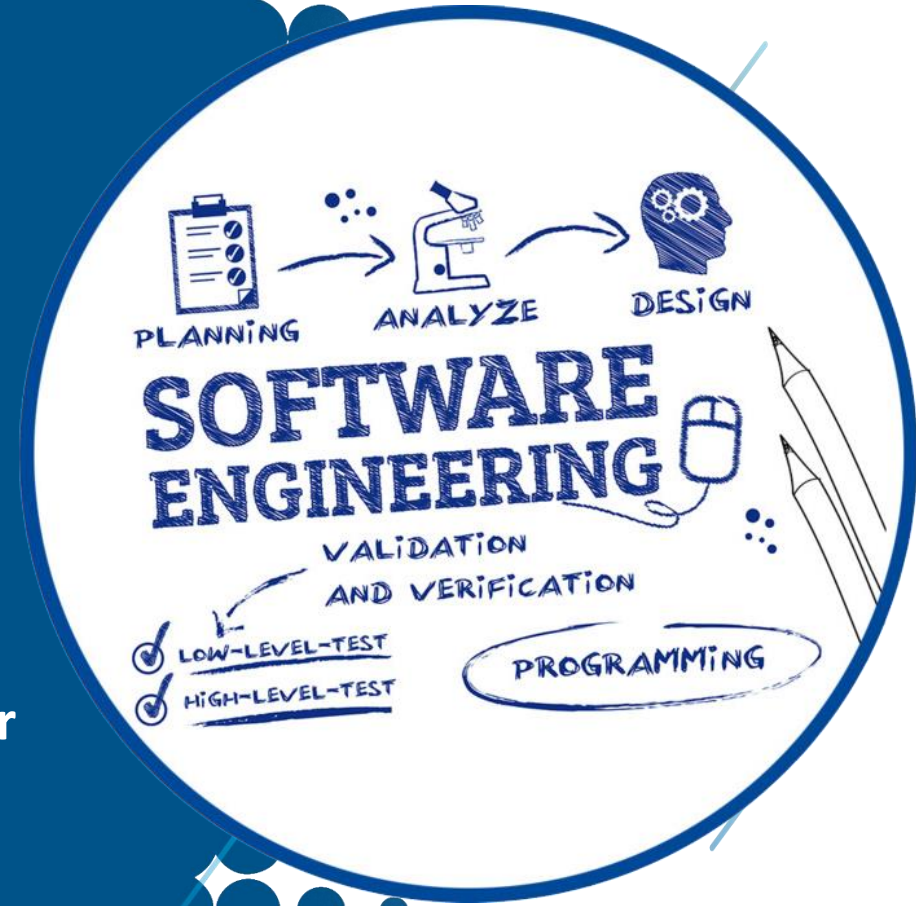


# Engenharia de Software I

Prof<sup>ª</sup>. Me. Cynara Leão Garcia  
[cynara.garcia@unicesumar.edu.br](mailto:cynara.garcia@unicesumar.edu.br)



An illustration depicting software development. A person in a yellow shirt and blue pants is climbing a ladder on the right side, reaching towards a large screen displaying code. The screen shows various lines of code in different colors (blue, grey, white) on a dark background. A blue puzzle piece is visible on the left side of the screen. A blue speech bubble containing the code symbol '</>' is positioned near the person. The overall scene is set against a dark background with a blue and orange color scheme.

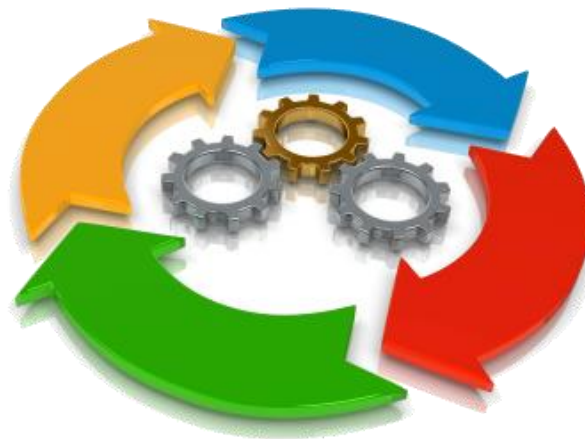
# Processos de Software

# O que é um processo de software?



## O que é um Processo de Software?

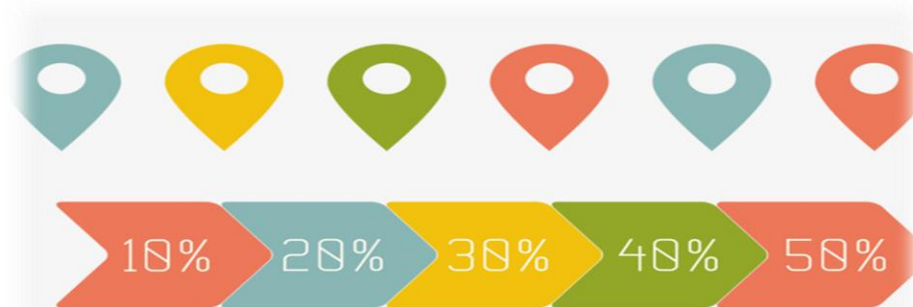
- Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Essas atividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem padrão de programação como C# ou Java por exemplo.



## Modelos de Processo de Software

O modelo de um processo de software é uma representação simplificada de um processo de software, e existem vários modelos ou paradigmas de engenharia de software.

Cada uma representa uma tentativa de colocar ordem em uma atividade inerentemente caótica.



## Vantagens

- Oferecem um roteiro útil para o trabalho de engenharia de software.

Mas nenhum modelo de processo é perfeito.

- Padronização dos artefatos;
- Melhor comunicação da equipe;
- Menos treinamento de pessoal.

# Exemplos de Modelos de Processos

- Modelos de processo mais gerais
  - Modelo Cascata;
  - Desenvolvimento Incremental;
  - Engenharia de Software Orientada a Reuso
- Modelos que lidam com mudanças
  - Prototipagem;
  - Entrega Incremental;
  - Modelo Espiral.



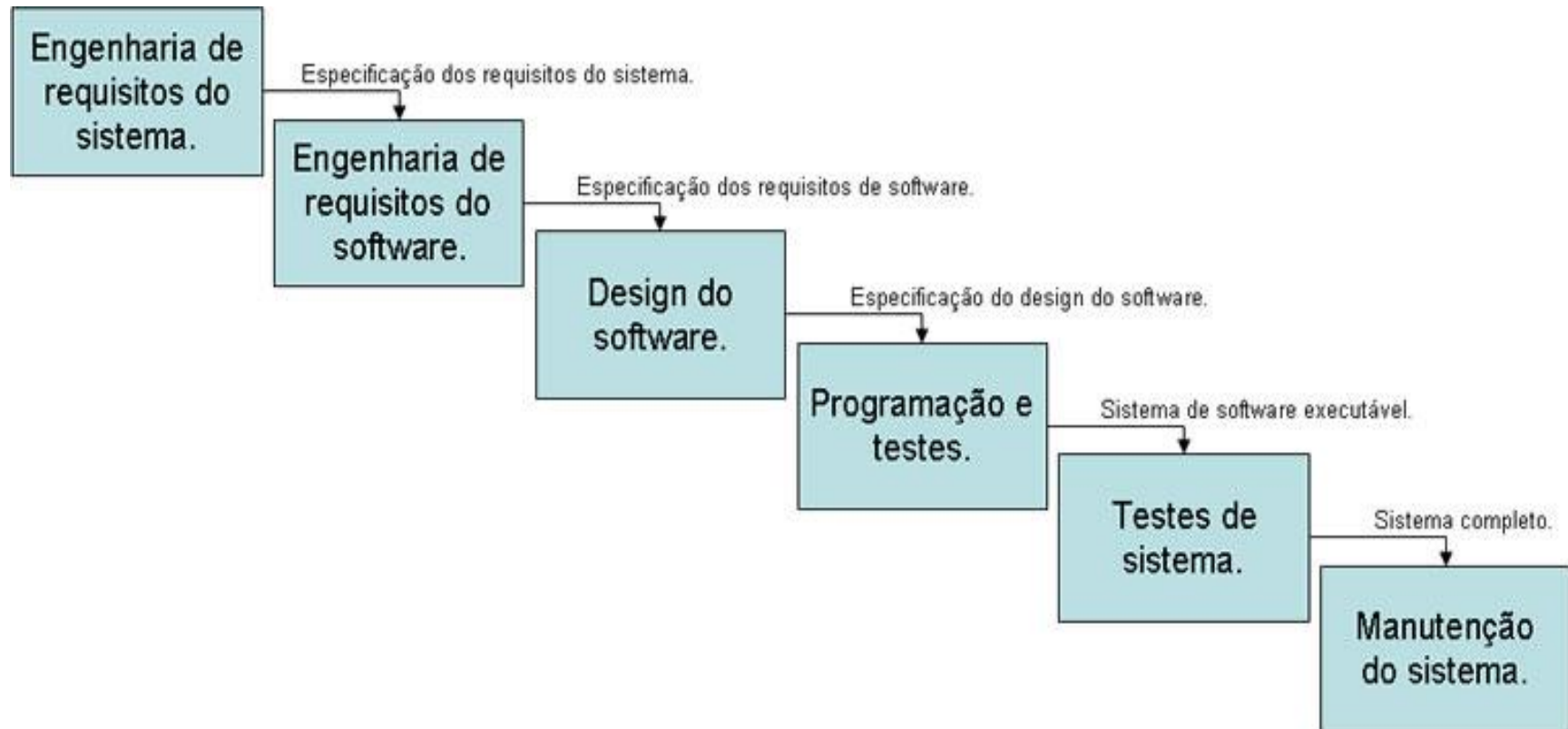


# Modelo em Cascata (Waterfall)

---



# Modelo em Cascata (Waterfall)



# Modelo em Cascata (Waterfall)

- Características
  - Composto por uma sequência de atividades;
  - Uma atividade começa a executar quando a outra termina;
  - Resultado de uma etapa é utilizado na etapa seguinte;
  - Guiado por documentos;
  - Ciclo de vida mais antigo e mais utilizado.

# Modelo em Cascata (Waterfall)

## 1- Engenharia de Requisitos do Sistema

Objetivos, funções e restrições são definidos, com a ajuda de clientes e usuários, e servem como uma especificação do sistema, indicando o que deve ser implementado.

## 2 – Engenharia de Requisitos do Software

Como o software será desenvolvido, linguagem de programação, mão de obra, softwares secundários necessários, equipamentos.

# Modelo em Cascata (Waterfall)

## **3 – Design do Software**

Envolve a descrição do sistema e do software em termos de unidades abstratas e de suas relações, indicando como o software será implementado.

## **4 – Programação e Testes**

As unidades do software devem ser codificadas e testadas individualmente.

# Modelo em Cascata (Waterfall)

## 5 – Testes de Sistema

As unidades são integradas e testadas.

## 6 – Manutenção do Sistema

O sistema é instalado e colocado em operação. A manutenção envolve a correção de erros e evolução do sistema para atender a novos requisitos.

# Modelo em Cascata (Waterfall)

## Vantagens

- Documentação rígida (idealmente completa) em cada atividade;
- Reflete abordagens adotadas em outras engenharias;
- Aderência a outros modelos de processo.



# Modelo em Cascata (Waterfall)

## Problemas comuns no modelo cascata

- Projetos reais raramente seguem o fluxo sequencial que o modelo propõe;
- Logo no início é difícil estabelecer explicitamente todos os requisitos. No começo dos projetos sempre existe uma incerteza natural;
- O cliente deve ter paciência. Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento (instalação)
- Difícil identificação de sistemas legados.



# Modelo em Cascata (Waterfall)

## Quando aplicar o modelo cascata?

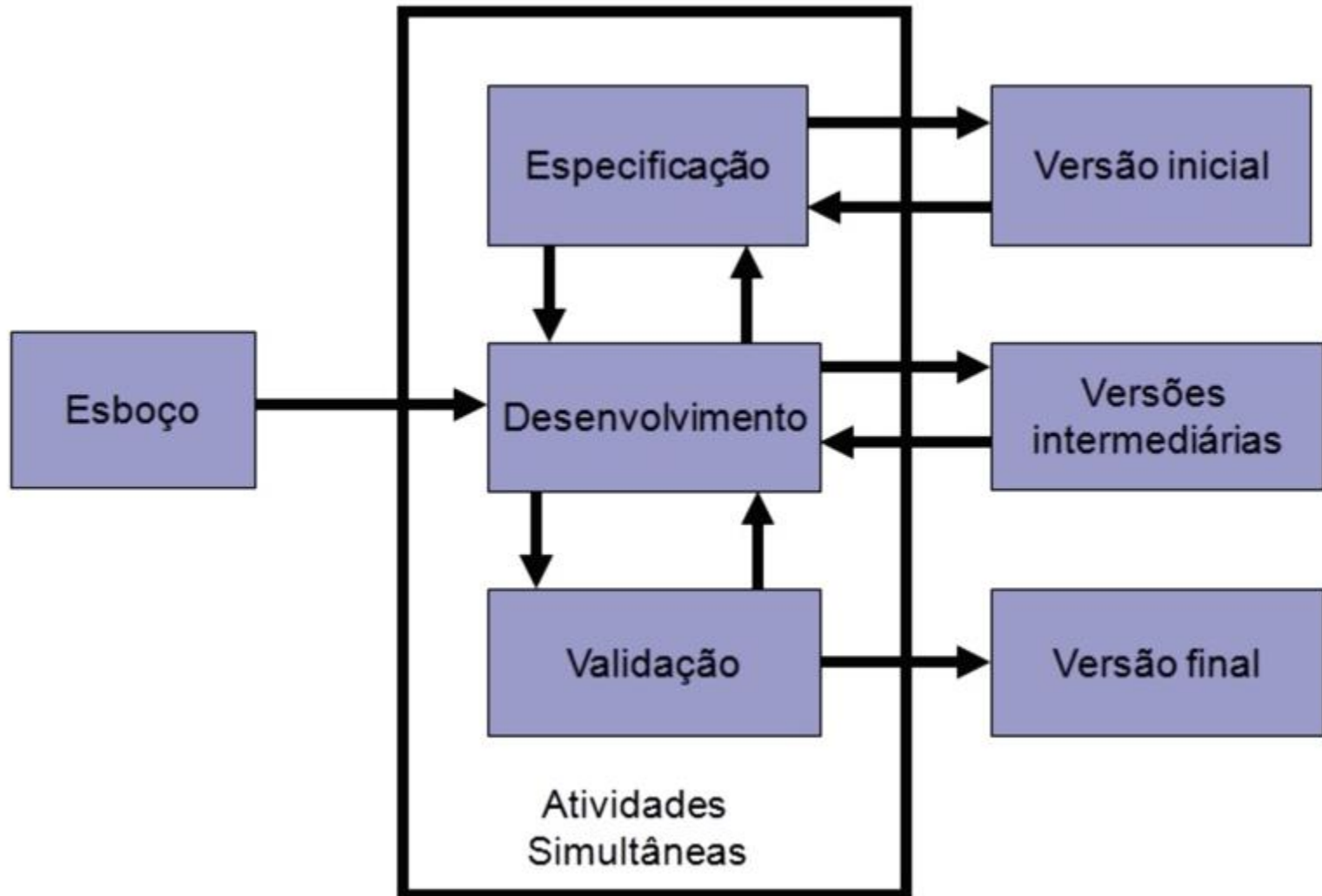
- Sistemas Críticos;
- Quando os requisitos são bem compreendidos;
- Quando há pouca probabilidade dos requisitos mudarem.





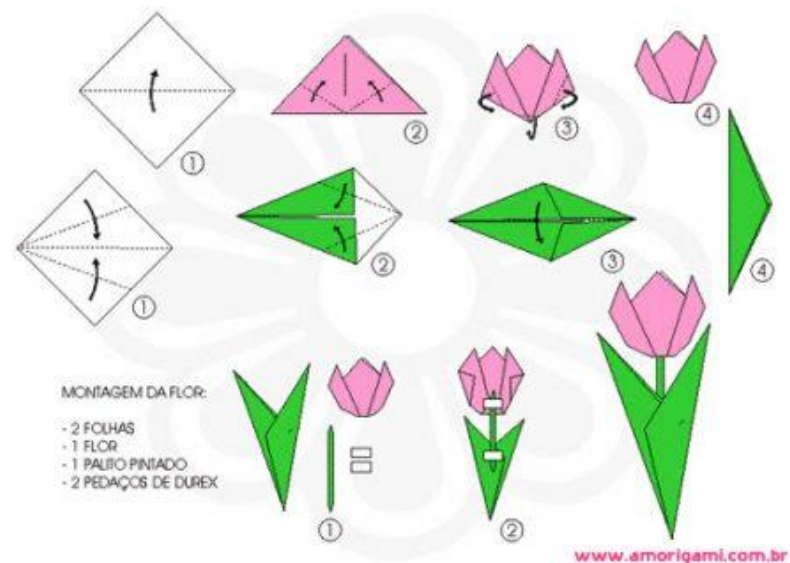
Modelo Incremental

# Modelo Incremental



## Modelo Incremental

O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões, até que um sistema adequado seja desenvolvido.

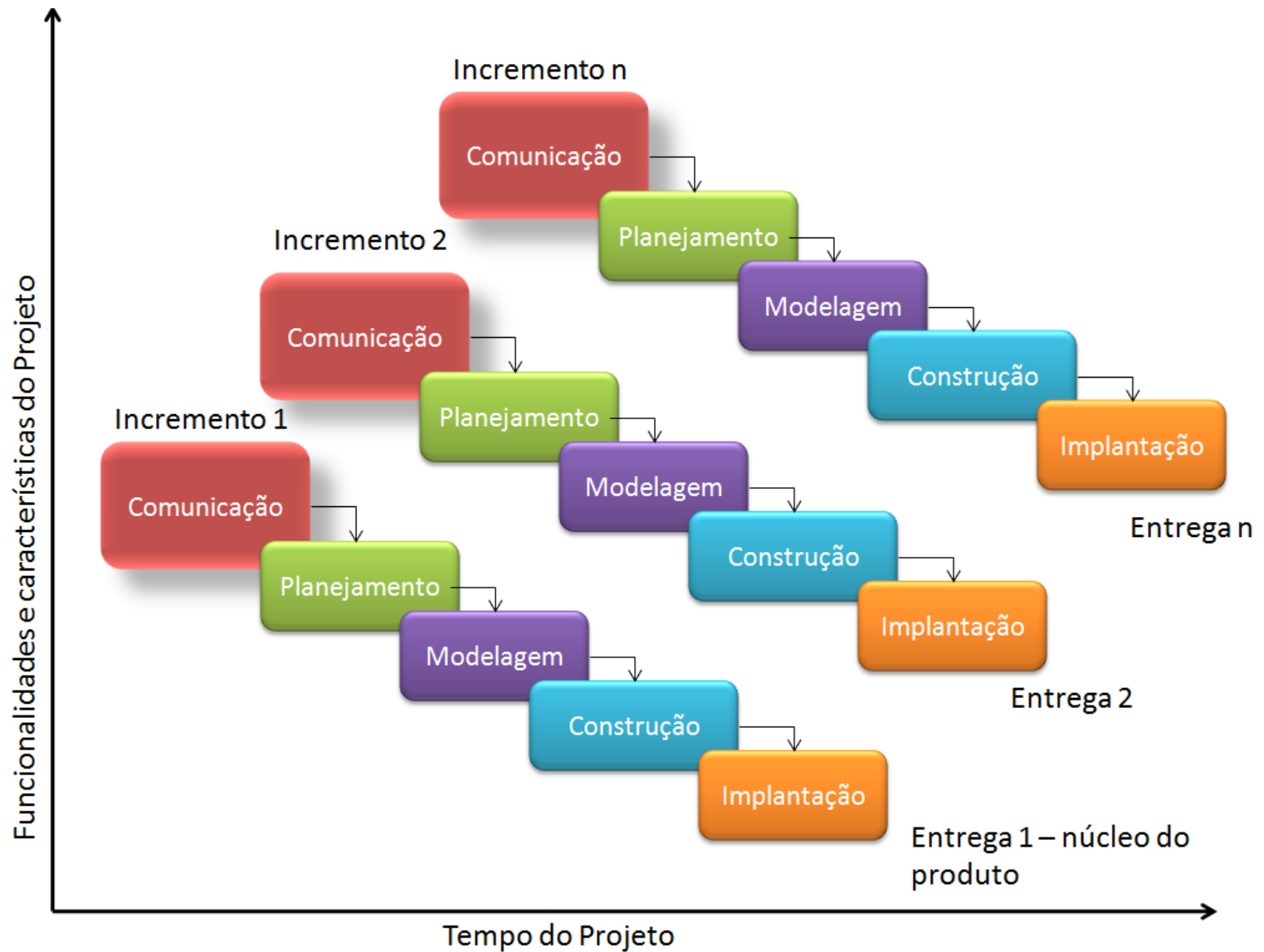


# Modelo Incremental

- Atividades são intercaladas;
- Objetivo: dar feedback rápido ao cliente.



# Modelo Incremental



## Vantagens

- Custo de acomodar mudanças nos requisitos é reduzido;
- Mais fácil obter feedback do cliente;
- Permite trabalhar com o cliente o entendimento dos requisitos;
- Pode-se começar o sistema pelas partes melhor entendidas.



## Desvantagens

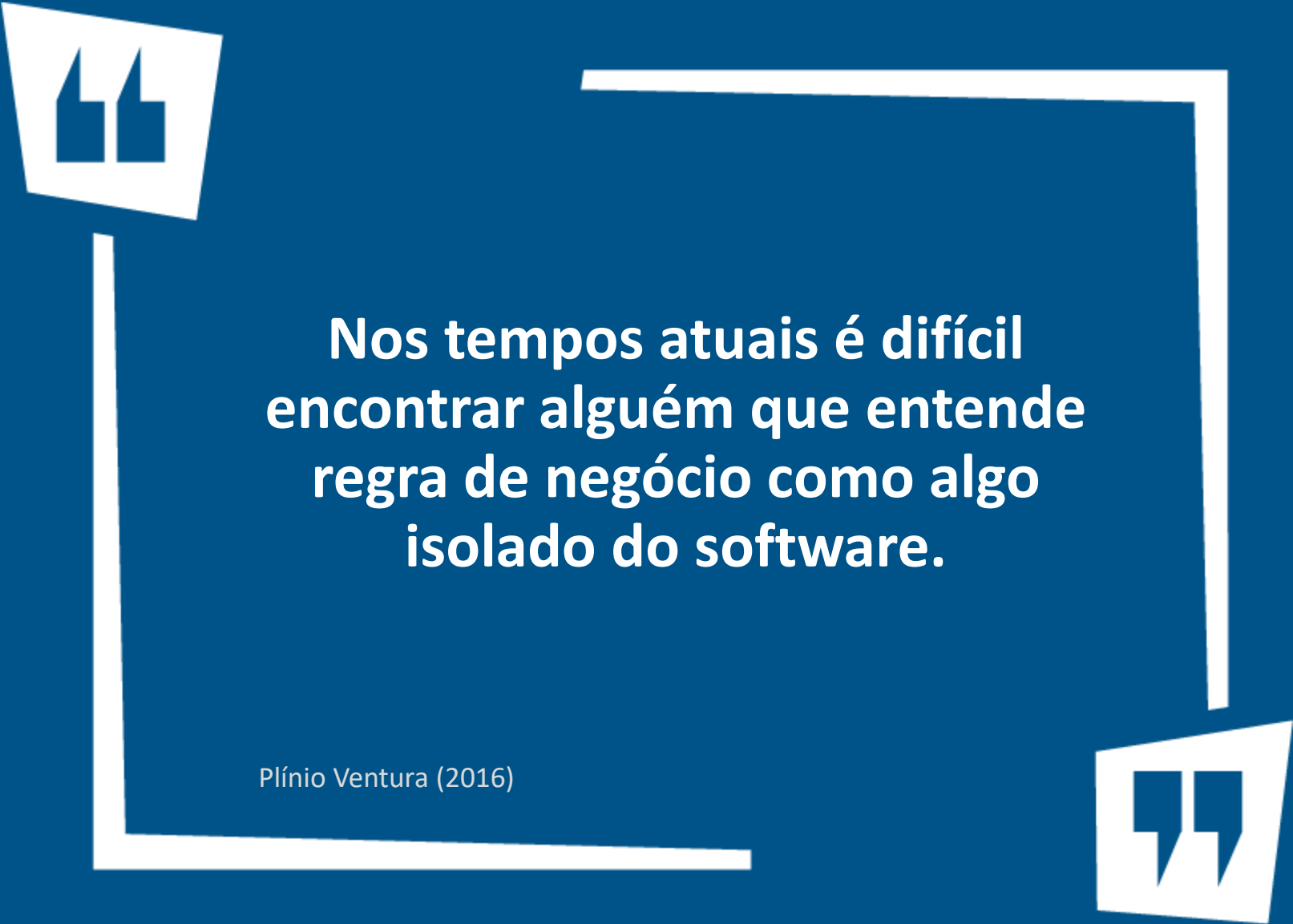
- O processo pode não ser muito claro;
- A gerência do software é complicada;
  - O sistema não é completamente especificado à priori.
- A estrutura do produto tende a se corromper com a adição de incrementos
  - O produto final pode se tornar mal estruturado.





# Atividade



A decorative graphic consisting of white lines and icons on a blue background. It includes a top-left icon of two upward-pointing chevrons, a top-right icon of two downward-pointing chevrons, and a bottom-right icon of two upward-pointing chevrons. A white line runs horizontally across the top, and another runs vertically on the right side, meeting at a corner. A third line runs vertically on the left side, and a fourth runs horizontally across the bottom, meeting at a corner.

**Nos tempos atuais é difícil  
encontrar alguém que entende  
regra de negócio como algo  
isolado do software.**

Plínio Ventura (2016)

