projeto, é preciso levar em conta os custos gerais da organização, salário, espaço de escritório, custos de hardware e software, manutenção, custos de viagem e treinamento entre outros.
- O custo de esforço, na maioria dos projetos, sempre é o de maior custo. Deve se estimar o esforço total(em pessoas-mês) que sera necessario para o projeto.
- A principio, o preço de um sistema de software desenvolvido é uma soma do custo mais o lucro do desenvolvedor, mas na prática são levados em conta vários aspectos organizacionais, politicos, econômicos e comerciais mais amplos.





**FIGURA 23.1** Fatores que afetam a precificação do software.

Fator	Descrição
Termos contratuais	Um cliente pode estar disposto a deixar que o desenvolvedor mantenha a propriedade do código-fonte e o reuse em outros projetos. O preço cobrado poderia então ser reduzido para refletir o valor do código-fonte para o desenvolvedor.
Incerteza da estimativa de custos	Se uma organização não tiver certeza de sua estimativa de custo, ela pode aumentar o seu preço por uma contingência além do seu lucro normal.
Saúde financeira	As empresas com problemas financeiros podem baixar seu preço para ganhar um contrato. É melhor ter um lucro abaixo do normal ou ficar no zero a zero do que falir. O fluxo de caixa é mais importante do que o lucro em períodos econômicos difíceis.
Oportunidade de mercado	Uma organização de desenvolvimento pode cotar um preço baixo porque deseja entrar para um novo segmento do mercado de software. Aceitar um lucro baixo em um projeto pode dar à organização a oportunidade de ter um lucro maior no futuro. A experiência adquirida também pode ajudar a empresa a desenvolver novos projetos.
Volatilidade de requisitos	Se os requisitos forem propensos a mudar, uma organização pode reduzir o seu preço para ganhar um contrato. Após conquistar o contrato, podem ser cobrados preços elevados pelas mudanças nos requisitos.





02

# Desenvolvimento dirigido por plano



# O que é desenvolvimento dirigido por plano?

---



- Um projeto **desenvolvido por plano** é planejado com detalhes.
- Registra o trabalho a ser feito, quem deve fazê-lo, cronograma de desenvolvimento e os produtos do trabalho.
- Usado para tomar **decisões de projeto** e métrica de progresso.
- Considerado uma forma **tradicional** de dirigir projetos.



# Vantagens e desvantagens

---



- A **desvantagem** é que as decisões iniciais devem ser revisadas mediante ao ambiente de desenvolvimento.
- A **vantagem** é que o planejamento inicial facilita a organização, identificação de problemas potenciais logo no início do desenvolvimento.



# O que constitui os planos de projeto ?

---



- Um bom plano de projeto deve estabelecer os **recursos disponíveis**, divisão do **trabalho, cronograma** de desenvolvimento.
- Além de estabelecer uma **abordagem de gerenciamento de riscos**, e identificação dos possíveis **riscos**.



# O que constitui os planos de projeto ?

---



→ Detalhes normalmente abordados nos planos de projeto:

- Introdução.
- Organização do projeto.
- Análise de riscos.
- Requisitos de recursos de hardware e software.
- Divisão do trabalho.
- Cronograma de projeto.
- Mecanismos de monitoramento e de divulgação





# Planos suplementáres

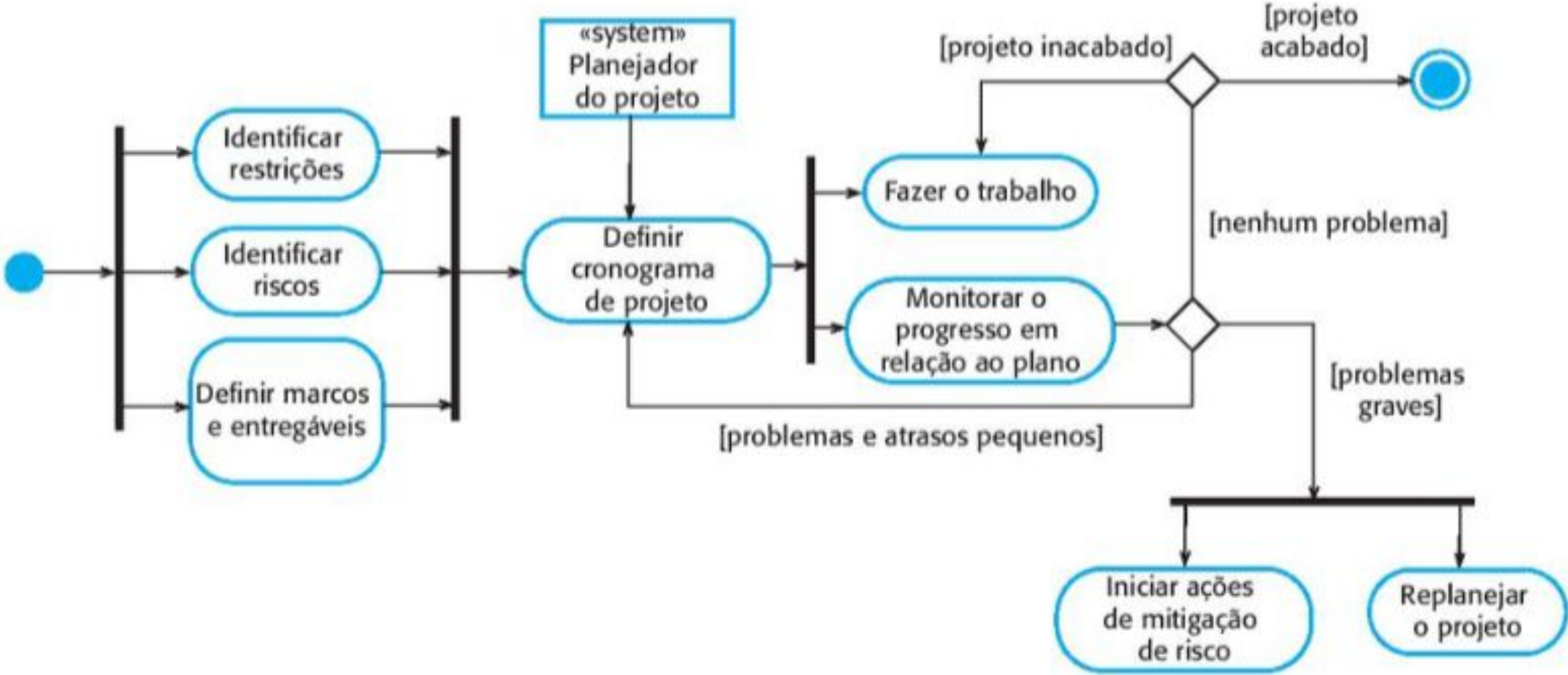


→ Além dos planos principais, é possível desenvolver **planos suplementares** como testes e gerenciamento de configuração.

Plano	Descrição
Plano de gerenciamento de configuração	Descreve os procedimentos de gerenciamento de configuração e as estruturas a serem utilizadas.
Plano de implantação	Descreve como o software e o hardware associado (se necessário) serão implantados no ambiente do cliente. Esse plano deve incluir um plano para migrar os dados dos sistemas existentes.
Plano de manutenção	Prevê os requisitos, os custos e o esforço de manutenção.
Plano de qualidade	Descreve os procedimentos e os padrões de qualidade que serão utilizados no projeto.
Plano de validação	Descreve as abordagens, os recursos e o cronograma utilizados na validação do sistema.

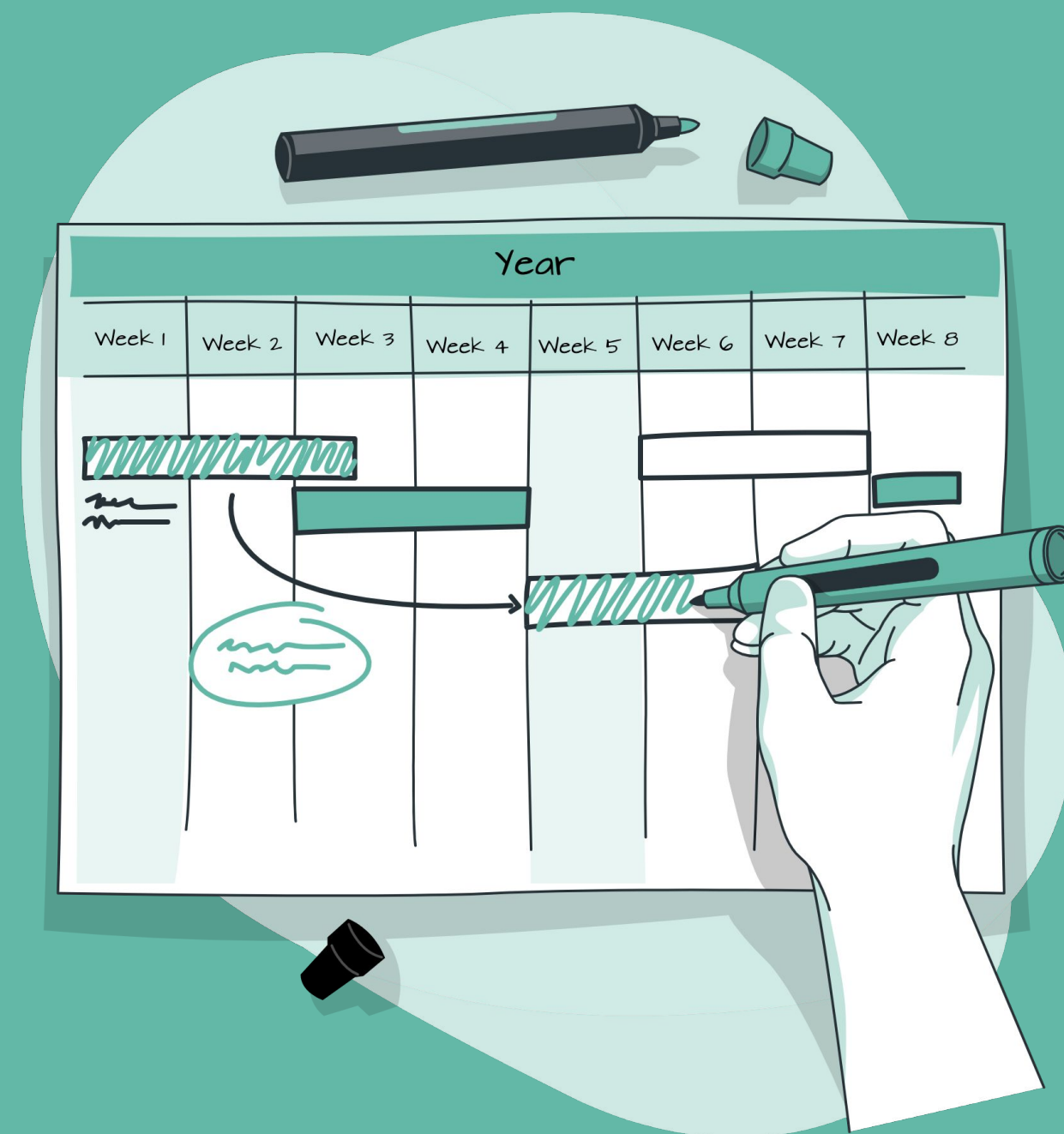


# Processo de planejamento



03

# Definição do Cronograma de Projeto



# O que é cronograma de projeto?

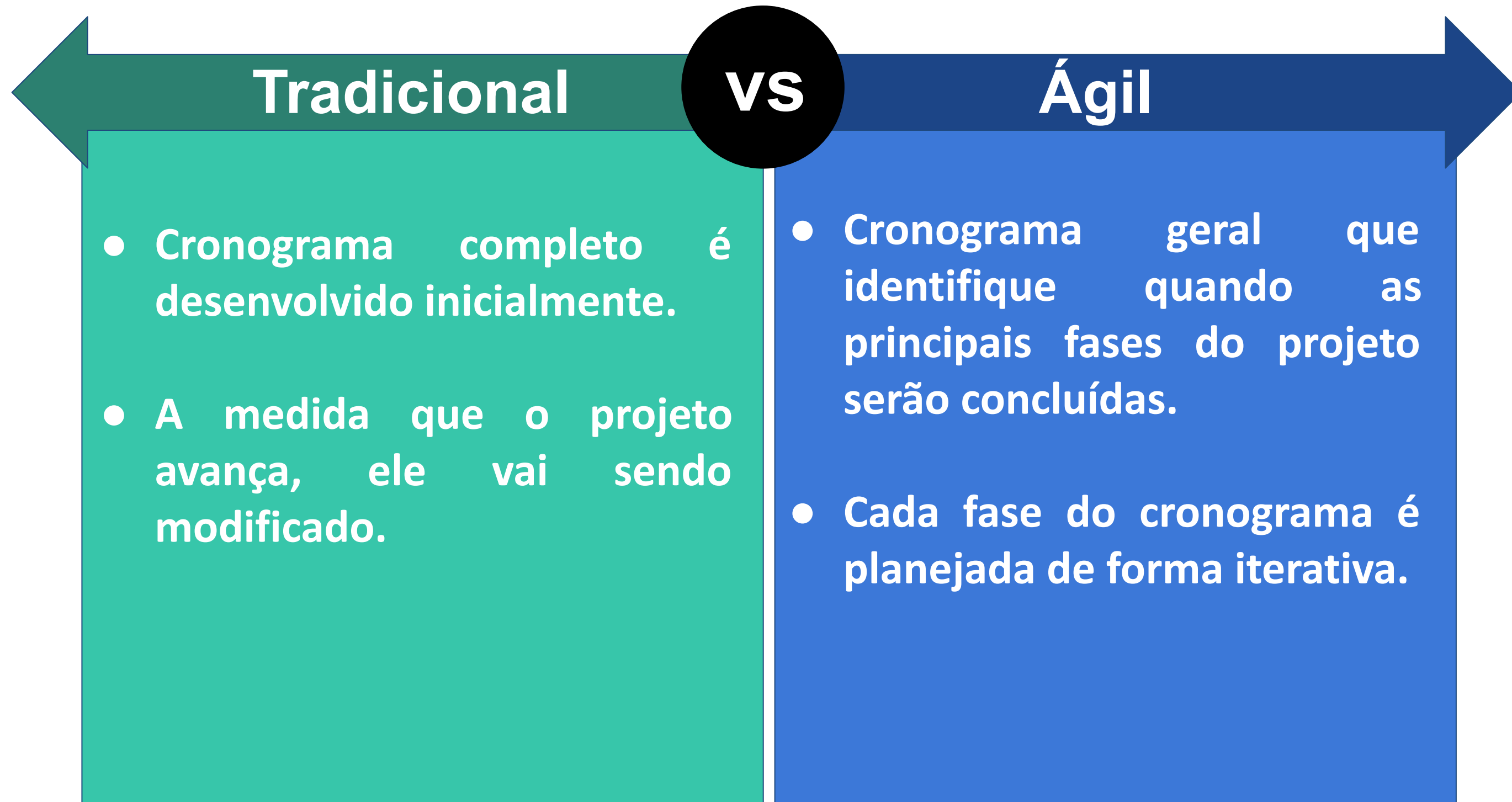


- Organização de um projeto: **quando** e **de que maneira** as tarefas serão executadas.
- Estimação de **prazos** e **esforços**.
- Estimação de **recursos de hardware e software**.
- Alocação de pessoas nas tarefas identificadas.





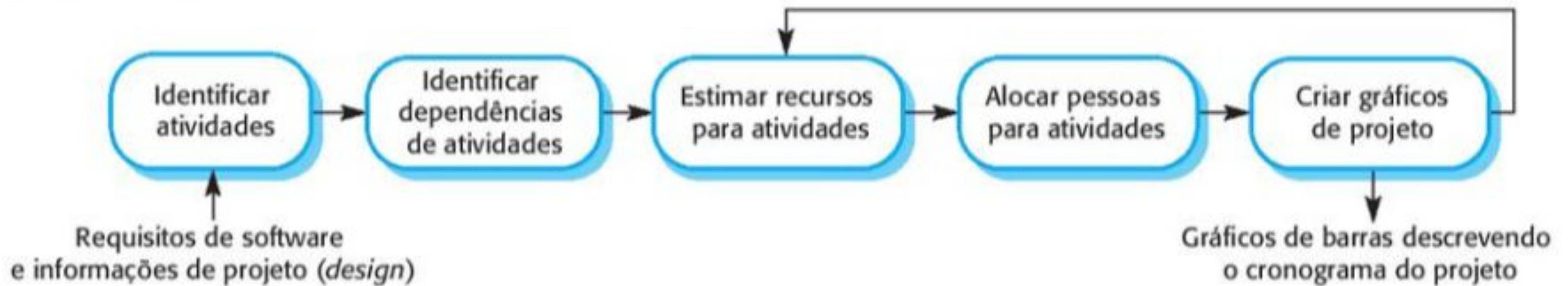
# Processos Tradicionais x Processos Ágil







**FIGURA 23.4** Processo de definição do cronograma do projeto.

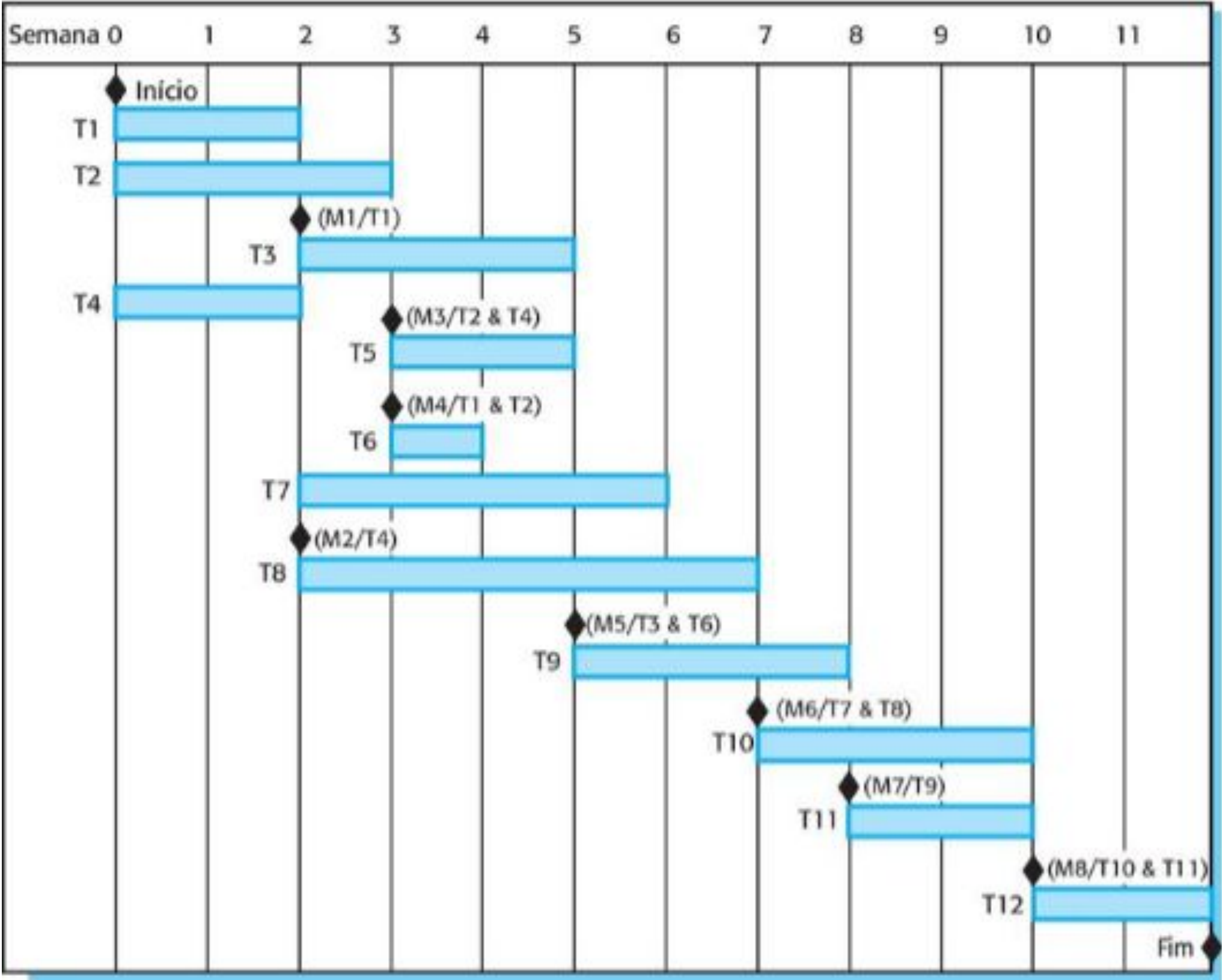


# Apresentação do Cronograma



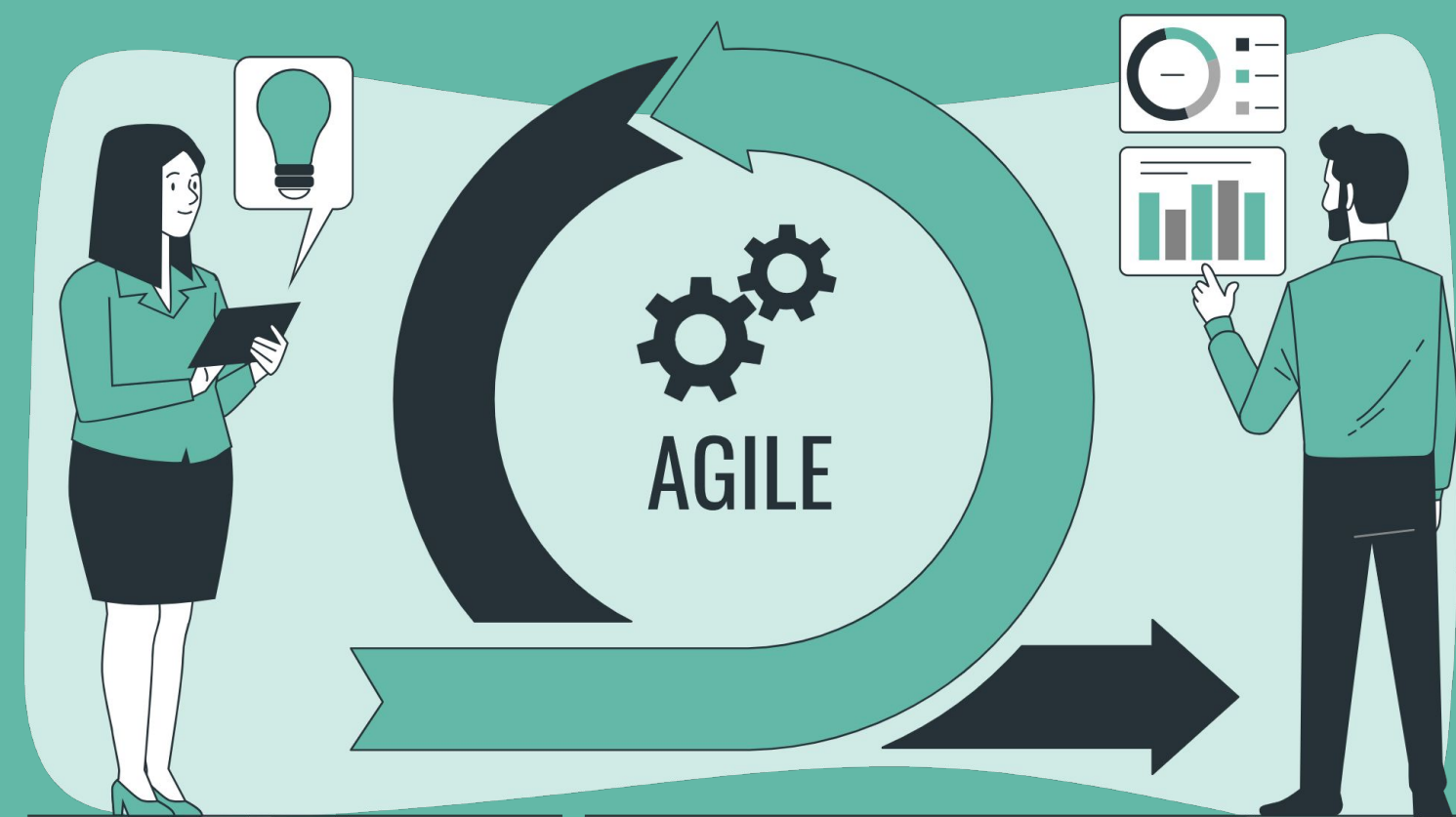
01

Gráficos de barras baseado em calendário



04

# Planejamento Ágil



# O que são métodos ágeis?

---



→ Os **métodos ágeis de desenvolvimento de software** são abordagens interativas nas quais o software é **desenvolvido e entregue aos clientes em incrementos**. A funcionalidade desses incrementos não é planejada com antecedência, mas decidida durante o desenvolvimento.

**“Prioridades e as necessidades do cliente mudam e, assim, faz sentido ter um plano flexível capaz de acomodar essas mudanças”**





# O Manifesto ágil



→ O Manifesto Ágil é uma declaração de valores e princípios essenciais para o desenvolvimento de software





# Extreme Programming (XP)



# Extreme Programming (XP)



05

# Técnicas de Estimativa





- As estimativas de cronograma de projeto são **difíceis**.
- Existem tantas incertezas que é **impossível** estimar com precisão os custos de desenvolvimento de sistema durante os estágios iniciais de um projeto.  
Existe até uma dificuldade fundamental na avaliação da precisão das diferentes abordagens para a estimativa de custo e esforço. Geralmente, as estimativas de projeto são de autopreenchimento.
- A estimativa é usada para definir o orçamento de projeto e o produto é ajustado para que o orçamento seja realizado. Um projeto que está dentro do orçamento pode ter alcançado isso em detrimento dos recursos do software que está sendo desenvolvido





**1. Técnicas baseadas em experiências:** As estimativas de requisitos de futuros esforços se baseiam na experiência do gerente em projetos anteriores e em seu domínio de aplicação. Essencialmente, o gerente faz uma avaliação informada do que os requisitos de esforço podem ser.







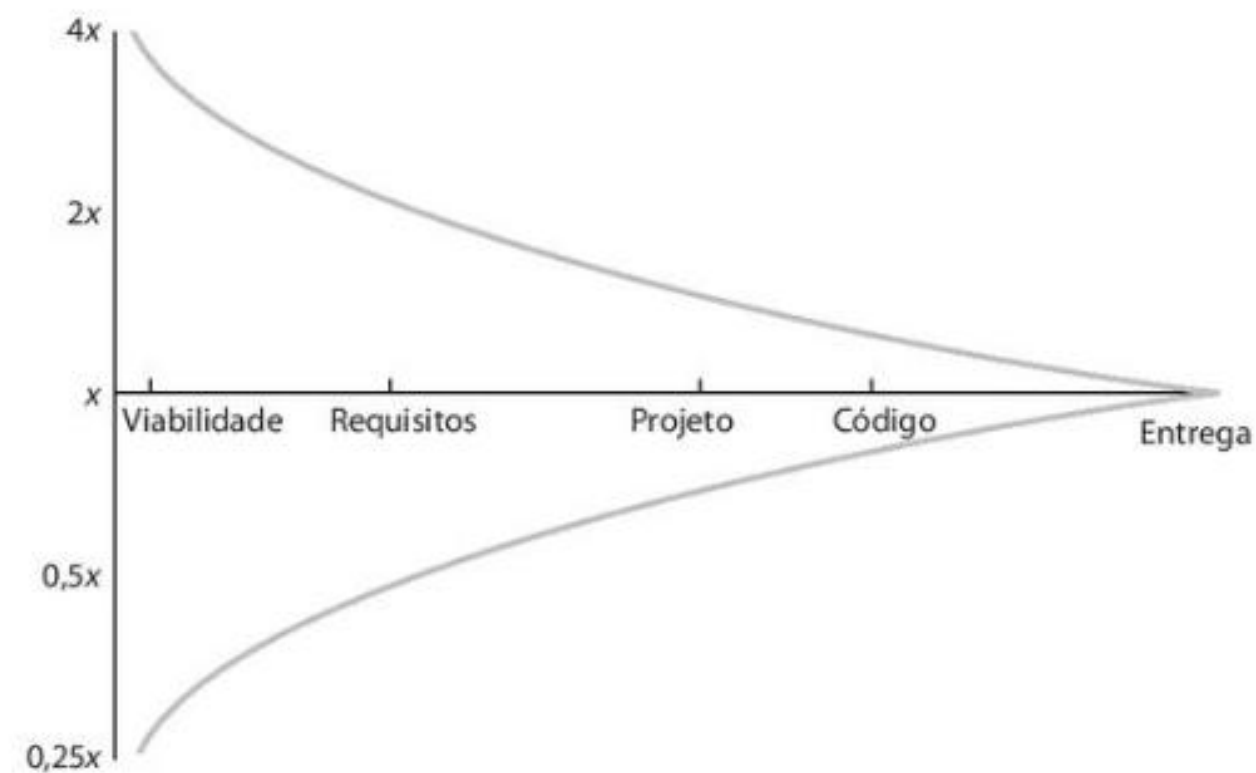
**2. Modelagem algorítmica de custos:** Uma abordagem usada em muitas situações para calcular o esforço de projeto com base em estimativas de atributos de produto, como tamanho e características do processo, por exemplo a experiência do pessoal envolvido.

$$Esforço = A \cdot Tamanho^B \cdot M$$

- **A** é um fator constante que depende das práticas organizacionais locais e do tipo de software em desenvolvimento.
- **Tamanho** pode ser uma avaliação do tamanho do código do software.
- **B** encontra-se geralmente entre 1 e 15.
- **M** é um multiplicador feito pela combinação de atributos de processo, produtos e de desenvolvimento, como os requisitos de confiança para o software e a experiência da equipe de desenvolvimento.

Em ambos os casos, você precisará usar seu julgamento para estimar o esforço, ou estimar as características de projeto e de produto. Na fase de iniciação de um projeto, essas estimativas têm uma grande margem de erro.

**Figura 23.6** Incerteza de estimativa



# Pontos importantes

---

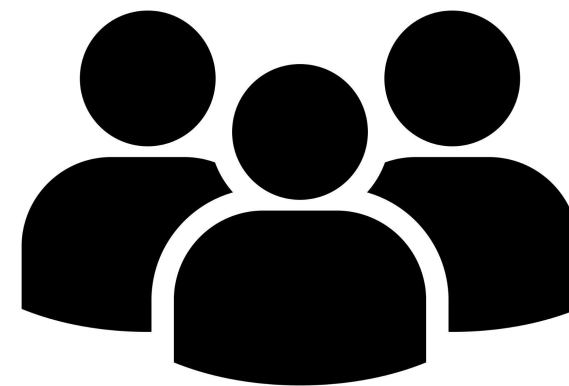


O preço cobrado por um sistema não depende apenas dos custos estimados de desenvolvimento e do lucro exigido pela empresa de desenvolvimento. Fatores organizacionais podem significar que o preço seja aumentado para compensar riscos altos, ou reduzido para ganhar vantagem competitiva. Muitas vezes, o preço do software é definido para ganhar um contrato e, em seguida, a funcionalidade do sistema é ajustada para atender ao preço estimado.





Geralmente, é útil contar com um grupo de pessoas envolvidas na estimativa de esforço e pedir a cada membro do grupo para explicar sua estimativa. Isso muitas vezes revela fatores que outros não levaram em consideração. Em seguida, você itera a uma estimativa acordada pelo grupo.



06

# Modelagem de Custos COCOMO

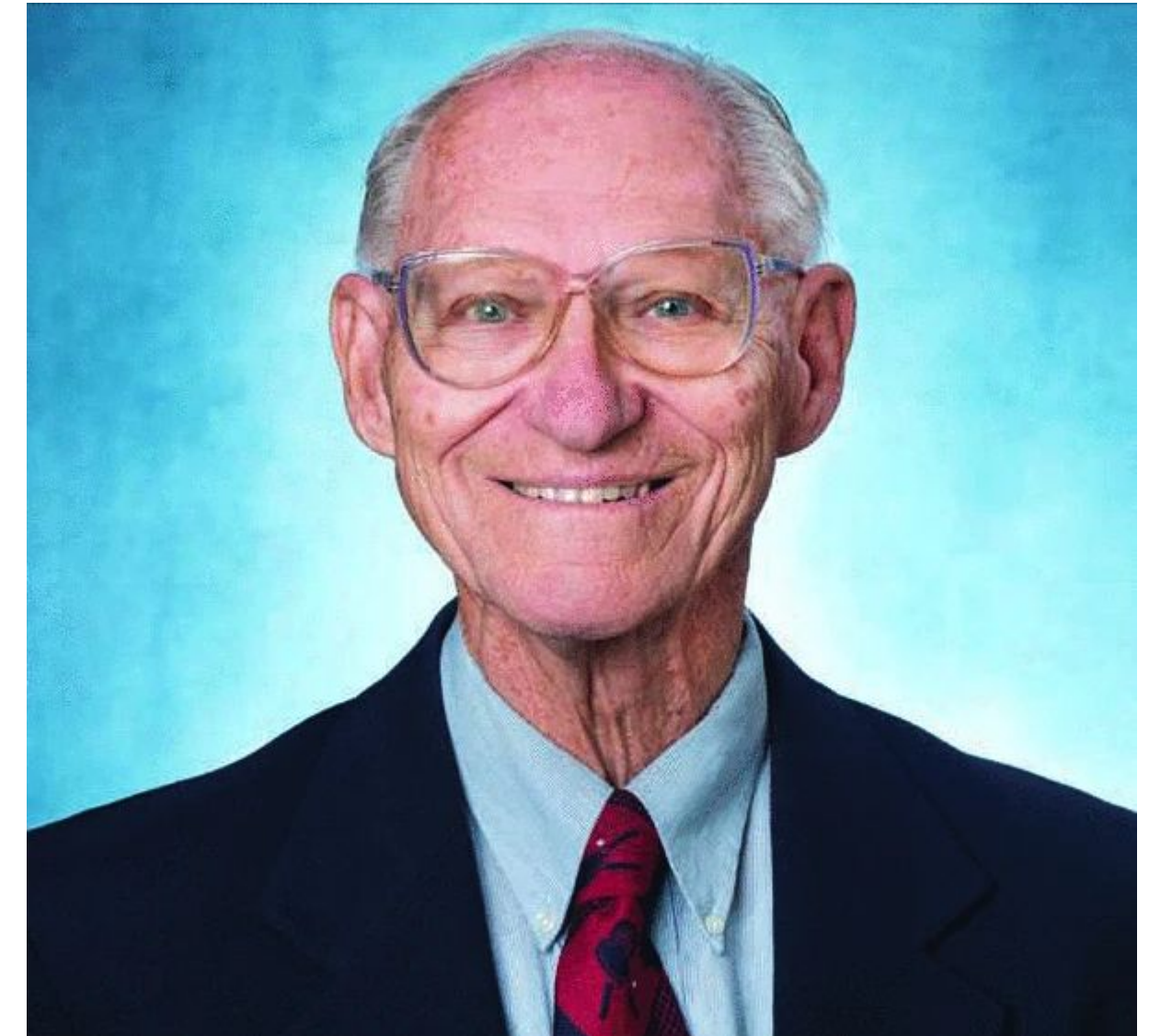




# O que é COCOMO ?



- Constructive Cost Model (Modelo de Construção de Custos)
- Método de estimativa de tempo de desenvolvimento de um software



Barry Boehm

☆ 1935- † 2022



# Divisão do COCOMO



O modelo COCOMO consistem em três implementações

- Básico
- Intermediário
- Avançado

O projeto de software pode ser definida em três tipos

- Orgânica
- Embutido
- Semidestacado



# Básico



Cocomo básico é bom por ser rápido em estimativas e custos de software, mas sua exatidão é limitada por causa de sua falta de fatores para explicar as diferenças entre ferramentas, qualidade de pessoal e experiência, uso de ferramentas modernas e técnicas, e outros atributos de projeto que influenciam nos custos de software.

$$E = a_b(KLOC)^{b_b}$$

$$D = c_b(E)^{d_b}$$

$$P = E / D$$

Projeto de Software	ab	bb	cb	db
Orgânico	2.4	1.05	2.5	0.38
Semi-Destacado	3.0	1.12	2.5	0.35
Embutido	3.6	1.20	2.5	0.32

**E** → Esforço aplicado pela pessoa no mês

**D** → Tempo de desenvolvimento em meses cronológicos

**KLOC** → Número calculado de linhas de códigos para o projeto (expressado em milhares)

**P** → Número de pessoas recomendadas

**ab, bb, cb e db** → coeficientes



# Intermediário



O método intermediário é uma extensão do método básico, mas com mais categorias de controle como:

Atributos do produto, Atributos de hardware, Atributos pessoais, Atributos do projeto.

$$E = a_i(KLOC)^{b_i}.EAF$$

Projeto de Software		$a_i$	$b_i$
Início	(orgânico)	3.2	1.05
Meio	(semidestacado)	3.0	1.12
Fim	(embutido)	2.8	1.50

**E** → Esforço aplicado pela pessoa no mês

**KLOC** → Número calculado de linhas de códigos para o projeto (expressado em milhares)

**EAF** → Fator calculado (0,9 e 1,4)

**$a_i$  e  $b_i$**  → coeficientes



Motoristas de Custos	Classificações					
	Muito Baixo	Baixo	Nominal	Alto	Muito Alto	Extra Alto
Atributos do produto						
Confiabilidade de software necessária	0,75	0,88	1.00	1.15	1.40	
Tamanho do banco de dados de aplicativos		0,94	1.00	1.08	1.16	
Complexidade do produto	0,70	0,85	1.00	1.15	1.30	1,65
Atributos de hardware						
Restrições de desempenho em tempo de execução			1.00	1.11	1.30	1.66
Restrições de memória			1.00	1.06	1.21	1,56
Volatilidade do ambiente de máquina virtual		0,87	1.00	1.15	1.30	
Tempo de resposta necessário		0,87	1.00	1.07	1.15	
Atributos da pessoa						
Capacidade do analista	1.46	1.19	1.00	0,86	0,71	
Experiência em aplicações	1.29	1.13	1.00	0,91	0,82	
Capacidade do engenheiro de software	1.42	1.17	1.00	0,86	0,70	
Experiência em máquina virtual	1.21	1.10	1.00	0,90		
Experiência em linguagem de programação	1.14	1.07	1.00	0,95		
Atributos do projeto						
Aplicação de métodos de engenharia de software	1.24	1.10	1.00	0,91	0,82	
Utilização de ferramentas de software	1.24	1.10	1.00	0,91	0,83	
Calendário de desenvolvimento necessário	1.23	1.08	1.00	1.04	1.10	





# Avançado



O COCOMO avançado incorpora todas as características da versão intermediária com uma avaliação do impacto dos drivers de custo em cada etapa do processo de engenharia de software. As seis fases do COCOMO Avançado são:

- Planejamento e requisitos
- Projeto do sistema
- Design detalhado
- Código do módulo e teste
- Integração e teste
- Modelo Construtivo de Custo



# Referências

---



Extreme programming (xp) - [Extreme Programming Values](#)

O Manifesto ágil - <https://agilemanifesto.org/iso/ptbr/manifesto.html>

COCOMO - JavaTPoint. COCOMO Model. JavaTPoint. Disponível em: <https://www.javatpoint.com/cocomo-model>. Acesso em: 19 de novembro de 2023.

SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo: Pearson, 2018. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 21 nov. 2023.



**OBRIGADO!**