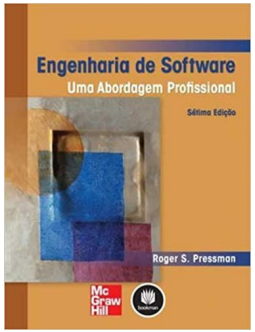
Engenharia de Software

Prof. Dr. Erik Aceiro Antonio

Objetivos

- Conceitos **Pressman (2010, c.1)**
- Apresentar os conceitos de Engenharia de Software **Pressman (2010, c.2)**
  - Definição de Software
  - Domínios de Aplicações dos Softwares
  - Software Legados
- Engenharia de Software
- Processos Genéricos & Frameworks
- Processo de Software
  - Modelos de Processos de Software
  - Tipos
- Desenvolvimento Ágil **Pressman (2010, c.3)**



Pressman, R.S. (2010) Software Engineering: A Practitioner's Approach. 7th Edition, McGraw Hill, New York.

Exemplos de uso de referência

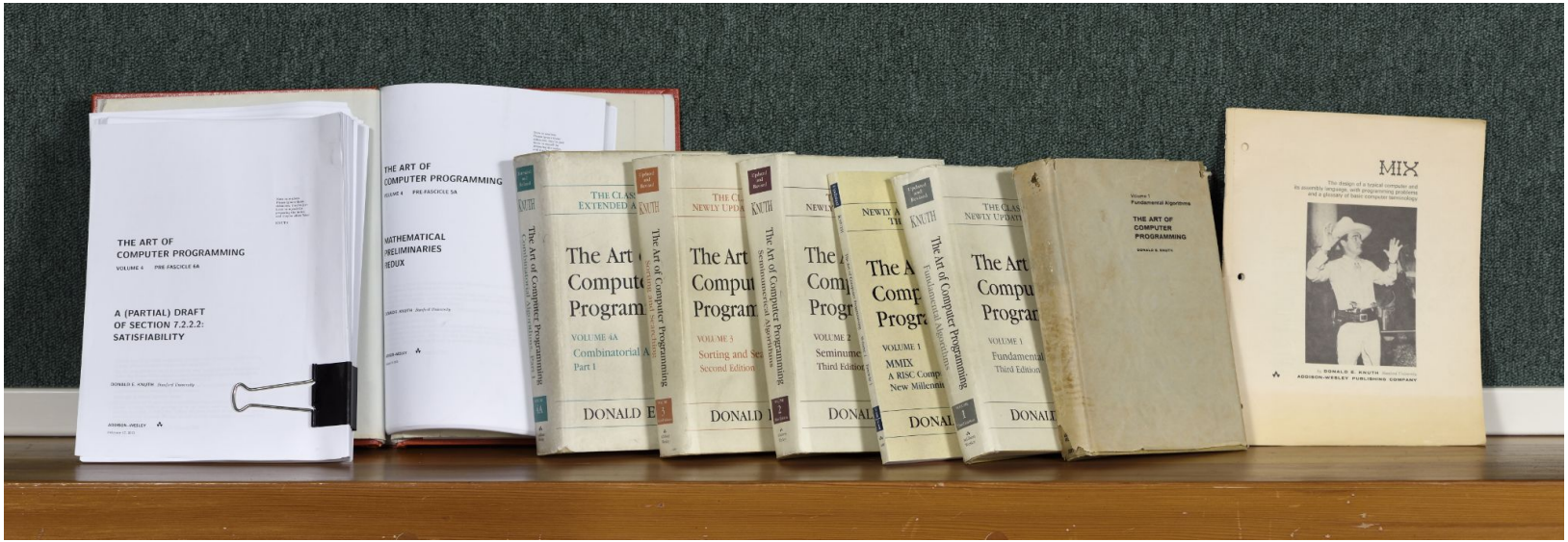
Pressman (2010, c.1) ou Pressman (2010, p.1)

mais específica

Pressman (2010)

mais geral

Software is Art (Donald Knuth)



Definição de Software

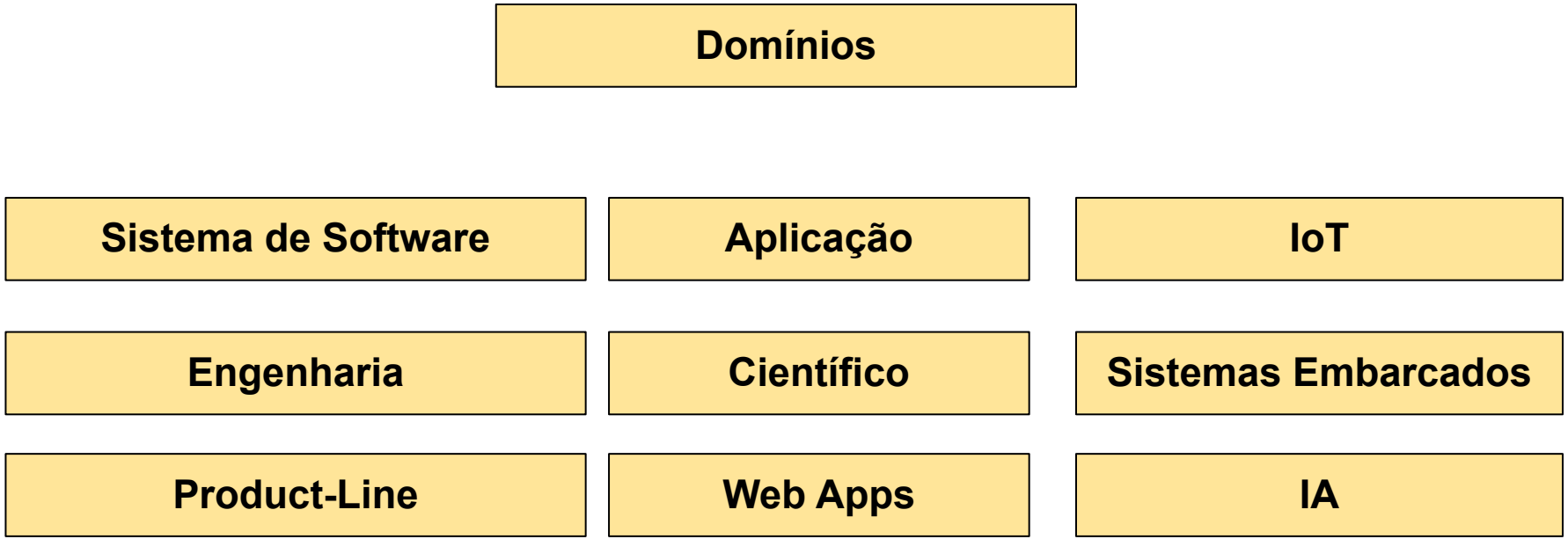
Software é

(1) Instruções
(2) Estrutura de dados
(3) Informação Descritiva

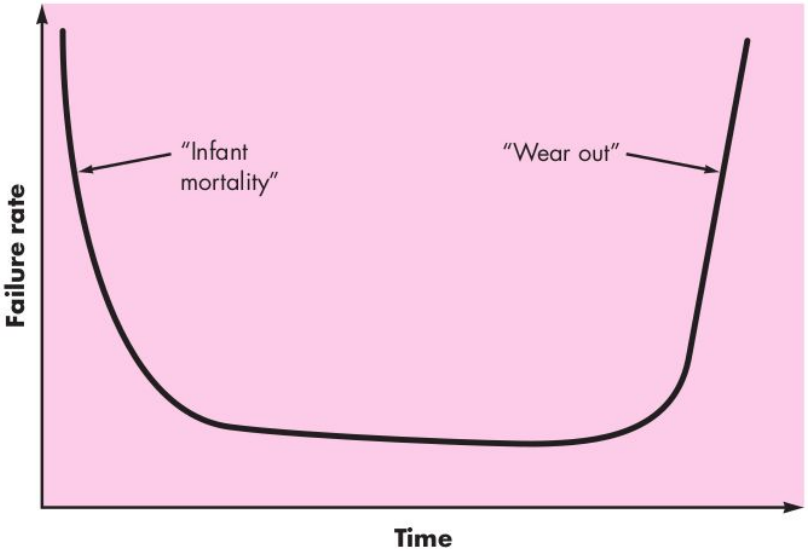
(*) Software é engenheirado, mas não fabricado!

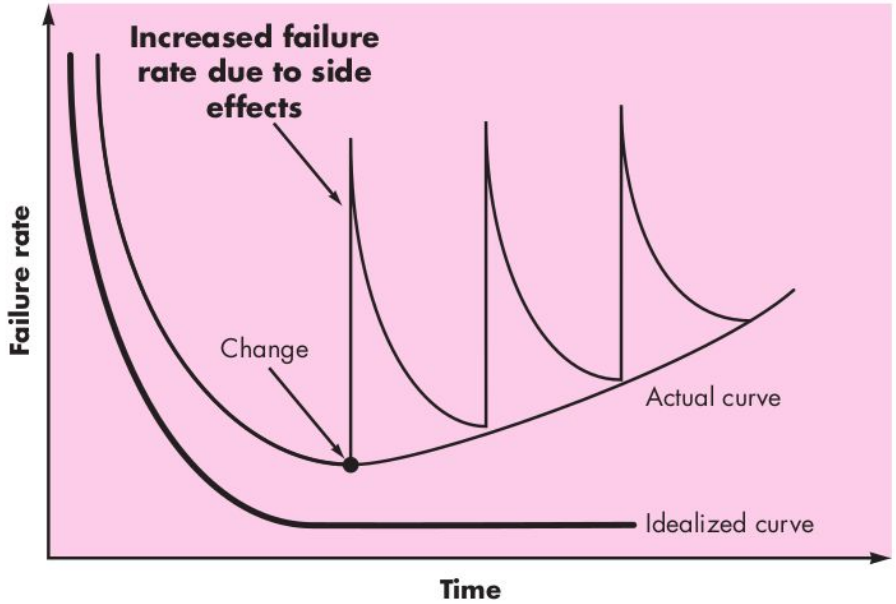
Pressman (2010, c.1)

Domínios de Aplicação do Software



Pressman (2010, c.1)



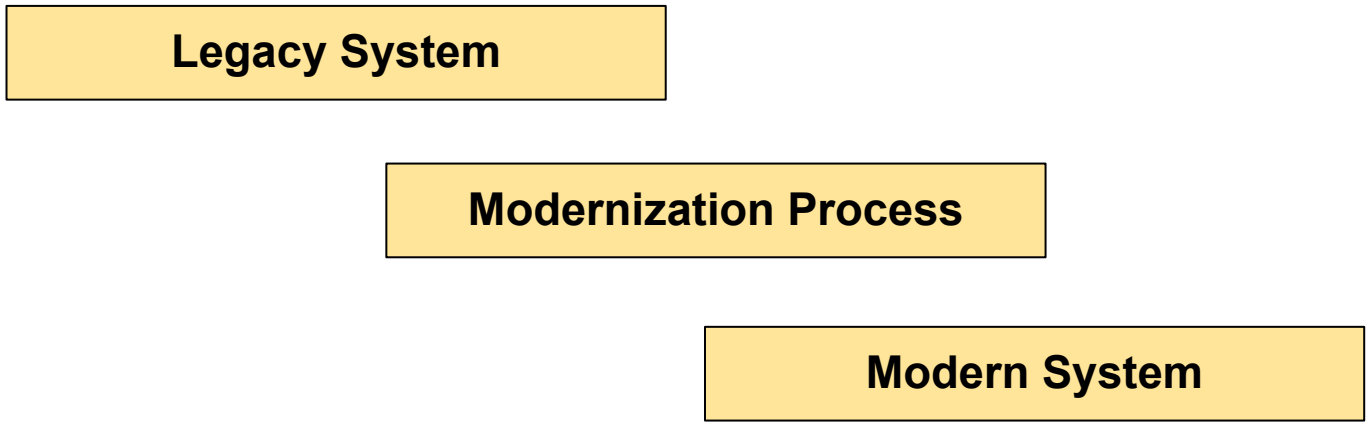


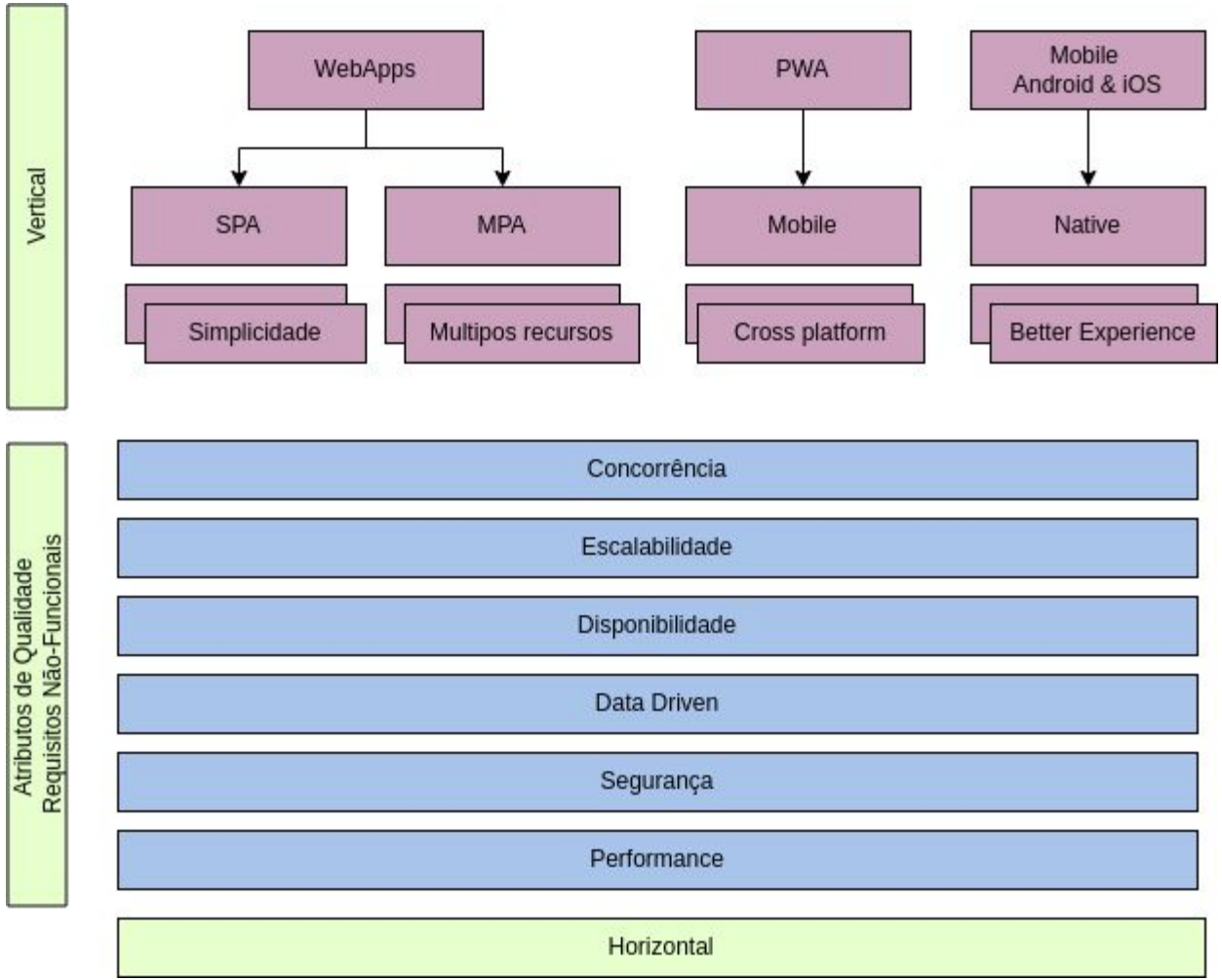
Pressman (2010, c.1)

Pressman, R.S. (2010) Software Engineering: A Practitioner's Approach. 7th Edition, McGraw Hill, New York.

Software Legado

Software legado consiste em sistemas que foram engenheirados e construídos e precisam ser continuamente alterados, reestruturados para acompanhar os requisitos de negócios.




Vertical
WebApps
PWA
Mobile
Android & iOS
SPA
MPA
Mobile
Native
Simplicidade
Multipos recursos
Cross platform
Better Experience
Atributos de Qualidade
Requisitos Não-Funcionais
Concorrência
Escalabilidade
Disponibilidade
Data Driven
Segurança
Performance
Horizontal

? How do we define software engineering?

The IEEE [IEE93a] has developed a more comprehensive definition when it states:

Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

Pressman (2010, c.1)

KEY POINT

Software engineering encompasses a process, methods for managing and engineering software, and tools.

Pressman (2010, c.1)

Pressman, R.S. (2010) Software Engineering: A Practitioner's Approach. 7th Edition, McGraw Hill, New York.

Modelo de Processo Genérico

Processo

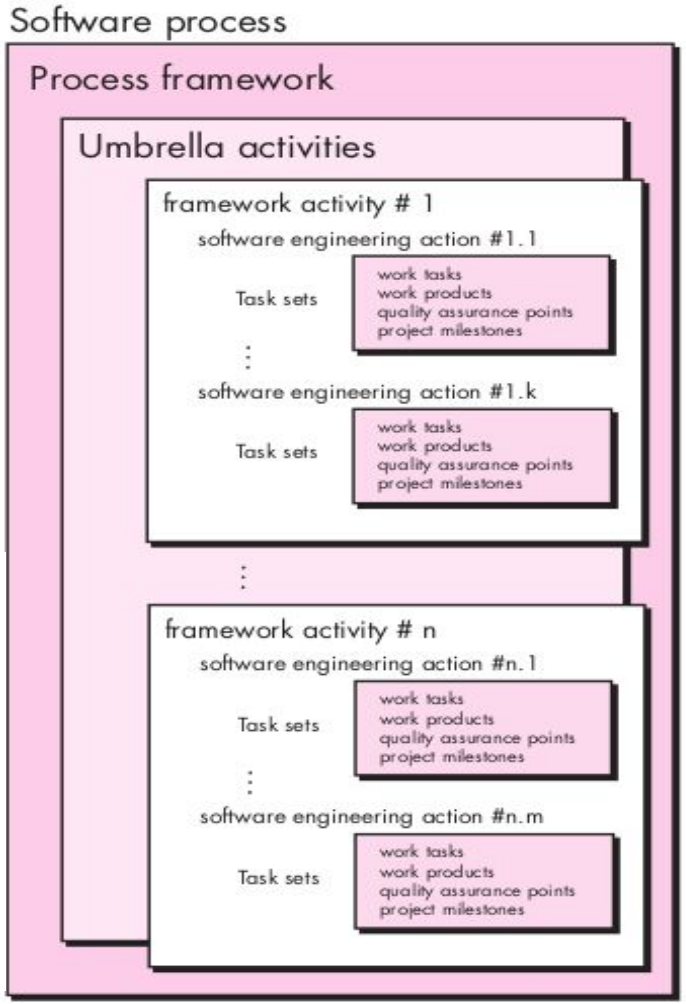
Framework

Modelo

Atividade

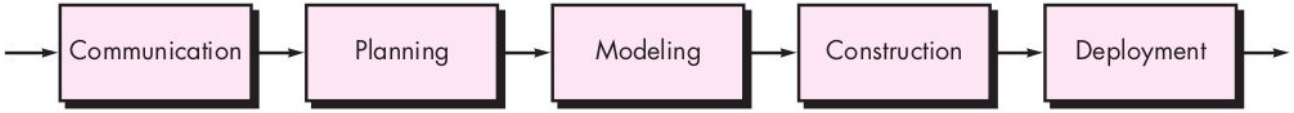
Flow (Fluxo)

Tasks (Tarefas)

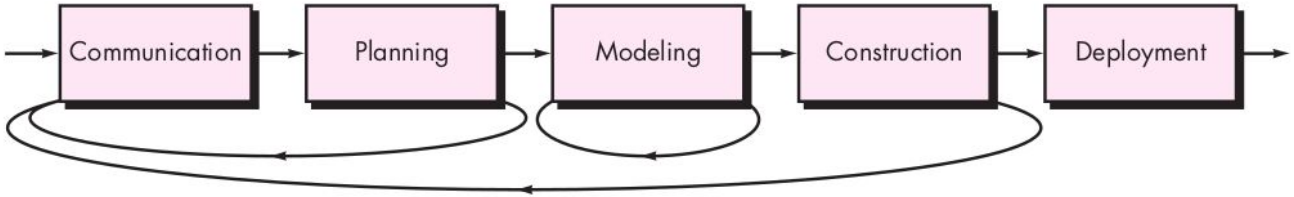


KEY POINT

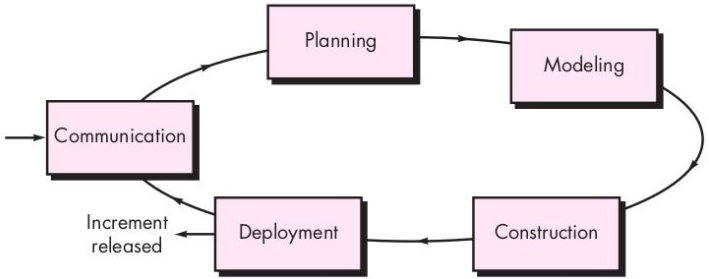
The hierarchy of technical work within the software process is activities, encompassing actions, populated by tasks.



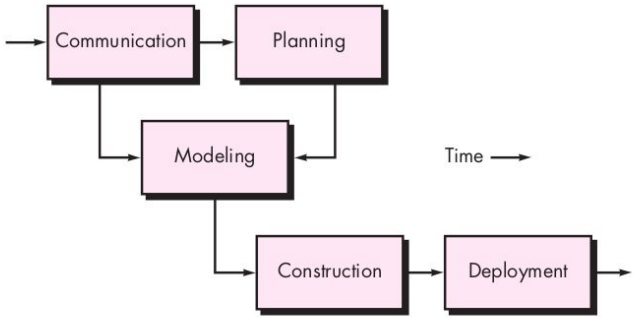
(a) Linear process flow



(b) Iterative process flow



(c) Evolutionary process flow



(d) Parallel process flow

Process Patterns

Um Padrão de Template fornece um consistente significado para descrever um padrão

Um padrão deve conter um par

- **Problema & Solução**
- **Contexto**

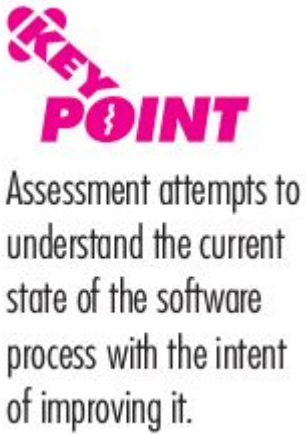
Pattern Name. The pattern is given a meaningful name describing it within the context of the software process (e.g., **TechnicalReviews**).

Forces. The environment in which the pattern is encountered and the issues that make the problem visible and may affect its solution.

Type. The pattern type is specified. Ambler [Amb98] suggests three types:

1. *Stage pattern*—defines a problem associated with a framework activity for the process. Since a framework activity encompasses multiple actions and work tasks, a stage pattern incorporates multiple task patterns (see the following) that are relevant to the stage (framework activity). An example of a stage pattern might be **EstablishingCommunication.** This pattern would incorporate the task pattern **RequirementsGathering** and others.
2. *Task pattern*—defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice (e.g., **RequirementsGathering** is a task pattern).
3. *Phase pattern*—define the sequence of framework activities that occurs within the process, even when the overall flow of activities is iterative in nature. An example of a phase pattern might be **SpiralModel** or **Prototyping.**[3]

Qualidade - Avaliação de Processos & Melhorias



A existência de um processo de NÃO É GARANTIA que o software vai ser entregue no tempo certo ou que vai reunir as necessidades dos consumidores, ou que o software produzido possui as características de qualidade necessárias.

Use Normas Internacionais para medir e avaliar a qualidade do Software e Processo

- Standard CMMI Assessment Method for Process Improvement (SCAMPI)
- CMM-Based Appraisal for Internal Process Improvement (CBA IPI)
- SPICE (ISO/IEC 15504)
- ISO 9001:2000 for Software
- DO-178 A/B para Sistemas Embarcados

Exemplo de Aplicação
UL-98, DO-178 C, IEEE-1044/2009

ANTONIO. (2014) RTSS: Uma Família de Técnicas de Leitura Para Suporte à Inspeção de Modelos SysML e Simulink. Tese. Universidade Federal de São Carlos.

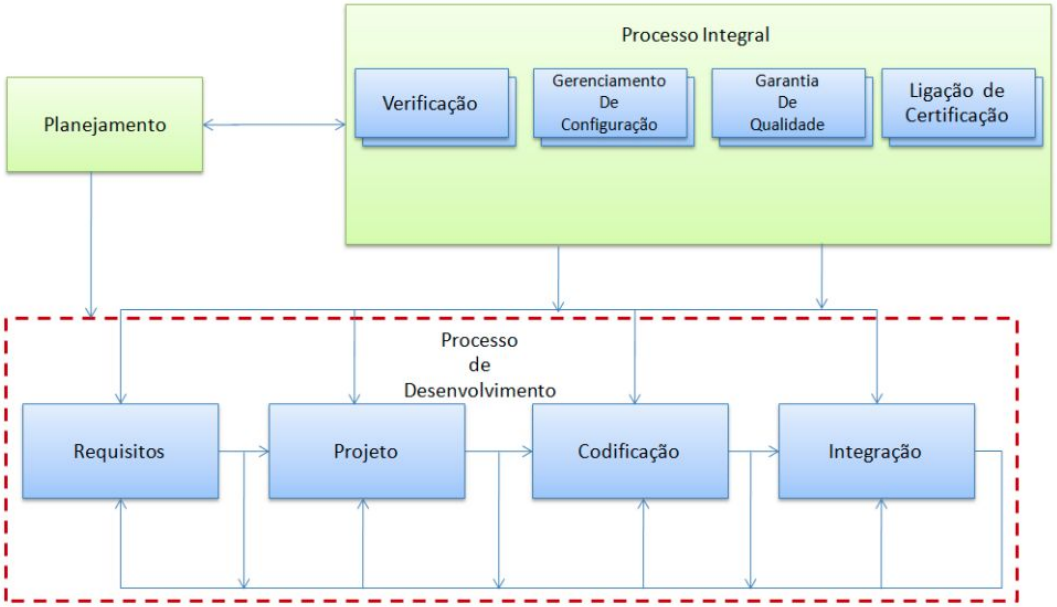


Figura 2.5 - Processo de desenvolvimento segundo norma de certificação DO-178B/ED-12B (adaptado(HAYHURST et al., 2001)).

Antonio (2014)

ANTONIO. (2014) RTSS: Uma Família de Técnicas de Leitura Para Suporte à Inspeção de Modelos SysML e Simulink. Tese. Universidade Federal de São Carlos.

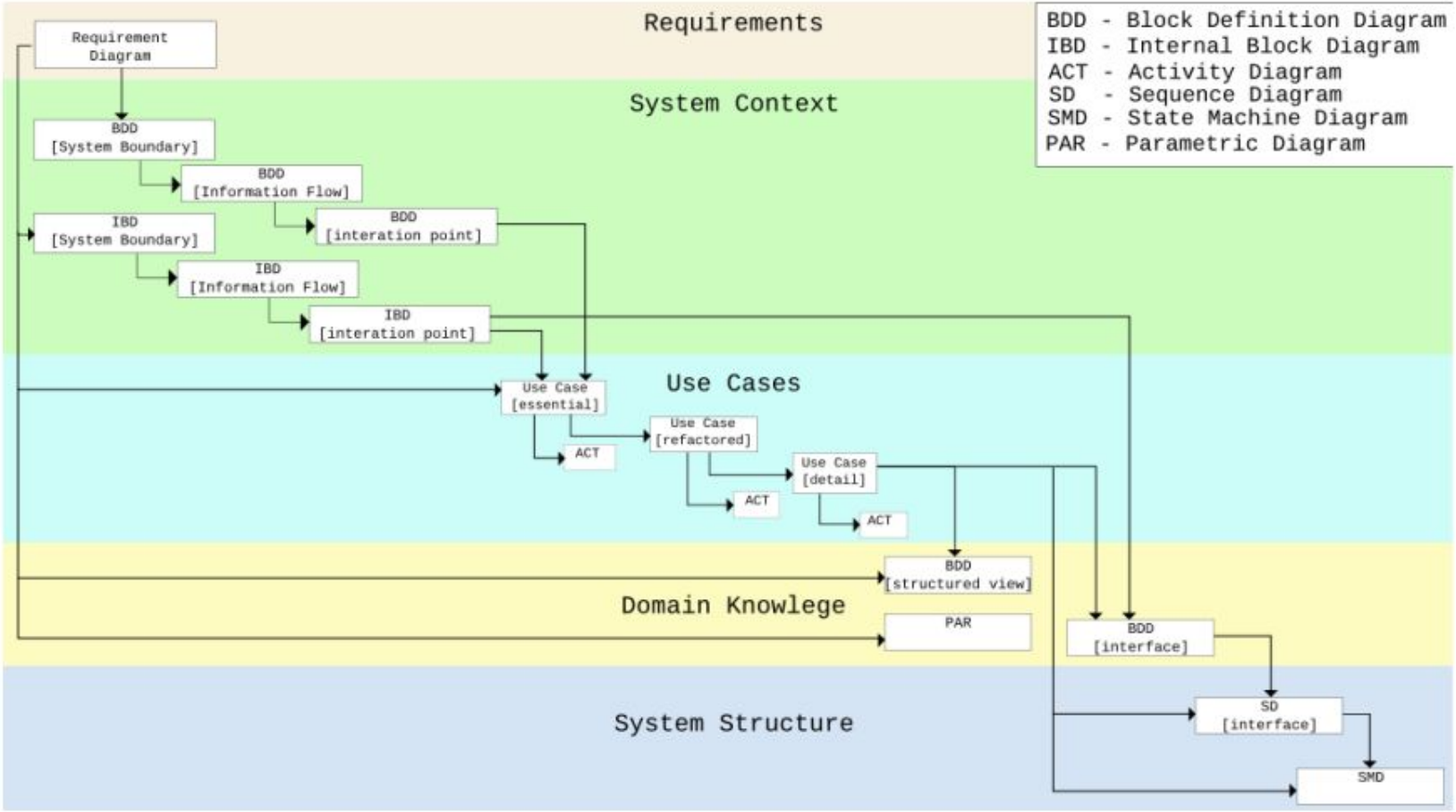


Figura 2.7 - Processo SYSMOD adaptado de (WEILKIENS, 2008).

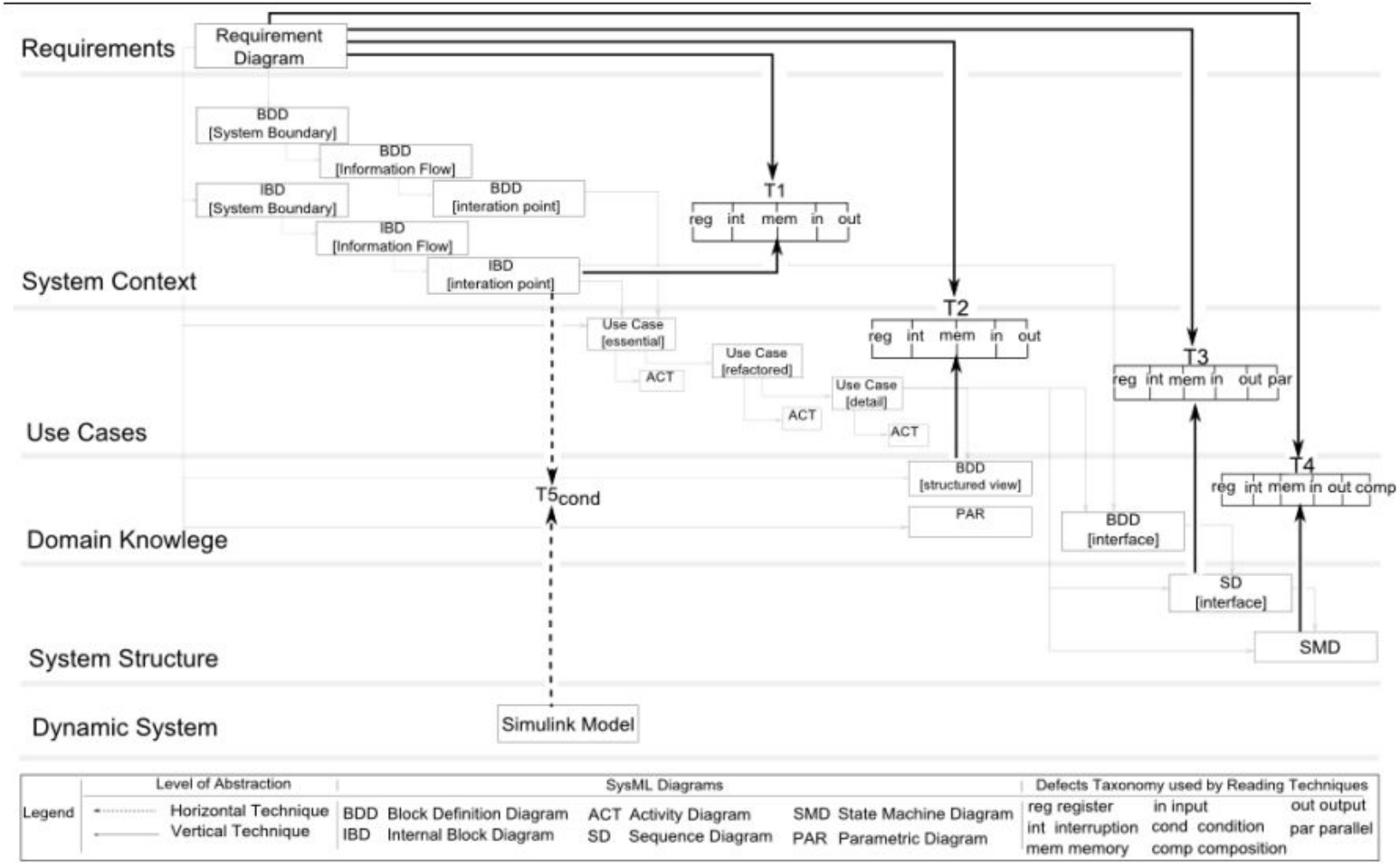
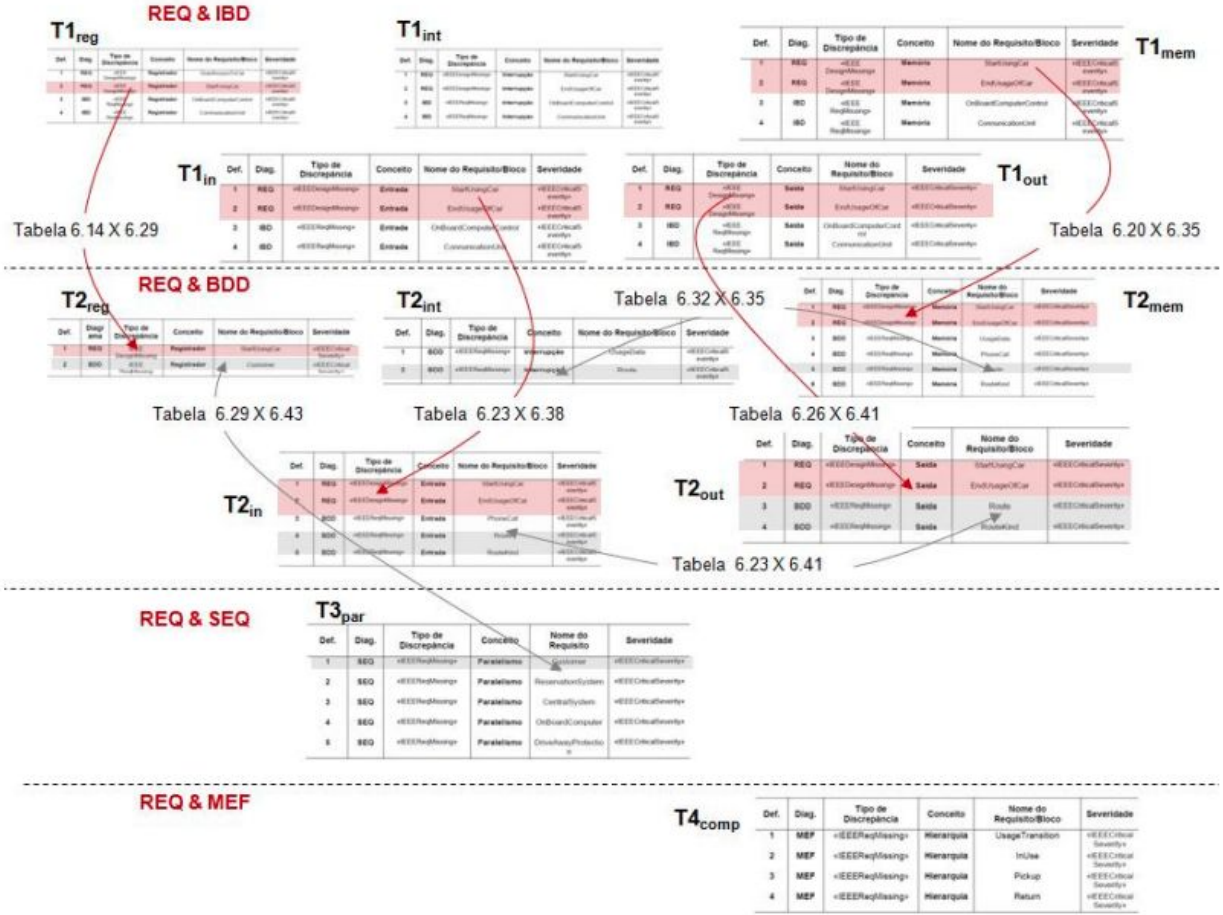


Figura 3.9 - Processo SYSMOD permeado com as técnicas RTSS.

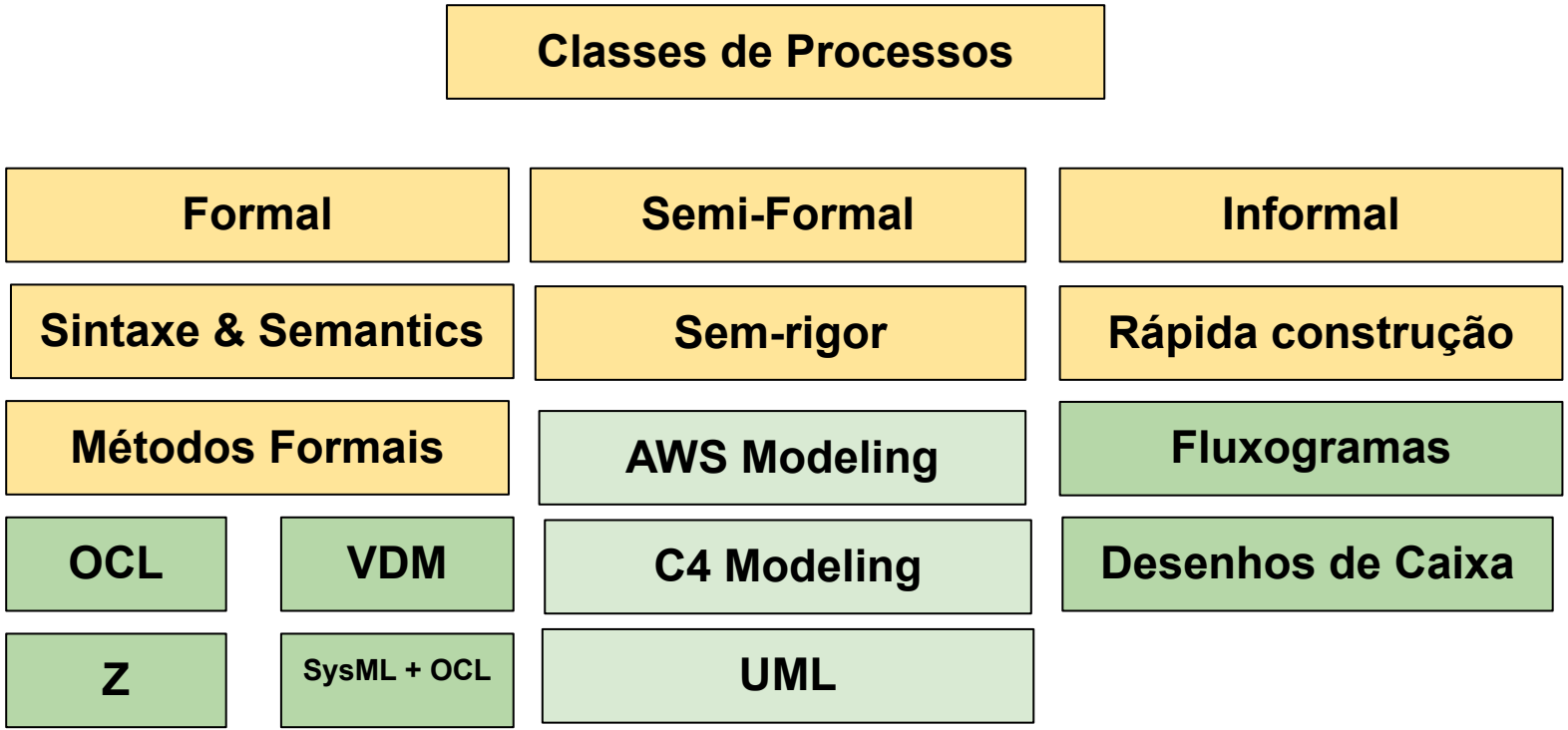


Propagação de defeitos caracterizados pela Engenharia de Software Experimental

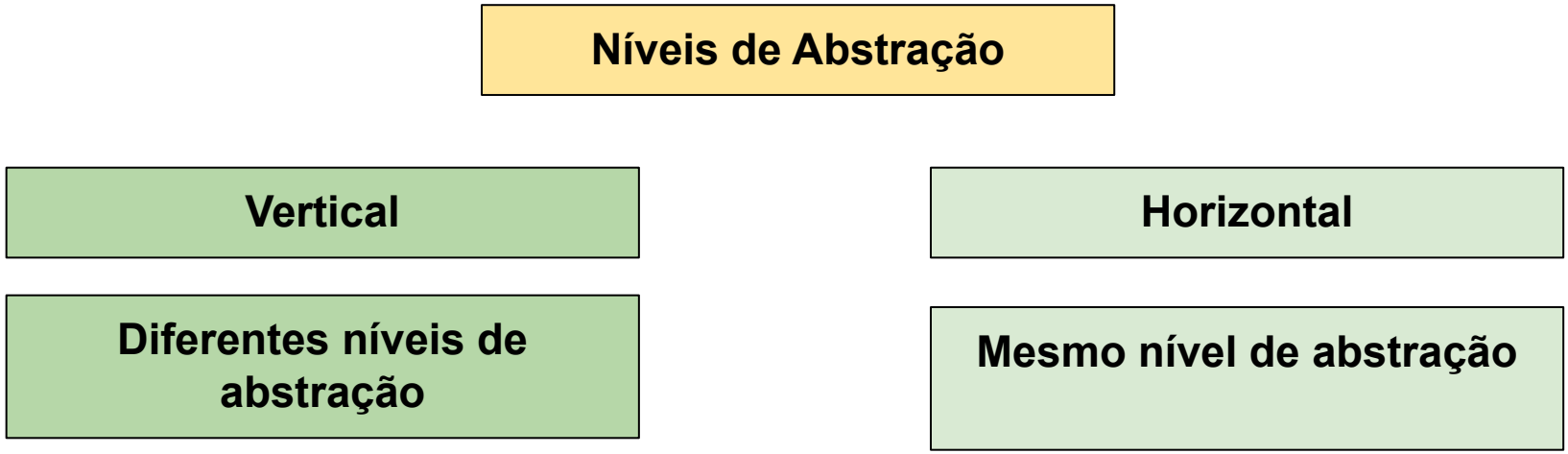
Conceitos importantes

- Classes de Formalismo
- Modelos de Processos em Engenharia de Software
- Níveis de Abstração
- Abordagens
  - Top-down
  - Bottom-up

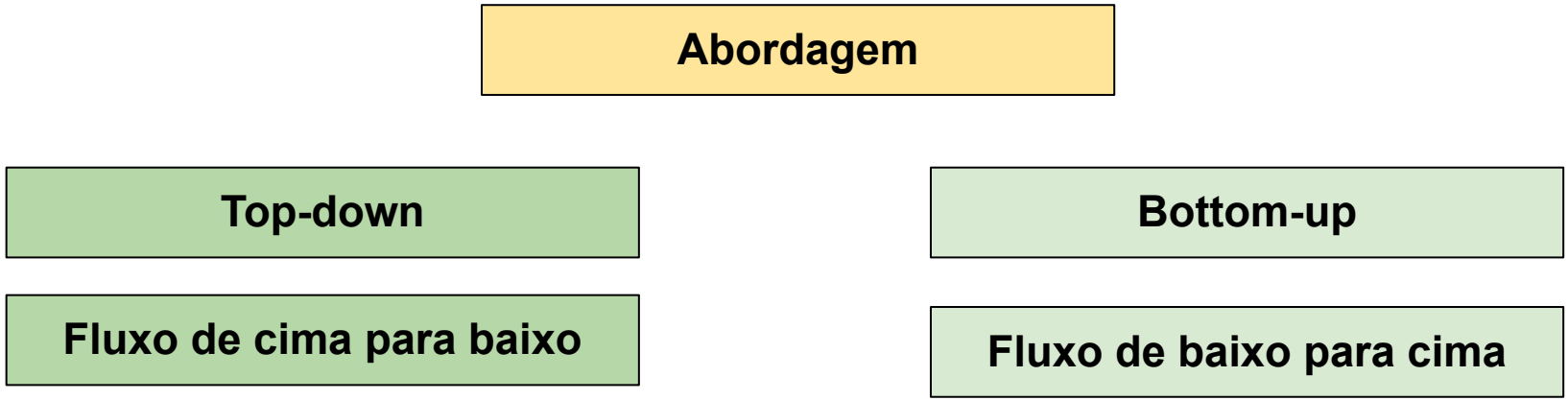
Classes de Processos



Níveis de Abstração



Abordagens


Abordagem
Top-down
Fluxo de cima para baixo
Bottom-up
Fluxo de baixo para cima

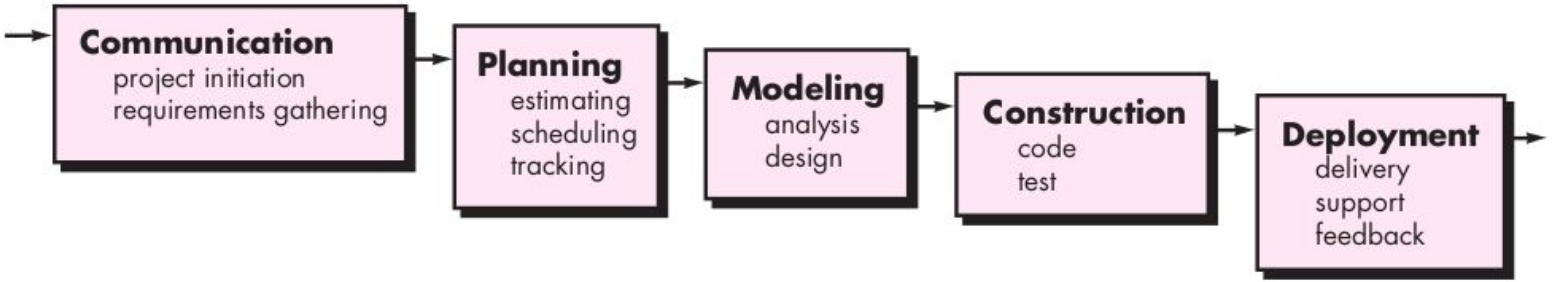
Modelo de Processo Prescritivo

Define-se um conjunto formal ou semi-formal de atividades e passos que devem ser seguidos com um objetivo

Modelos de Processos Prescritivo

- **Workflow**
- **Step-by-step**
- **Incremental**
- **Waterfall Model**
- **V-Model**
- **Evolucionário**
- **Spiral Model**
- **Concurrent Model**
- **Pipeline**

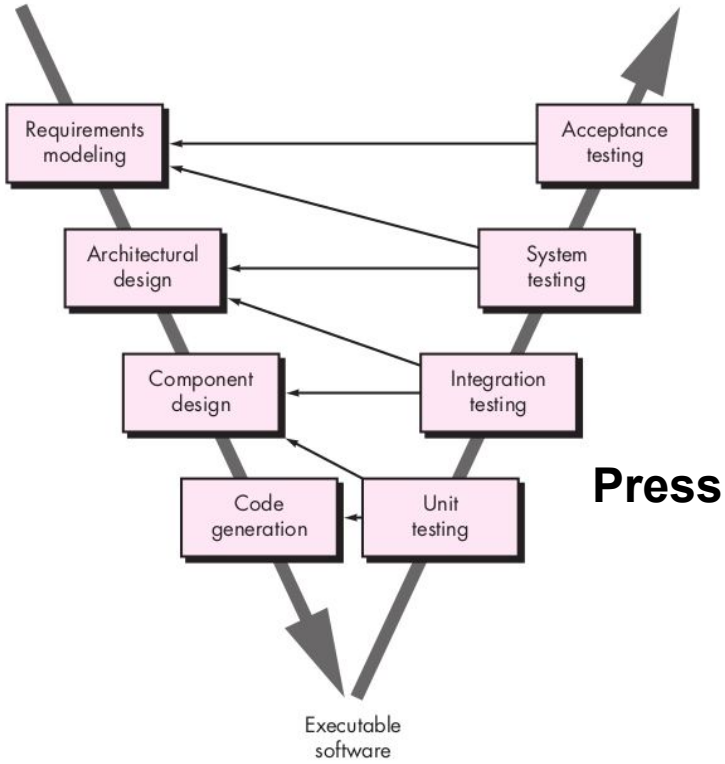
FIGURE 2.3 The waterfall model



Pressman (2010, c.2)

FIGURE 2.4

The V-model



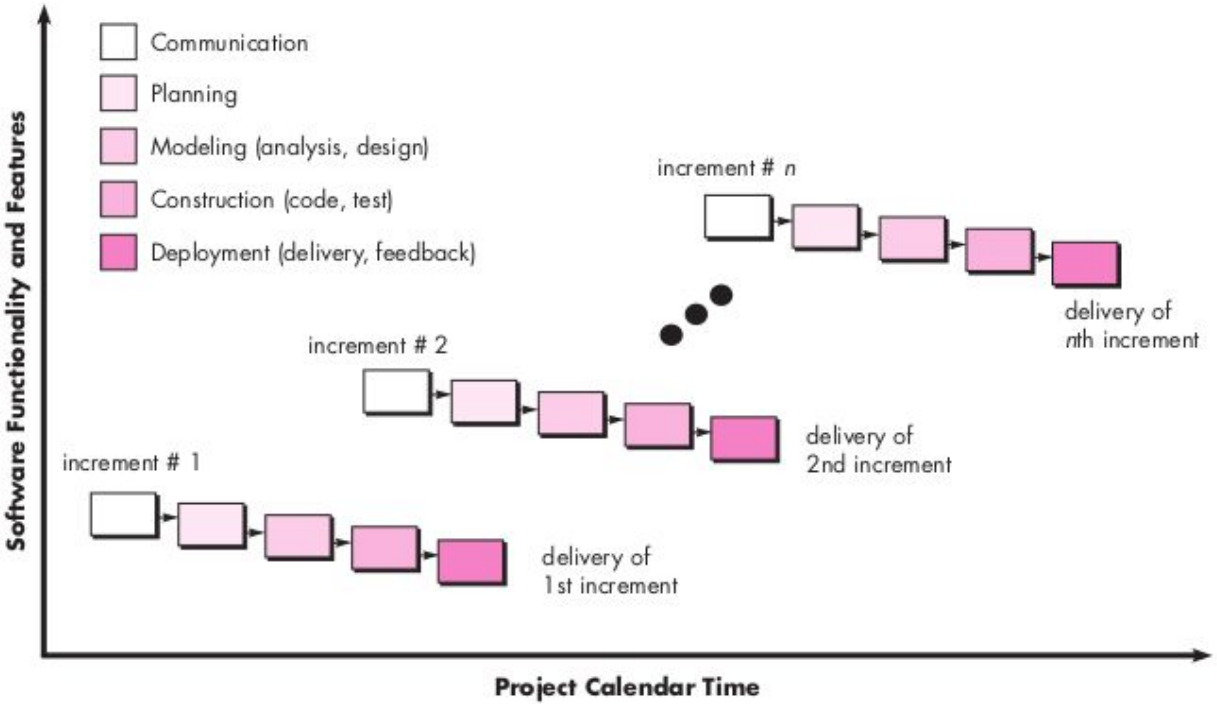
Pressman (2010, c.2)

Quiz

Indique pontos **positivos** e **negativos** no uso de processos em cascata

FIGURE 2.5

The incremental model



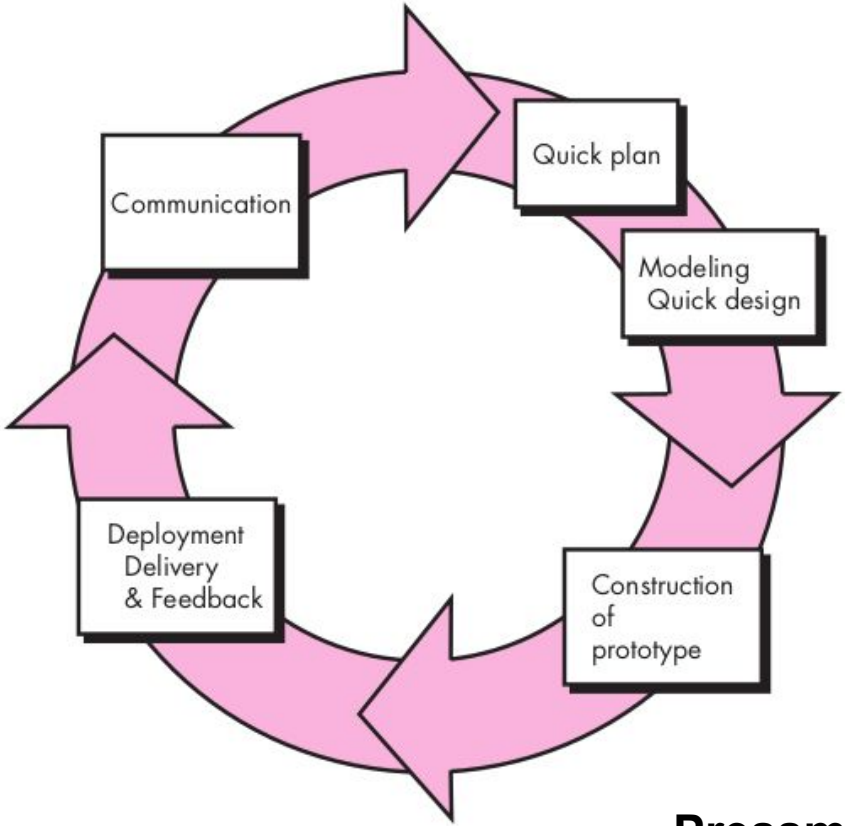
KEY POINT

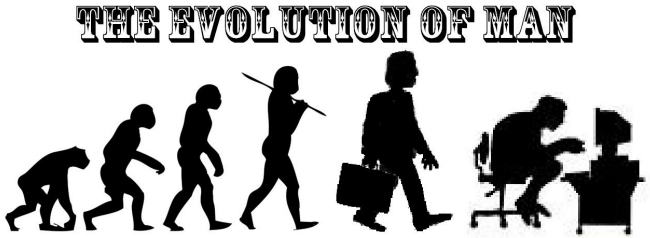
The incremental model delivers a series of releases, called increments, that provide progressively more functionality for the customer as each increment is delivered.

Pressman (2010, c.2)

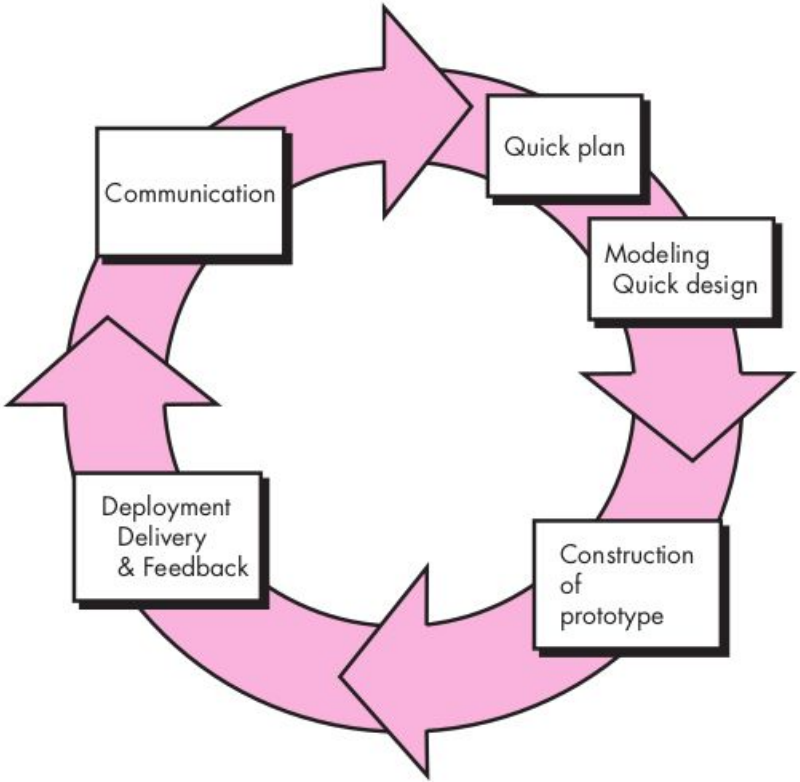
FIGURE 2.6

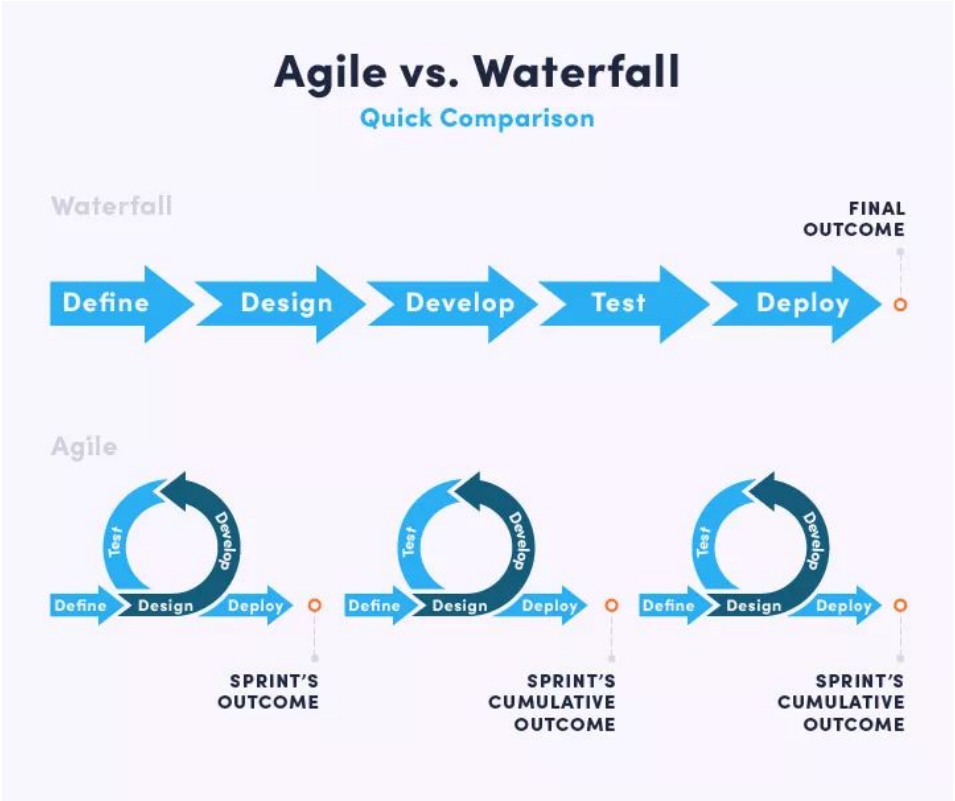
The prototyping paradigm





Pressman (2010, c.2)



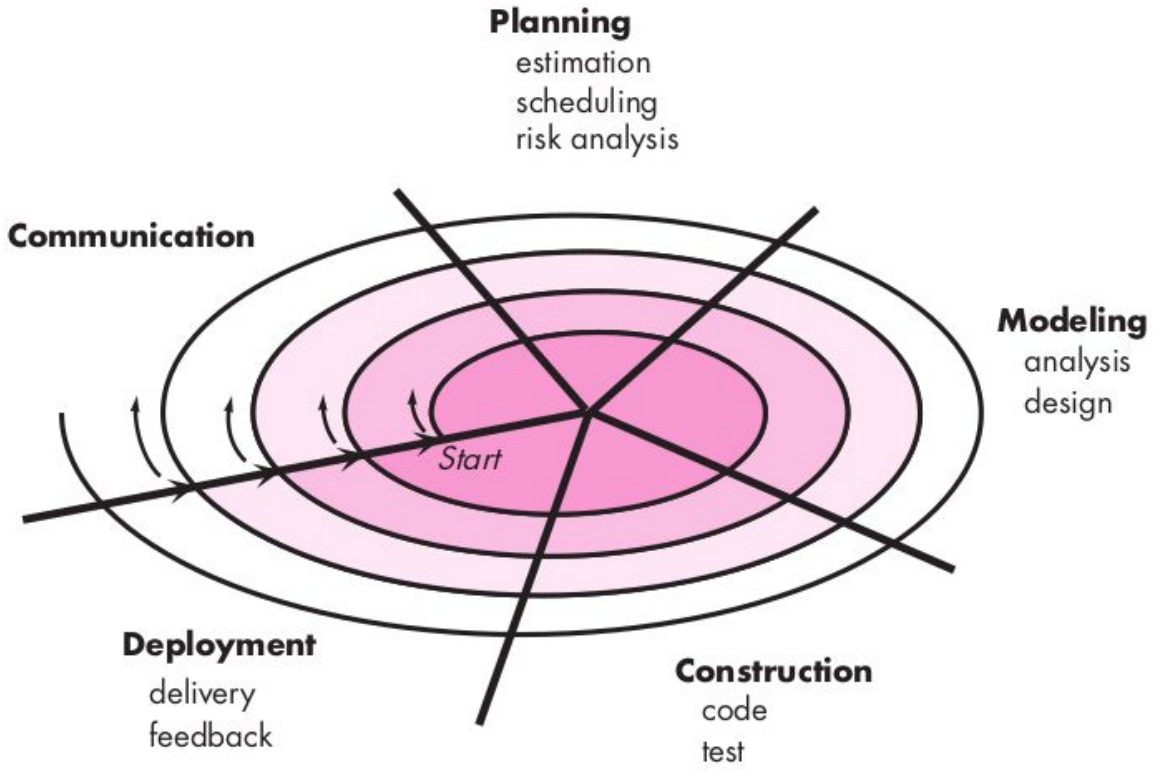


Agile

has(Incremental && Iterative)

FIGURE 2.7

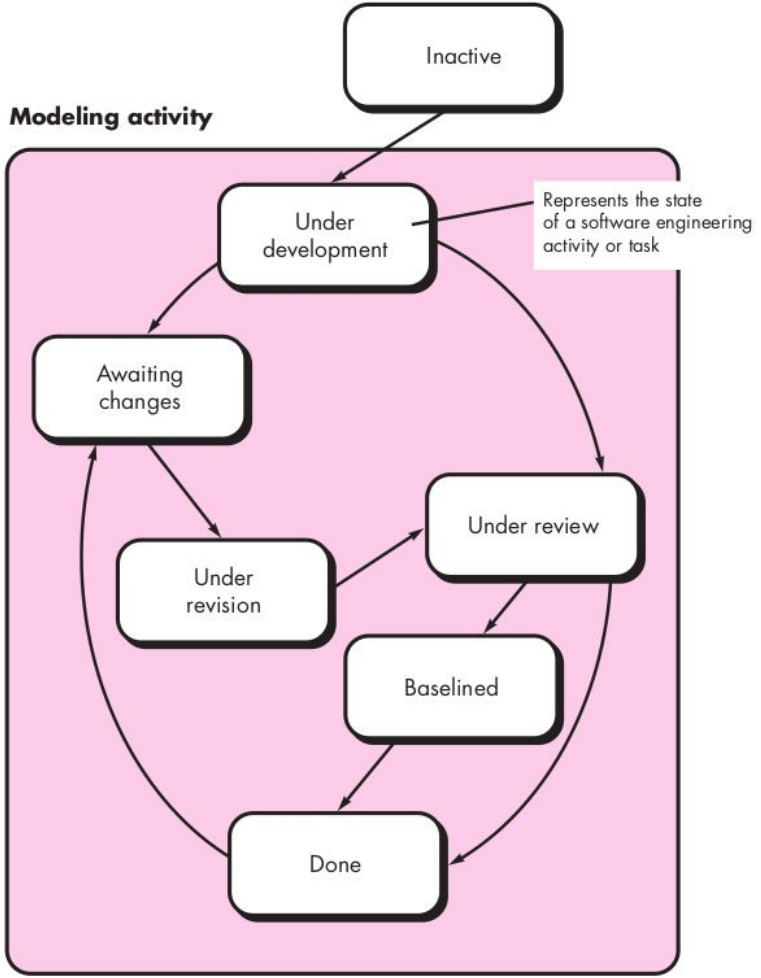
A typical spiral model



Modelo Espiral de Barry Boehm (1988)

Pressman (2010, c.2)

Modelos Concorrentes

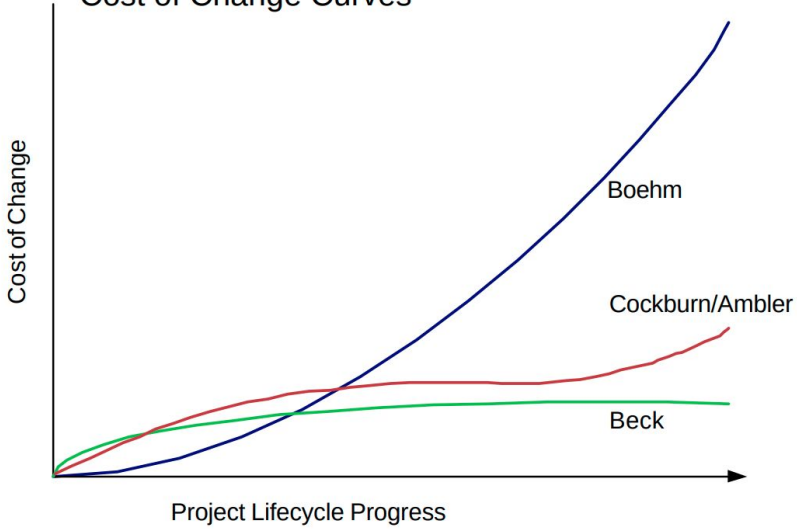


Quiz

Indique pontos **positivos** e **negativos** no uso de processos evolucionários como a Prototipação e o Processo em Espiral

Agilidade != Velocidade

Figure 3:
Cost of Change Curves



SZALVAY, Victor. An introduction to agile software development. **Danube technologies**, v. 3, 2004.

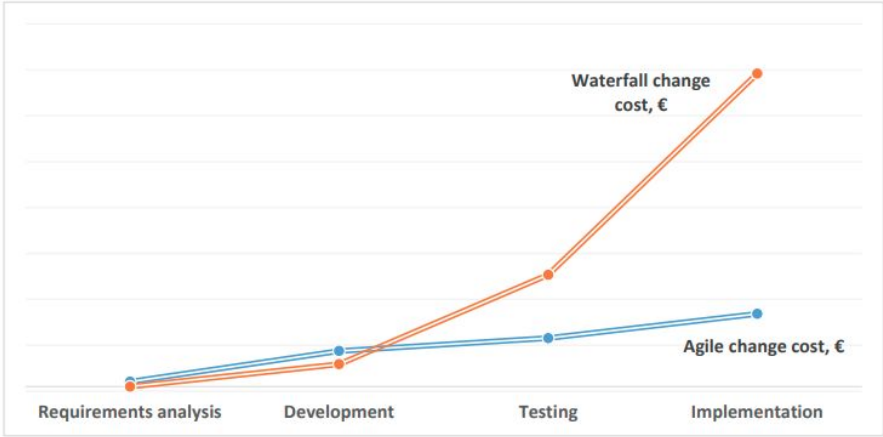


Fig. 9. Cost change curves of case study projects.

BORMANE, Līga et al. Impact of requirements elicitation processes on success of information system development projects. **Information Technology and Management Science**, v. 19, n. 1, p. 57-64, 2016.

Leituras recomendadas

- **Capítulo 1 & 2 do Pressman**
- **Curva de custo de mudança de software**
  - **http://www.agilemodeling.com/essays/costOfChange.htm**

- **Artigos**
  - **https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.451.9579&rep=rep1&type=pdf**
  - **https://itms-journals.rtu.lv/article/view/itms-2016-0012**

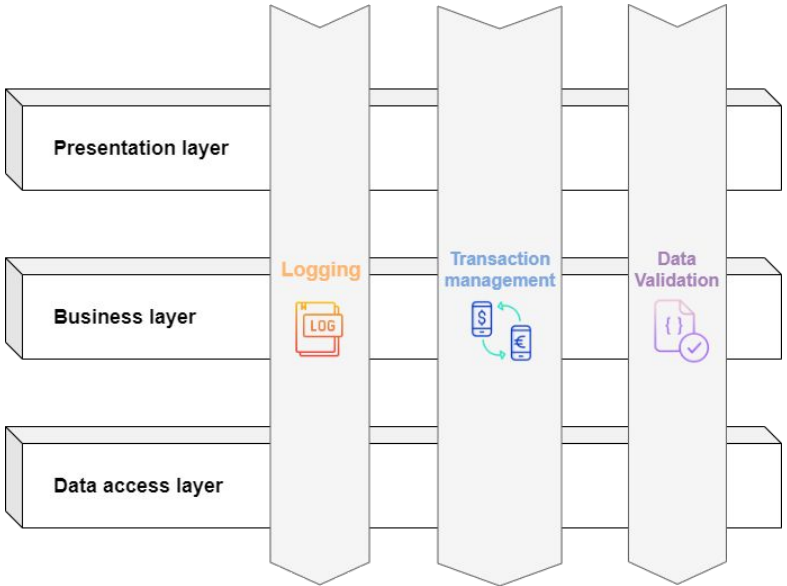
Atividade em Grupo

Faça a leitura do artigo original da curva de custo de software e responda a seguinte pergunta.

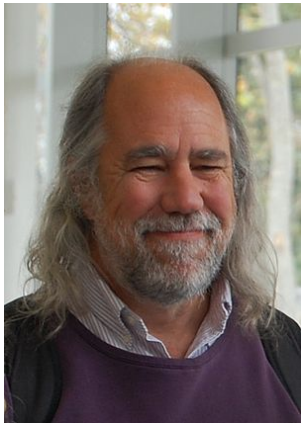
Suponha que você precise ajustar um processo que atualmente tem um alto custo operacional. Quais são os atributos de Métodos Ágeis que podem ser utilizados para incluir ao longo de um processo ágil para que este tenha um menor custo operacional de entrega de valor ?

Modelos de Processo de Software (outros)

- Component-Based Development
  - COTS System (Sistemas de Prateleira)
- Formal Methods Model
  - Métodos Matemáticos para Especificar, Avaliar e Verificar
- Aspect-Oriented Software Development
  - AOSD
  - Interesses transversais



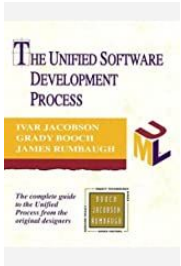
https://saigontechnology.com/blog/an-introduction-to-aspect-oriented-programming



Grady Booch

Ivar Jacobson

James Rumbaugh

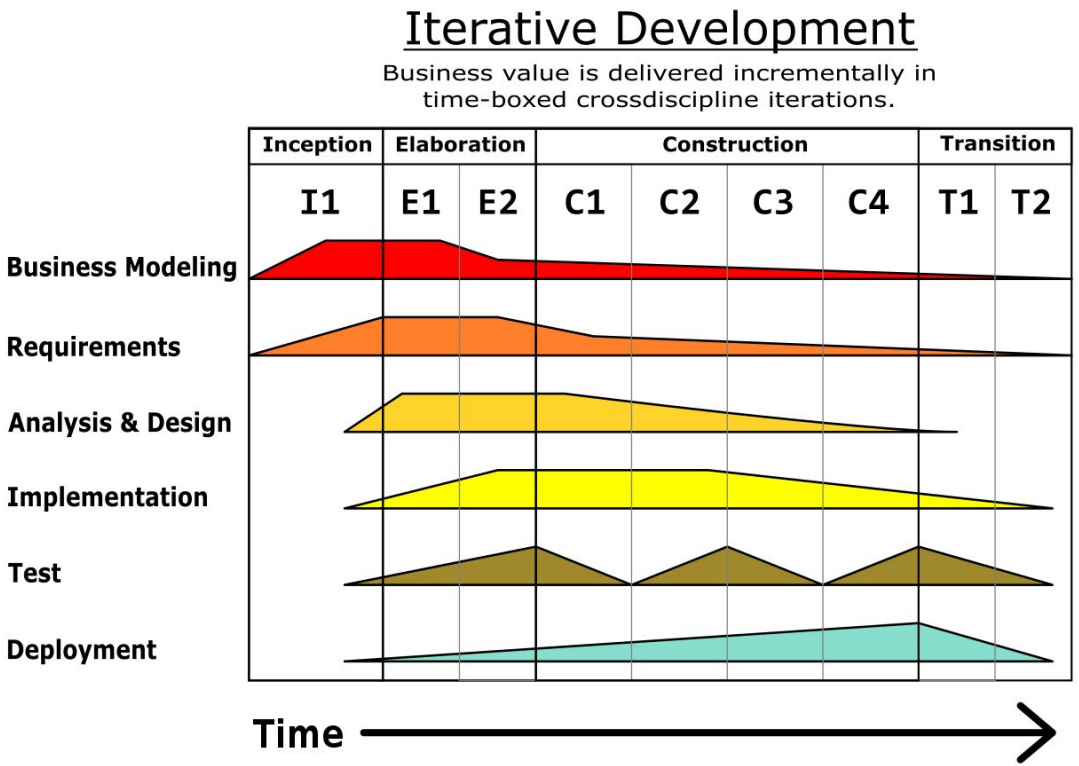


UNIFIED MODELING LANGUAGE™

In their seminal book on the *Unified Process,* Ivar Jacobson, Grady Booch, and James Rumbaugh [Jac99] discuss the need for a "use case driven, architecture-centric, iterative and incremental" software process when they state:

> Today, the trend in software is toward bigger, more complex systems. That is due in part to the fact that computers become more powerful every year, leading users to expect more from them. This trend has also been influenced by the expanding use of the Internet for exchanging all kinds of information. . . . Our appetite for ever-more sophisticated software grows as we learn from one product release to the next how the product could be improved. We want software that is better adapted to our needs, but that, in turn, merely makes the software more complex. In short, we want more.

UP - Unified Process



Atividade - Ajuste o Diagrama 1 com o Diagrama 2

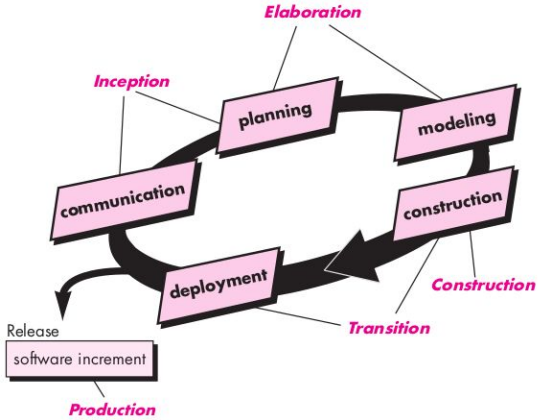


Diagrama 1 (PRESSMAN, 2010)

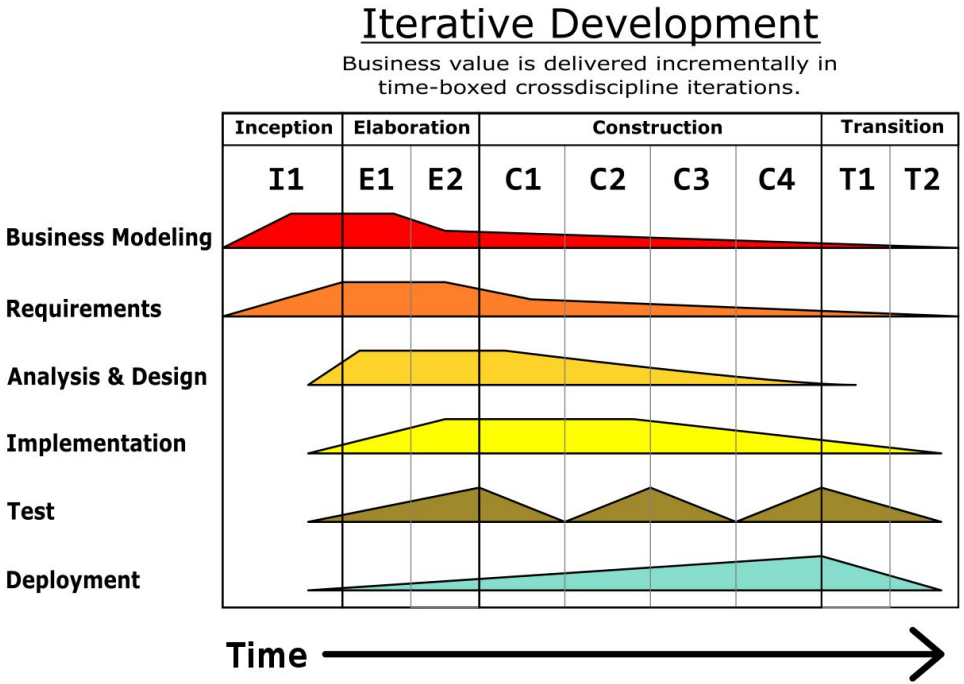


Diagrama 2 - UP