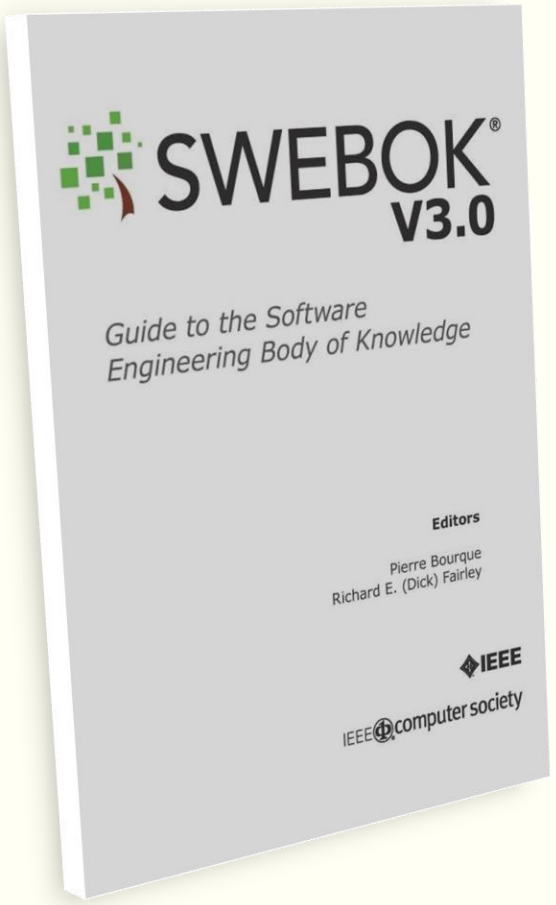
9
GERENCIAMENTO DA ENGENHARIA DE SOFTWARE

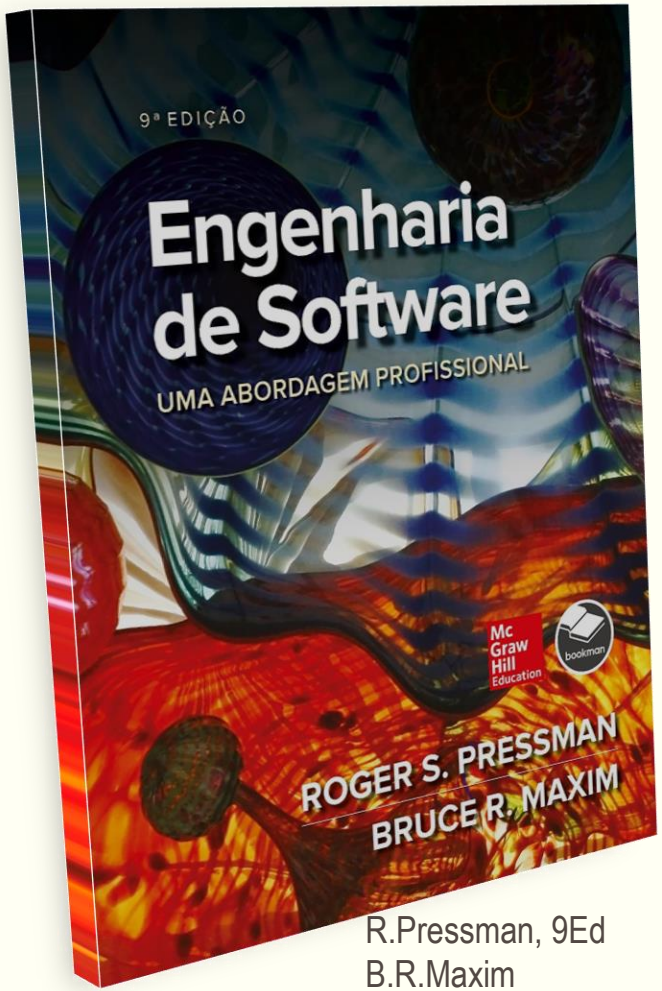
Prof. Ausberto S. Castro Vera
UENF – CCT – LCMAT
Ciência da Computação

Bibliografia Básica

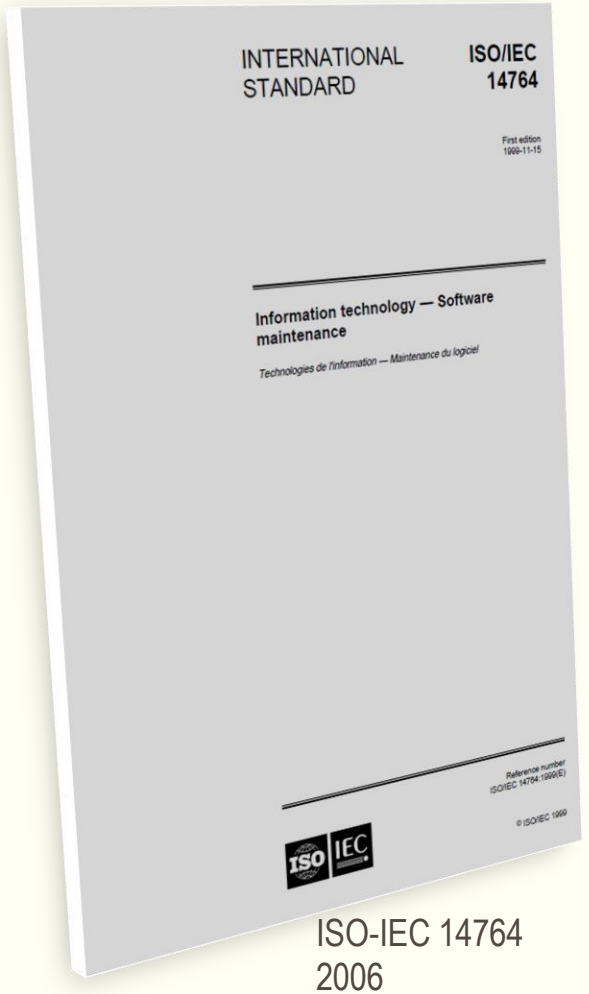


Bibliografia Básica

I.Sommerville, 10Ed
Pearson
2018

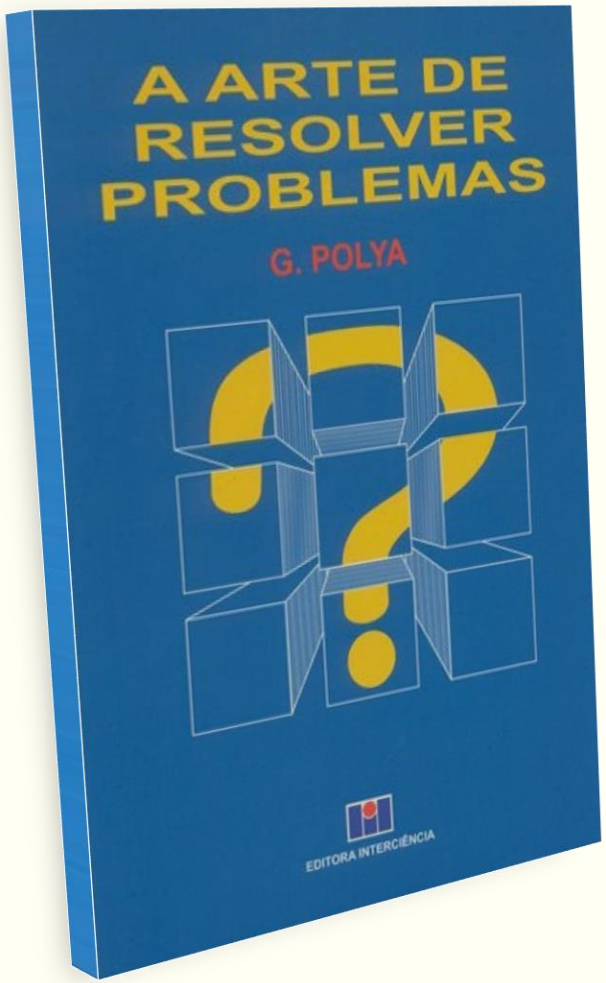


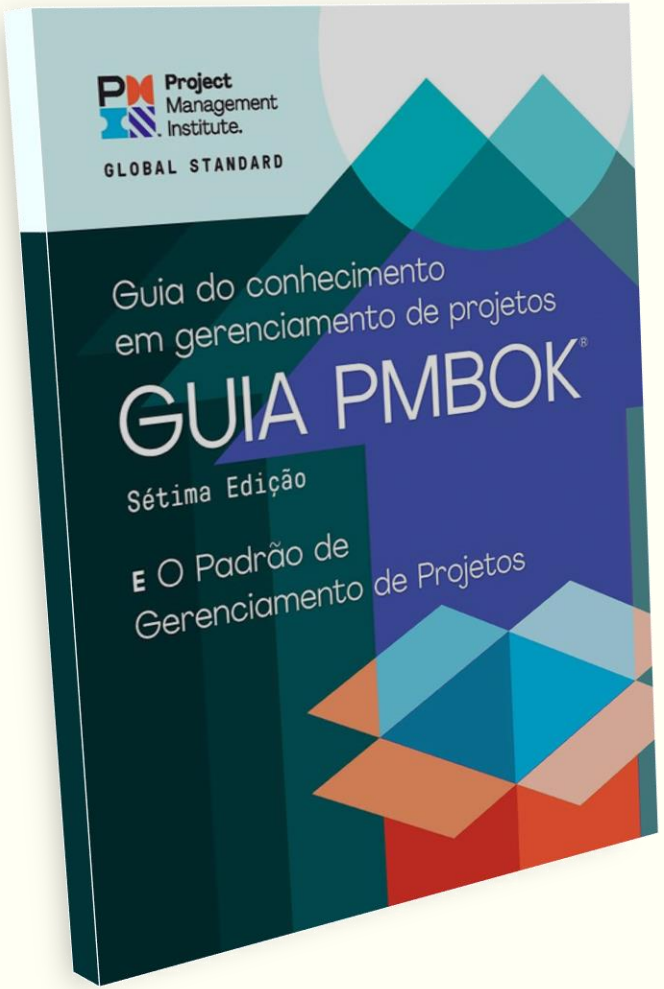
R.Pressman, 9Ed
B.R.Maxim
AMGH Editora
2021

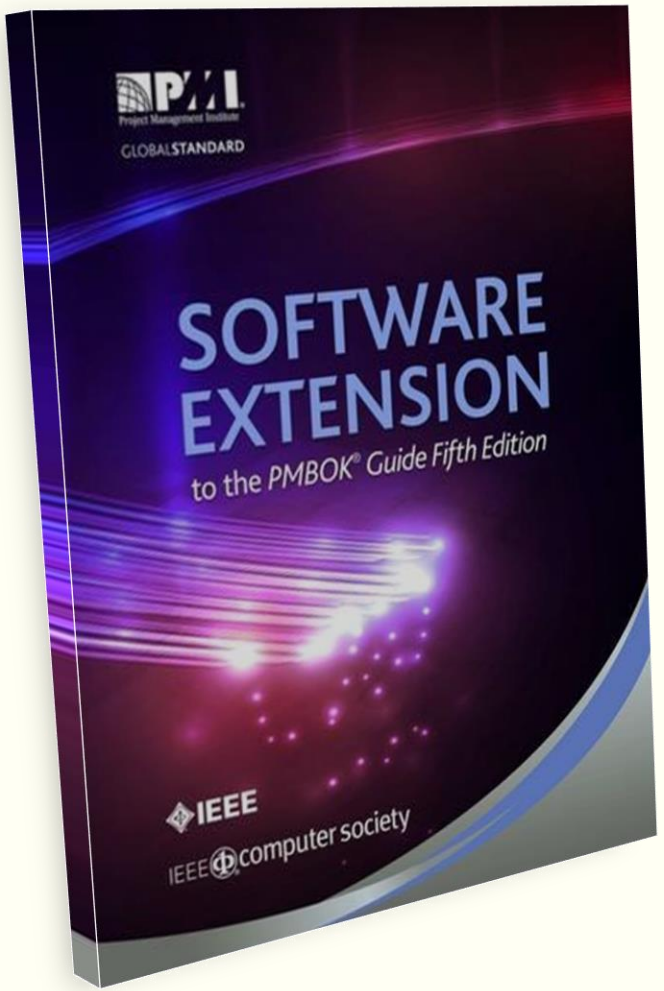


ISO-IEC 14764
2006

Bibliografia complementar







Gerenciamento da Engenharia de Software:

é definido como a **aplicação de atividades de gerenciamento** – *planejamento, coordenação, medição, monitoramento, controle e relatórios* – para garantir que os **produtos de software** e os **serviços de engenharia de software** sejam entregues de forma eficiente, eficaz e em benefício dos stakeholders..

Gerenciamento da Engenharia de Software

As **atividades** de *Gerenciamento de Engenharia de Software* ocorrem em três níveis:

- Gerenciamento organizacional e de infraestrutura,
- Gerenciamento de projetos e
- Gerenciamento do programa de medição.

Gerente de Engenharia Organizacional de Software (GEOS)

Gerente de Engenharia Organizacional de Software (GEOS):
é um profissional crucial para o sucesso de empresas que dependem de software em larga escala. Ele atua como um maestro, orquestrando o trabalho de equipes multidisciplinares para garantir que o desenvolvimento, a entrega e a manutenção de softwares atendam às necessidades da organização de forma eficiente, eficaz e alinhada com os objetivos estratégicos.

- Devem estar familiarizados com o *conhecimento de gerenciamento* de projetos e medição de software
- Devem possuir algum *conhecimento do domínio* alvo.
- Entender a *importância* das *políticas e procedimentos de gestão de pessoal* para contratação, treinamento e orientação de pessoal para o desenvolvimento de carreira, não apenas no nível do projeto, mas também para o sucesso de uma organização a longo prazo.
- Apresentar desafios únicos de *treinamento ou gerenciamento de pessoal*.
- Conhecer o *gerenciamento da comunicação* para uma compreensão precisa das necessidades do usuário, dos requisitos de software e dos designs de software.

Gerente de Engenharia Organizacional de Software (GEOS)

Responsabilidades:

- **Liderança estratégica:** O GEOS define a visão e a estratégia de engenharia de software para toda a organização, garantindo que ela esteja alinhada com os objetivos de negócio e as necessidades dos stakeholders.
- **Gerenciamento de equipes:** O GEOS lidera e gerencia equipes multidisciplinares de engenheiros de software, garantindo que eles estejam trabalhando de forma coesa e produtiva para alcançar os objetivos da organização.
- **Definição de processos e padronização:** O GEOS define e implementa processos e padrões de engenharia de software para garantir a qualidade, a consistência e a eficiência do desenvolvimento de software em toda a organização.
- **Gerenciamento de ferramentas e tecnologias:** O GEOS seleciona, implementa e gerencia as ferramentas e tecnologias necessárias para o desenvolvimento, a entrega e a manutenção de software.
- **Melhoria contínua:** O GEOS promove a cultura de melhoria contínua na equipe de engenharia de software, identificando oportunidades para otimizar processos, ferramentas e tecnologias.
- **Comunicação e colaboração:** O GEOS se comunica efetivamente com stakeholders internos e externos, garantindo que todos estejam informados sobre o progresso dos projetos de software e colaborando com outras áreas da organização para alcançar os objetivos comuns

Gerente de Engenharia Organizacional de Software (GEOS)

Habilidades e conhecimentos:

- **Sólida formação em engenharia de software:** O GEOS precisa ter um profundo conhecimento dos princípios, práticas e metodologias de engenharia de software.
- **Visão estratégica:** O GEOS precisa ter uma visão estratégica para poder definir a direção da engenharia de software da organização e alinhá-la com os objetivos de negócio.
- **Habilidades de liderança excepcionais:** O GEOS precisa ser um líder inspirador e motivador que possa liderar e gerenciar equipes multidisciplinares de forma eficaz.
- **Excelentes habilidades de comunicação:** O GEOS precisa ter excelentes habilidades de comunicação escrita e oral para se comunicar efetivamente com stakeholders internos e externos.
- **Habilidades de resolução de problemas:** O GEOS precisa ter excelentes habilidades de resolução de problemas para identificar e resolver problemas que possam surgir durante o desenvolvimento de software.
- **Pensamento crítico e analítico:** O GEOS precisa ter um pensamento crítico e analítico para avaliar diferentes soluções e tomar decisões informadas.
- **Conhecimento de negócios:** O GEOS precisa ter um bom conhecimento de negócios para entender as necessidades dos stakeholders e como o software pode agregar valor à organização.

Gerentes de Projetos e Programas Complexos

Um **Gerente de Projetos e Programas Complexos** é um profissional altamente especializado e experiente que *lidera* e *gerencia* projetos e programas de grande porte, abrangência e impacto.

Responsabilidades:

- **Planejar e definir os objetivos estratégicos do projeto ou programa:** trabalha em estreita colaboração com stakeholders para definir os objetivos estratégicos do projeto ou programa, garantindo que eles estejam alinhados com a missão e visão da organização.
- **Desenvolvimento do plano de gerenciamento:** é responsável por desenvolver um plano de gerenciamento abrangente que detalha o escopo do projeto, o cronograma, o orçamento, os recursos necessários, os riscos potenciais e as estratégias de mitigação.
- **Liderança da equipe:** lidera e motiva uma equipe multidisciplinar de profissionais de diversas áreas, garantindo que todos estejam trabalhando em conjunto para alcançar os objetivos do projeto ou programa.
- **Gerenciamento de riscos:** identifica, avalia e gerencia os riscos potenciais que podem afetar o projeto ou programa.
- **Monitoramento e controle:** monitora o progresso do projeto ou programa e implementa as medidas corretivas necessárias para garantir que ele esteja dentro do prazo, do orçamento e dos requisitos de qualidade.
- **Comunicação e stakeholder management:** se comunica efetivamente com stakeholders internos e externos, mantendo-os informados sobre o progresso do projeto ou programa e gerenciando suas expectativas.

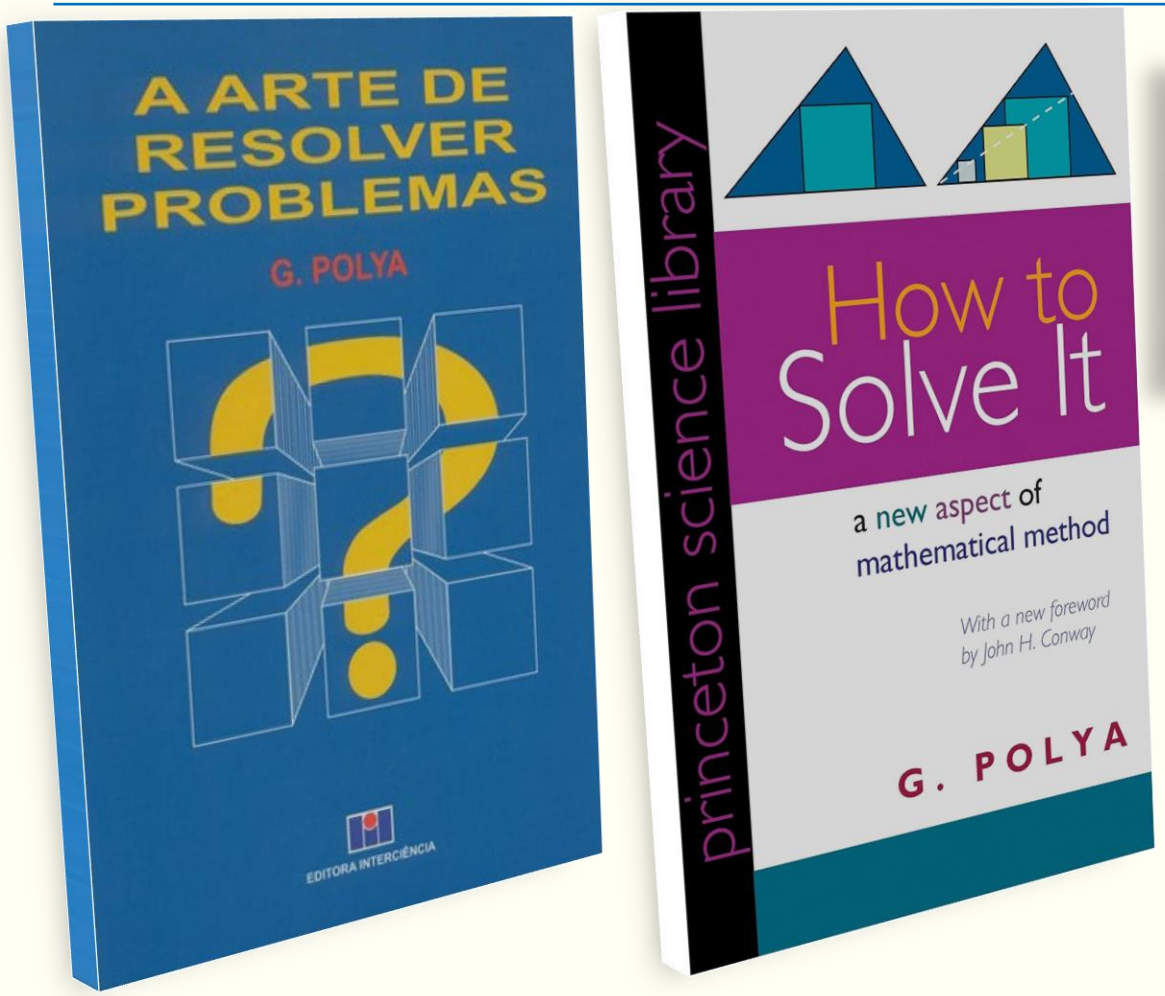
Gerentes de Projetos e Programas Complexos

Um **Gerente de Projetos e Programas Complexos** é um profissional altamente especializado e experiente que *lidera* e *gerencia* projetos e programas de grande porte, abrangência e impacto.

Habilidades e conhecimentos:

- **Liderança:** precisa ser um líder inspirador e motivador que possa liderar e motivar uma equipe multidisciplinar.
- **Comunicação:** precisa ter excelentes habilidades de comunicação escrita e oral para se comunicar efetivamente com stakeholders internos e externos.
- **Gerenciamento de tempo:** precisa ter excelentes habilidades de gerenciamento de tempo para garantir que o projeto ou programa esteja dentro do prazo.
- **Resolução de problemas:** precisa ter excelentes habilidades de resolução de problemas para identificar e resolver problemas que possam surgir durante o projeto ou programa.
- **Negociação:** precisa ter excelentes habilidades de negociação para negociar com stakeholders e fornecedores.
- **Conhecimento técnico:** precisa ter um bom conhecimento técnico da área do projeto ou programa.

Resolução de Problemas - Métodos



1. Compreensão do Problema
2. Estabelecimento de um Plano
3. Execução do Plano
4. Retrospecto

1. Identificação do Problema
2. Análise das Causas Possíveis
3. Coleta de Dados
4. Análise dos Dados
5. Formulação de Hipóteses
6. Teste das Hipóteses
7. Implementação da Solução
8. Monitoramento e Controle

Método de Resolução de Problemas em Engenharia de Software

EXEMPLO

Cenário:
Imagine que você está trabalhando como desenvolvedor em um aplicativo móvel de gerenciamento de tarefas. Um dos usuários relata que o aplicativo está travando frequentemente ao tentar adicionar novas tarefas.

Problema:
O aplicativo móvel de gerenciamento de tarefas está travando com frequência ao adicionar novas tarefas.

Método de Resolução de Problemas em Engenharia de Software

1. **Identificação do Problema:**
   - **Descrição do Problema:** O aplicativo trava ao adicionar novas tarefas.
   - **Impacto do Problema:** Os usuários não conseguem adicionar novas tarefas, o que limita a funcionalidade do aplicativo.
   - **Frequência do Problema:** O problema ocorre com frequência, afetando vários usuários.

2. **Análise das Causas Possíveis:**
   - **Causas Técnicas:**
     - Bug no código que lida com a adição de novas tarefas.
     - Vazamento de memória.
     - Problema de compatibilidade com o dispositivo ou sistema operacional do usuário.
   - **Causas Externas:**
     - Falha na rede do usuário.
     - Problema com o servidor do aplicativo.

3. **Coleta de Dados:**
   - Registrar os passos do usuário que levam ao travamento.
   - Obter logs do aplicativo e do dispositivo do usuário.
   - Reproduzir o problema em diferentes dispositivos e sistemas operacionais.
   - Coletar feedback de outros usuários que relataram o mesmo problema.

4. **Análise dos Dados:**
   - Analisar os logs do aplicativo e do dispositivo para identificar possíveis erros.
   - Comparar os passos dos usuários que relataram o problema para identificar padrões.
   - Tentar correlacionar o problema com outros eventos, como atualizações recentes do aplicativo ou do sistema operacional.

Método de Resolução de Problemas em Engenharia de Software

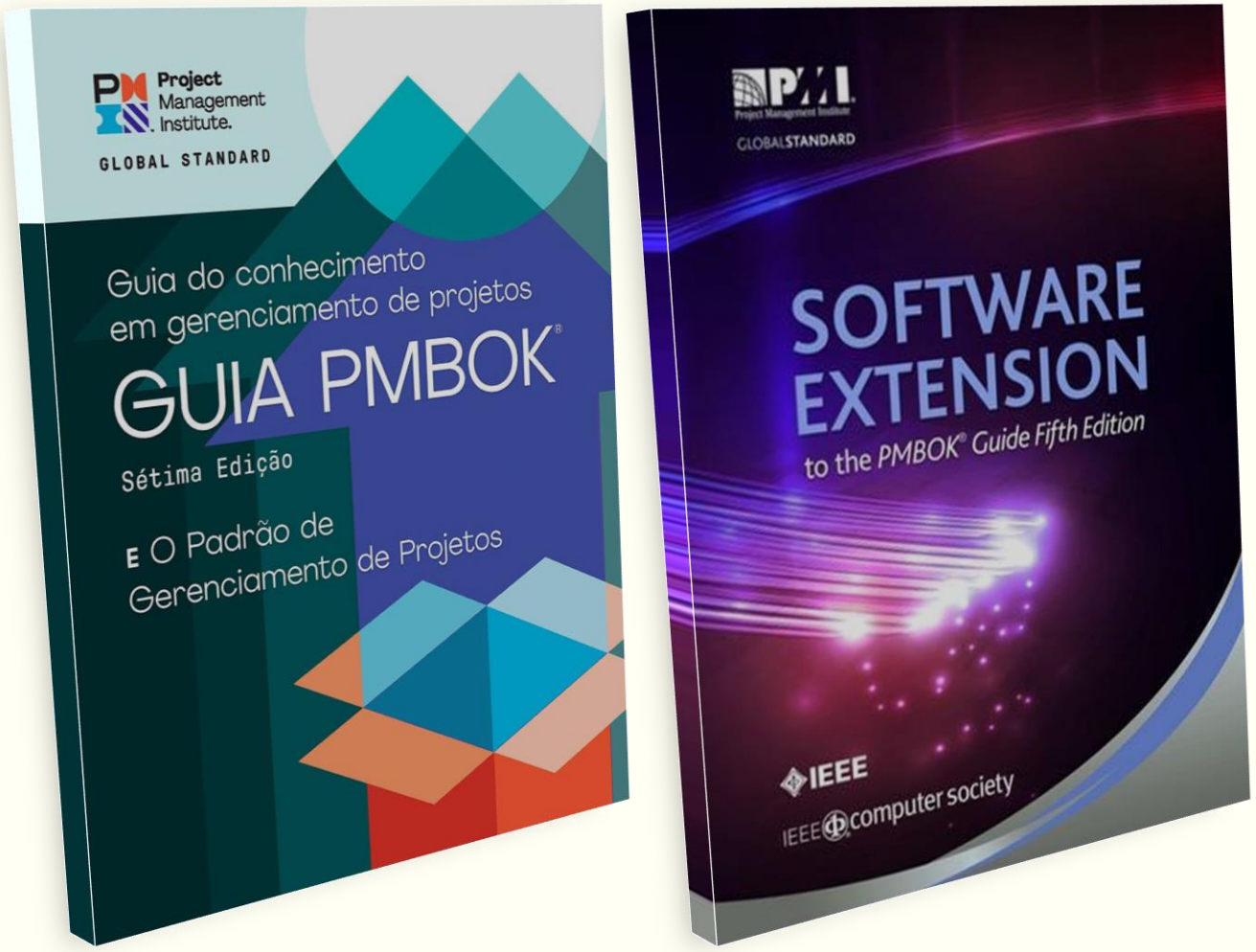
5. **Formulação de Hipóteses:**
   - Com base na análise dos dados, formular hipóteses sobre as causas mais prováveis do problema.
   - Priorizar as hipóteses com base na probabilidade e no impacto potencial.

6. **Teste das Hipóteses:**
   - Desenvolver testes para cada hipótese.
   - Executar os testes para verificar se as hipóteses estão corretas.
   - Refinar as hipóteses com base nos resultados dos testes.

7. **Implementação da Solução:**
   - Com base na análise e nos testes, identificar a solução mais provável para o problema.
   - Desenvolver e implementar a solução.
   - Testar a solução para garantir que o problema foi corrigido.

8. **Monitoramento e Controle:**
   - Monitorar o aplicativo após a implementação da solução para verificar se o problema foi realmente corrigido.
   - Coletar feedback dos usuários para garantir que a solução atendeu às suas expectativas.
   - Documentar o processo de resolução de problemas para referência futura.

Gerencia de Projetos



1. Gerenciamento de integração de projetos,
2. Gerenciamento de escopo de projetos,
3. Gerenciamento de tempo de projetos,
4. Gerenciamento de custos de projetos,
5. Gerenciamento de qualidade de projetos,
6. Gerenciamento de recursos humanos de projetos,
7. Gerenciamento de comunicações de projetos,
8. Gerenciamento de riscos de projetos,
9. Gerenciamento de aquisições de projetos e
10. Gerenciamento de stakeholders do Projeto

Infelizmente ...

… uma *percepção comum* da indústria de software é que os **produtos de software** são entregues

- *com atraso,*
- *acima do orçamento,*
- *de baixa qualidade e*
- *com funcionalidade incompleta*.

O gerenciamento informado sobre **medições** pode *ajudar a melhorar* a percepção e a realidade.

Em essência,

- a ***gestão sem medição*** (qualitativa e quantitativa) sugere uma falta de disciplina, e
- a ***medição sem gestão*** sugere uma falta de propósito ou contexto.

A gestão eficaz requer uma combinação de medição e experiência.

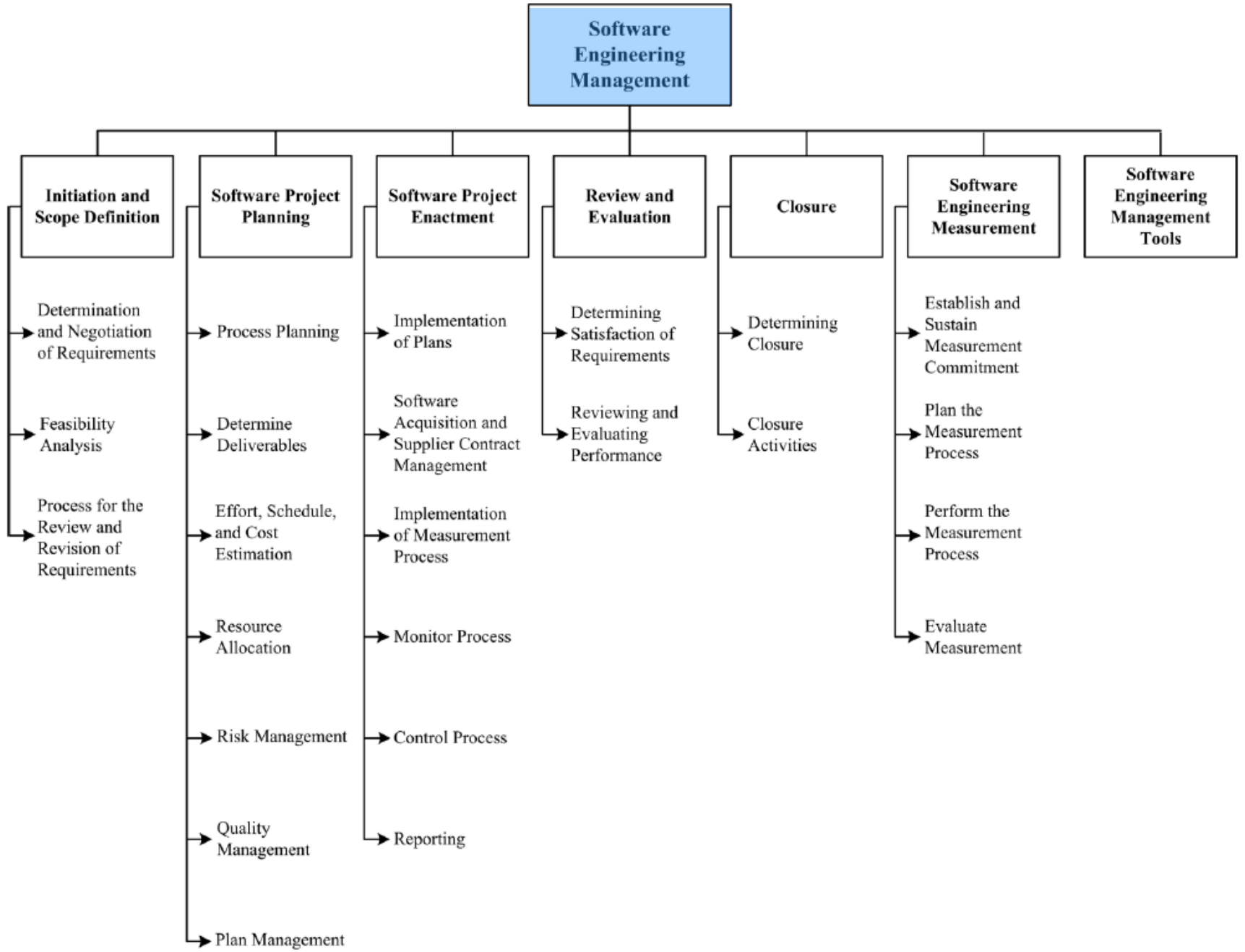
Gestão e Medição

A **gestão** é um sistema de *processos* e *controles* necessários para atingir os objetivos estratégicos definidos pela organização.

A **medição** refere-se à *atribuição de valores* e *rótulos* a **produtos, processos e recursos de trabalho** de engenharia de software, além dos modelos derivados deles, sejam esses modelos desenvolvidos usando técnicas estatísticas ou outras

Áreas de Conhecimento e Engenharia de Software

- A área **Fundamentos de Engenharia** descreve alguns conceitos gerais de medição que são diretamente aplicáveis à seção Medição de Engenharia de Software
- A área de **Requisitos de Software** descreve algumas das atividades que devem ser executadas durante a fase de Iniciação e definição do Escopo do projeto.
- A área de **Gerenciamento de Configuração de Software** lida com identificação, controle, contabilidade de status e auditoria de configurações de software, juntamente com gerenciamento de liberação e entrega de software e ferramentas de gerenciamento de configuração de software.
- A área **Processo de Engenharia de Software** descreve modelos de ciclo de vida de software e as relações entre processos e produtos de trabalho.
- A área **Qualidade do Software** enfatiza a qualidade como objetivo de gerenciamento e como objetivo de muitas atividades de engenharia de software.
- A área **Economia da Engenharia de Software** discute como tomar decisões relacionadas a software em um contexto de negócios.


Software Engineering Management
Initiation and Scope Definition
Determination and Negotiation of Requirements
Feasibility Analysis
Process for the Review and Revision of Requirements
Software Project Planning
Process Planning
Determine Deliverables
Effort, Schedule, and Cost Estimation
Resource Allocation
Risk Management
Quality Management
Plan Management
Software Project Enactment
Implementation of Plans
Software Acquisition and Supplier Contract Management
Implementation of Measurement Process
Monitor Process
Control Process
Reporting
Review and Evaluation
Determining Satisfaction of Requirements
Reviewing and Evaluating Performance
Closure
Determining Closure
Closure Activities
Software Engineering Measurement
Establish and Sustain Measurement Commitment
Plan the Measurement Process
Perform the Measurement Process
Evaluate Measurement
Software Engineering Management Tools

Gerenciamento da Engenharia de Software - Conteúdo

1. **Iniciação e Definição de Escopo,**
   que tratam da decisão de embarcar em um projeto de engenharia de software;
2. **Planejamento de Projeto de Software,**
   que aborda as atividades realizadas para preparar um projeto de engenharia de software bem-sucedido do ponto de vista gerencial;
3. **Aprovação de Projeto de Software,**
   que trata das atividades de gerenciamento de engenharia de software geralmente aceitas que ocorrem durante a execução de um projeto de engenharia de software;
4. **Revisão e Avaliação,**
   que tratam de garantir que as atividades de engenharia técnica, de cronograma, de custo e de qualidade sejam satisfatórias;
5. **Encerramento,**
   que aborda as atividades realizadas para concluir um projeto;
6. **Medição de Engenharia de Software,**
   que trata do desenvolvimento e implementação eficazes de programas de medição em organizações de engenharia de software;
7. **Ferramentas de gerenciamento de engenharia de software,**
   que descreve a seleção e o uso de ferramentas para gerenciar um projeto de engenharia de software.

Gerenciamento da Engenharia de Software - Conteúdo

1. **Iniciação e Definição de Escopo**
   - 1.1. Determinação e Negociação de Requisitos
   - 1.2. Análise de Viabilidade
   - 1.3. Processo para Revisão e Análise de Requisitos

2. **Planejamento de Projeto de Software**
   - 2.1. Planejamento de Processos
   - 2.2. Determinar os resultados
   - 2.3. Estimativa de esforço, cronograma e custos
   - 2.4. Alocação de recursos
   - 2.5. Gerenciamento de riscos
   - 2.6. Gestão da Qualidade
   - 2.7. Gestão do Plano

3. **Aprovação do Projeto de Software**
   - 3.1. Implementação de Planos
   - 3.2. Aquisição de Software e Gestão de Contratos de Fornecedores
   - 3.3. Implementação do Processo de Medição
   - 3.4. Monitorar Processo
   - 3.5. Processo de controle
   - 3.6. Comunicando

4. **Revisão e avaliação**
   - 4.1. Determinando a satisfação dos requisitos
   - 4.2. Revendo e avaliando o desempenho

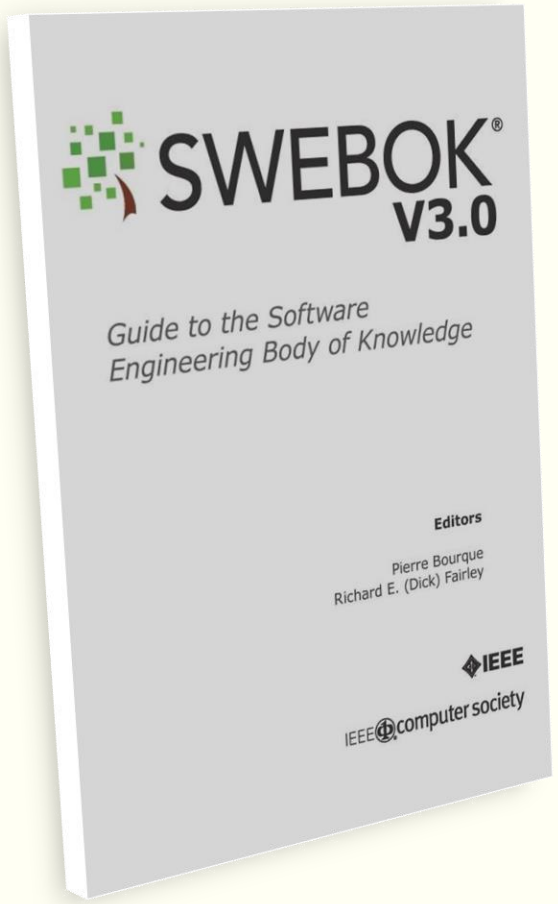
5. **Encerramento**
   - 5.1. Determinando o Fechamento
   - 5.2. Atividades de encerramento

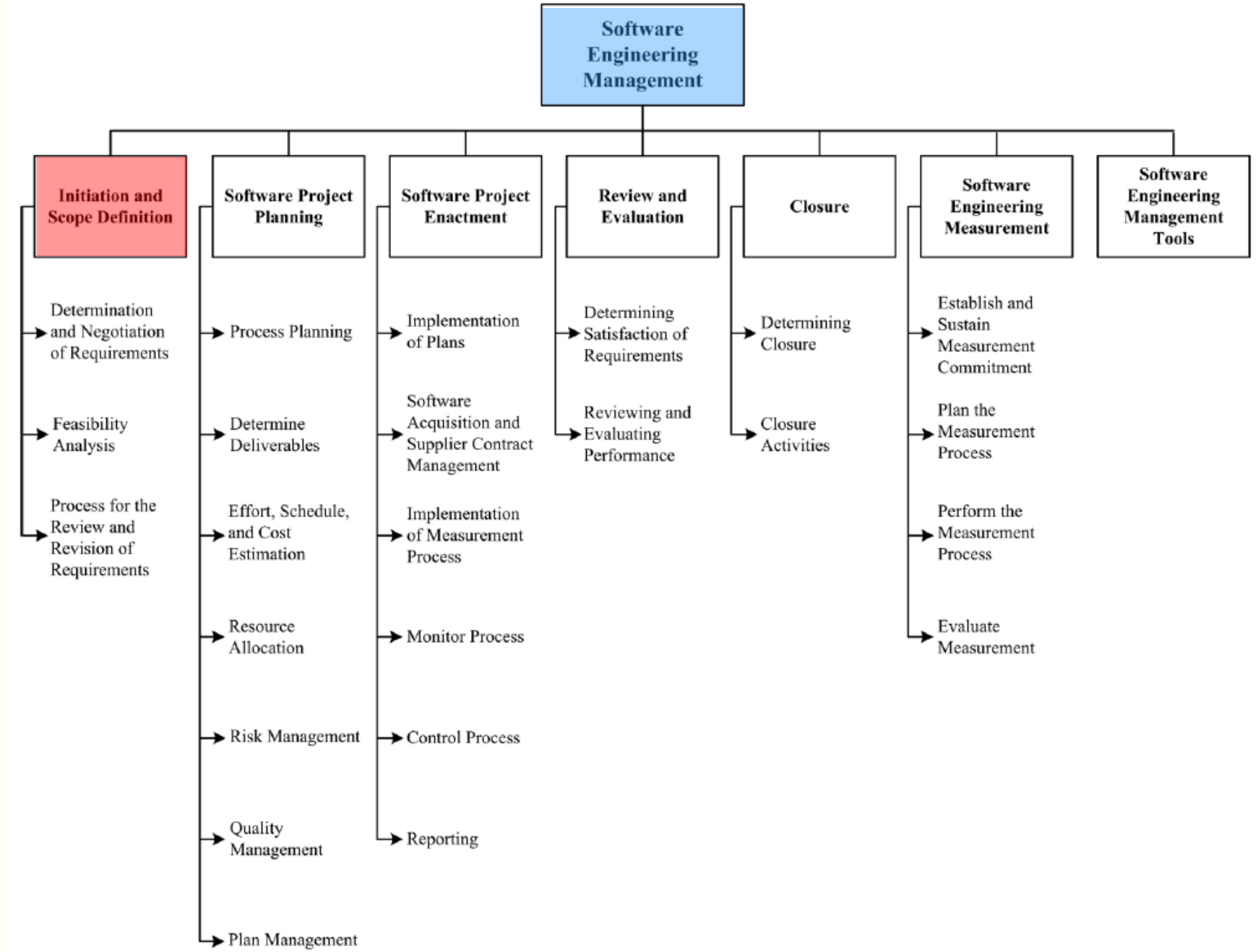
6. **Medição de Engenharia de Software**
   - 6.1. Estabelecer e manter o compromisso de medição
   - 6.2. Planeje o processo de medição
   - 6.3. Execute o processo de medição
   - 6.4. Avaliar medição

7. **Ferramentas de gerenciamento de Engenharia de Software**

Gerenciamento da Engenharia de Software

Gerenciamento da Engenharia de Software é definido como a **aplicação de atividades de gerenciamento** – *planejamento, coordenação, medição, monitoramento, controle e relatórios* – para garantir que os **produtos de software** e os **serviços de engenharia de software** sejam entregues de forma eficiente, eficaz e em benefício dos stakeholders.




Software Engineering Management
Initiation and Scope Definition
Determination and Negotiation of Requirements
Feasibility Analysis
Process for the Review and Revision of Requirements
Software Project Planning
Process Planning
Determine Deliverables
Effort, Schedule, and Cost Estimation
Resource Allocation
Risk Management
Quality Management
Plan Management
Software Project Enactment
Implementation of Plans
Software Acquisition and Supplier Contract Management
Implementation of Measurement Process
Monitor Process
Control Process
Reporting
Review and Evaluation
Determining Satisfaction of Requirements
Reviewing and Evaluating Performance
Closure
Determining Closure
Closure Activities
Software Engineering Measurement
Establish and Sustain Measurement Commitment
Plan the Measurement Process
Perform the Measurement Process
Evaluate Measurement
Software Engineering Management Tools

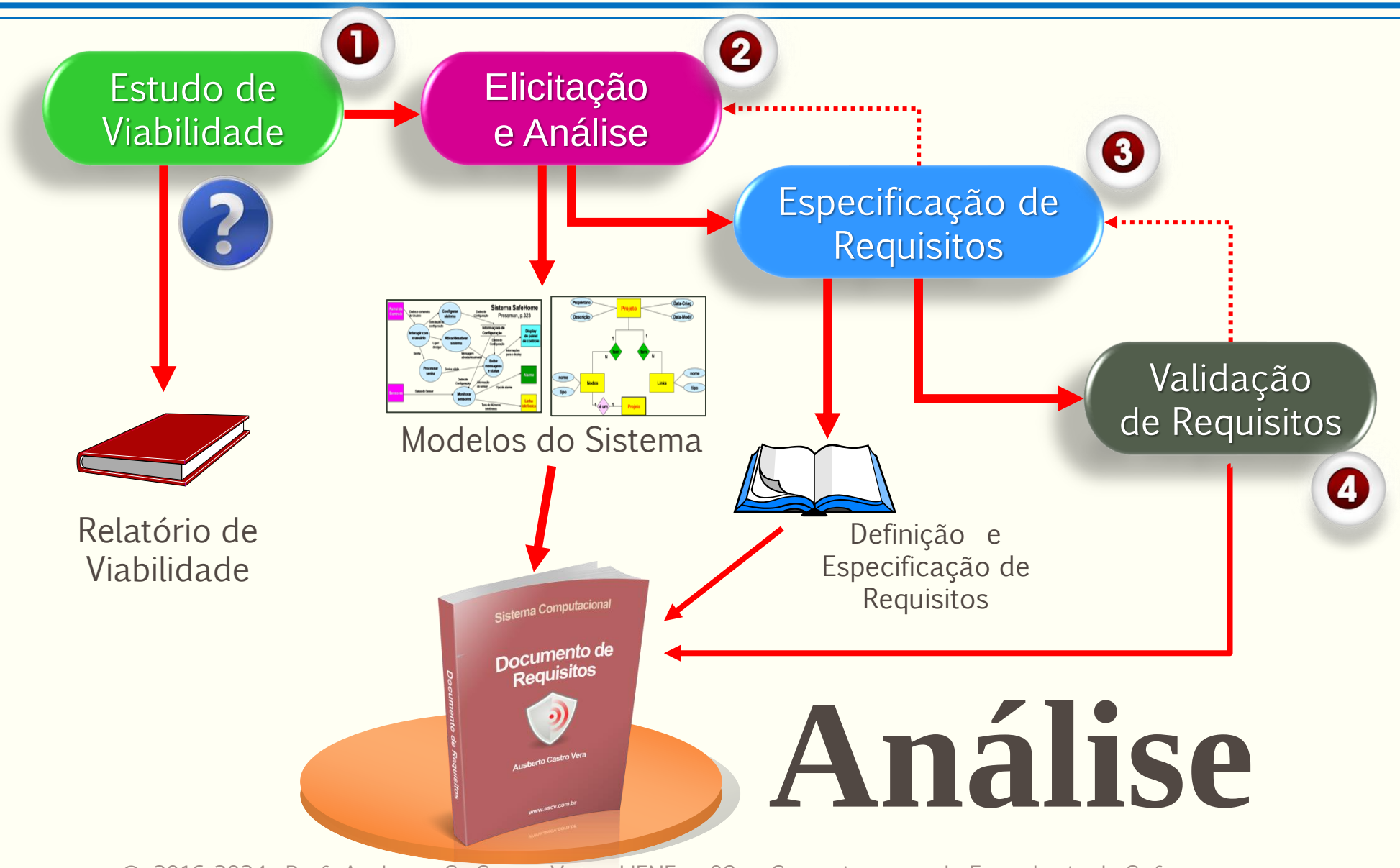
1. Iniciação e Definição de Escopo

O foco dessas atividades está na *determinação eficaz dos requisitos de software* usando vários *métodos de elicitação* e na avaliação da *viabilidade do projeto* sob vários pontos de vista.

Uma vez estabelecida a viabilidade do projeto, as tarefas restantes nesta seção são a *especificação dos requisitos* e a seleção dos *processos para revisão* (revisión) e análise (review) dos requisitos.

1.1. Determinação e Negociação de Requisitos
1.2. Análise de Viabilidade
1.3. Processo para Revisão e Análise de Requisitos

Processo de Engenharia de Requisitos



1.1. Determinação e Negociação de Requisitos

- A *determinação e a negociação* de requisitos estabelecem os *limites visíveis* para o conjunto de tarefas que estão sendo realizadas (consulte Área de Conhecimento Requisitos de Software).

- As atividades incluem *elicitação*, *análise*, *especificação e validação* de requisitos.

- Os *métodos e técnicas* devem ser seleccionados e aplicados, tendo em conta as diversas perspectivas dos stakeholders. Isso leva à *determinação do escopo do projeto*, a fim de cumprir os objetivos e satisfazer as restrições.

1.2 Análise de Viabilidade

❖ **Objectivo**

desenvolver uma descrição clara dos objectivos do projecto e avaliar abordagens alternativas, a fim de determinar se o projecto proposto é a melhor alternativa, dadas as restrições de tecnologia, recursos, finanças e considerações sociais/políticas.

❖ **Documentação**

- declaração inicial do escopo do projeto e do produto,
- as entregas do projeto,
- as restrições de duração do projeto e
- uma estimativa dos recursos necessários.

❖ **Recursos**

- número suficiente de pessoas que possuem as competências,
- instalações,
- infra-estruturas e apoio necessários (interna ou externamente).

- A análise de viabilidade muitas vezes requer *estimativas aproximadas* de esforço e *custo baseadas* em métodos apropriados

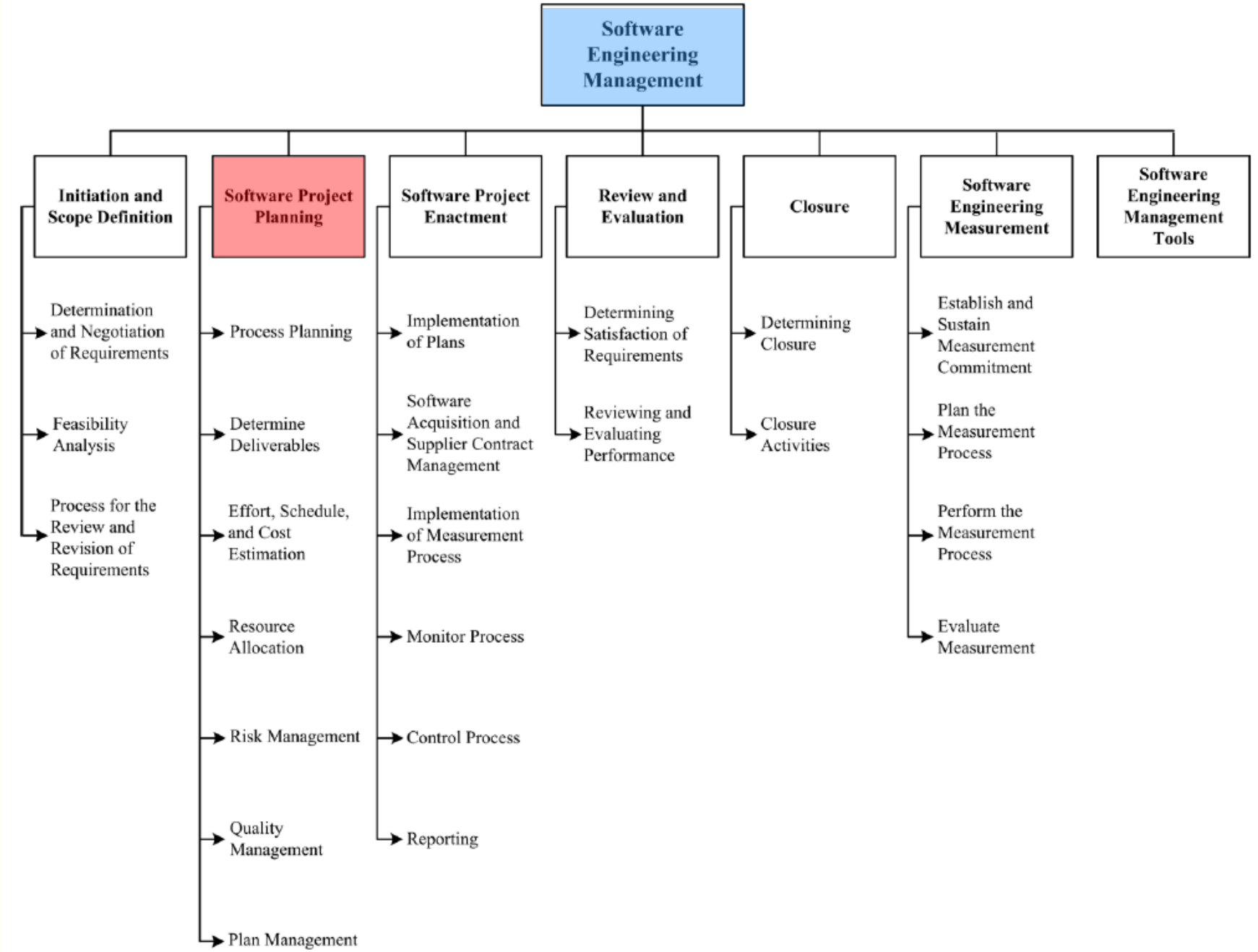
1.3. Processo para Revisão e Análise de Requisitos

1. **Acordo**
   Dada a inevitabilidade da mudança, as partes interessadas devem chegar a acordo sobre os meios pelos quais os requisitos e o âmbito devem ser revistos e revisados (por exemplo, procedimentos de gestão de mudanças, retrospetivas de ciclo iterativo).

2. **Revisão pontual**
   Isto implica claramente que o âmbito e os requisitos não serão "definidos em pedra", mas podem e devem ser revistos em pontos predeterminados à medida que o projecto se desenvolve (por exemplo, no momento em que as prioridades do backlog são criadas ou nas revisões de marcos).

3. **Análise de rastreabilidade e risco**
   Se as alterações forem aceitas, alguma forma de *análise de rastreabilidade* e *análise de risco* deverá ser usada para determinar o *impacto dessas alterações* (Ver Gerenciamento de riscos)

- Uma abordagem de ***mudança gerenciada*** também pode formar a base para a avaliação do sucesso durante o encerramento de um ciclo incremental ou de um projeto inteiro, com base nas mudanças que ocorreram ao longo do caminho (ver Encerramento).


Software Engineering Management
Initiation and Scope Definition
Software Project Planning
Software Project Enactment
Review and Evaluation
Closure
Software Engineering Measurement
Software Engineering Management Tools
Determination and Negotiation of Requirements
Feasibility Analysis
Process for the Review and Revision of Requirements
Process Planning
Determine Deliverables
Effort, Schedule, and Cost Estimation
Resource Allocation
Risk Management
Quality Management
Plan Management
Implementation of Plans
Software Acquisition and Supplier Contract Management
Implementation of Measurement Process
Monitor Process
Control Process
Reporting
Determining Satisfaction of Requirements
Reviewing and Evaluating Performance
Determining Closure
Closure Activities
Establish and Sustain Measurement Commitment
Plan the Measurement Process
Perform the Measurement Process
Evaluate Measurement

2. Planejamento de Projeto de Software

2.1. Planejamento de Processos
2.2. Determinar produtos entregáveis
2.3. Estimativa de esforço, cronograma e custos
2.4. Alocação de recursos
2.5. Gerenciamento de Riscos
2.6. Gestão da Qualidade
2.7. Gestão do Plano

2. Planejamento de Projeto de Software

1. **Modelo de CVDS**

   Seleção de um modelo apropriado e talvez adaptá-lo com base
   - no escopo do projeto,
   - nos requisitos de software
   - em uma avaliação de riscos.
   - natureza do domínio do aplicativo,
   - a complexidade funcional e técnica
   - requisitos de qualidade de software (consulte Requisitos de qualidade de software no KA de qualidade de software).

2. **Avaliação de riscos**
   - deve ser um elemento do planeamento inicial do projecto,
   - o “perfil de risco” do projecto deve ser discutido e aceite por todas as partes interessadas relevantes.
   - Os processos de gerenciamento de qualidade de software
     - parte do processo de planejamento
     - procedimentos e responsabilidades para garantia de qualidade de software,
     - verificação e validação,
     - revisões e auditorias
     - análise e revisão contínua do plano do projeto

2.1. Planejamento de Processos

Modelos de ciclo de vida de desenvolvimento de software (CVDS) :

- **CVDS s preditivos** são caracterizados
  - pelo desenvolvimento de *requisitos de software detalhados*,
  - *planejamento detalhado* do projeto e
  - planejamento mínimo para *iteração entre as fases de desenvolvimento*.

  - Um CVDS altamente preditivo executa os processos em uma sequência linear com revisões nas fases anteriores apenas quando necessário

- **CVDS s adaptativos** são projetados para acomodar *requisitos de software emergentes* e *ajustes iterativos* de planos.
  - CVDSs adaptativos são caracterizados por ciclos de desenvolvimento iterativos

- **CVDS conhecidos**: modelos em cascata, incremental e espiral, desenvolvimento ágil

- **Métodos e ferramentas relevantes** devem ser selecionados como parte do planejamento

2.2. Determinar produtos entregáveis

- **Identificação de Produtos**
  O(s) produto(s) de trabalho de cada atividade do projeto devem ser identificados e caracterizados.
  - documentos de design de arquitetura de software,
  - relatórios de inspeção,
  - software testado

- **Reutilização:**
  Devem ser avaliadas oportunidades para *reutilizar componentes de software* de projetos anteriores ou para utilizar *produtos de software prontos* para uso.
- **Aquisição de software**
- **Uso de terceiros** para desenvolver resultados devem ser planejados
- **Seleção de fornecedores**

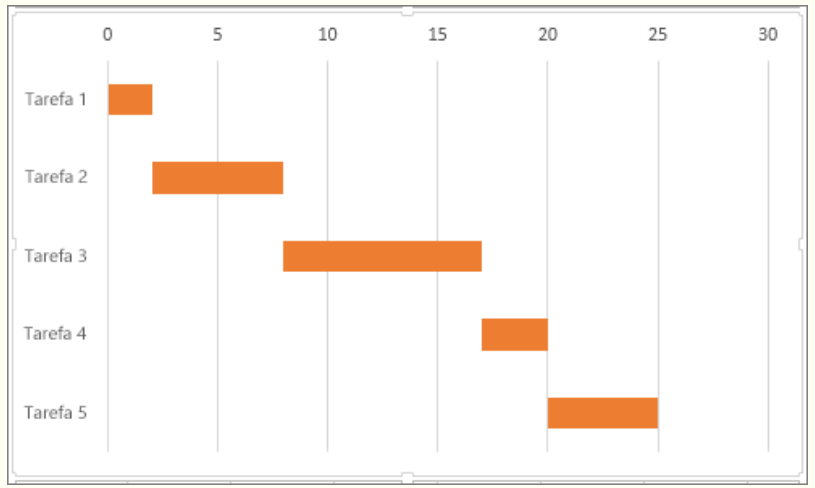
2.3. Estimativa de esforço, cronograma e custos

A ***estimativa de esforço, cronograma e custos*** (EEC) em Engenharia de Software é o processo de prever o esforço, o cronograma e os custos de um *projeto de software*. Essa é uma *parte crucial do planejamento* de software, pois permite que **gerentes de projeto, desenvolvedores e clientes** tomem decisões informadas sobre o projeto.

- **Esforço** é a quantidade de trabalho que precisa ser feito para concluir o projeto. Isso é geralmente medido em horas-homem ou dias-homem.

- **Cronograma** é o tempo que levará para concluir o projeto. Isso é geralmente medido em semanas, meses ou anos.

- **Custos** são os recursos financeiros que serão necessários para concluir o projeto. Isso inclui os **salários** dos desenvolvedores, o **custo** do hardware e software e outros **custos indiretos**.



2.3. Estimativa de esforço, cronograma e custos

- **Estimativa de Esforço**
  - Determinada usando um *modelo de estimativa* calibrado com base em *dados históricos* de tamanho e esforço (quando disponíveis) e outros métodos relevantes, como *julgamento de especialistas* e analogia.
  - As dependências de tarefas podem ser estabelecidas e oportunidades potenciais para completar tarefas simultânea e sequencialmente podem ser identificadas e documentadas usando um *gráfico de Gantt*, por exemplo.
- **Cronograma**

  Para *projetos CVDS preditivos*, o cronograma esperado de tarefas com horários de início, durações e horários de término projetados normalmente é produzido durante o planejamento.

  Para *projetos CVDS adaptativos*, uma estimativa geral de esforço e cronograma é normalmente desenvolvida a partir do entendimento inicial dos requisitos ou, alternativamente, restrições de esforço e cronograma globais podem ser especificadas e usadas para determinar uma estimativa inicial do número de ciclos iterativos e estimativas de esforço e outros recursos alocados a cada ciclo.
- **Custos**
  - Os requisitos de recursos (pessoas e ferramentas) podem ser traduzidos em estimativas de custos.

A **estimativa inicial** de esforço, cronograma e custo é uma **atividade iterativa** que deve ser negociada e revisada entre as partes interessadas afetadas até que seja alcançado consenso sobre os recursos e o tempo disponível para a conclusão do projeto.

2.4 Alocação de Recursos

- **Equipamentos, instalações e pessoas** devem ser alocados para as tarefas identificadas, incluindo a *atribuição de responsabilidades* para a conclusão de vários elementos de um projeto e do projeto global.

- Para cada uma das *tarefas* pode ser utilizada uma **matriz** que mostre
  - Quem é responsável,
  - Quem é consultado,
  - Quem é informado.

- A **alocação de recursos** baseia-se e é limitada pela *disponibilidade de recursos* e pela sua *utilização óptima*, bem como por questões relacionadas com o pessoal

Prof. Dr. Ausberto S. Castro Vera
Ciência da Computação
UENF-CCT-LCMAT
Campos, RJ

ascv@uenf.br