

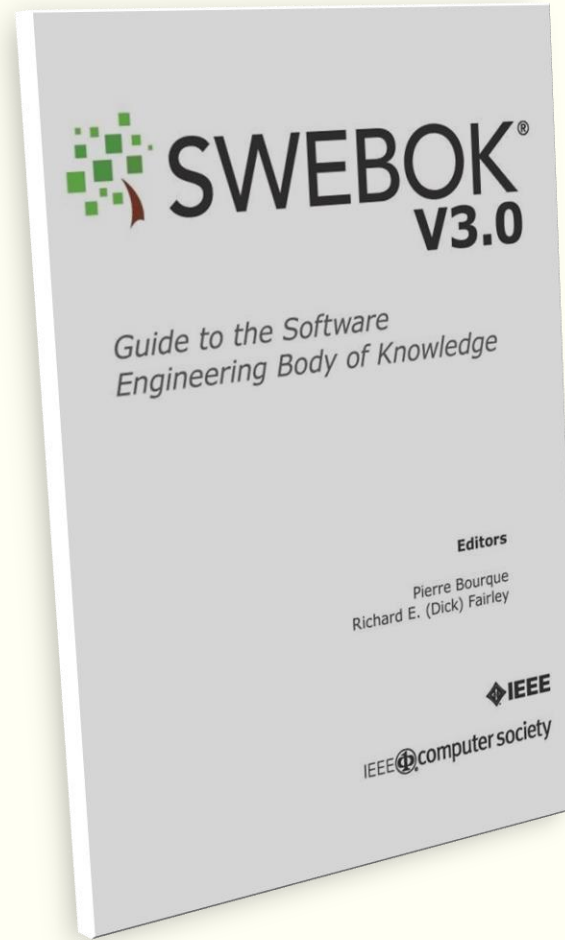
PROCESSOS DE ENGENHARIA DE SOFTWARE

Prof. Ausberto S. Castro Vera
ascv@uenf.br

9 de junho de 2024



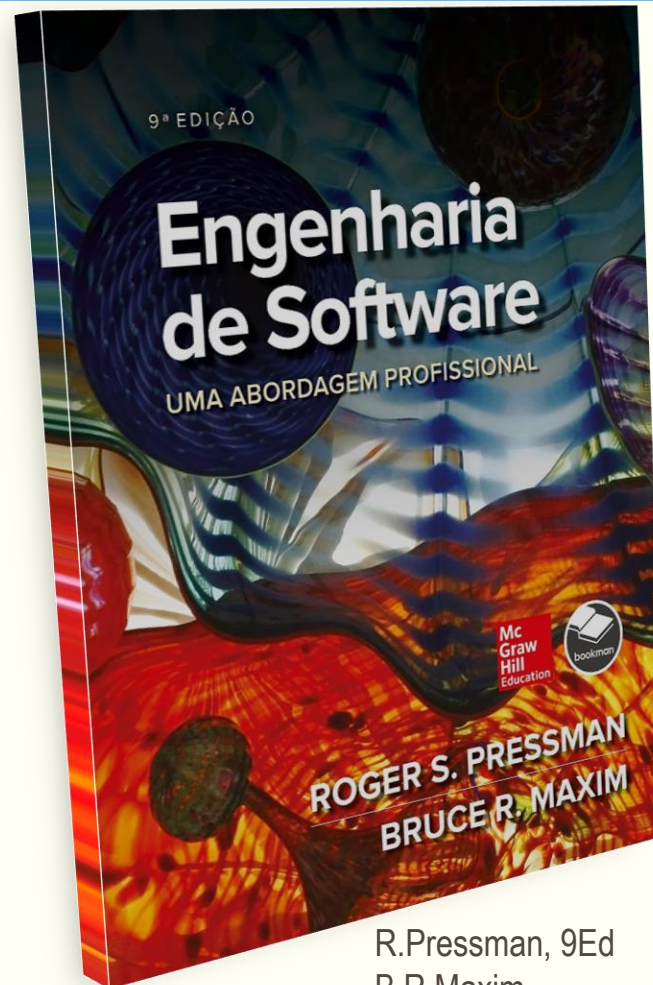
Bibliografia Básica



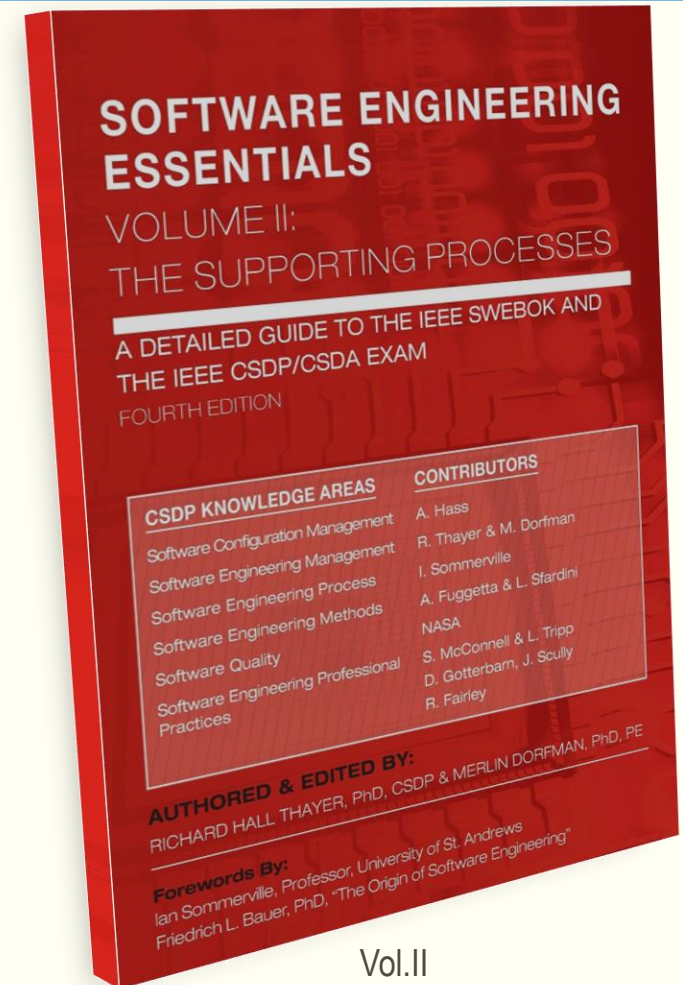
Bibliografia Básica



I.Sommerville, 10Ed
Pearson
2018
Cap 2



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



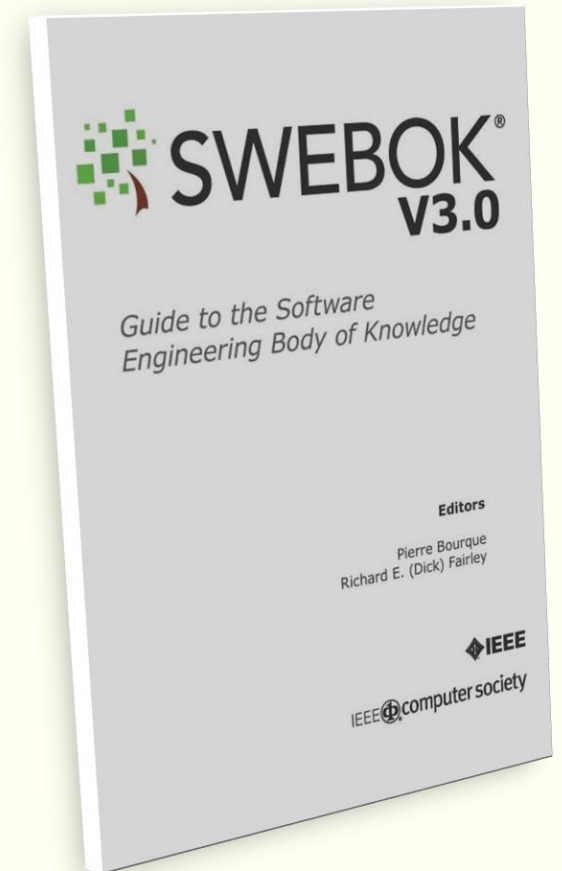
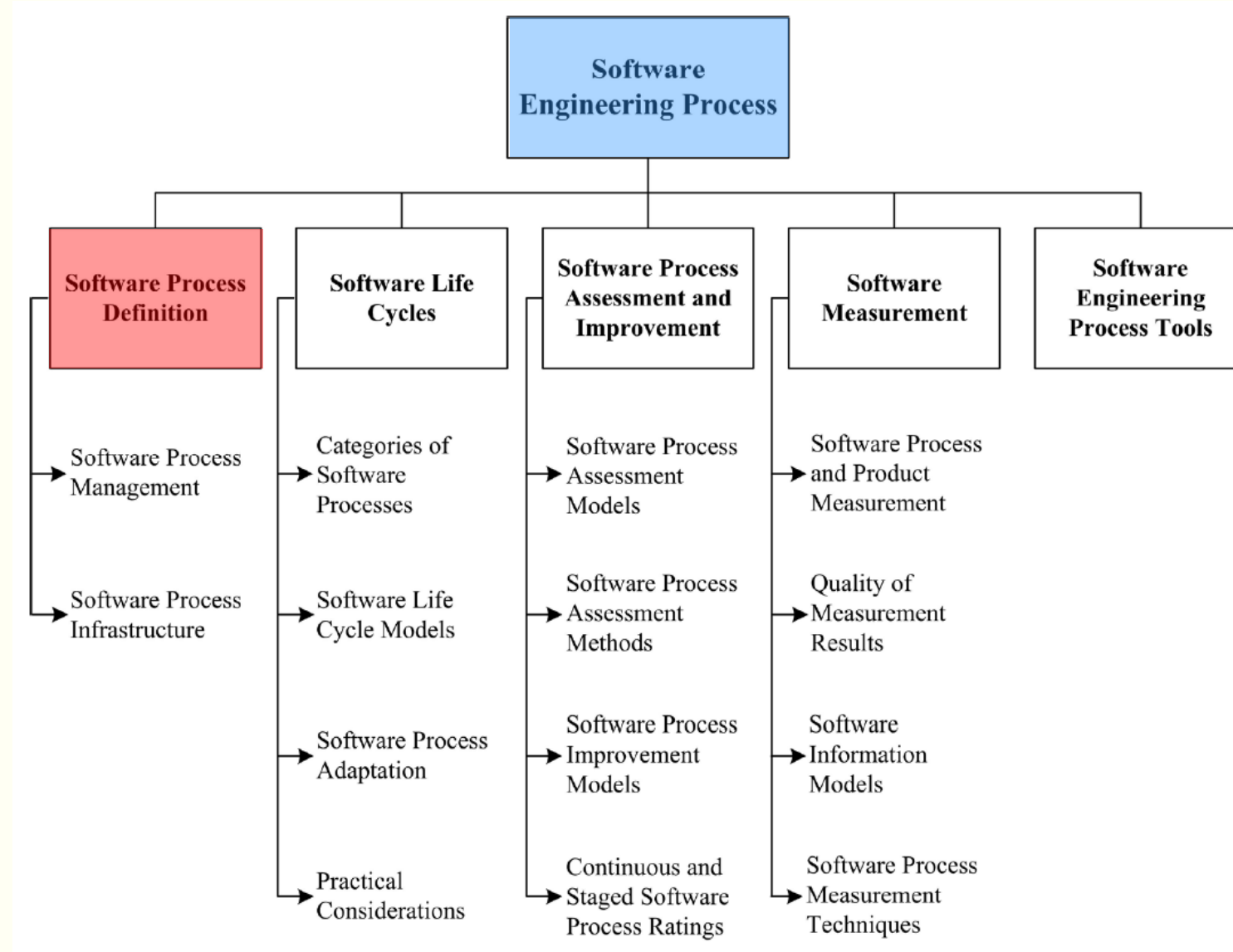
Vol.II
2013
Cap. 8

Bibliografia complementar

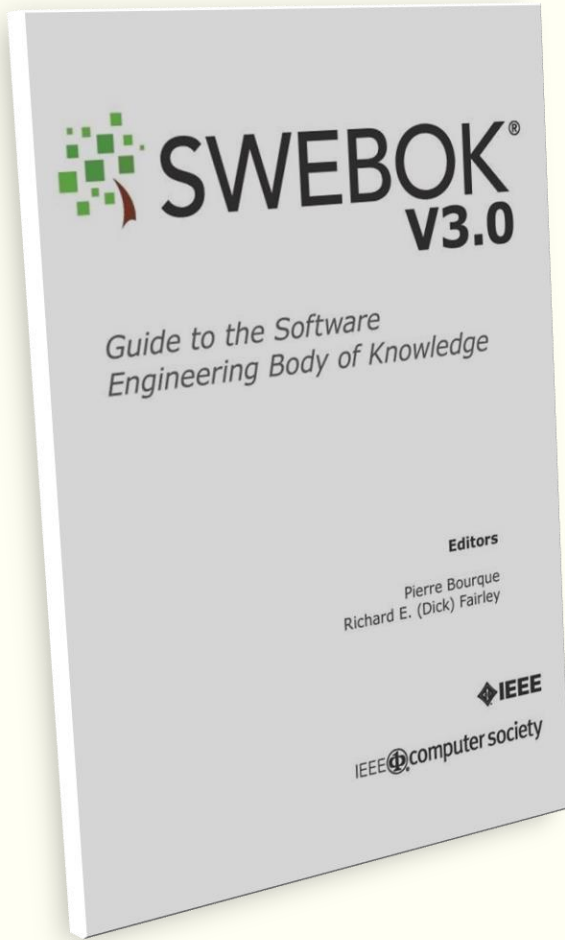
Software Engineering Process

- **1. Software Process Definition**
 - **1.1. Software Process Management**
 - **1.2. Software Process Infrastructure**
- **2. Software Life Cycles**
 - **2.1. Categories of Software Processes**
 - **2.2. Software Life Cycle Models**
 - **2.3. Software Process Adaptation**
 - **2.4. Practical Considerations**
- **3. Software Process Assessment and Improvement**
 - **3.1. Software Process Assessment Models**
 - **3.2. Software Process Assessment Methods**
 - **3.3. Software Process Improvement Models**
 - **3.4. Continuous and Staged Software Process Ratings**
- **4. Software Measurement**
 - **4.1. Software Process and Product Measurement**
 - **4.2. Quality of Measurement Results**
 - **4.3. Software Information Models**
 - **4.4. Software Process Measurement Techniques**
- **5. Software Engineering Process Tools**

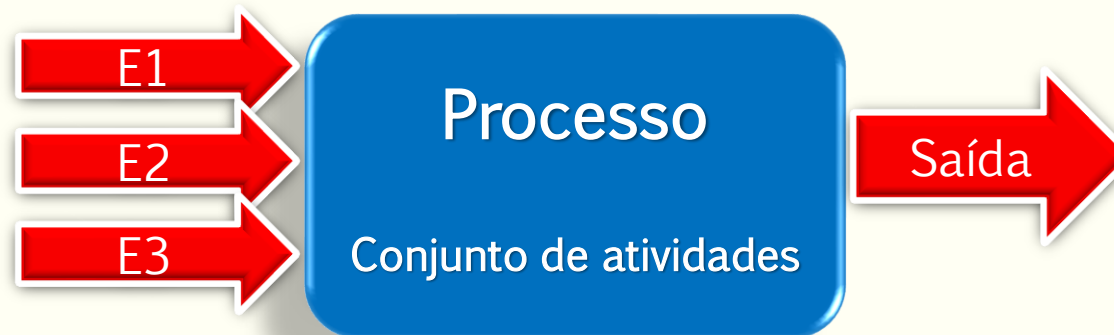
8. Processos de Engenharia de Software



Definição de Processo



- Um **processo** de *engenharia* consiste em um conjunto de *atividades inter-relacionadas* que transformam uma ou mais entradas em saídas enquanto consomem recursos para realizar a transformação.
- Os **processos de engenharia de software** preocupam-se com as *atividades de trabalho* realizadas por engenheiros de software para *desenvolver, manter e operar software*, como requisitos, projeto, construção, testes, gerenciamento de configuração e outros processos de engenharia de software.
- Um processo de software pode incluir vários *subprocessos*



Processo de Software

- **Processo**

Um *processo* é um conjunto de atividades, ações e tarefas realizadas na criação de algum artefato (de software).

- **Atividade**

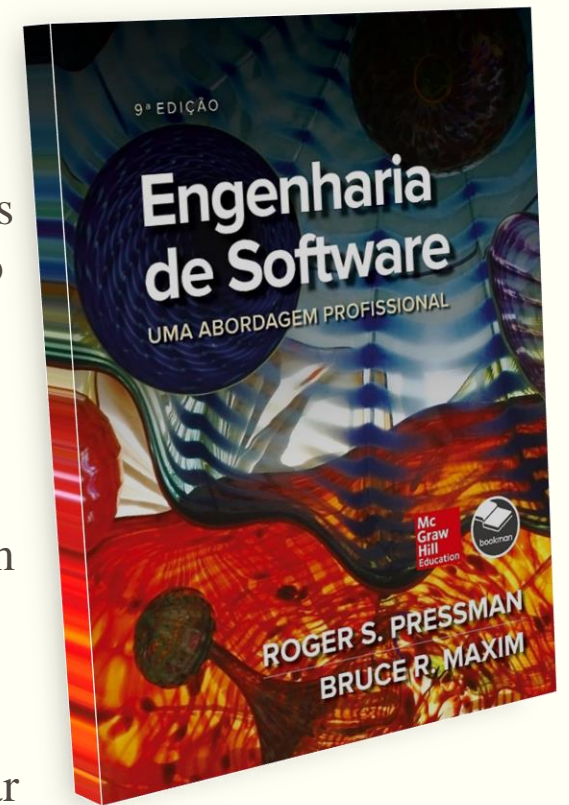
Uma *atividade* se esforça para atingir um objetivo amplo (p. ex., comunicar-se com os envolvidos) e é utilizada independentemente do domínio de aplicação, do tamanho do projeto, da complexidade dos esforços ou do grau de rigor com que a engenharia de *software* será aplicada.

- **Ação**

Uma *ação* (p. ex., projeto de arquitetura) envolve um conjunto de tarefas que resultam em um artefato de *software* fundamental (p. ex., um modelo arquitetural).

- **Tarefa**

Uma *tarefa* se concentra em um objetivo pequeno, porém bem-defnido (p. ex., realizar um teste de unidades), e produz um resultado tangível.



Processo de Software

Processo de software

Metodologia do processo

Atividades de apoio

Atividade metodológica nº 1

ação de engenharia de software nº 1.1

Conjuntos
de tarefas

tarefas de trabalho
artefatos
fatores de garantia da qualidade
marcos do projeto

⋮

ação de engenharia de software nº 1.k

Conjuntos
de tarefas

tarefas de trabalho
artefatos
fatores de garantia da qualidade
marcos do projeto

⋮

Atividade metodológica nº n

ação de engenharia de software nº n.1

Conjuntos
de tarefas

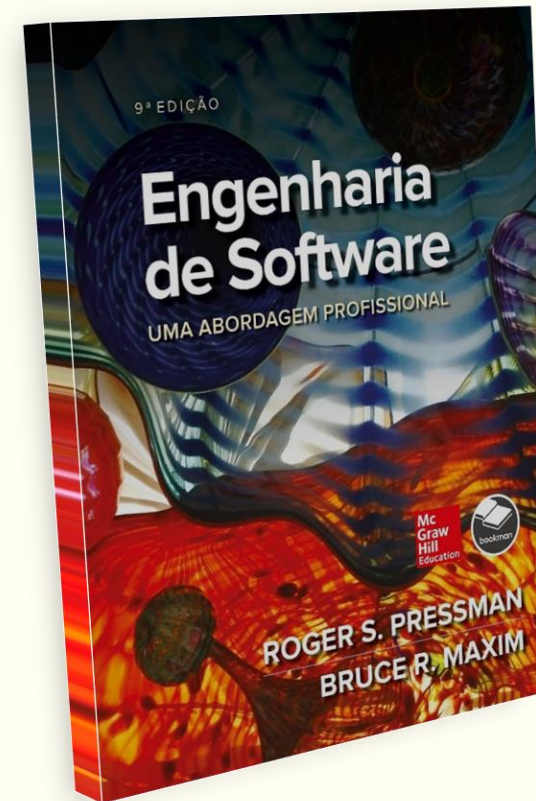
tarefas de trabalho
artefatos
fatores de garantia da qualidade
marcos do projeto

⋮

ação de engenharia de software nº n.m

Conjuntos
de tarefas

tarefas de trabalho
artefatos
fatores de garantia da qualidade
marcos do projeto



Definição de Processos de Software

- Um **processo** de software é um conjunto de atividades relacionadas que levam à produção de um sistema de software
- Existem *muitos tipos diferentes* de sistemas de software e *não há um método universal* de engenharia de software que seja aplicável a todos eles.
- O *processo* depende de:
 - Tipo de software a ser desenvolvido
 - Requisitos do cliente
 - Habilidades dos desenvolvedores
- Todos os processos devem incluir *quatro atividades fundamentais*:
 - *Especificação*: funcionalidades e restrições
 - *Desenvolvimento*
 - *Validação*: requisitos do cliente
 - *Evolução*: atender mudanças nas necessidades do cliente



1. Definição de Processos de Software

- **1.1 Gerenciamento de Processos de Software**
- **1.2 Infraestrutura dos Processos de Software**

1.1 Gerenciamento de Processos de Software

DEFINIÇÃO

O **gerenciamento de processos de software** envolve a *aplicação de práticas e métodos* para planejar, monitorar e controlar o desenvolvimento de software, garantindo que ele *atenda aos requisitos* e seja entregue dentro do *prazo e orçamento*.

1. Aplicação de práticas e métodos,
2. Planejar, monitorar e controlar o desenvolvimento de software,
3. atenda aos requisitos,
4. Entregar dentro do prazo e orçamento.

1.1 Gerenciamento de Processos de Software

- **Objetivos** do Gerenciamento de Processos de Software
 - *Alcançar a eficiência e eficácia* que resultam de uma *abordagem sistemática* para realizar processos de software e
 - *Produzir produtos de trabalho* – seja no nível individual, de projeto ou organizacional – e introduzir processos novos ou aprimorados.
- Os ***processos são alterados*** com a expectativa de que um processo novo ou modificado melhore a *eficiência e/ou eficácia* do processo e a *qualidade dos produtos* de trabalho resultantes.
 - Mudança para um novo processo,
 - Melhoria de um processo existente,
 - Mudança organizacional e
 - Mudança de infraestrutura
- A ***mudança no processo tem impactos*** não apenas no produto de software; muitas vezes levam a mudanças organizacionais.
 - Alterar um processo ou introduzir um novo processo pode ter efeitos em cascata em toda a organização. Estas situações também podem ser denominadas “evolução do processo”.
- Se as ***modificações forem extensas***, então provavelmente serão necessárias mudanças na *cultura organizacional* e no *modelo de negócios* para acomodar as mudanças no processo.



1.2 Infraestrutura dos Processos de Software

- **PROJETOS INDIVIDUAIS**

O estabelecimento, implementação e gerenciamento de **processos de software (PS)** e **modelos de ciclo de vida de software (M CVS)** geralmente ocorrem no nível de projetos de software individuais.

- **PROJETOS NA ORGANIZAÇÃO**

A aplicação sistemática de PS e M CVS em uma **organização** pode trazer benefícios para todo o trabalho de software **dentro da organização**

- **INFRA-ESTRUTURA**

Uma **infra-estrutura de processos de software** pode fornecer definições de processos, políticas para interpretar e aplicar os processos e descrições dos procedimentos a serem usados para implementar os processos.

Uma *infraestrutura de processo* de software pode fornecer *financiamento, ferramentas, treinamento e funcionários* aos quais foram atribuídas responsabilidades para estabelecer e manter a infraestrutura de processo de software.

1.2 Infraestrutura dos Processos de Software

- **TAMANHO E COMPLEXIDADE DA INFRE-ESTRUTURA**

A infraestrutura de processos de software varia, dependendo do tamanho e da complexidade da organização e dos projetos realizados dentro da organização.

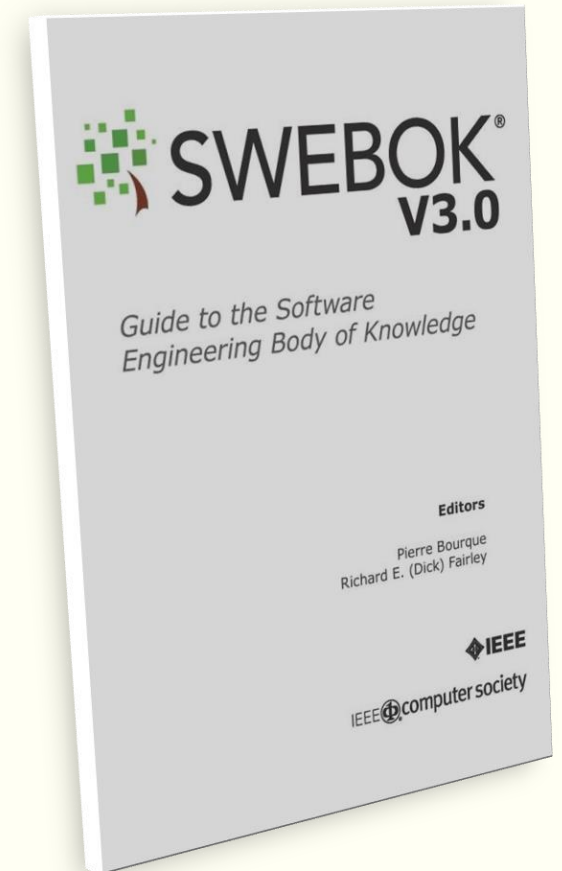
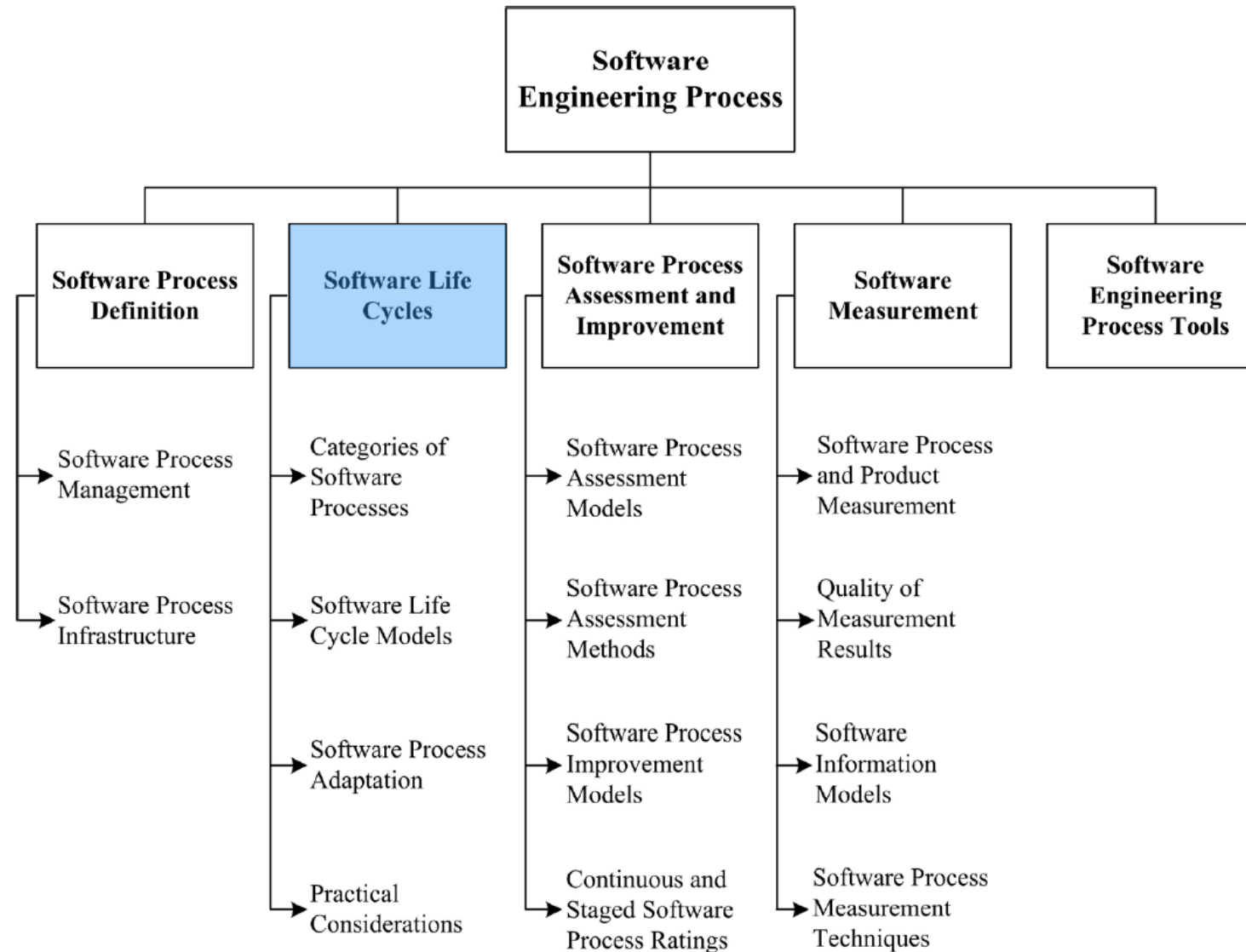
- Organizações e projetos **pequenos e simples** têm necessidades de infraestrutura pequenas e simples
- Organizações e projetos **grandes e complexos**, por necessidade, possuem infraestruturas de processos de software maiores e mais complexas.
- Várias unidades organizacionais podem ser estabelecidas (como um grupo de processos de engenharia de software ou um comitê diretor) para supervisionar a implementação e melhoria dos processos de software.

- **BENEFÍCIOS DA INFRA-ESTRUTURA**

- Um *equívoco comum*: estabelecer uma infraestrutura de processos de software e implementar processos de software repetíveis *adicionará tempo e custo* ao desenvolvimento e manutenção de software.
- Existe um **custo associado à introdução ou melhoria de um processo de software**; no entanto, a experiência tem mostrado que a *implementação de melhorias sistemáticas* nos processos de software tende a resultar em:
 - ❖ *custos mais baixos* através de maior eficiência, prevenção de retrabalho e
 - ❖ *software mais confiável e acessível*
 - ❖ *qualidade* do produto de software.



8.2 Ciclo de Vida do Software



2 Ciclo de Vida do Software

Ciclo de Vida do Software (CVS)

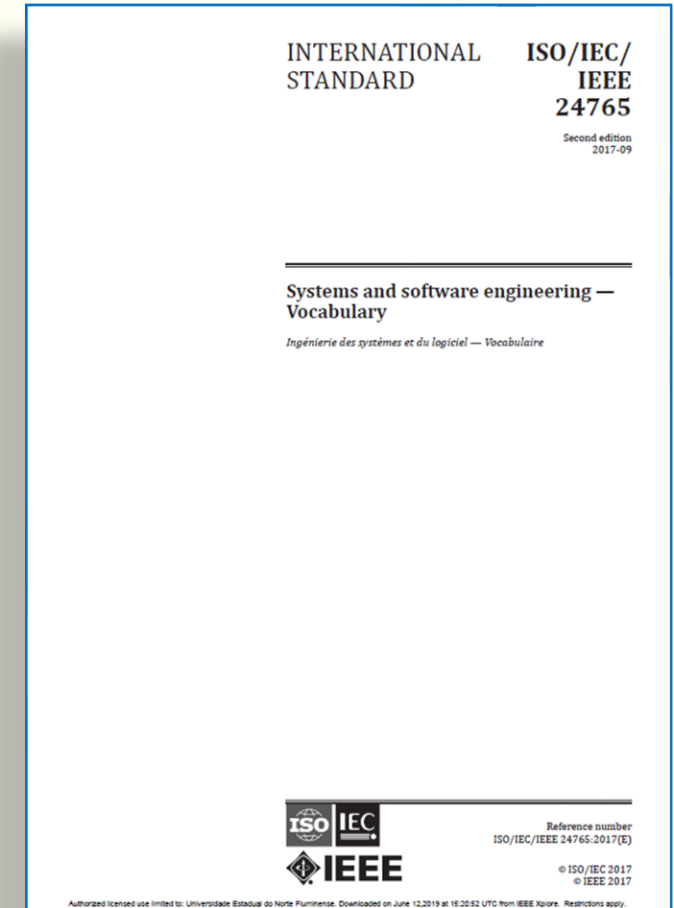
Software Life Cycle (SLC)

Ciclo de Vida de Desenvolvimento de Software (CVDS)

1. **Sequência de atividades** específica do projeto que é criada mapeando as atividades de um padrão em um modelo de ciclo de vida de software (MCVS) selecionado

[IEEE 730-2014 IEEE Standard for Software Quality Assurance Processes, 3.2]

2. **Ciclo de sistema de software** ou **produto de software** *iniciado* por uma necessidade do usuário ou necessidade percebida do cliente e *encerrado* pelo uso descontinuado do produto ou quando o software não está mais disponível para uso.
3. Um **ciclo de vida de desenvolvimento de software** (SDLC) inclui os *processos de software* usados para especificar e *transformar requisitos de software* em um *produto de software entregável*



2 Ciclo de Vida do Software

- Um **ciclo de vida de desenvolvimento de software** (CVDS) inclui os *processos de software* usados para especificar e transformar *requisitos de software* em um produto de software entregável.
- Um **ciclo de vida de produto de software** (CVPS) inclui um *ciclo de vida de desenvolvimento de software* mais *processos de software adicionais* que fornecem implantação, manutenção, suporte, evolução, desativação e todos os outros processos do *início à desativação* de um produto de software, incluindo o
 - gerenciamento de configuração de software
 - processos de software de garantia de qualidade que são aplicados ao longo do ciclo de vida de um produto de software.
- O **ciclo de vida de um produto** de software pode incluir **vários ciclos de vida de desenvolvimento de software** para evoluir e aprimorar o software.



Ciclo de Vida de Desenvolvimento de Software (CVDS-SDLC)

- **Definição**

O **Ciclo de Vida de Desenvolvimento de Software**

(SDLC, do inglês *Software Development Life Cycle*) é uma estrutura que descreve as *fases envolvidas na criação de um software*, desde a *concepção inicial* até a *descontinuação do produto*.

- **Objetivo**

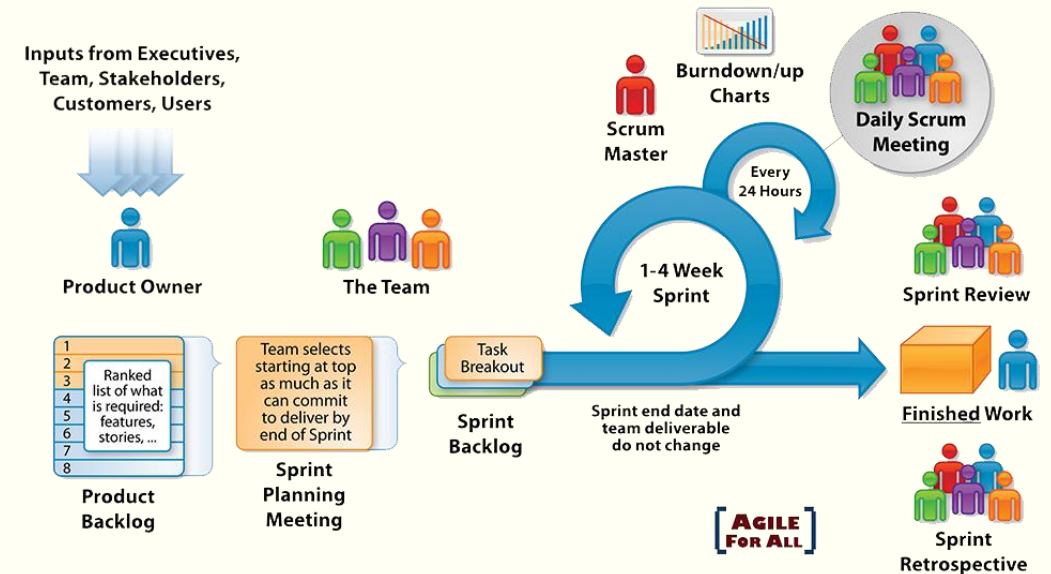
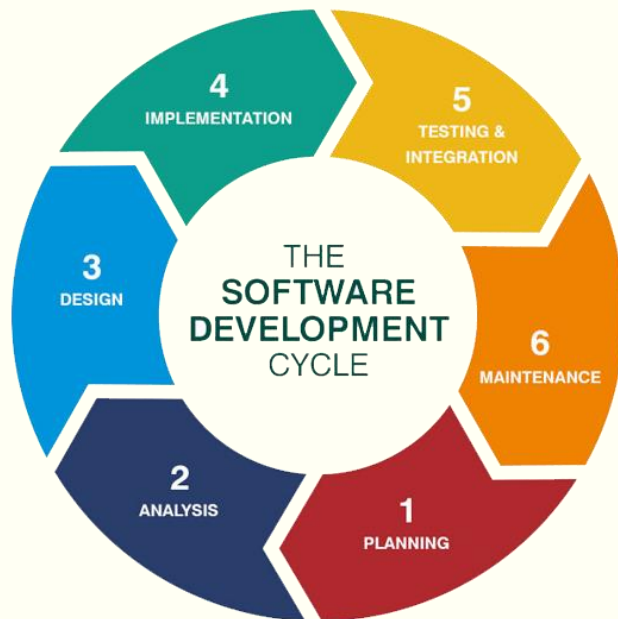
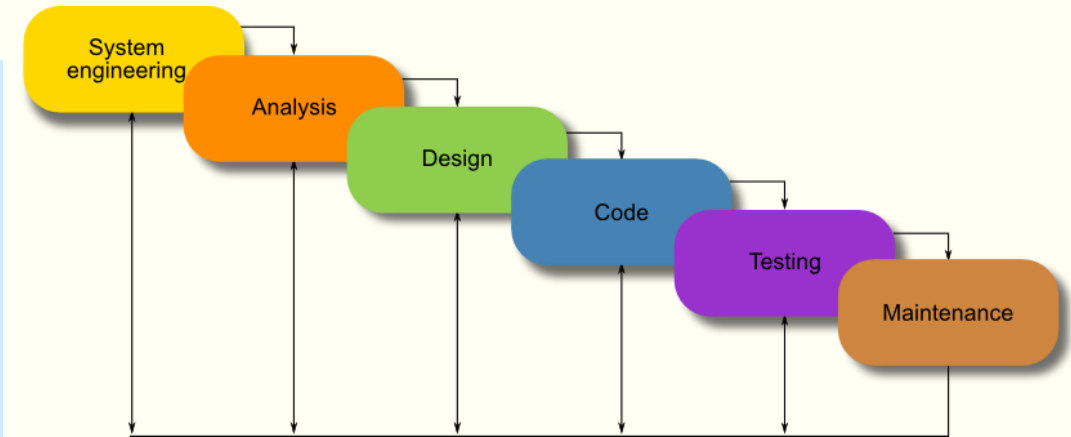
O objetivo do SDLC é fornecer um processo eficiente e eficaz para o desenvolvimento de software, garantindo que ele seja bem planejado, executado e mantido.

FASES do CVDS:

1. Planejamento
2. Análise de Requisitos
3. Desenho (Projeto)
4. Implementação (Codificação)
5. Testes
6. Implantação (Deploy)
7. Manutenção
8. Descontinuação

2 Ciclo de Vida do Software

- 2.1 Categorias de Processos de Software
- 2.2 Modelos de Ciclo de Vida do Software
- 2.3 Adaptação dos Processos de Software
- 2.4 Considerações Práticas



<https://www.pngwing.com/en/free-png-xqgjv>

2.1 Categorias de Processos de Software

- Muitos *processos de software distintos* foram definidos para uso nas diversas partes dos ciclos de vida de desenvolvimento e manutenção de software.
- **Categorias**
 1. Os **processos primários** incluem processos de software para *desenvolvimento, operação e manutenção* de software.
 2. Os **processos de suporte** são aplicados de forma intermitente ou contínua ao longo do ciclo de vida de um produto de software para dar suporte aos processos primários; eles incluem processos de software como *gerenciamento de configuração, garantia de qualidade e verificação e validação*.
 3. Os **processos organizacionais** fornecem suporte à engenharia de software; eles incluem *treinamento, análise de medição de processos, gerenciamento de infraestrutura, gerenciamento de portfólio e reutilização, melhoria de processos organizacionais e gerenciamento de modelos* de ciclo de vida de software.
 4. **Processos entre projetos**, como reutilização, linha de produtos de software e engenharia de domínio; eles envolvem mais de um único projeto de software em uma organização.

4
categorias

2.1 Categorias de Processos de Software

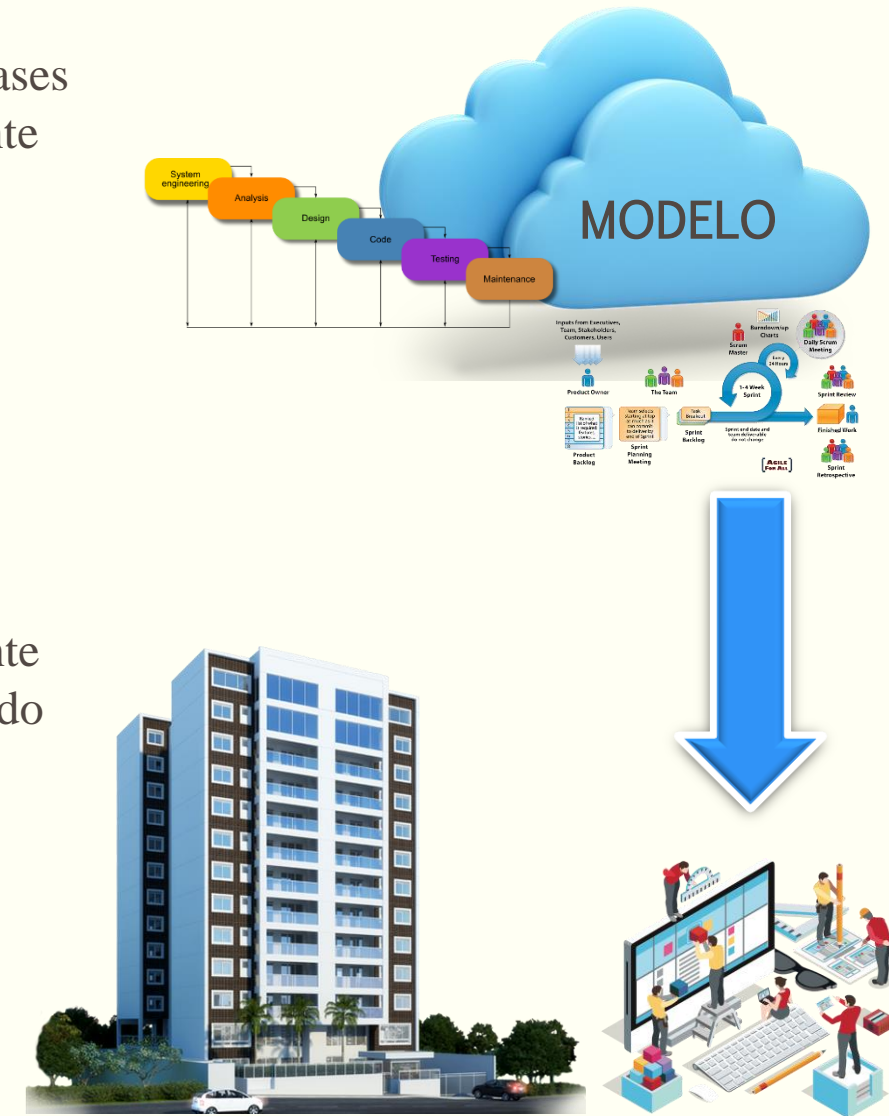
Os **processos de software** podem ser categorizados de várias maneiras, geralmente com base em seus *objetivos, atividades e resultados* esperados

7 categorias

1. **Processos de Engenharia de Software**
 - Desenvolvimento de Requisitos
 - Design de Software
 - Implementação
 - Testes
 - Manutenção
2. **Processos de Gerenciamento de Projetos de Software**
3. **Processos de Garantia de Qualidade de Software**
 - Revisões e Inspeções
 - Auditorias de Qualidade
 - Gestão de Configuração
4. **Processos de Suporte e Manutenção de Software**
5. **Processos de Melhoria de Processos de Software**
6. **Processos de Gestão de Recursos Humanos em Software**
 - Treinamento e Desenvolvimento
 - Gestão de Equipes
7. **Processos de Gestão de Riscos em Software**
 - Identificação de Riscos
 - Análise de Riscos
 - Mitigação de Riscos

2.2 Modelos de Ciclo de Vida do Software

- **Modelos lineares ou incrementais (preditivos)** nos quais as fases de desenvolvimento de software são realizadas sequencialmente com feedback e iteração conforme necessário, seguido de integração, teste e entrega de um único produto.
- **Modelos iterativos (adaptativo)** nos quais o software é desenvolvido em incrementos de funcionalidade crescente em ciclos iterativos
- **Modelos ágeis (adaptativos)** que normalmente envolvem demonstrações frequentes de software funcional para um cliente ou representante do usuário que direciona o desenvolvimento do software em ciclos iterativos curtos que produzem pequenos incrementos de software funcional e entregável.
- Modelos incrementais, iterativos e ágeis podem entregar subconjuntos iniciais de software funcional ao ambiente do usuário, se desejado.



2.3 Adaptação dos Processos de Software

- Processos predefinidos de CVDS e CVPS e processos de software individuais predefinidos muitas vezes *precisam ser adaptados* para melhor atender às necessidades locais.
- As **adaptações necessárias** ocorrem devido:
 - Ao contexto organizacional,
 - As inovações em tecnologia,
 - O tamanho do projeto,
 - A criticidade do produto,
 - Os requisitos regulamentares,
 - As práticas da indústria e
 - A cultura corporativa.
- A adaptação de processos de software individuais e modelos de ciclo de vida de software (desenvolvimento e produto) pode consistir
 - na **adição de mais detalhes** aos processos, atividades, tarefas e procedimentos de software para abordar questões críticas.
 - no uso de **um conjunto alternativo de atividades** que atinja o propósito e os resultados do processo de software.
- A adaptação também **pode incluir a omissão de processos** ou atividades de software de um modelo de desenvolvimento ou de ciclo de vida de produto

2.4 Considerações Práticas

- Na prática, os processos e atividades de software são frequentemente intercalados, sobrepostos e aplicados simultaneamente.
- **Modelos como idealizações**

Os modelos de ciclo de vida de software que especificam processos de software discretos, com critérios de entrada e saída rigorosamente especificados e limites e interfaces prescritos, devem ser reconhecidos como idealizações que devem ser adaptadas para refletir as realidades do desenvolvimento e manutenção de software dentro do contexto organizacional e do ambiente de negócios.
- **Adaptação.**

os processos de software (como gerenciamento de configuração, construção e teste) podem ser adaptados para facilitar a operação, suporte, manutenção, migração e desativação do software.
- **Fatores adicionais**
 - conformidade exigida com padrões, diretivas e políticas;
 - demandas dos clientes;
 - criticidade do produto de software; e
 - maturidade e competências organizacionais.
 - a natureza do trabalho (por exemplo, modificação de software existente versus novo desenvolvimento)
 - domínio da aplicação (por exemplo, aeroespacial versus gestão hoteleira).

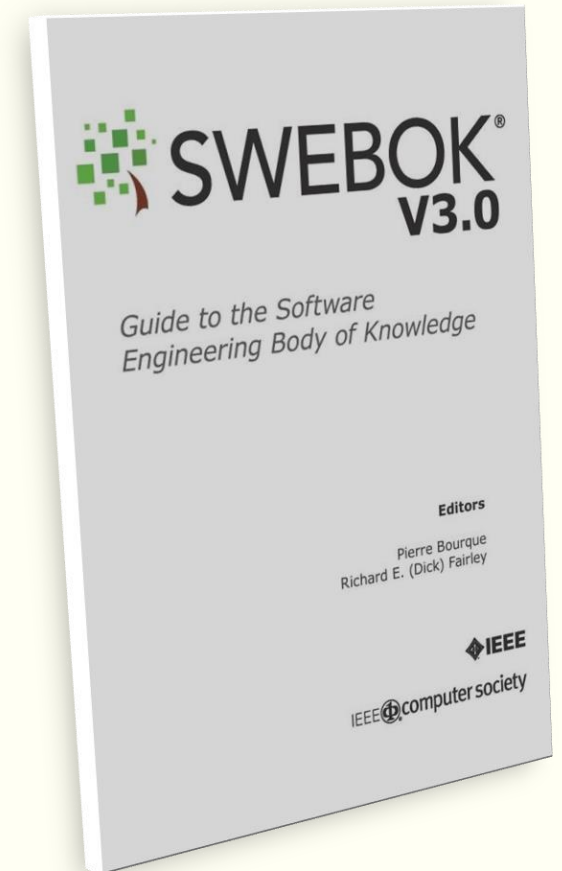
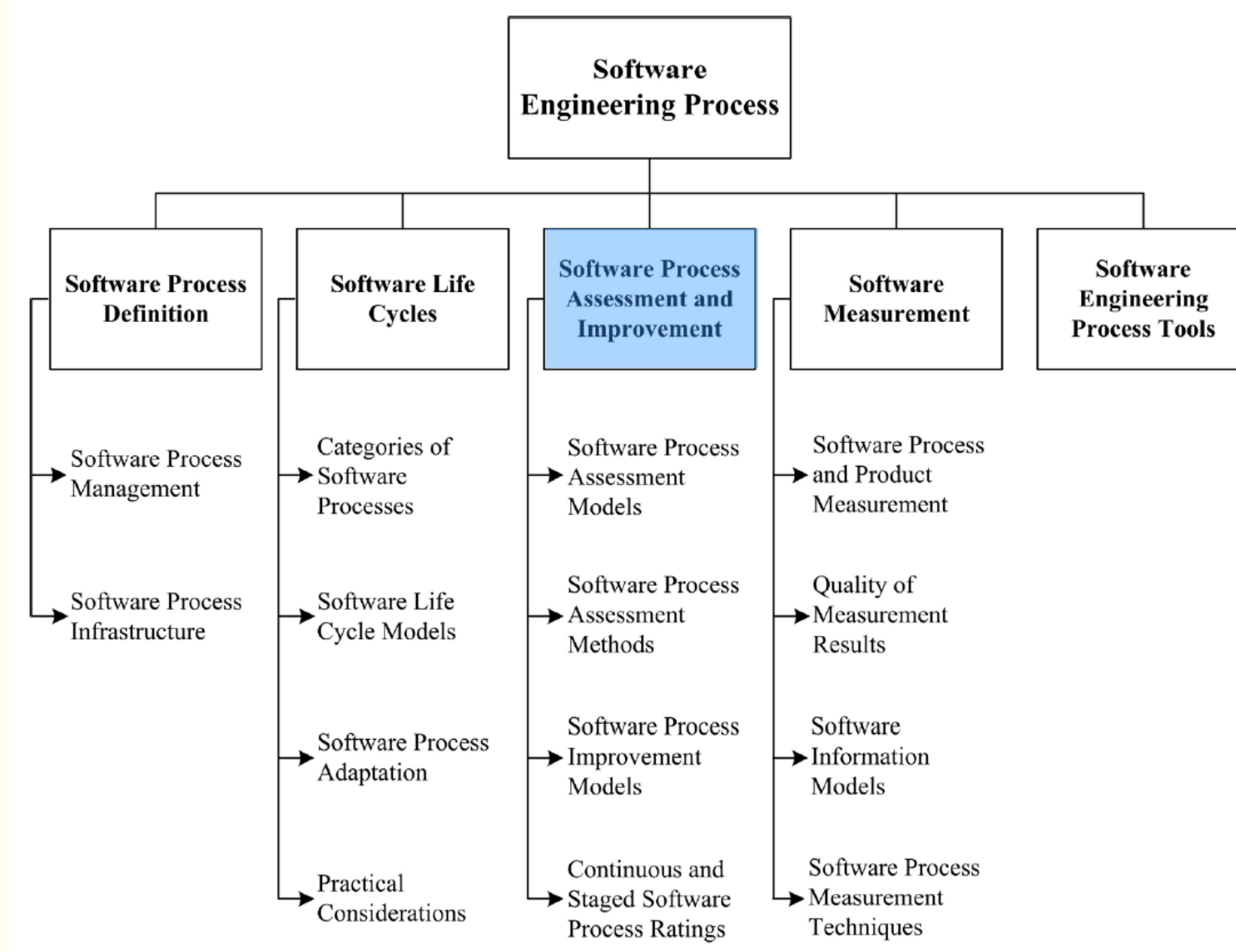


Prof. Dr. Ausberto S. Castro Vera
Ciência da Computação
UENF-CCT-LCMAT
Campos, RJ

ascv@uenf.br



8.3 Estimativa e Melhoria dos Processos de Software



8.3 Estimativa e Melhoria dos Processos de Software

- **8.3.1 Modelos de Estimativa dos Processos de Software**
- **8.3.2 Métodos de Estimativa dos Processos de Software**
- **8.3.3 Modelos de Melhoria dos Processos de Software**
- **8.3.4 Classificações dos Processos de Software Contínuas e Graduais**

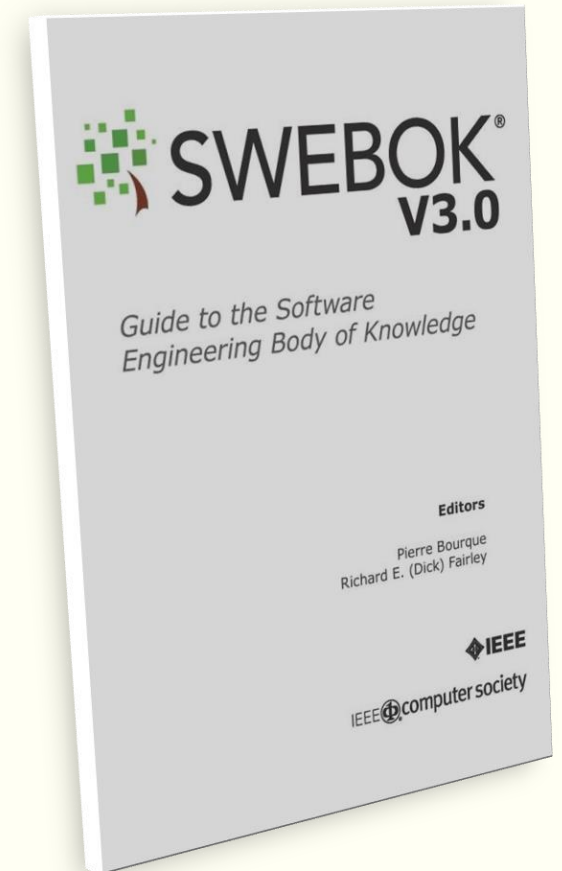
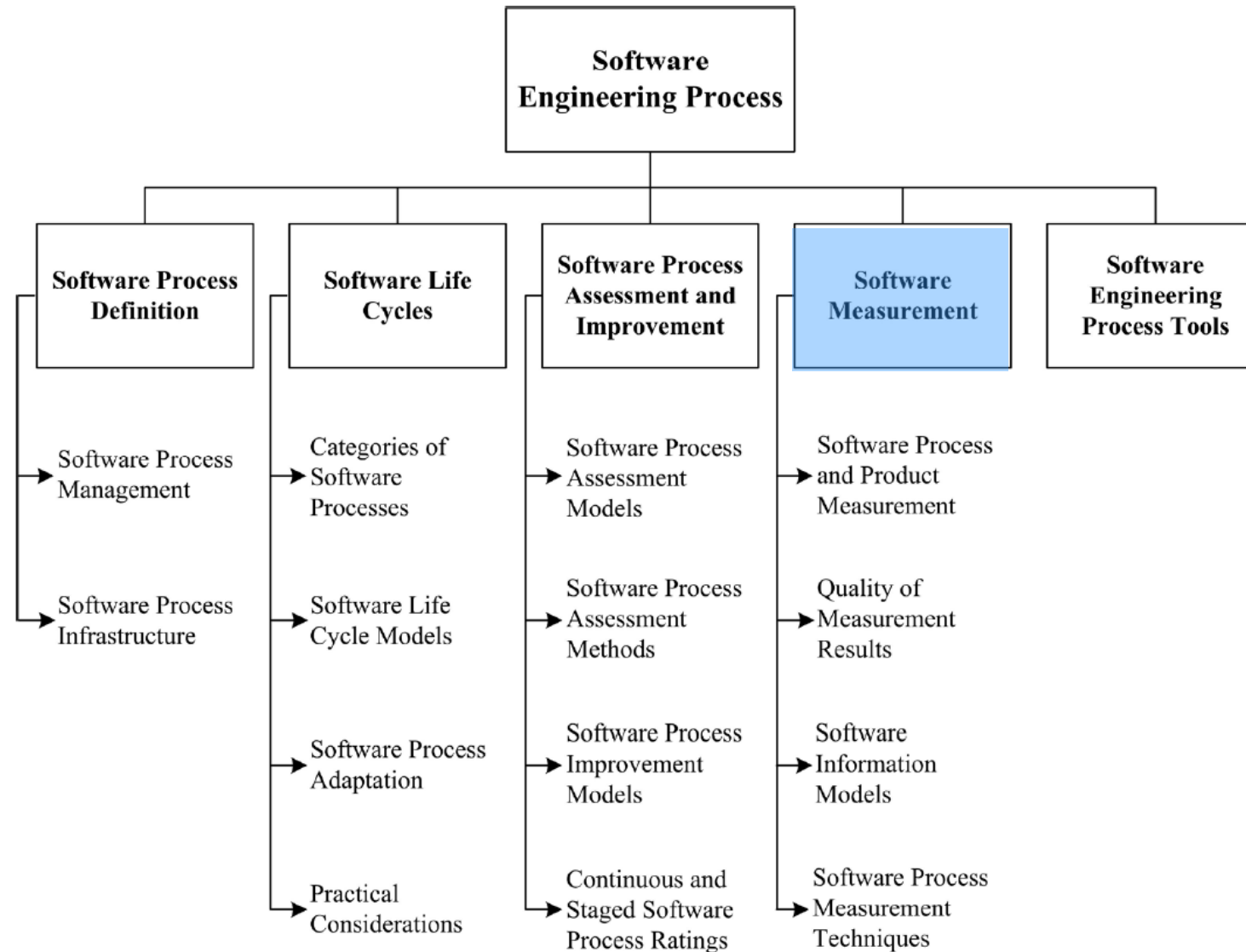
8.3.1 Modelos de Estimativa dos Processos de Software

8.3.2 Métodos de Estimativa dos Processos de Software

8.3.3 Modelos de Melhoria dos Processos de Software

8.3.4 Classificações dos Processos de Software Contínuas e Graduais

8.4 Medição de Software



8.4 Medição de Software

- **8.3.1 Medição do Processo e Produto de Software**
- **8.3.2 Qualidade dos Resultados da Medição**
- **8.3.3 Modelos de Informação de Software**
- **8.3.4 Técnicas de Medição do Processo de Software**

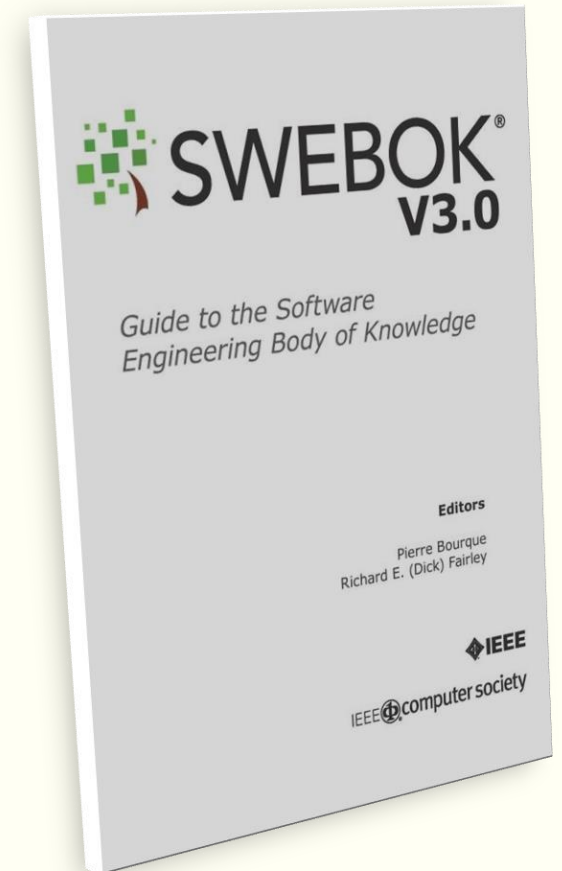
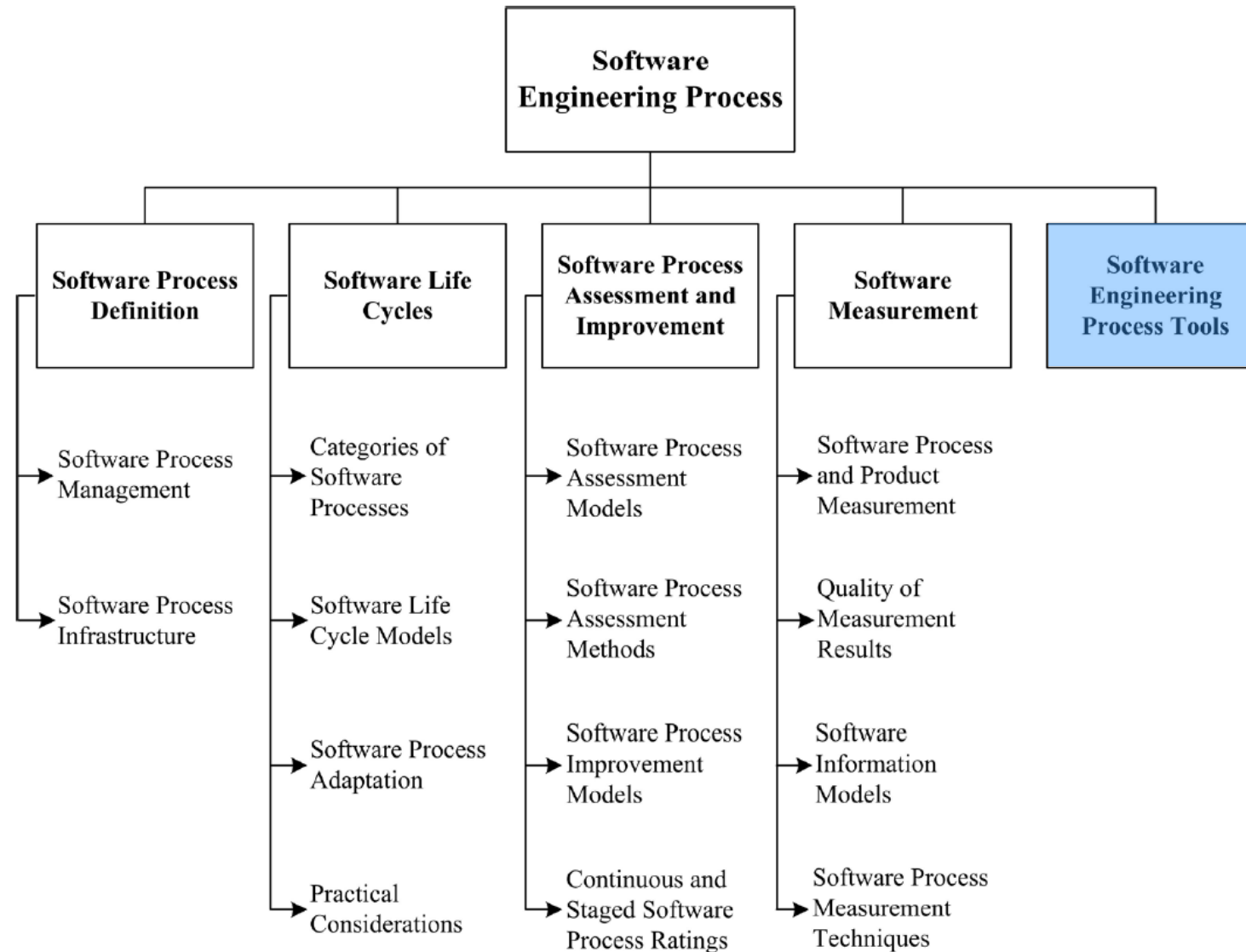
8.3.1 Medição do Processo e Produto de Software

8.3.2 Qualidade dos Resultados da Medição

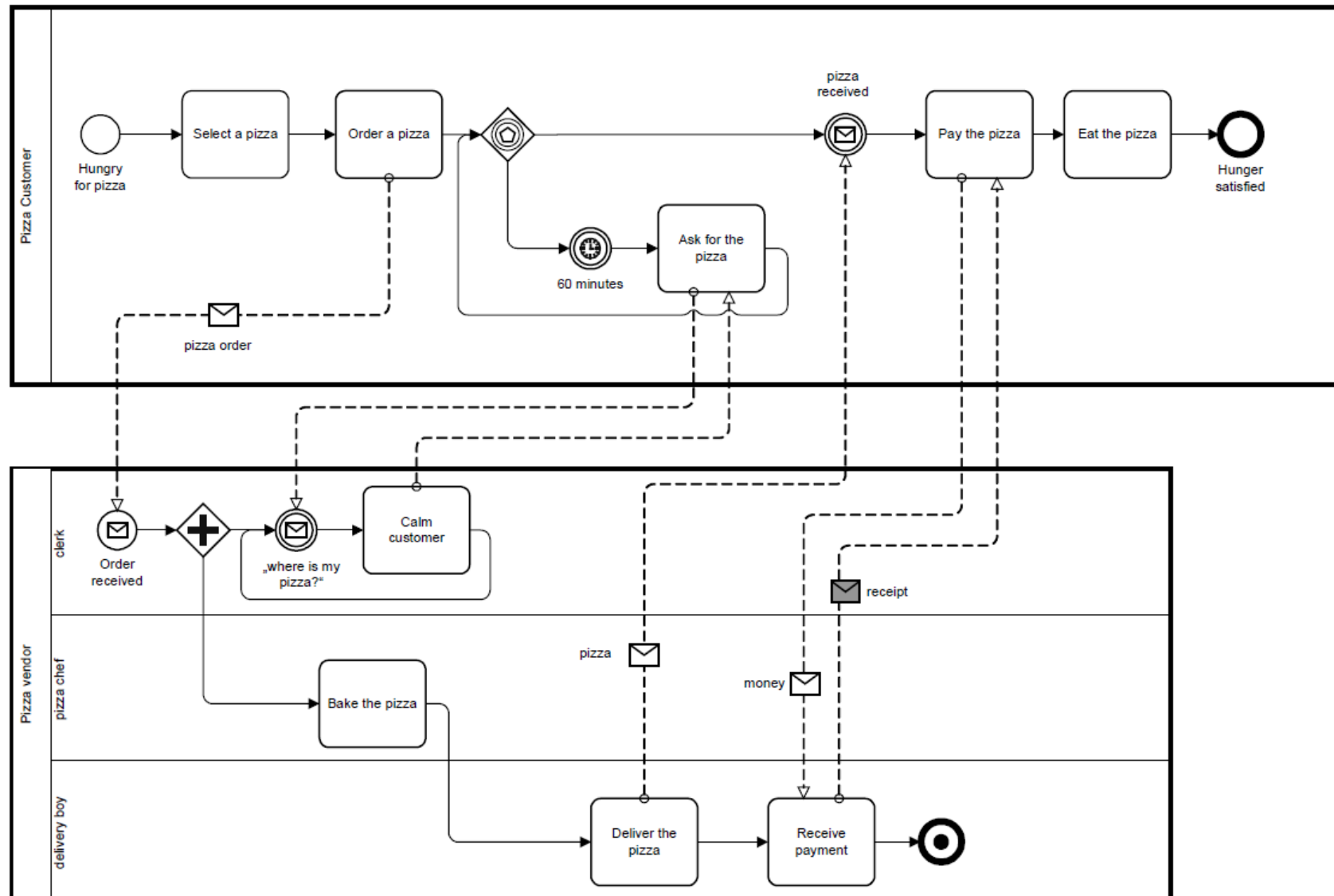
8.3.3 Modelos de Informação de Software

8.3.4 Técnicas de Medição do Processo de Software

8.5 Ferramentas do Processo de Engenharia de Software

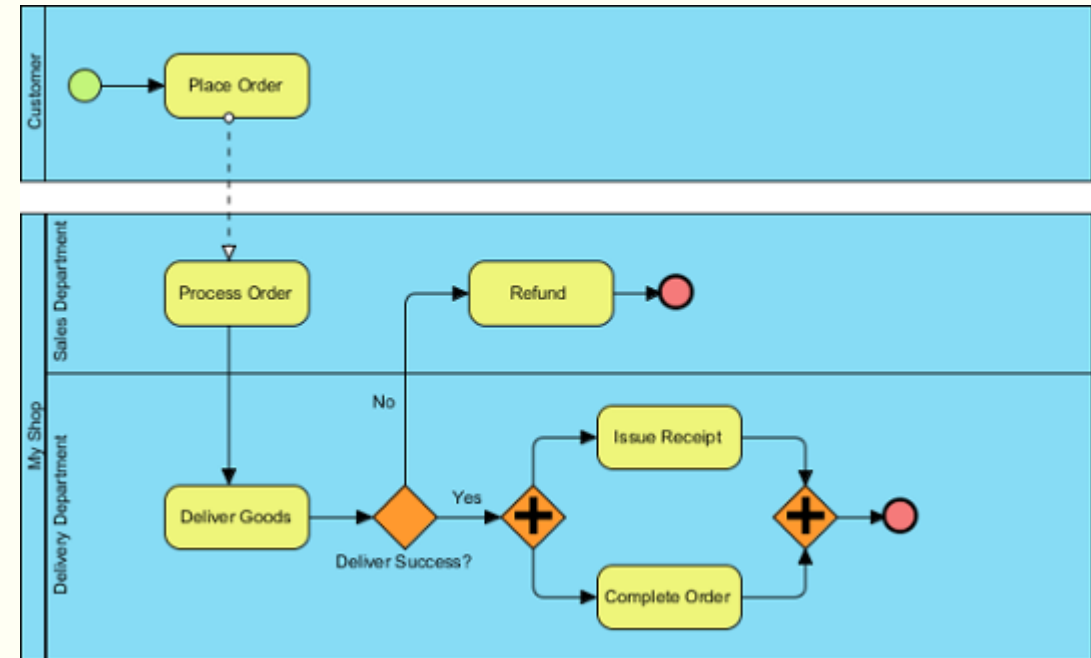
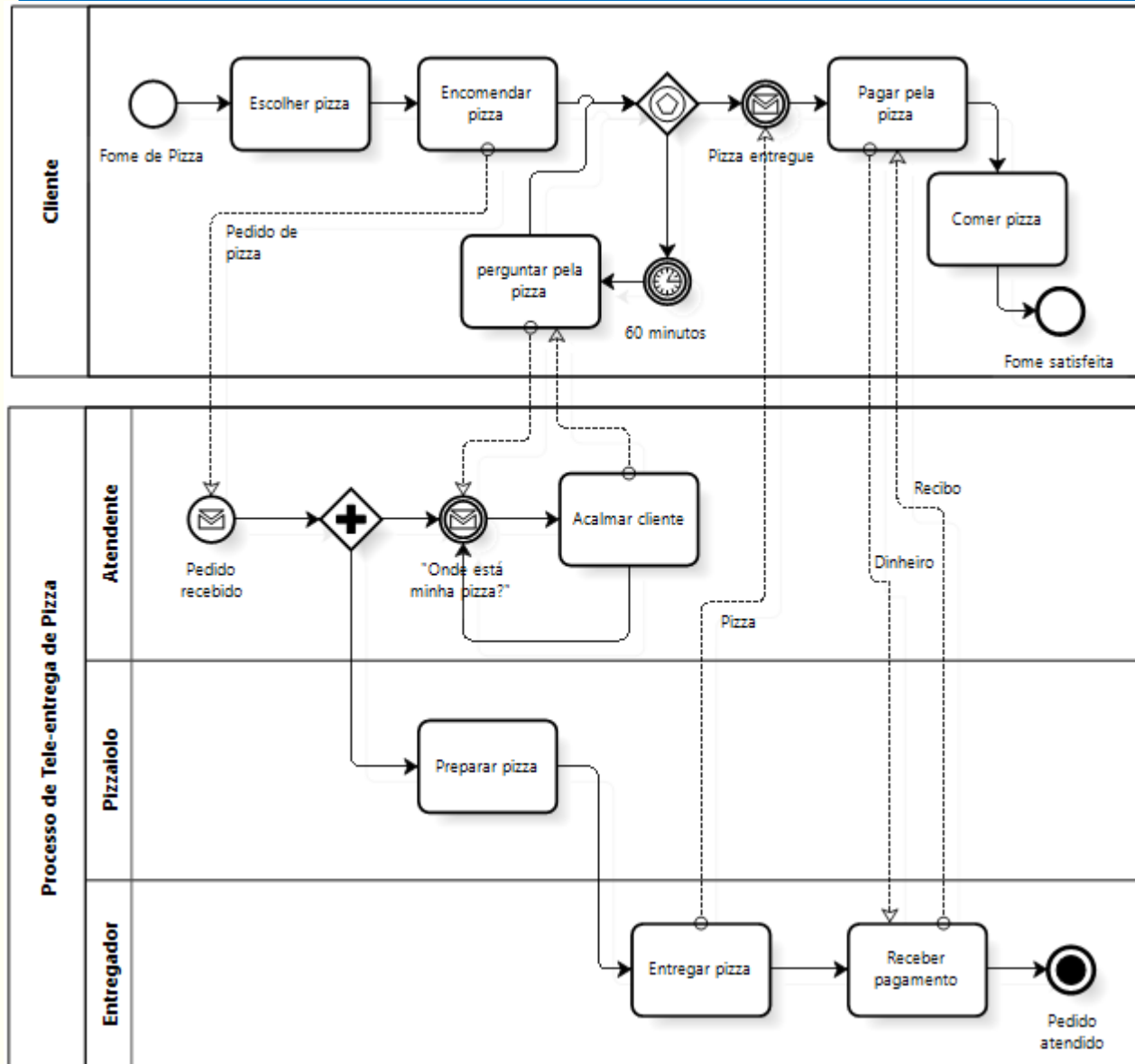


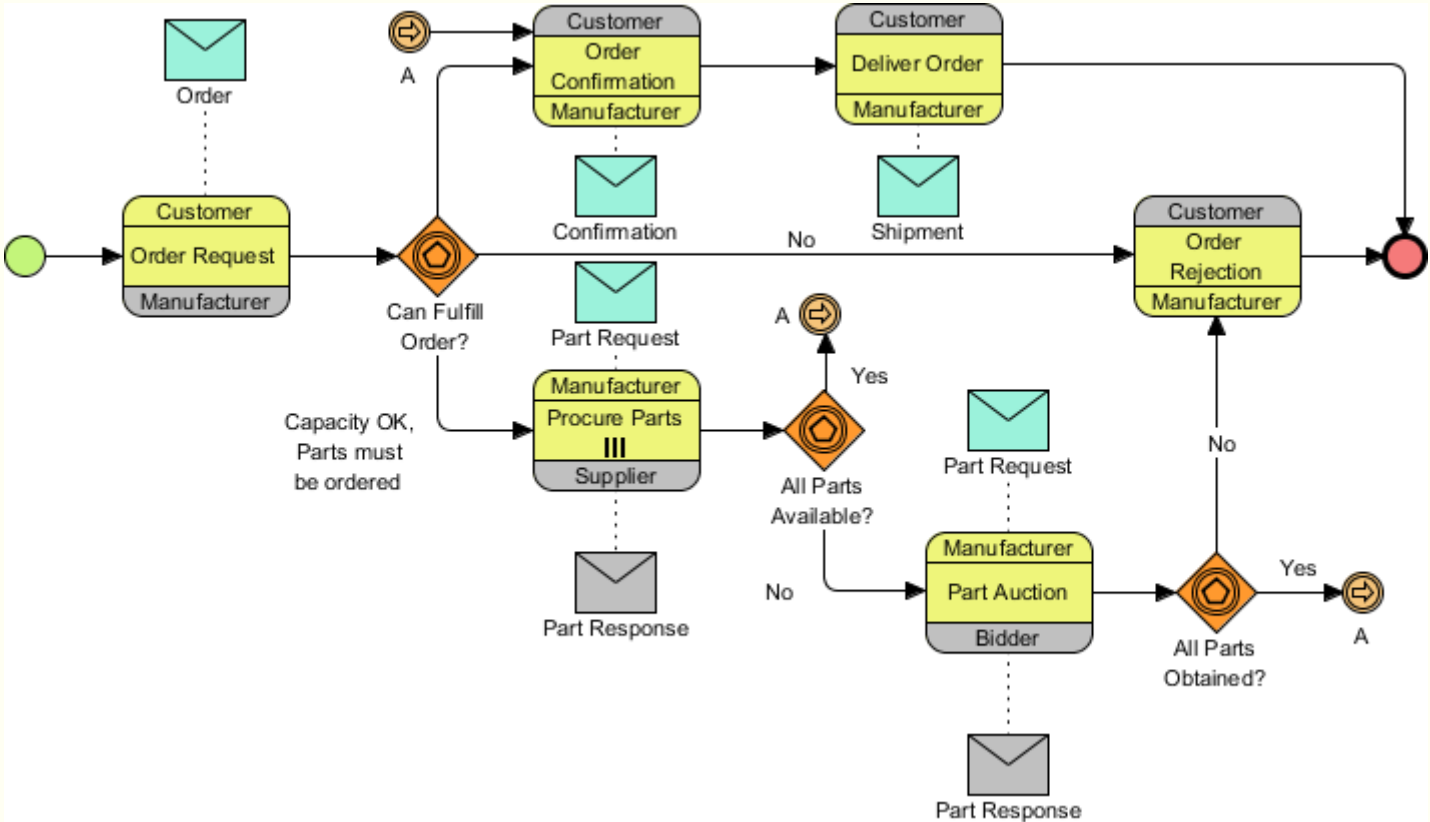
8.5 Ferramentas do Processo de Engenharia de Software



BPMN

BPMN





Petri Nets

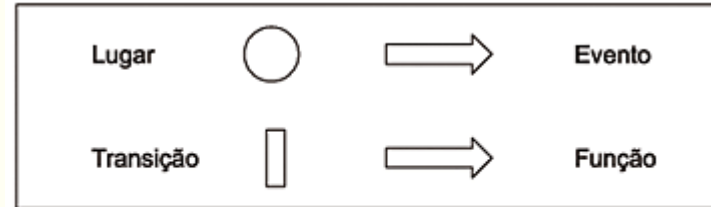


Figura 1 – Elementos básicos das redes de Petri.

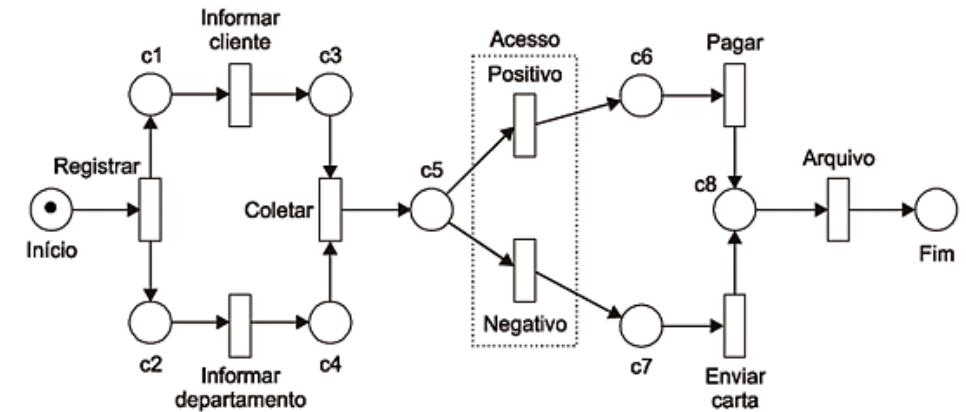


Figura 4 – O processo de gerenciar reclamação modelado em redes de Petri. Fonte: Aalst & Hee (2002, p. 50).

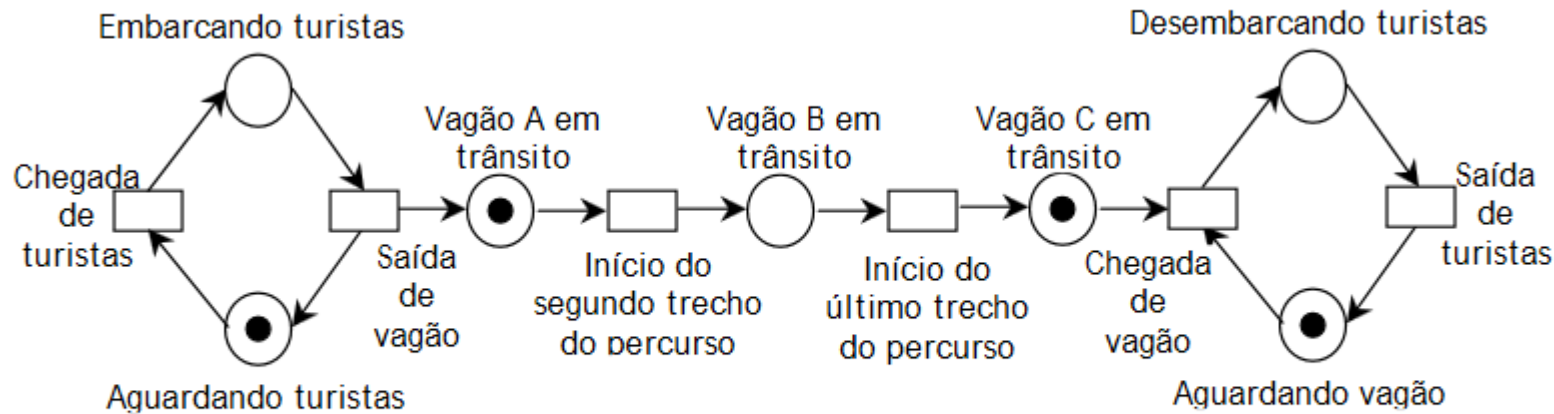


Figura-10: Rede elementar modelando um teleférico com três vagões e uma única linha.

Petri Nets

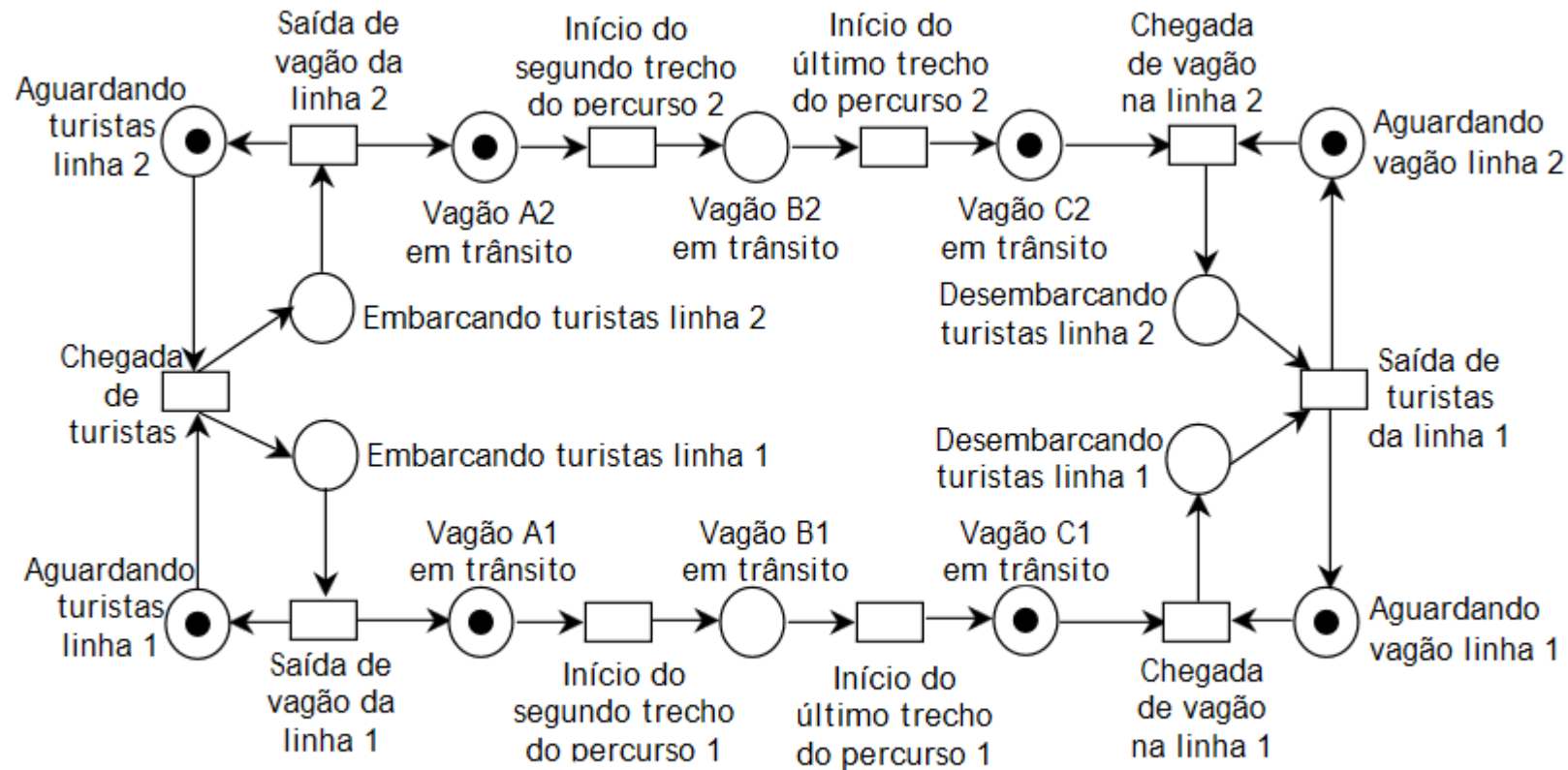


Figura-11: Rede elementar modelando um teleférico com seis vagões e duas linhas.