
Engenharia de Software

Guide to the SWEBOK

(Guide to the Software Engineering Body of Knowledge)

IEEE Computer Society

Rational[®]
the e-development company™



National Research
Council Canada

Conseil national
de recherches Canada

- Institute of Electric and Electronic Engineers
- Associação profissional mundial sem fins lucrativos para suporte ao avanço tecnológico em sua área de atuação.
- Missão: O IEEE promove o processo de engenharia de criação, desenvolvimento, integração, compartilhamento e aplicação do conhecimento sobre tecnologia eletroeletrônica, tecnologia da informação e ciências, para o benefício da humanidade e da profissão (*The IEEE promotes the engineering process of creating, developing, integrating, sharing, and applying knowledge about electro and information technologies and sciences for the benefit of humanity and the profession*).



IEEE Computer Society

- Aproximadamente 100.000 membros.
- Organização de suporte aos profissionais da computação, provendo informação técnica, serviços a comunidade e aos profissionais.
- Fundada em 1946.
- A maior das 39 sociedades do IEEE.
- Dedicada ao avanço teórico, prático e da aplicação de computadores e da tecnologia da informação.
- Aproximadamente 40% dos membros vive e trabalha fora dos EUA, fomentando a comunicação internacional, a cooperação e a troca de informação.
- Tem um escritório central de serviços em Tokyo, Japão; um escritório de publicação em Los Alamitos, California; e a sede em Washington, DC.

The Guide to the SWEBOK

- Não é o conhecimento em si, mas uma síntese e uma referência ao material disponível em diversas publicações que se complementam para formar o corpo de conhecimento.
- Descreve que parte do conhecimento é **geralmente aceito** pela comunidade profissional.
- Organiza o conhecimento.
- Provê acesso por tópicos.
- Patrocinado por empresas como: Rational (IBM), SAP, Boeing, entre outras.

Categorização do Conhecimento

Geralmente Aceito

Specialized Practices used only for certain types of software	Generally Accepted Established traditional practices recommended by many organizations
	Advanced and Research Innovative practices tested and used only by some organizations and concepts still being developed and tested in research organizations

Objetivos

1. Promover uma **visão consistente** da engenharia de software em todo o mundo (500 revisores de 42 países na fase Stoneman, versão Trial, e 120 revisores de 21 países na fase Ironman, versão 2004).
2. Definir as **fronteiras de atuação** da engenharia de software e as áreas de interseção com outras disciplinas como: engenharia da computação, ciência da computação, gestão de negócios, matemática, gerenciamento de projetos, gestão da qualidade, ergonomia (acessibilidade e usabilidade) e engenharia de sistemas (SWEBOK, capítulo 12).

Objetivos (Continuação)

1. Caracterizar o **conteúdo** da disciplina engenharia de software, subdividindo-o hierarquicamente em áreas de conhecimento (o Apêndice A descreve como as AC devem ser organizadas).
2. Prover **acesso por tópicos** a base de conhecimento da engenharia de software (material de referência e matriz em cada AC).
3. Fornecer um alicerce para desenvolvimento do currículo, certificação individual e licenciamento de material (**conhecimento geralmente aceito**: aplica-se a maioria dos projetos e das equipes pelo **consenso** e pela **efetividade**).

Material de Referência e Matriz

- Exemplo de matriz que relaciona os tópicos de cada AC com as fontes recomendadas.

	[Dav93]	[Gog93]	[IEEE830-98]	[IEEE14143.1-00]	[Kot00]	[Lou95]	[Pff01]	[Rob99]	[Som97]	[Som05]	[Tha97]	[You01]
1. Software Requirements Fundamentals												
<i>1.1 Definition of a Software Requirement</i>					*		*			c5	c1	
<i>1.2 Product and Process Requirements</i>					*				c1			
<i>1.3 Functional and Non-functional Requirements</i>					*				c1			
<i>1.4 Emergent Properties</i>										c2		
<i>1.5 Quantifiable Requirements</i>	c3s4									c6		
<i>1.6 System Requirements and Software Requirements</i>												
2. Requirements Process	*									c5		
<i>2.1 Process Models</i>					c2s1			*	c2	c3		

Material de Referência e Matriz

- Exemplo de referências recomendadas em cada AC.

RECOMMENDED REFERENCES FOR SOFTWARE REQUIREMENTS

[Dav93] A.M. Davis, *Software Requirements: Objects, Functions and States*, Prentice Hall, 1993.

[Gog93] J. Goguen and C. Linde, "Techniques for Requirements Elicitation," presented at International Symposium on Requirements Engineering, 1993.

[IEEE830-98] IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE, 1998.

(IEEE14143.1-00) IEEE Std 14143.1-2000//ISO/IEC14143-1:1998, *Information Technology—Software Measurement—Functional Size Measurement—Part 1: Definitions of Concepts*, IEEE, 2000.

[Kot00] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, John Wiley & Sons, 2000.

[Lou95] P. Loucopulos and V. Karakostas, *Systems Requirements Engineering*, McGraw-Hill, 1995.

[Pfl01] S.L. Pfleeger, "Software Engineering: Theory and Practice," second ed., Prentice Hall, 2001, Chap. 4.

[Rob99] S. Robertson and J. Robertson, *Mastering the Requirements Process*, Addison-Wesley, 1999.

[Som97] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, John Wiley & Sons, 1997, Chap. 1-2.

[Som05] I. Sommerville, *Software Engineering*, seventh ed., Addison-Wesley, 2005.

[Tha97] R.H. Thayer and M. Dorfman, eds., *Software Requirements Engineering*, IEEE Computer Society Press, 1997, pp. 176-205, 389-404.

[You01] R.R. You, *Effective Requirements Practices*, Addison-Wesley, 2001.

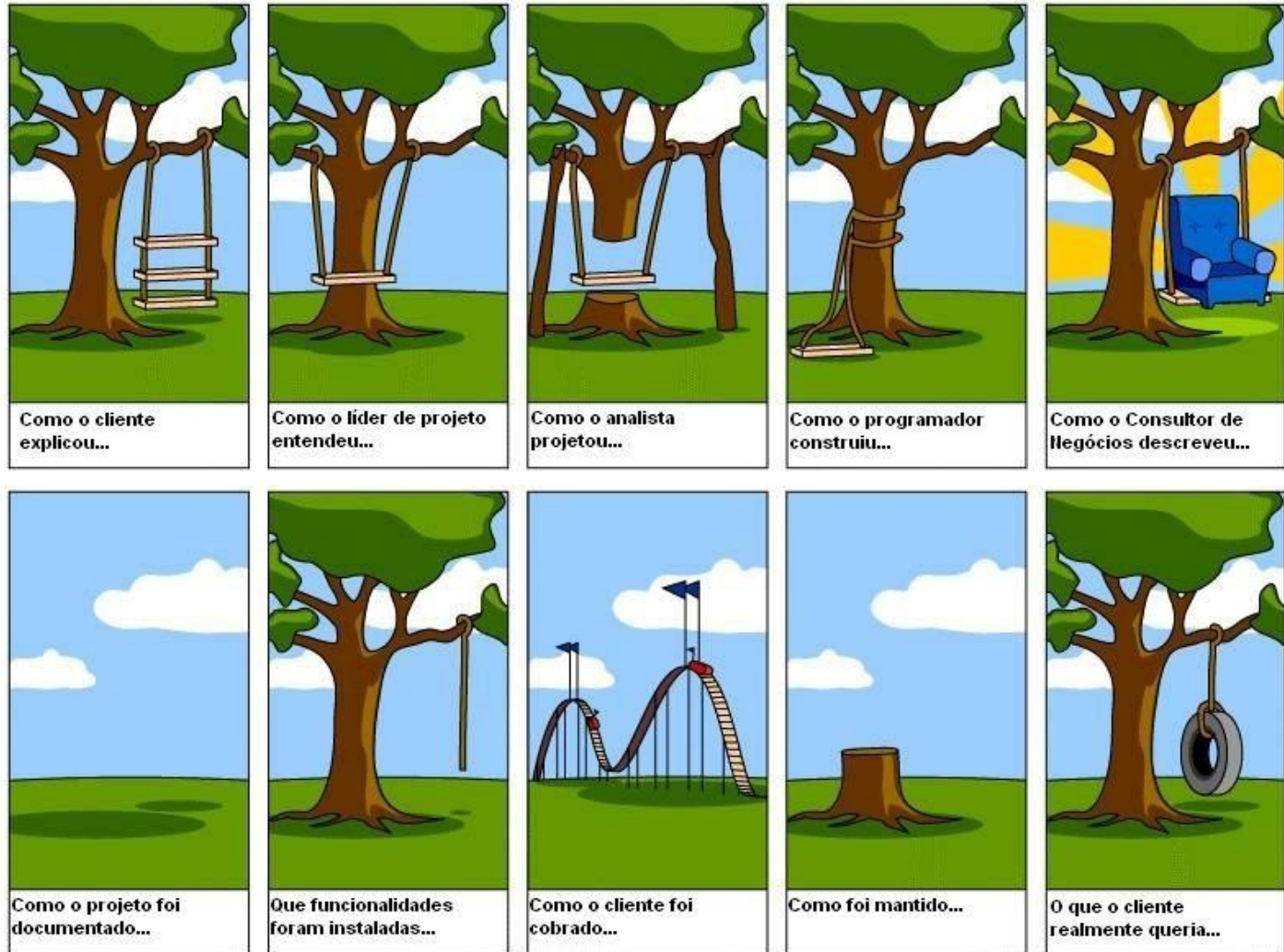
Áreas de Conhecimento (AC)

1. Software Requirements (Requisitos)
2. Software Design (Desenho / Projeto)
3. Software Construction (Construção / Implementação)
4. Software Testing (Testes)
5. Software Maintenance (Manutenção)
6. Software Configuration Management (Configuração)
7. Software Engineering Management (Gerenciamento)
8. Software Engineering Process (Processo)
9. Software Engineering Tools and Methods (Ferramentas e Métodos)
10. Software Quality (Qualidade)

Requisitos (SWEBOK, Capítulo 2)

- São problemas do mundo real que devem ser explicitados para serem solucionados (O que será feito!).
- A AC está dividida em sete subáreas.
 1. Fundamentos dos Requisitos
 2. Processo de Requisitos
 3. Declaração de Requisitos
 4. Análise de Requisitos
 5. Especificação de Requisitos
 6. Validação de Requisitos
 7. Considerações Práticas

Requisitos (O que “pode” acontecer!?)



Desenho (SWEBOK, Capítulo 3)

- É o processo de definição da arquitetura, dos componentes, da interface e de outras características dos componentes do sistema e do resultado esperado, para resolver o problema identificado na fase de requisitos (Como será feito!).
- A AC está dividida em seis subáreas:
 1. Fundamentos do Desenho de Software
 2. Questões Chave no Desenho de Software
 3. Estrutura e Arquitetura de Software
 4. Evolução e Análise de Qualidade do Desenho de Software
 5. Notações do Desenho de Software
 6. Estratégias e Métodos para o Desenho de Software

Construção (SWEBOK, Capítulo 4)

- Refere-se a criação do conjunto de programas que compõe o software e a verificação de conformidade com o desenho, da funcionalidade individual (teste unitário), da funcionalidade integrada e a depuração de erros.
- A AC está dividida em três subáreas:
 1. Fundamentos da Construção de Software
 2. Gerenciamento da Construção de Software
 3. Considerações Práticas (e **padrões**) da Construção de Software

Teste (SWEBOK, Capítulo 5)

- Consiste na verificação do comportamento de um programa ou grupo de programas, a partir de um conjunto de casos selecionados no domínio de negócio, comparando-o ao comportamento esperado, definido na especificação de requisitos.
- A AC está dividida em cinco subáreas:
 1. Fundamentos do Teste de Software
 2. Níveis de Teste de Software
 3. Técnicas de Teste de Software
 4. Medidas de Teste de Software
 5. Processo de Teste e Considerações Práticas

Manutenção

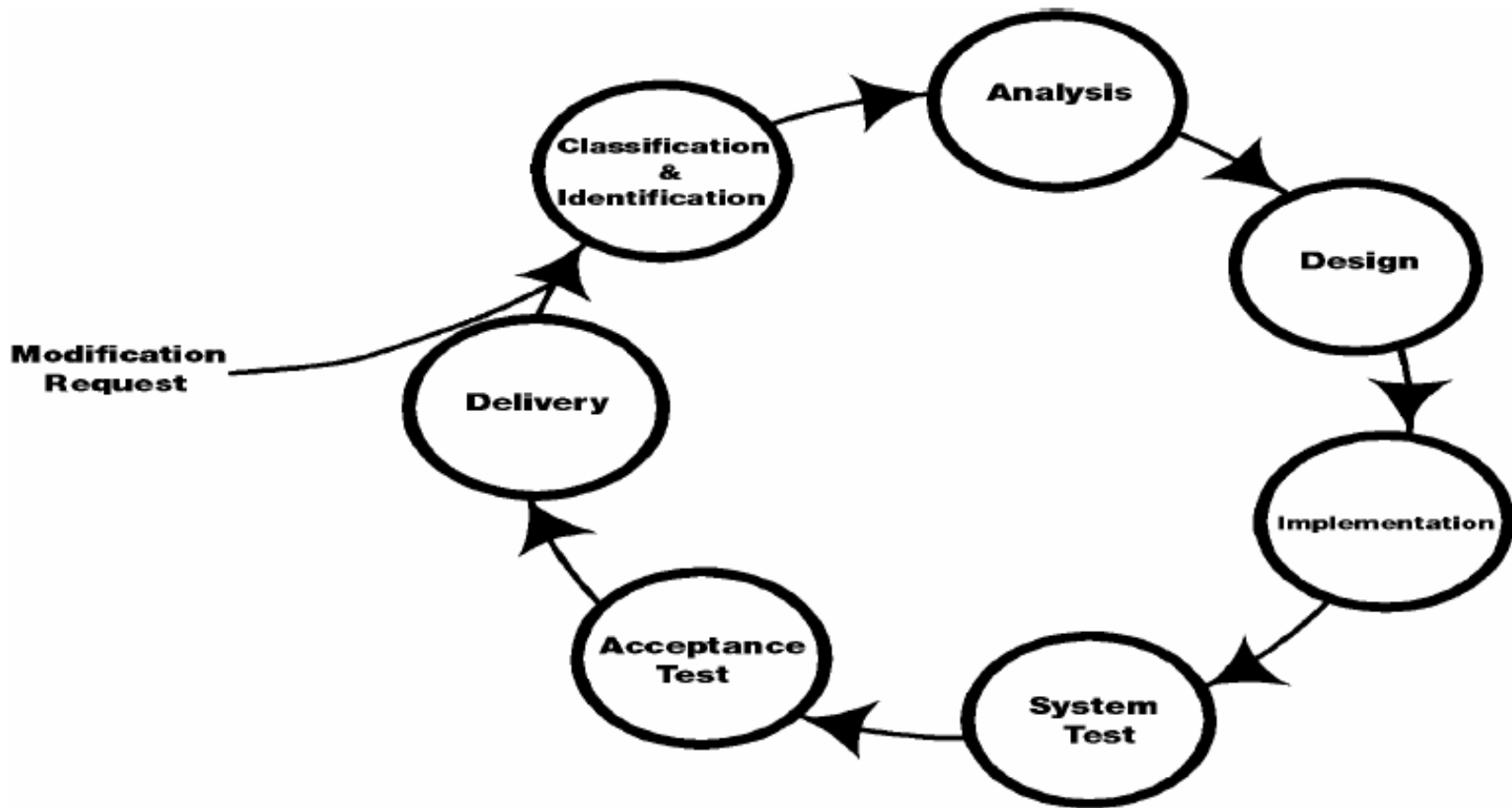
(SWEBOK, Capítulo 6)

- Um vez em operação, anomalias são descobertas, ambientes operacionais mudam e novos requisitos surgem. A fase de manutenção inicia-se a partir da entrega, mas as atividades de manutenção começam antes mesmo da entrega.
- A AC está dividida em quatro subáreas:
 1. Fundamentos da Manutenção de Software
 2. Questões Chave sobre Manutenção de Software
 3. Processo de Manutenção de Software
 4. Técnicas de Manutenção de Software

Manutenção (Categorias)

	Correção	Evolução
Pró-ativa	Correction	Enhancement
Reativa	Proactive Preventive	Perfective
	Reactive Corrective	Adaptive

Manutenção (Atividades)



Gerenciamento de Configuração (SWEBOK, Capítulo 7)

- Consiste em identificar a configuração do software em diferentes momentos com o propósito de controlar sistematicamente as mudanças ocorridas e garantir a integridade e a rastreabilidade das configurações de **todos os artefatos** ao longo do ciclo de vida do software.
- A AC está dividida em seis subáreas:
 1. Processo de Gerenciamento de Configurações
 2. Identificação da Configuração
 3. Controle da Configuração
 4. Registro do Estado da Configuração
 5. Auditoria da Configuração
 6. Entrega e Gerenciamento de Versões

Gerenciamento de Projeto de Software (SWEBOK, Capítulo 8)

- Consiste no gerenciamento (PDCA) da engenharia de software. Gerenciamento é um aspecto fundamental e presente em todas as AC, mas nesta AC são apresentados os tópicos específicos sobre o gerenciamento.
- A AC está dividida em seis subáreas:
 1. Iniciação e Definição de Escopo
 2. Planejamento do Projeto
 3. Declaração do Plano de Projeto
 4. Monitoramento e Controle
 5. Fechamento

Gerenciamento de Projeto de Software

Projeto, segundo o PMBOK:

- “É um empreendimento temporário que tem por finalidade criar um produto, serviço ou resultado único”.
- Características:
 - Não repetitivo (início, meio e fim)
 - Seqüência lógica de eventos (ciclo de vida)
 - Acompanhamento e controle
 - Objetivo claro e definido
 - Conduzido por pessoas
 - Parâmetros pré-definidos (tempo, recursos e qualidade).
 - É bem-sucedido quando atende às expectativas dos stakeholders.
- Para saber se temos um projeto, perguntamos:
 1. É único?
 2. Tem prazo limitado?
 3. Existe uma maneira de saber se está concluído?
 4. É possível saber se os **stakeholders** estão satisfeitos?

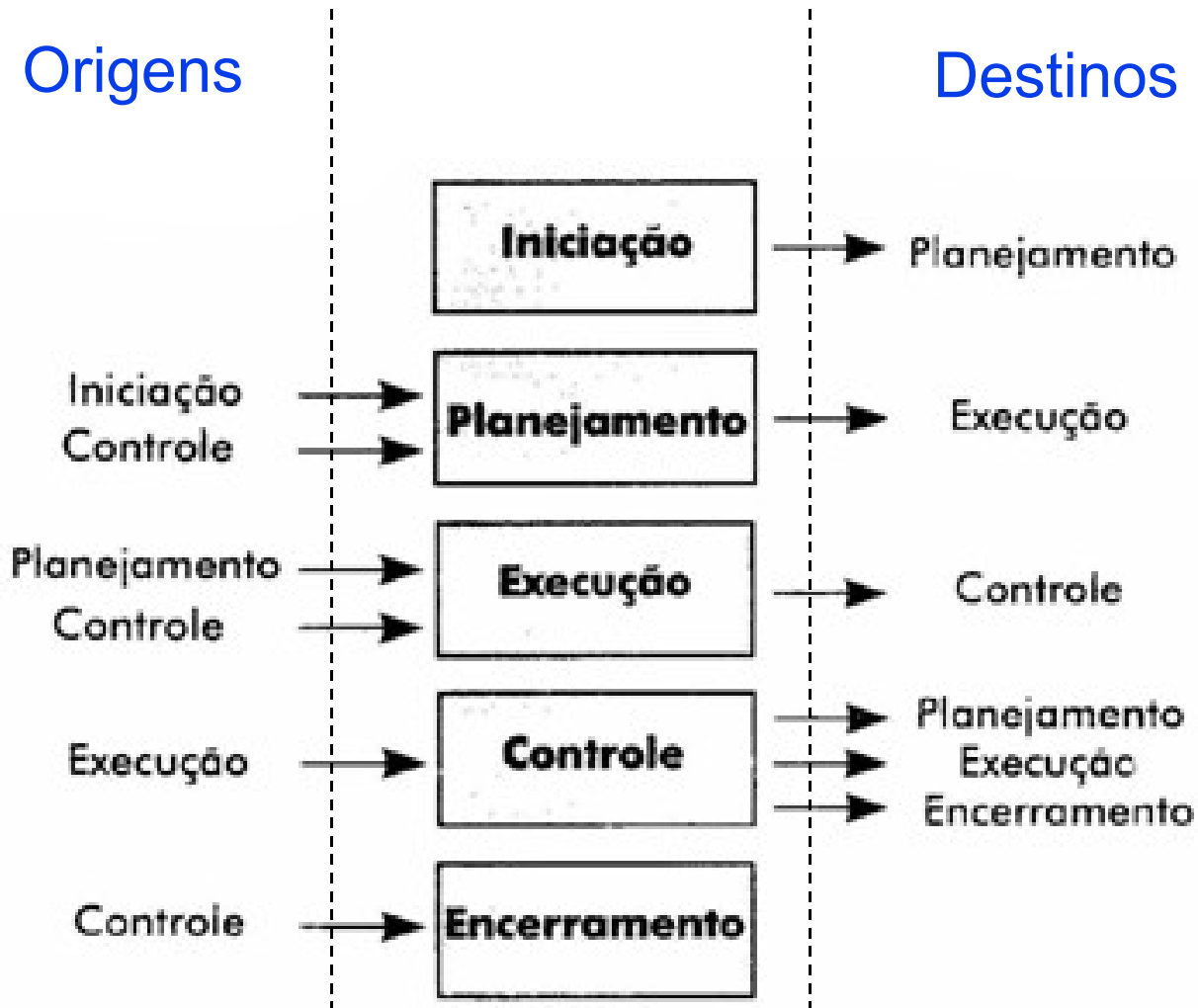
Gerenciamento de Projeto de Software

Áreas de Conhecimento do PMBOK



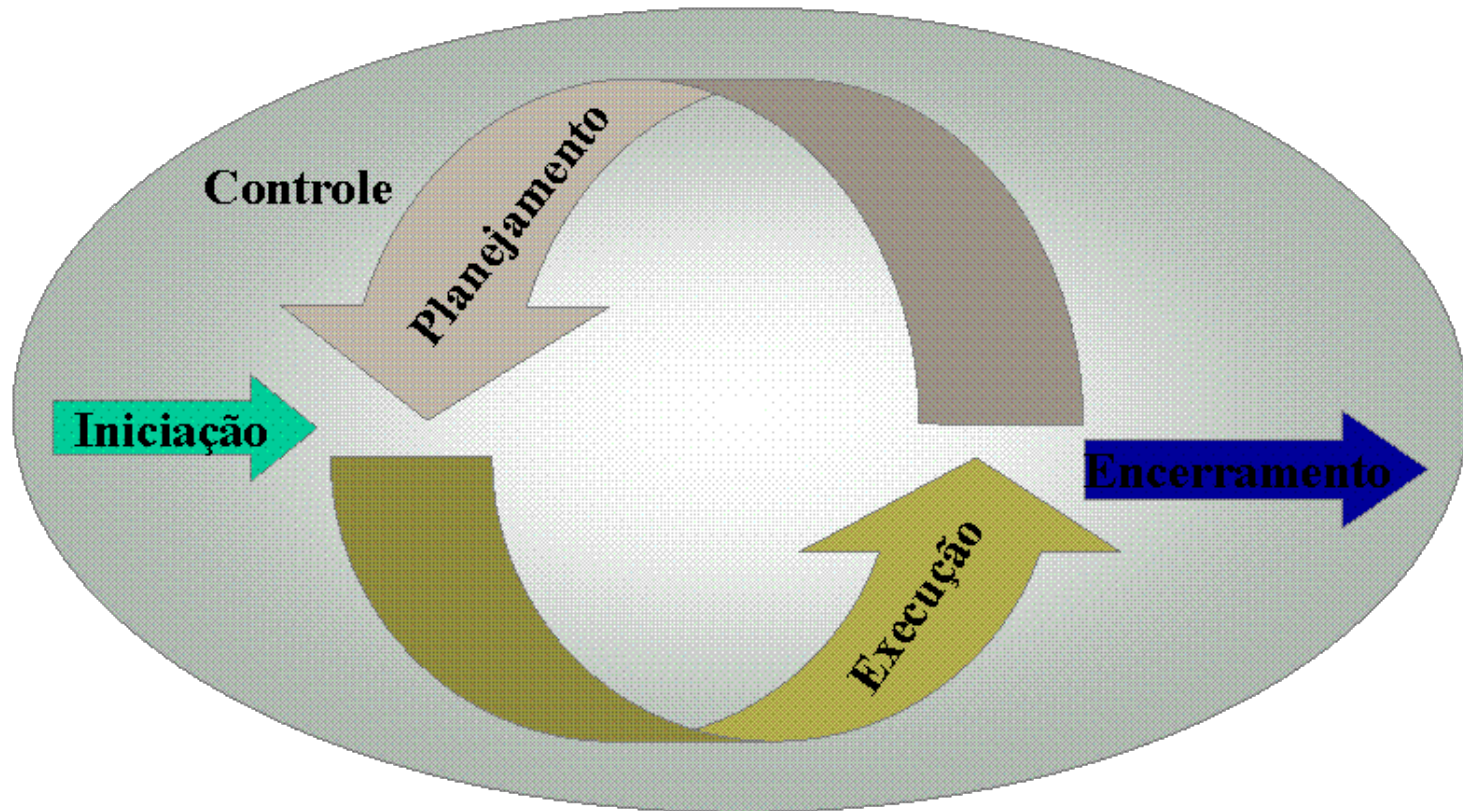
Gerenciamento de Projeto de Software

Grupos de Processos do PMBOK

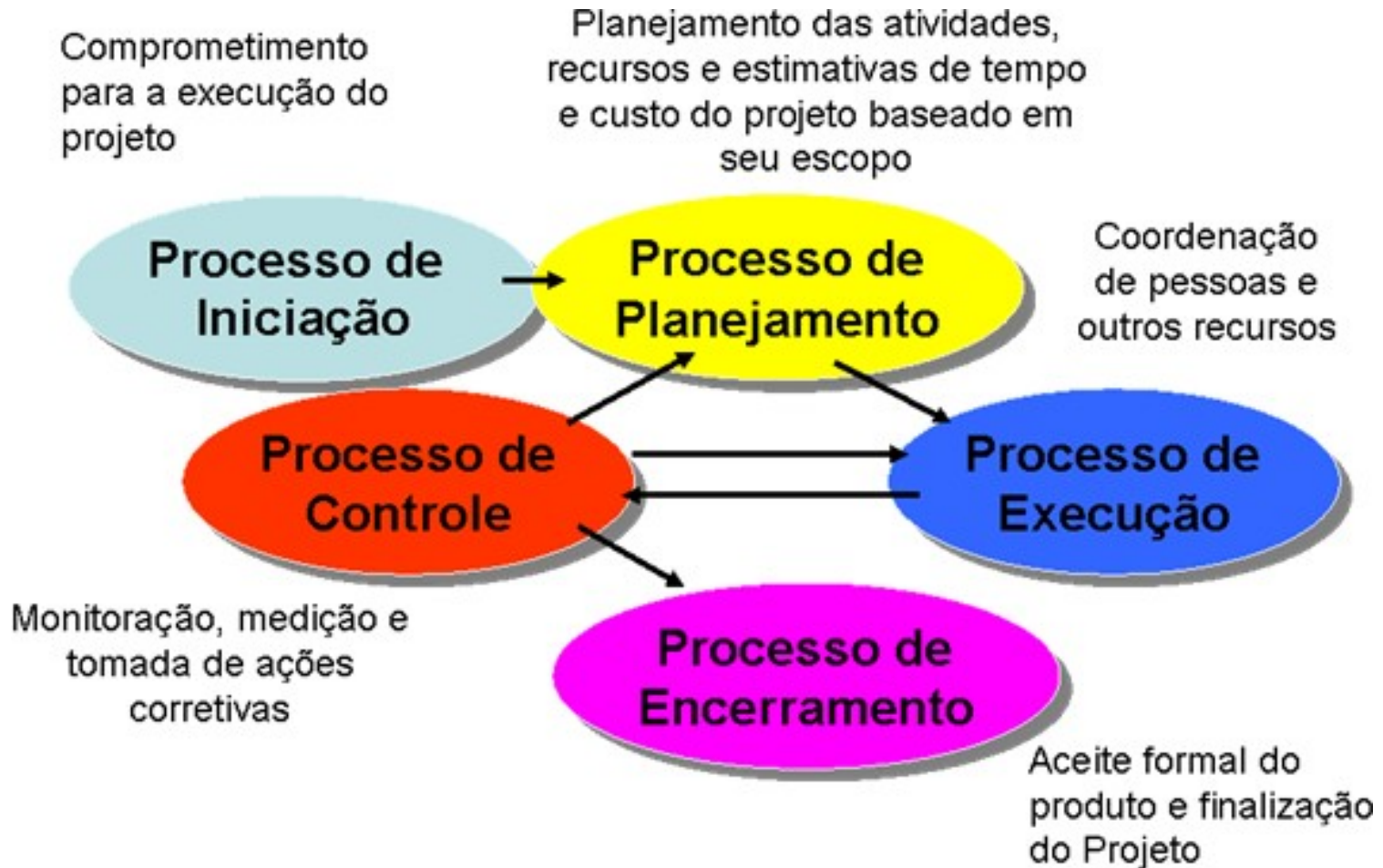


Gerenciamento de Projeto de Software

Ciclo Genérico



Gerenciamento de Projeto de Software (Ciclo Genérico Detalhado)

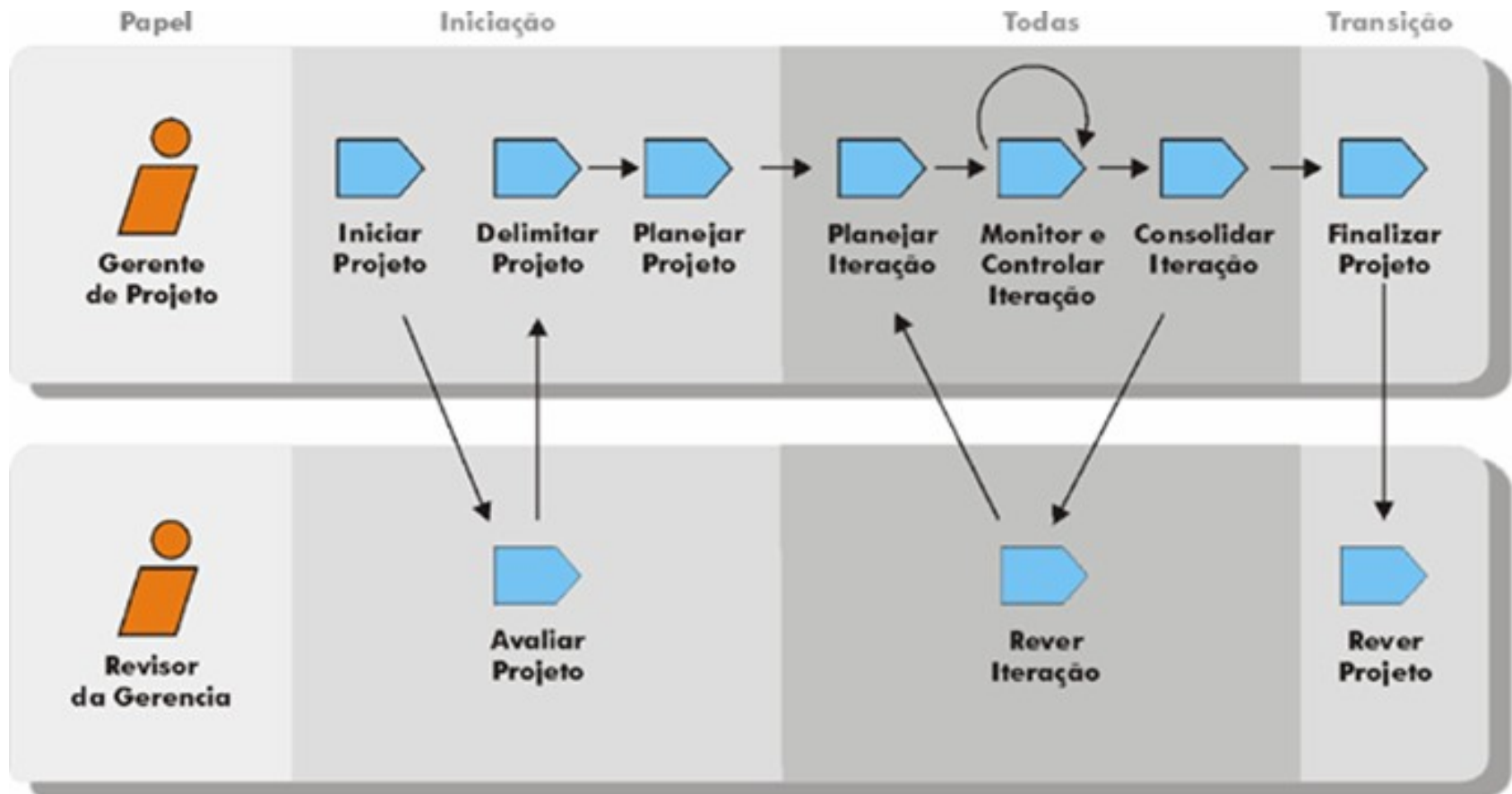


Gerenciamento de Projeto de Software (Ciclo **P**lan**D**o**C**ontrol**A**ct do PMBOK)



Gerenciamento de Projeto de Software

Exemplo



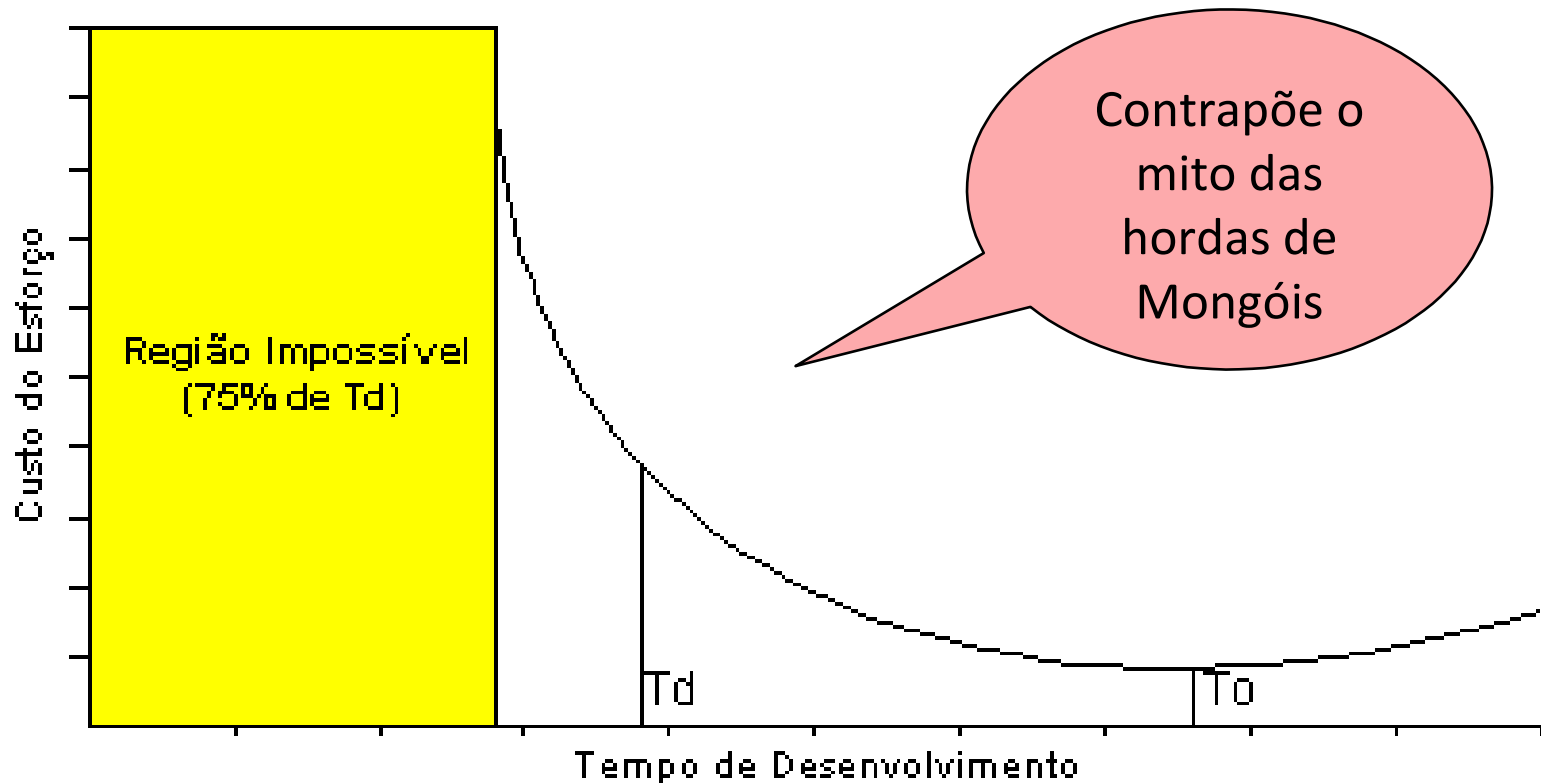
Gerenciamento de Projeto de Software

GP x AC do PMBOK

	Inicição	Planejamento	Execução	Monitoramento e Controle	Encerramento
Integração	XX	X	X	XX	X
Escopo		XXX		XX	
Tempo		XXXXX		X	
Custos		XX		X	
Qualidade		X	X	X	
Recursos Humanos		X	XX	X	
Comunicação		X	X	XX	
Riscos		XXXXX		X	
Aquisições		XX	XX	X	X

Gerenciamento de Projeto de Software

Relação Tempo x Custos



Processo de Software (SWEBOK, Capítulo 9)

- Consiste na definição, implementação, avaliação, medição, gerenciamento, mudanças e evolução do próprio processo de software. É uma **abstração** do **projeto** de software; uma definição de como o projeto será dividido em suas diversas fases e atividades (metodologia).
- A AC está dividida em quatro subáreas:
 1. Mudança e Implementação do Processo
 2. Definição do Processo
 3. Avaliação do Processo
 4. Medidas de Produtos e Processos

Processo de Software (Exemplos)

- Processo Unificado (Jacobson, Booch, Rumbaugh)
- RUP – Rational Unified Process (Rational/IBM)
- PRAXIS – PRocesso para Aplicativos eXtensíveis InterativoS (Wilson de Pádua Paula Filho)
 - Para projetos realizados individualmente ou por pequenas equipes, com duração de seis meses a um ano.
- XP – eXtreme Programming
 - Projetos cujos requisitos mudam com muita frequência, orientados a objetos, desenvolvido por equipes de até 12 desenvolvedores, com processo incremental (ou interativo).
- SEA UP (SEA Tecnologia)
- MetoDeS – Desenvolvida por Alunos como TCC

Ferramentas e Métodos (SWEBOK, Capítulo 10)

- Classifica ferramentas e métodos para a Engenharia de Software.
- Na sub-área Ferramentas, há uma tópico para cada uma das outras nove AC.
- Na sub-área Métodos, há três tópicos:
 - I. Heurísticos*, para a abordagem informal
 - II. Matemáticos, para a abordagem formal
 - III. Protótipos, para a abordagem do software baseada em formulários

* Metodologia, ou algoritmo, usado para resolver problemas por métodos que, embora não rigorosos, geralmente refletem o conhecimento humano e permitem obter uma solução satisfatória.

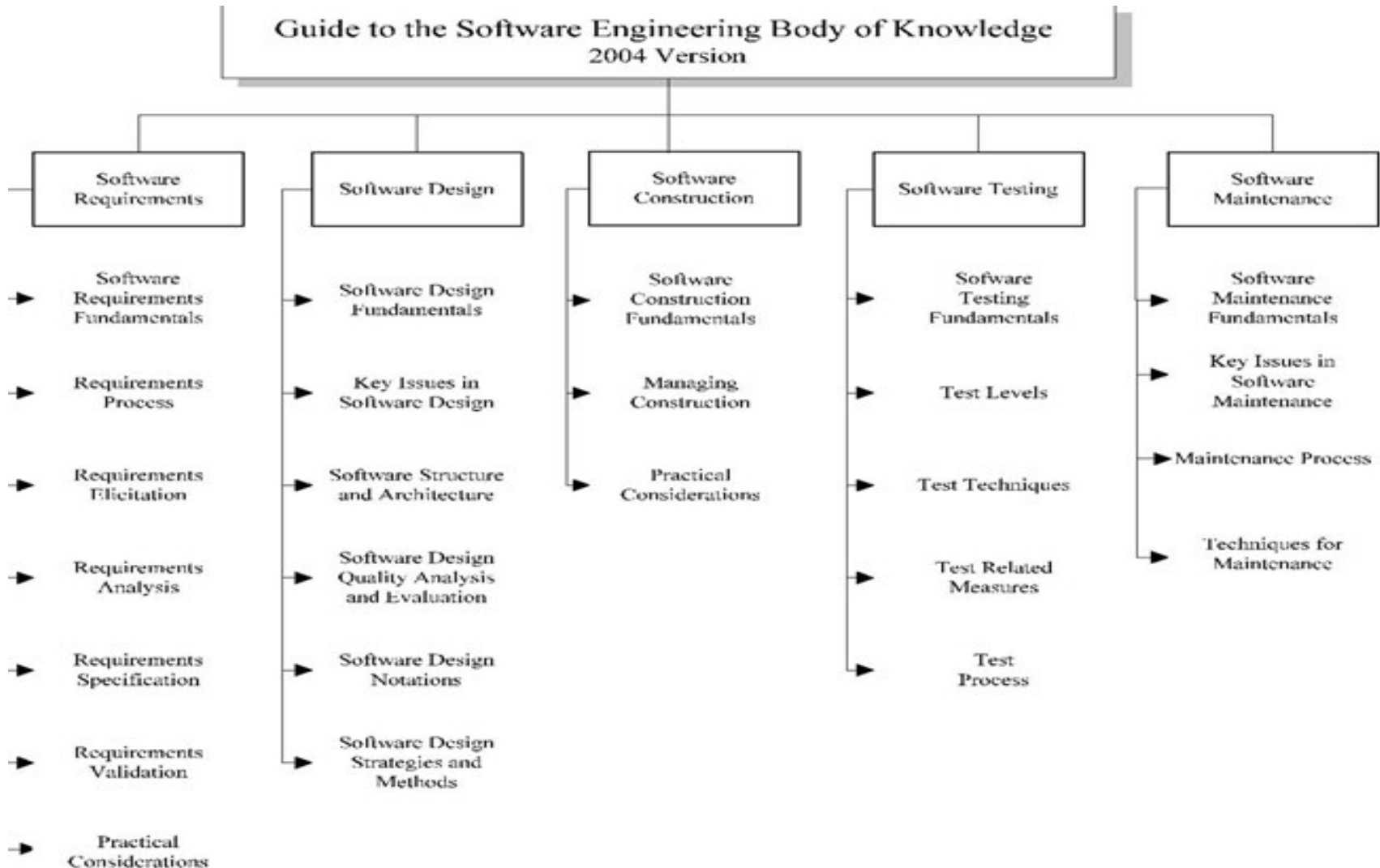
Qualidade de Software (SWEBOK, Capítulo 11)

- Relaciona-se com a qualidade de software e transcende o ciclo de vida do seu desenvolvimento. Como a preocupação com qualidade está presente toda a engenharia de software (e em toda a gestão de projetos), também é observada em todas as outras AC.
- A AC está dividida em três subáreas:
 1. Fundamentos da Qualidades de Software
 2. Processo de Gerenciamento da Qualidade de Software
 3. Considerações Práticas sobre a Qualidade de Software

Disciplinas Relacionadas (SWEBOK, Capítulo 12)

1. Engenharia da Computação
2. Ciência da Computação
3. Gestão de Negócios
4. Matemática
5. Gerenciamento de Projetos
6. Gestão da Qualidade
7. Ergonomia (acessibilidade e usabilidade)
8. Engenharia de Sistemas

Hierarquia das Cinco Primeiras Áreas de Conhecimento



Hierarquia das Cinco Últimas Áreas de Conhecimento e Disciplinas

Guide to the Software Engineering Body of Knowledge (2004 Version)

