



WBA0450_v1.0

Gerenciamento Ágil dos Sistemas





Gestão de riscos e de custos com COCOMO II

Bloco 1

Marco Ikuro Hisatomi



Desafio da gestão de custos em projetos de software

- ▶ CoCoMo II é a Modelagem Construtiva de Custo desenvolvida pela Universidade do Sul da Califórnia (Los Angeles – USA).
 - Abordagens modernas.
 - Desenvolvimento rápido com linguagens dinâmicas.
 - Reuso de código-fonte.
 - Programação em banco de dados.

Em um cenário hipotético, após apurar o total de Ponto de Função: 210 (não ajustados, conforme é considerado para o CoCoMo II).

CoCoMo II modelo de composição da aplicação

Estimativa ajustada de acordo com a dificuldade de desenvolver cada ponto da aplicação, usado na fase de prototipagem.

Considera:

- ▶ Produtividade (PROD) dependendo da experiência e capacidade do desenvolvedor e das ferramentas de software (CASE).
 - Muito baixa até muito alta (4, 7, 13, 25 e 50).
- ▶ Percentual de reuso do software existente.
- ▶ NAP – número total de pontos de aplicação.

$$PM = (NAP \times (1 - \%reuso/100))/PROD.$$

CoCoMo II modelo de projeto (design) inicial

Estimativa de esforço do design inicial a partir dos requisitos.

Considera:

- ▶ **A** - Coeficiente proposto por Bohem (2001): 2,94 (de acordo com histórico próprio).
- ▶ **Tamanho** - KSLOC (1000 *Source Line Of Code*), com base em Pontos de Função no software.
- ▶ **B** - Maior esforço necessário à medida que o tamanho do projeto aumenta: variando de 1,1 a 1,24 (dependendo de vários fatores, como: novidade, flexibilidade, resolução do risco, coesão do time e maturidade do processo).

$$\text{Esforço (PM)} = A \times \text{Tamanho}^B \times M$$

CoCoMo II modelo de projeto (design) inicial

► **M** – multiplicador (7) atributos de projeto e processo:

- PERS – capacidade da equipe.
- PREX – experiência da equipe.
- RCPX – confiabilidade e complexidade do produto.
- RUSE – reuso.
- PDIF – dificuldade da plataforma.
- SCED – cronograma.
- FSIL – recursos de apoio.

$$\text{Esforço (PM)} = A \times \text{Tamanho}^B \times M$$

CoCoMo II modelo de reuso

- ▶ Cálculo do esforço para integrar código reusável ou gerado (ESLOC), que é acrescido ao modelo de projeto inicial.
- ▶ Código caixa-preta: utilizando modelos ou bibliotecas (não acresce no esforço).
- ▶ Código de caixa-branca: adaptações de código existente, necessita:
 - Avaliar se o componente deve ou não ser reusado.
 - Entender o código que está sendo reusado.
 - Modificar o código reusado para integração ao sistema.

$$\text{ESLOC} = (\text{ASLOC} \times (1 - \text{AT}/100) \times \text{AAM})$$

CoCoMo II modelo de reuso

- ▶ Para obter o número equivalente do esforço para o reuso.
- ▶ **ASLOC** estimativa do número de linhas de código que necessitam de adequação.
- ▶ **AT** percentual de código reusado que pode ser modificado automaticamente.
- ▶ **AAM** multiplicador de ajuste de adaptação (*adaptation adjustment multiplier*).

$$ESLOC = (ASLOC \times (1 - AT/100)) \times AAM$$

CoCoMo II modelo pós-arquitetura

- ▶ Considerar o tamanho acrescentando aos três anteriores.
- ▶ SLOC estimativa do número total de linhas de código novo.
- ▶ ESLOC estimativa de reuso.
- ▶ Estimativa do número de linhas de código que podem ser alteradas pela mudança de requisitos do sistema.
- ▶ Considera o expoente **B** que o nível de complexidade do projeto (5 fatores).
- ▶ **M** – multiplicador (7) atributos de projeto inicial.

$$PM = A \times \text{Tamanho}^B \times M$$

CoCoMo II modelo pós-arquitetura

Sobre o expoente **B**:

- ▶ Arquitetura/resolução de risco: grau de análise de riscos.
- ▶ Flexibilidade de desenvolvimento: flexibilidade no processo, aceitando objetivos gerais do cliente para iniciar o projeto.
- ▶ Precedência: experiência anterior para o tipo de projeto.
- ▶ Coesão do time: se conhecem e trabalham juntos.
- ▶ Maturidade do processo: nível maturidade do processo da organização (CMM - *Capability Maturity Model*).

$$PM = A \times \text{Tamanho}^B \times M$$

CoCoMo II modelo pós-arquitetura

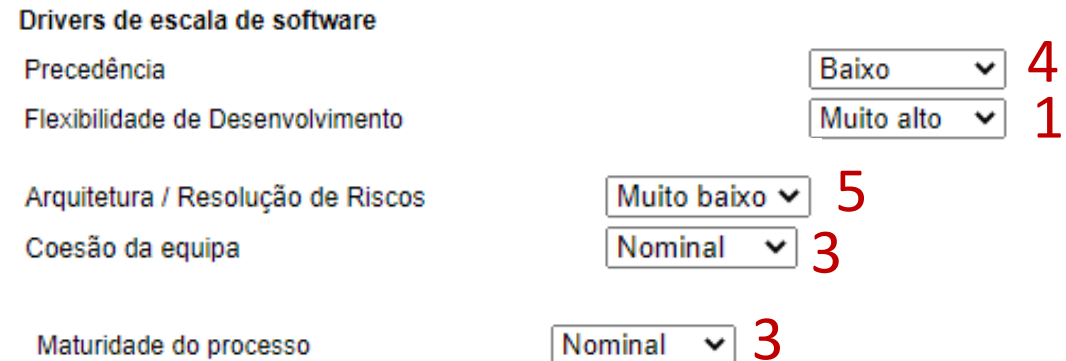
B, o nível de complexidade do projeto.

Cada atributo pode aumentar ou diminuir a estimativa.

Somando todos os pontos: exemplo 16, deve dividir por 100 e somar 0,01.

Ou seja, **B** = 1,17.

Figura 1 - Drivers de escala de software



Drivers de escala de software	
Precedência	Baixo 4
Flexibilidade de Desenvolvimento	Muito alto 1
Arquitetura / Resolução de Riscos	Muito baixo 5
Coesão da equipa	Nominal 3
Maturidade do processo	Nominal 3

Fonte: captura de tela do software Cocomo.

$$PM = A \times \text{Tamanho}^B \times M$$



Gestão de riscos e de custos com COCOMO II

Bloco 2

Marco Ikuro Hisatomi



COCOMO II – pós-arquitetura – refinamento

- ▶ A estimativa de esforço global, segundo Sommerville (2018), é refinada usando um conjunto extenso de 17 atributos de produto, de processo, da organização e de projeto.
- ▶ Sendo os principais direcionadores de custo no projeto de desenvolvimento:
 - Confiabilidade (RELY).
 - Complexidade (CPLX).
 - Armazenamento (STOR).
 - Ferramentas (TOOL).
 - Cronograma (SCED).

COCOMO II – softwarecost.org

Figura 2 - Drivers de custo de software - produto

Drivers de custo de software produtos

► Confiabilidade de software necessária	Nominal ▼
Tamanho da base de dados	Nominal ▼
► Complexidade do produto	Nominal ▼
Desenvolvido para Reutilização	Nominal ▼
Correspondência da documentação às necessidades do ciclo de vida	Nominal ▼

Fonte: captura de tela do software Cocomo.

COCOMO II – softwarecost.org

Figura 3 - Drivers de custo de software - Pessoal

Pessoal

Capacidade do analista

Nominal ▼

Capacidade do Programador

Nominal ▼

Continuidade do pessoal

Nominal ▼

Experiência de aplicação

Nominal ▼

Experiência de plataforma

Nominal ▼

Experiência de idioma e conjunto de ferramentas

Nominal ▼

Fonte: captura de tela do software Cocomo.

COCOMO II – softwarecost.org

Figura 4 - Drivers de custo de software – plataforma e projeto

Plataforma

Restrição de tempo

Nominal ▼

► Restrição de armazenamento

Nominal ▼

Volatilidade da plataforma

Nominal ▼

Projeto

► Uso de ferramentas de software

Nominal ▼

Desenvolvimento Multisite

Nominal ▼

► Cronograma de desenvolvimento necessário

Nominal ▼

Fonte: captura de tela do software Cocomo.



Gestão de riscos e de custos com COCOMO II

Bloco 3

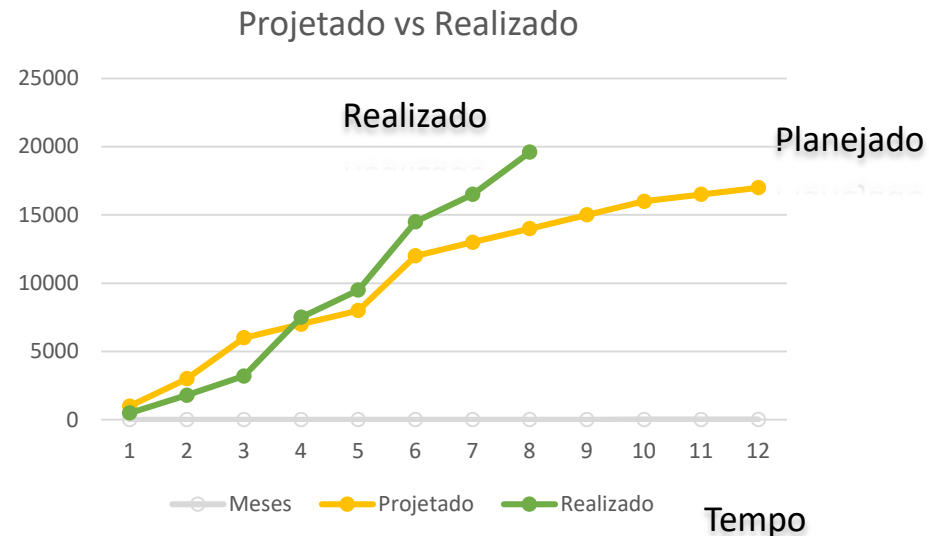
Marco Ikuro Hisatomi



Continuamos com o projeto?

- Risco \Rightarrow Prejuízo
- Custo \Rightarrow Prejuízo
- ▶ Gerentes sem experiências em ágil.
- ▶ Modelos burocráticos são incompatíveis.
- ▶ Uso de ferramentas operacionais na gestão de projetos.
- ▶ Processos convencionais em engenharia de sistemas.

Figura 5 – Projeção de custos

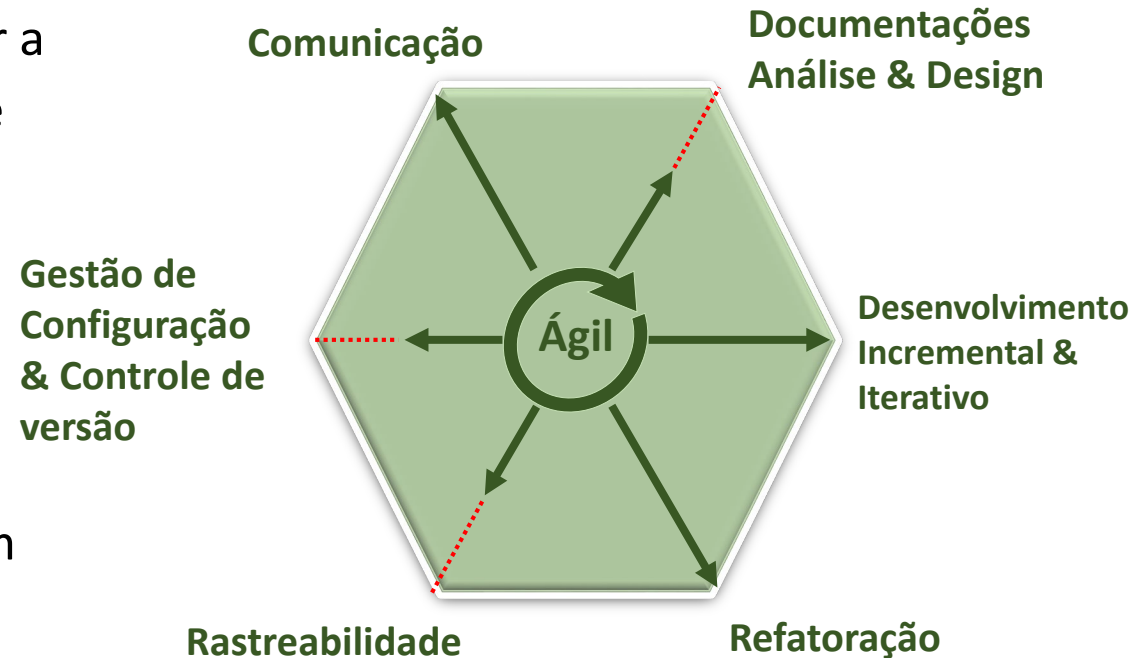


Fonte: elaborado pelo autor

Vantagens do ágil para minimizar riscos

- ▶ IID (*Iterative and Incremental Development*) facilita a reduzir a complexidade das Histórias de Usuários.
- ▶ Ênfase na Comunicação, um dos pontos altos de Métodos Ágeis.
- ▶ Alguns processos ainda podem ser aprimorados para sofrer menos com o impacto das mudanças de requisitos.

Figura 6 - Vantagens e carências dos métodos ágeis



Fonte: adaptado de LAI (2015 p. 32).

Classificação de riscos

Incerteza  Risco  Perdas

Pra melhor tratar os riscos, Pressman (2016, p. 779), além de registrar as perdas, afirma que devemos categorizar em:

De Projeto: relacionado ao plano do projeto, cronograma, orçamento, envolvidos, recursos, clientes e requisitos; complexidade e grau de incerteza.

Técnico: tem impacto na qualidade ou prazo de entrega do software, por falhas em design, implementação, interface e obsolescência técnica.

De Negócio: do produto (falta de interesse), de estratégia (incompatível com a organização), mudança de foco do negócio e falta de comprometimento.

Providenciar recursos para escalabilidade ágil

Pontos a serem providenciados, Sommerville (2018, p. 82):

- Requisitos antecipados.
- Vários representantes do cliente.
- Maior número de documentações.
- Ferramental comum aos times de projeto.
- Alinhamento das entregas entre os times.

Contemplar PMBoK em projeto de software

Muitos projetos tradicionais de grande porte, longos e, ainda, para os clientes que exigem documentações anexo ao contrato.

Exigências em monitoramento e controle.

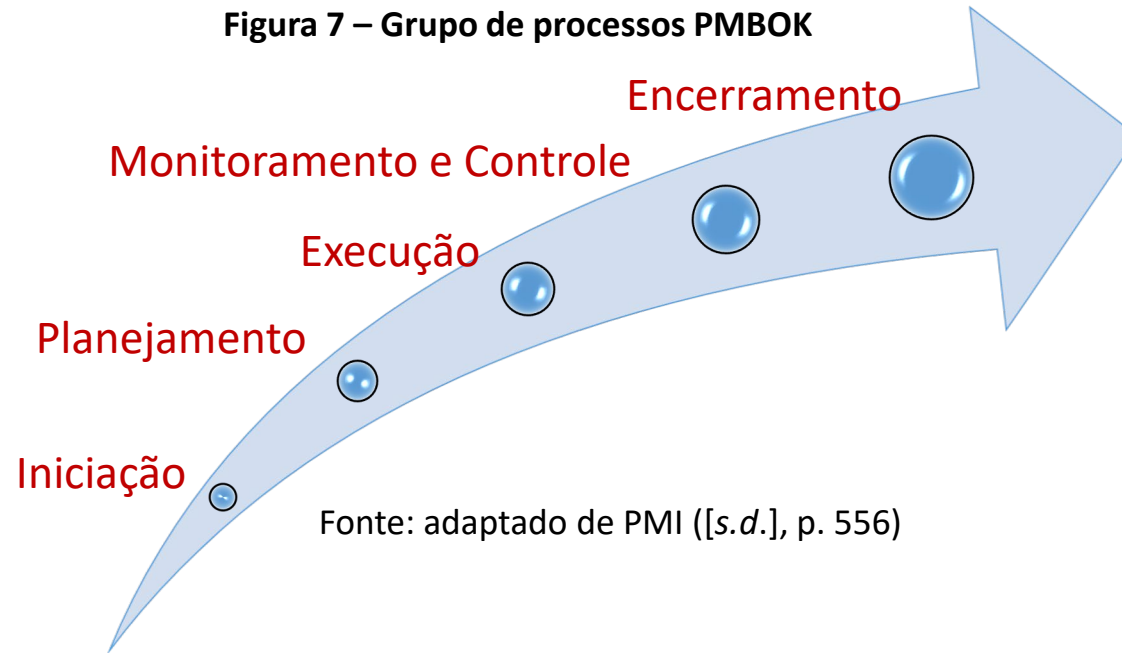
Registro de papéis e responsabilidades.

Identificação e mitigação de riscos.

Estrutura analítica do projeto.

Project Management Body of Knowledge

Figura 7 – Grupo de processos PMBOK



Fonte: adaptado de PMI ([s.d.], p. 556)

Plano *Risk Mitigation Monitoring and Management*

Usando o RIS (*Risk Information Sheet*).

Figura 8 – Ficha de informação de risco

#Risco: PGR-999	Data: 31/05/2020	Probabilidade: 75%	Impacto: Alto
Descrição: O índice de Refatoração está em 50%			
Refinamento: a) falta de um desenvolvedor b) falta de padrão de nomenclatura			
Mitigação/Monitoramento: a) Treinar membro para substituir o desenvolvedor; b) Reunir equipe para padronizar nomenclatura.			
Gerenciamento/Contingência: Alocação, programa e agendamento de capacitação de desenvolvedor. Criar incentivo para planejar faltas de desenvolvedor. Desenvolver plano de atividades de planejamento para padronização. Data limite 31/07/2020.			
Estado atual: Planejando atividades de capacitação. Reunindo para elaboração do plano de padrões.			
Autor: Marco Hisatomi		Responsável: Marco Hisatomi	

Fonte: adaptado de Pressman (2016, p. 791).



Teoria em Prática

Bloco 4

Marco Ikuro Hisatomi



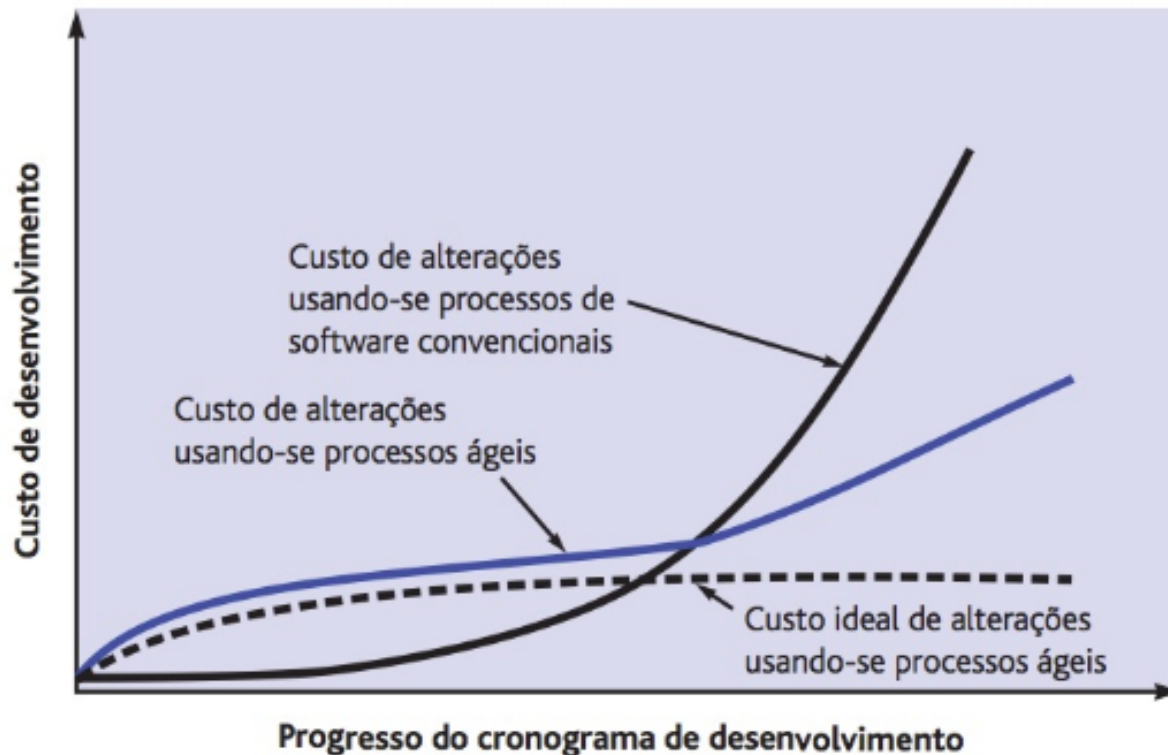
Refleta sobre a seguinte situação

O modelo de gerenciamento de projeto de software pode influenciar diretamente na gestão de custos? Caso façamos um comparativo entre os modelos tradicionais e Métodos Ágeis, segundo Pressman (2016 apud AMBLER, 2004), um processo ágil, quando aplicado adequadamente, propicia um crescimento menor em custos ao longo do ciclo de vida de um sistema, comparado com os modelos tradicionais.

- Acredita-se que os custos serão menores usando Métodos Ágeis? Quando ocorrer as mudanças de requisitos do sistema, qual é a tendência de aumento dos custos?

Tendências dos custos em Métodos Ágeis (PRESSMAN, 2016, p. 69)

Figura 9 – Comparativo de custos de alterações



Fonte: Pressman (2016, p. 69).

- A tendência dos custos em ágeis é ter uma curva “achatada”.
- Entregas **incrementais**, aliado aos testes contínuos de unidades ou em pares.
- **Debates contínuos** e histórias de usuário atualizadas aumentam a eficiência no design e na implementação.



Dica do Professor

Bloco 5

Marco Ikuro Hisatomi



Precificação de software (SOMMERVILLE, 2018, p. 634)

- ▶ Termos contratuais: quanto a propriedade do código-fonte e reuso em outros projetos (poderia ser um preço menor).
- ▶ Incerteza da estimativa de custos: falta de cultura organizacional, com histórico de estimativa e precificação (pode precificar a mais).
- ▶ Saúde financeira: com situação de dificuldade no momento (pode precificar com desconto para manter o fluxo de caixa).
- ▶ Oportunidade de mercado: para entrar no mercado para um segmento específico (o preço pode ser uma maneira de investimento).
- ▶ Volatilidade de requisitos: preço menor no início, com possibilidade de renegociar no futuro para compensar com as mudanças.

Referências

PMI. **PMBOK Guia do CONHECIMENTO EM GERENCIAMENTO DE PROJETOS.**

[s.d.]. Disponível em: <https://www.pmi.org/pmbok-guide-standards/practice-guides>. Acesso em: 4 set. 2020.

PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. Porto Alegre: AMGH, 2016.

SCHWABER, K. **Um guia definitivo para o scrum**: as regras do Jogo. Out. 2017. Disponível em:

<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>. Acesso em: 20 maio 2020.

Referências

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson Education do Brasil, 2018.

COCOMO II - CONSTRUCTIVE COST MODEL. Disponível em:
<http://softwarecost.org/tools/COCOMO/>. Acesso em: 30 maio 2020.



Bons estudos!

