# Engenharia de Software

**Professor:**
Zady Castaneda Salazar

# Aula 3

## Processos, Métodos e Modelos da Engenharia de Software

## Agenda:

- Princípios da Engenharia de software
- Métodos, ferramentas e processos de desenvolvimento de software
- Modelo de processo de software

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Princípios da Engenharia de Software

- **Formalidade**: produtos mais confiáveis, controlar seu custo e mais confiança no seu desempenho;
- **Abstração**: identificar os aspectos importantes, ignorando os detalhes
- **Decomposição**: subdividir o processo em atividades específicas, atribuídas a diferentes especialistas
- **Generalização**: sendo mais geral, é bem possível que a solução possa ser reutilizada
- **Flexibilização**: modificação com facilidade.

**Queremos construir um balanço no jardim para nosso filho?**

# Engenharia de software

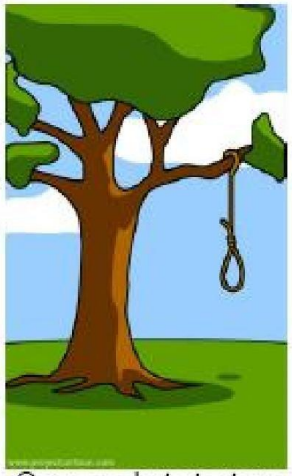



Como o cliente explicou

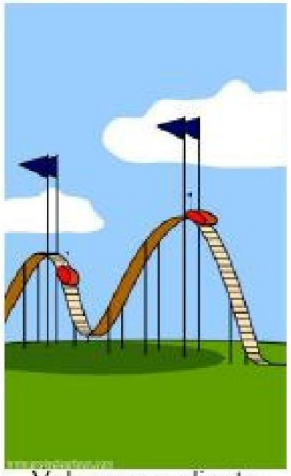

Como o lider de projeto entendeu

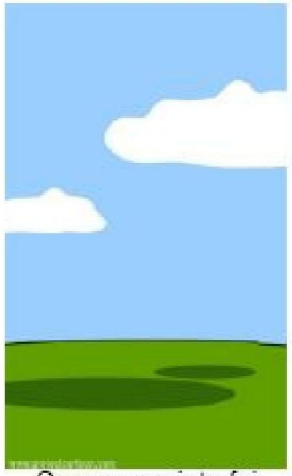

Como o analista planejou


Como o programador codificou

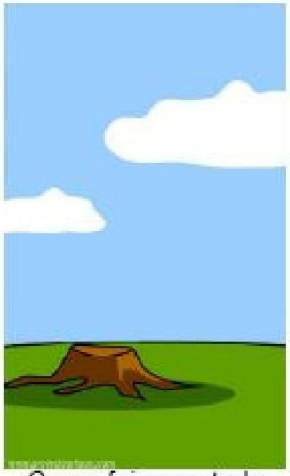

O que os beta testers receberam


Como o consultor de negocios descreveu


Valor que o cliente pagou


Como o projeto foi documentado


O que a assistencia tecnica instalou


Como foi suportado


Quando foi entregue


O que o cliente realmente necessitava

# Engenharia de software

- A Engenharia de Software é uma disciplina da Engenharia que se preocupa com todos os aspectos da produção de software.
- **Produtos de software** consistem no desenvolvimento de **programas** e **documentação** associada.
- **Manutenção**, **independência**, **validação** e **evolução** são os atributos essenciais.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

**Uso de métodos, ferramentas e processos de desenvolvimento de software.**

# Engenharia de software

- A Engenharia de Software é uma disciplina que reúne:
  - **Métodos:** prescrevem os detalhes sobre como fazer para construir o software.
  - **Ferramentas:** dão suporte automatizado aos métodos.
  - **Processos:** constituem o elo de ligação entre os métodos e ferramentas.
- Os métodos visam resolver problemas inerentes:
  - Ao processo.
  - Ao produto.

# Engenharia de software

- O processo de software consiste de atividades que são envolvidas no desenvolvimento de produtos de software. As atividades básicas deste desenvolvimento são:
  - especificação,
  - desenvolvimento,
  - validação,
  - evolução.

- **Em que consistem cada uma das atividades?**
  - Especificação: define o que o sistema deverá fazer,considerando as suas restrições.

  O processo da engenharia de requisitos leva a produção de um **DOCUMENTO DE REQUISITOS**.

  Os requisitos são geralmente apresentados em 2 níveis de detalhe:

  - Usuários finais e clientes: declaração de requisitos de alto nível.
  - Projetistas : uma especificação mais detalhada.

# Engenharia de software

- **Em que consistem cada uma das atividades?**
  - Desenvolvimento: produção do software.

  Envolve os processos de projeto e programação de software

- **Em que consistem cada uma das atividades?**
  - Validação: checagem se o software faz o que o usuário deseja.

Mostra que o sistema está em conformidade com sua especificação e que atende às expectativas do cliente que esta adquirindo o sistema.

Teste de Componente
Teste de Sistema
Teste de Aceitação

} Processo de teste

- **Em que consistem cada uma das atividades?**
  - Evolução: mudanças no software para atender às novas demandas.

Após a decisão de aquisição do hardware, mudanças podem ser feitas no software ao longo de sua vida, em resposta às mudança de requisitos e às necessidades do cliente.

# Engenharia de software

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Campinas

- Os métodos são formas organizadas de produzir software.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Quais são os métodos da Engenharia de Software?

São abordagens estruturadas para o desenvolvimento de software que incluem:

- modelos de software,
- notações,
- regras,
- maneiras de desenvolvimento.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Modelos de processo

- **Especificam as atividades** e **a ordem** em que, **de acordo com o modelo**, devem ser executadas.
- Produtos de software podem ser construídos utilizando-se de diferentes modelos de processo.
- Alguns modelos são mais adequados que outros para determinados tipos de aplicação.
- A opção por um determinado modelo deve ser feita levando-se em consideração o produto a ser desenvolvido.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# O que é um modelo de processo de software?

- É uma **representação simplificada** de um processo de software, apresentada sobre uma perspectiva específica.

# Engenharia de software

Os principais modelos de processo de software são:

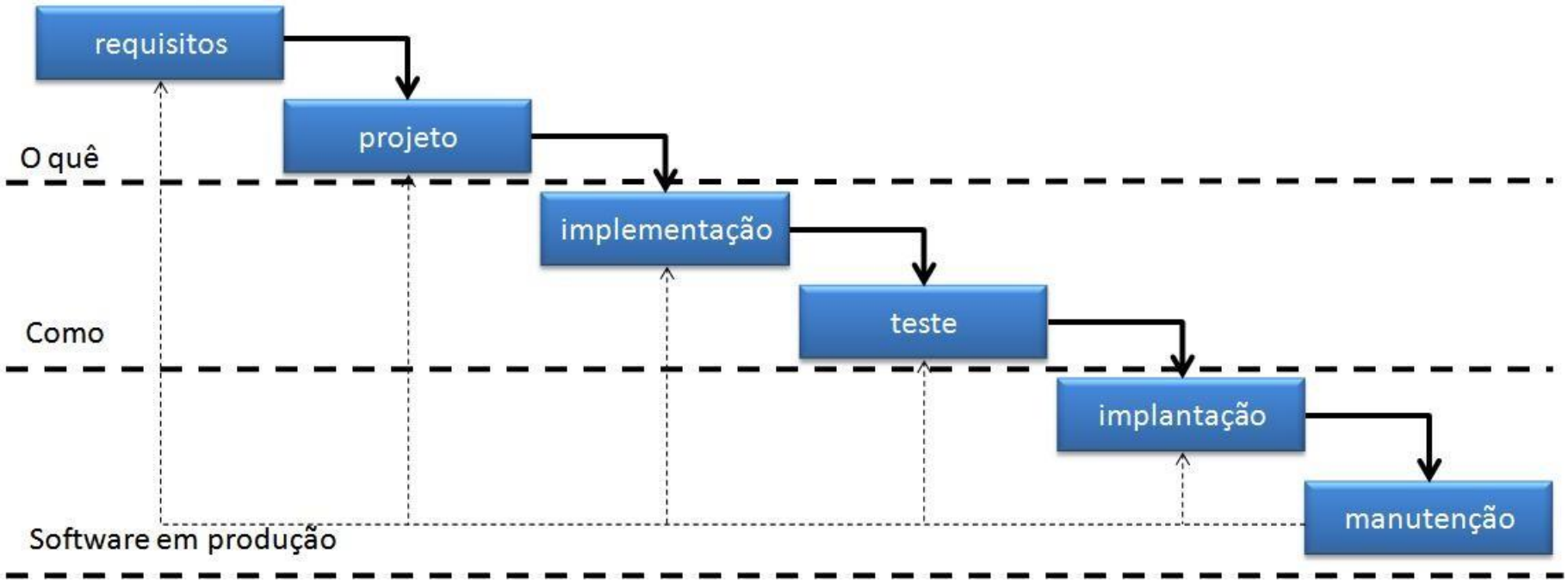
- Modelo sequencial Linear
  - Ciclo de vida clássico (Modelo Cascata).
- Modelo evolutivo
  - Modelo Incremental.
  - Modelo espiral.
  - Modelo baseado em componentes (reuso)
- Prototipação
- Modelo de Desenvolvimento formal
  - Uso de modelo matemático é formalmente transformado em uma implementação.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

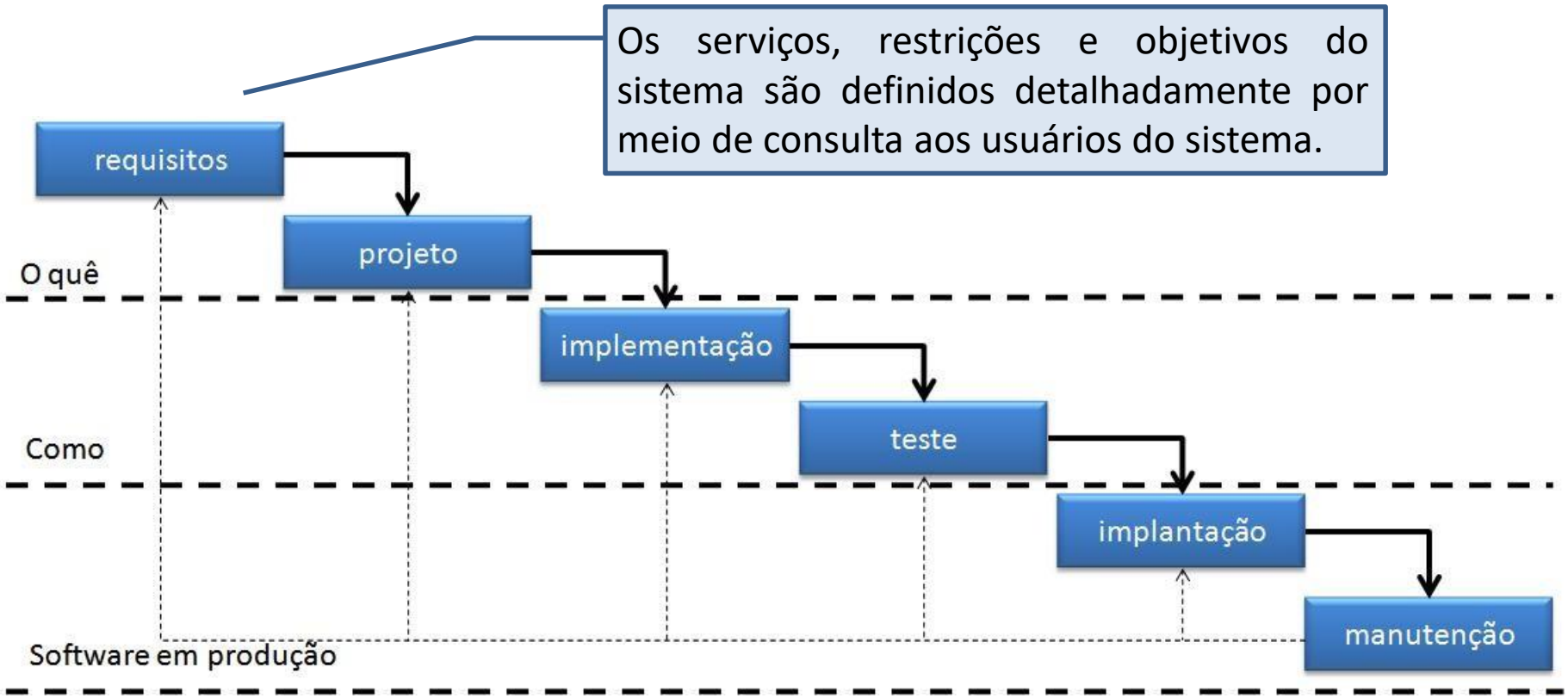
- **Objetivos dos modelos**
- Auxiliar no processo de produção **=>** produtos de alta qualidade, produzidos mais rapidamente e a um custo cada vez menor.
- Atributos: complexidade, visibilidade, aceitabilidade, confiabilidade, manutenção, segurança etc.
- Possibilitam:
  - Ao **gerente**: controlar o processo de desenvolvimento de sistemas de software.
  - Ao **desenvolvedor**: obter a base para produzir, de maneira eficiente, software que satisfaça os requisitos pré-estabelecidos.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

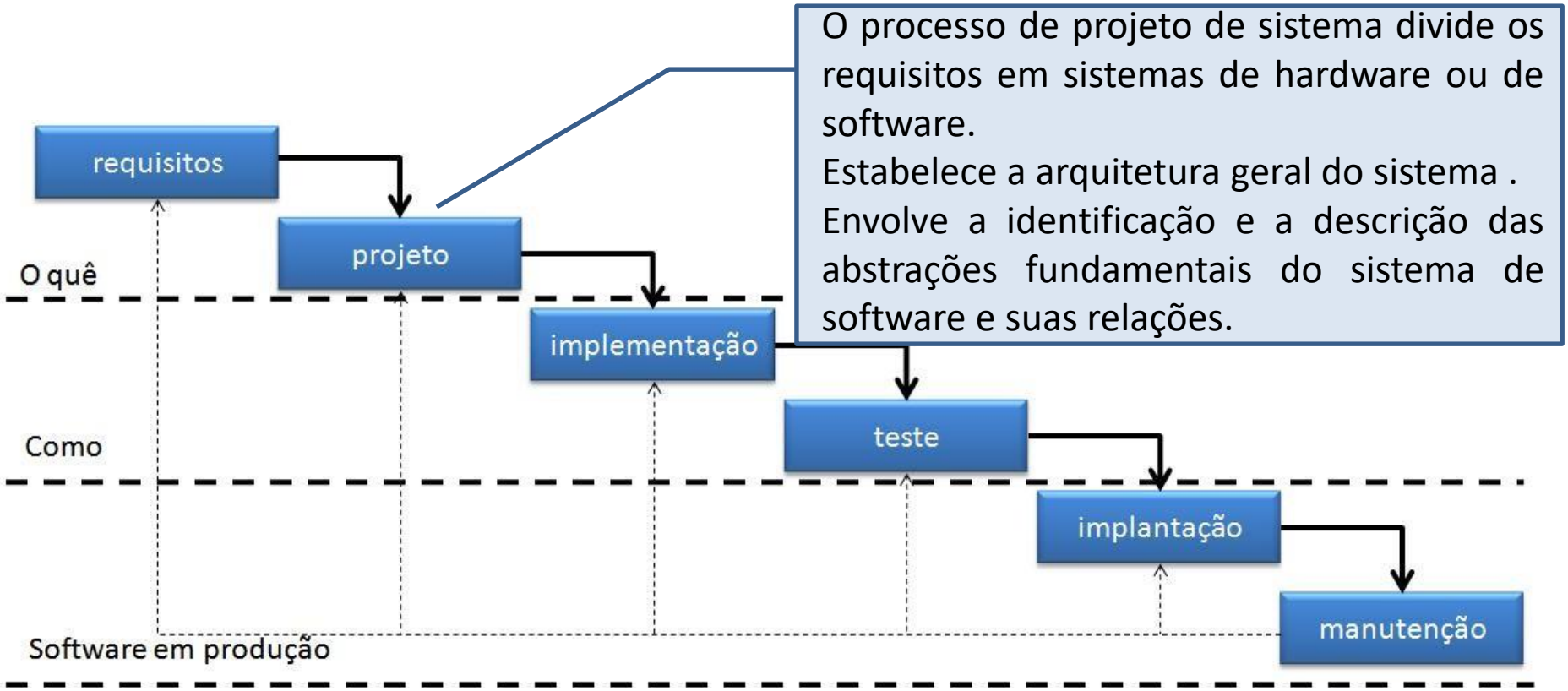
# Modelo Cascata ou sequencial



# Modelo Cascata ou sequencial

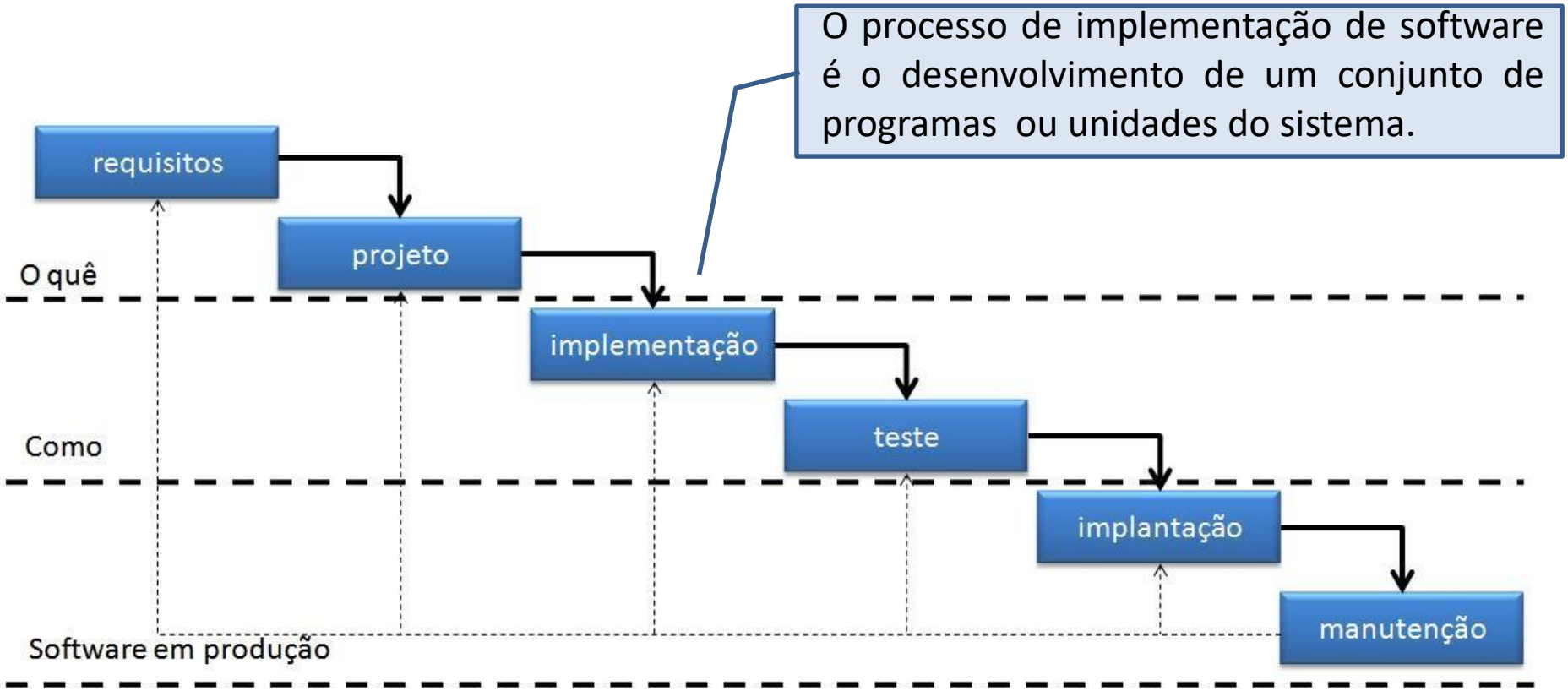


# Modelo Cascata ou sequencial



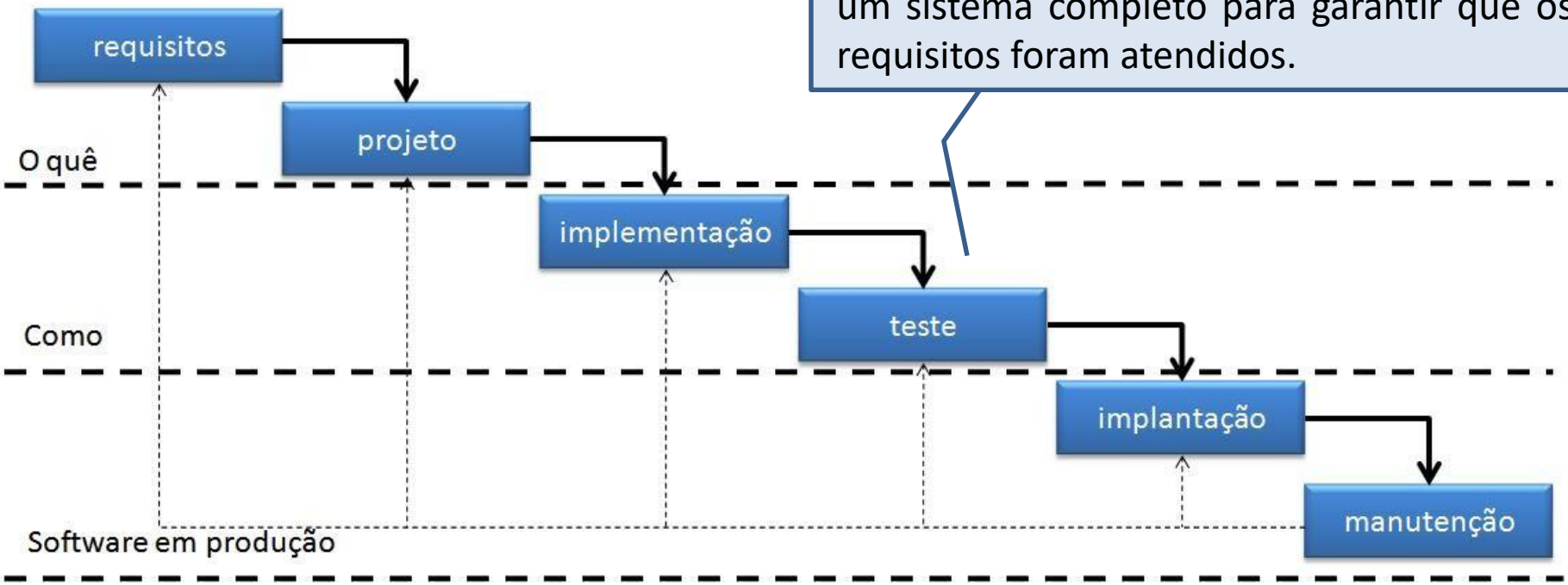

O processo de projeto de sistema divide os requisitos em sistemas de hardware ou de software.
Estabelece a arquitetura geral do sistema .
Envolve a identificação e a descrição das abstrações fundamentais do sistema de software e suas relações.

# Modelo Cascata ou sequencial

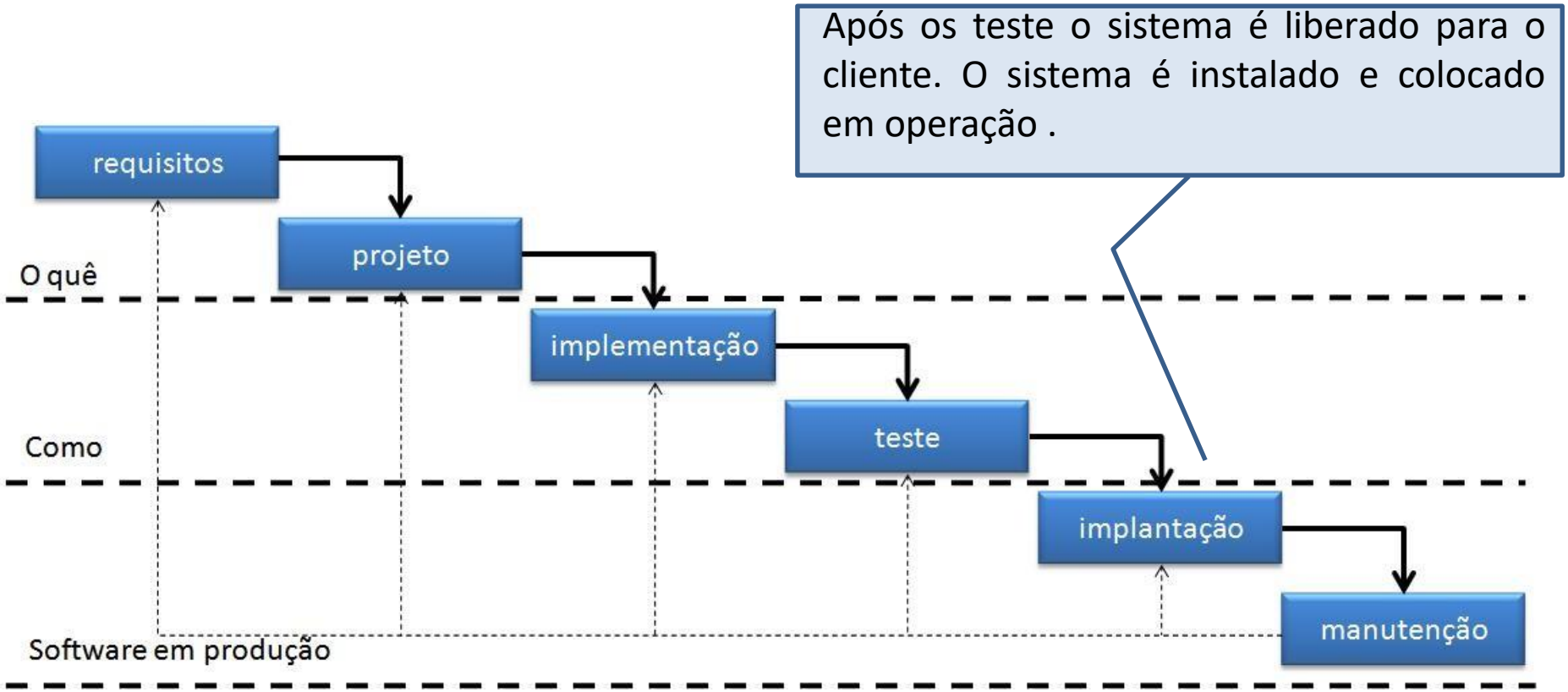


# Modelo Cascata ou sequencial

As unidades individuais de programas ou os programas são integrados e testados como um sistema completo para garantir que os requisitos foram atendidos.

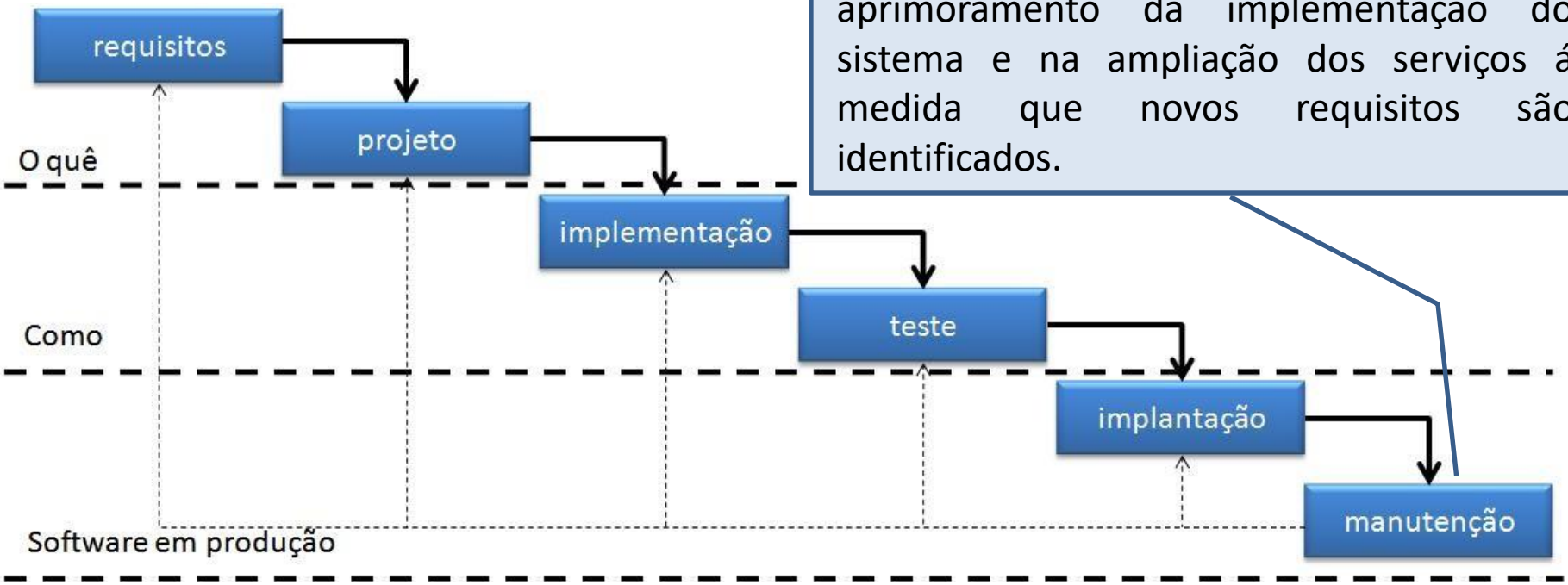


# Modelo Cascata ou sequencial



# Modelo Cascata ou sequencial

Envolve a correção de erros não detectados nos estágios anteriores do ciclo de vida no aprimoramento da implementação do sistema e na ampliação dos serviços á medida que novos requisitos são identificados.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo Cascata ou sequencial

- Método sistemático e sequencial.
- O resultado de uma fase constitui na entrada de outra.
- Cada fase é estruturada como um conjunto de atividades que podem ser executadas por pessoas diferentes, simultaneamente.

# Engenharia de software

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

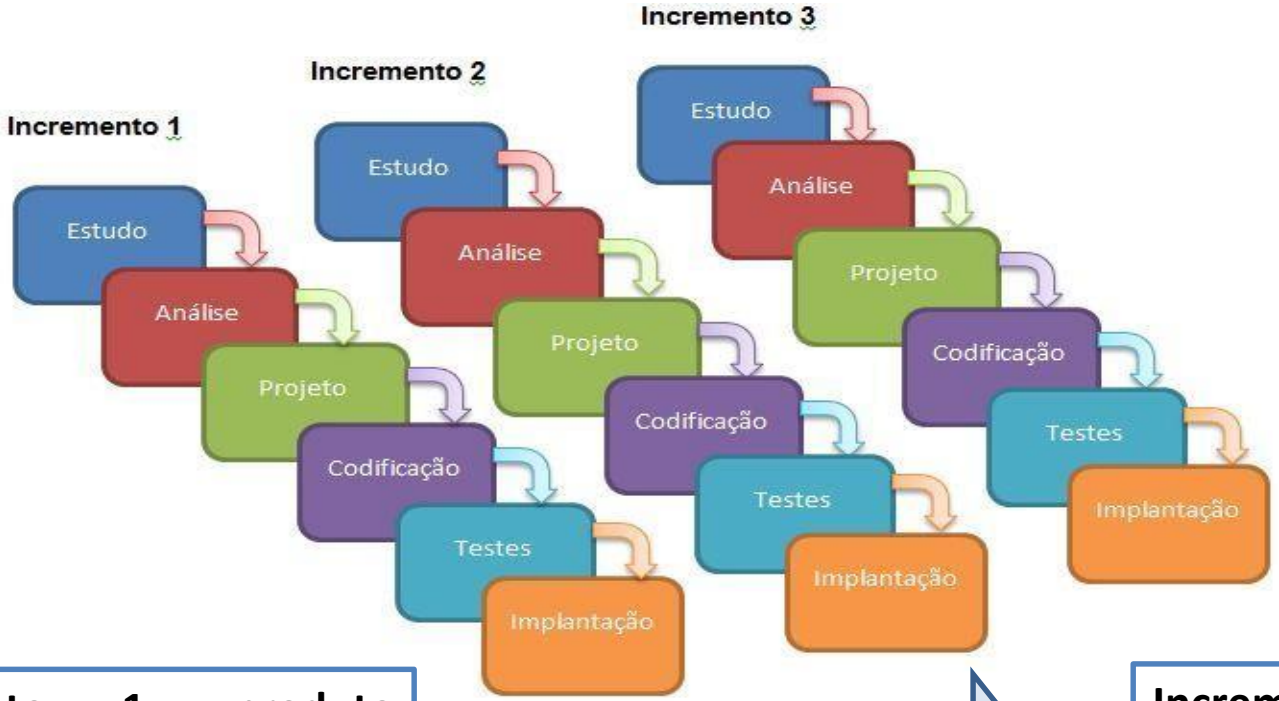
- **Problemas** :
  - O Modelo e sequencial, onde **a entrada de uma fase é o resultado da anterior.**
  - Modelo mais adequado quando os **requisitos estão muito bem entendidos.**
  - O **reinício do modelo** é a dificuldade de acomodar mudanças depois que o processo está no final.
  - **Dificuldade em atender às mudanças** exigidas posteriormente pelo cliente.
  - **Alto custo** de manutenção e correção de erros.
  - **Retrabalho constante.**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo evolutivo: Incremental

- Os modelos evolutivos são **caracterizados por iterações**, o que possibilita o desenvolvimento de várias versões do produto.
- Estas versões são colocadas em produção e os **requisitos sofrem constantes refinamentos.** É importante salientar também que as versões, quando implantadas, podem gerar novos requisitos.
- **O modelo incremental é classificado como um modelo evolutivo** dentro da engenharia de software.
- Ele é baseado no modelo cascata e diversas iterações, ou seja, várias “cascatinhas” são implementadas durante o desenvolvimento do produto – uma cada versão.

# Modelo evolutivo: Incremental

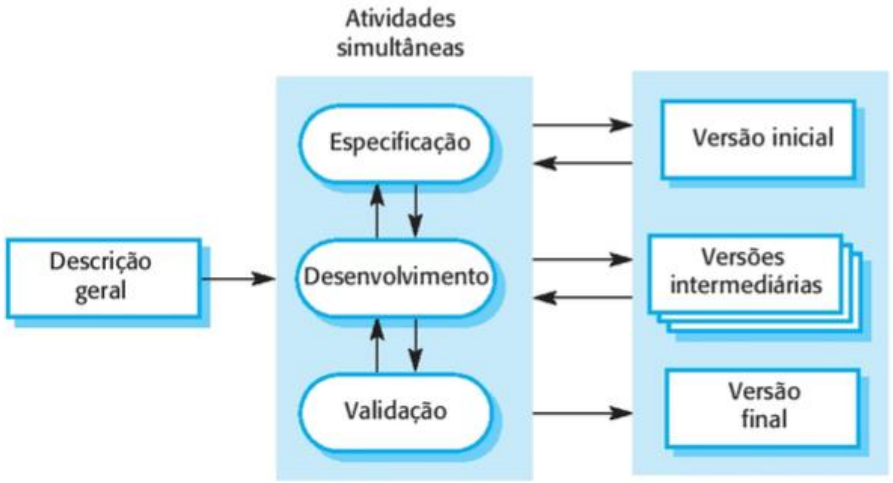



**Incremento 1: produto essencial, requisitos básicos são atendidos, porém alguns recursos complementares (conhecidos ou não) ainda não são entregues.**

Como resultado do uso e/ou avaliação e desenvolvido um planejamento para o incremento seguinte

**Incremento 2: produto essencial com recursos e funcionalidades adicionais , versão 2 do produto final**

# Modelo evolutivo: Incremental




**O modelo evolutivo incremental tem seu foco voltado para a entrega de um produto operacional em cada incremento.**
**É útil nos casos em que não há pessoal disponível para uma completa implementação na época de vencimento do prazo estabelecido para o projeto.**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo evolutivo: Incremental

O desenvolvimento incremental, em algumas de suas formas, é atualmente a **abordagem mais comum** para o desenvolvimento de aplicações e produtos de software.

Essa abordagem pode ser **dirigida por plano ao ágil:** na maioria das vezes, uma mistura de ambas.

Em uma abordagem de plano, os **incrementos do sistema são identificados antecipadamente.**
Se for adotada uma abordagem ágil, **os incrementos são identificados**, mas o desenvolvimento dos incrementos finais depende do progresso e das prioridades do cliente.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo evolutivo: Incremental

O desenvolvimento incremental tem três grandes vantagens em relação ao modelo em cascata:

1. O custo de implementação das mudanças nos requisitos é reduzido. A quantidade de análise e documentação que precisa ser refeita é significativamente menor do que a necessária ao modelo em cascata.
2. É mais fácil obter *feedback* do cliente sobre o trabalho de desenvolvimento. Os clientes podem comentar as demonstrações de software e ver o quanto foi implementado. Para eles, é mais difícil julgar o progresso a partir dos documentos do projeto (*design*) de software.
3. A entrega e a implantação antecipadas de um software útil para o cliente são possíveis, mesmo se toda a funcionalidade não tiver sido incluída. Os clientes são capazes de usar o software e de obter valor a partir dele mais cedo do que com um processo em cascata.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo evolutivo: Espiral

- Esse é um modelo de processo de **software evolucionário** que também é **iterativo** como a prototipação, porém com **aspectos sistemáticos** e controlados do modelo cascata.
- O modelo espiral fornece um grande potencial para que possamos ter rápido desenvolvimento de versão cada vez mais completas.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo evolutivo: Espiral

- Consiste em uma serie de ciclos que se repetem. Cada um tem as mesmas fases e quando termina entrega um produto ampliado com respeito ao ciclo anterior.
- Neste sentido é semelhante ao Modelo Incremental, com a diferencia que adiciona um novo elemento
  - **a análise de risco**.

# Risco

- Um **risco significa algo que pode dar errado**. É a ocorrência de um evento que possa comprometer o andamento do projeto Por exemplo:
  - Requisitos no compreendidos,
  - Mal desenho do projeto,
  - Erro na implementação,
  - Indisponibilidade da infraestrutura - Infra-estrutura de desenvolvimento (espaço físico, ferramentas de software e hardware) não disponível nos momentos delimitados no plano do projeto,
  - Dificuldade de comunicação devido a distribuição geográfica da equipe,
  - Indisponibilidade dos usuários das áreas de negócio para levantamento de informações durante a requisitos.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Risco

- Um risco pode causar problemas no projeto, tal como: ultrapassar o cronograma e os custos; por isso, a **minimização dos riscos** é uma atividade de gerenciamento de projeto **muito importante**.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Tipos de Risco

**Riscos de projeto**

– Cronograma
– Pessoal
– Orçamento

**Riscos técnicos**

– Análise, design, implementação e testes
– Ferramentas de hardware e software

**Riscos de negócios**

– Mudanças no Mercado, novas estratégias
– Requisitos e Restrições organizacionais

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Análise de Risco

Análise de Riscos é o processo de identificar, analisar e responder a estes eventos.

- **Identificação dos Riscos**
- **Estimativas**

– Probabilidade de ocorrência

– Impacto sobre o projeto – Prioridade (probabilidade x impacto)

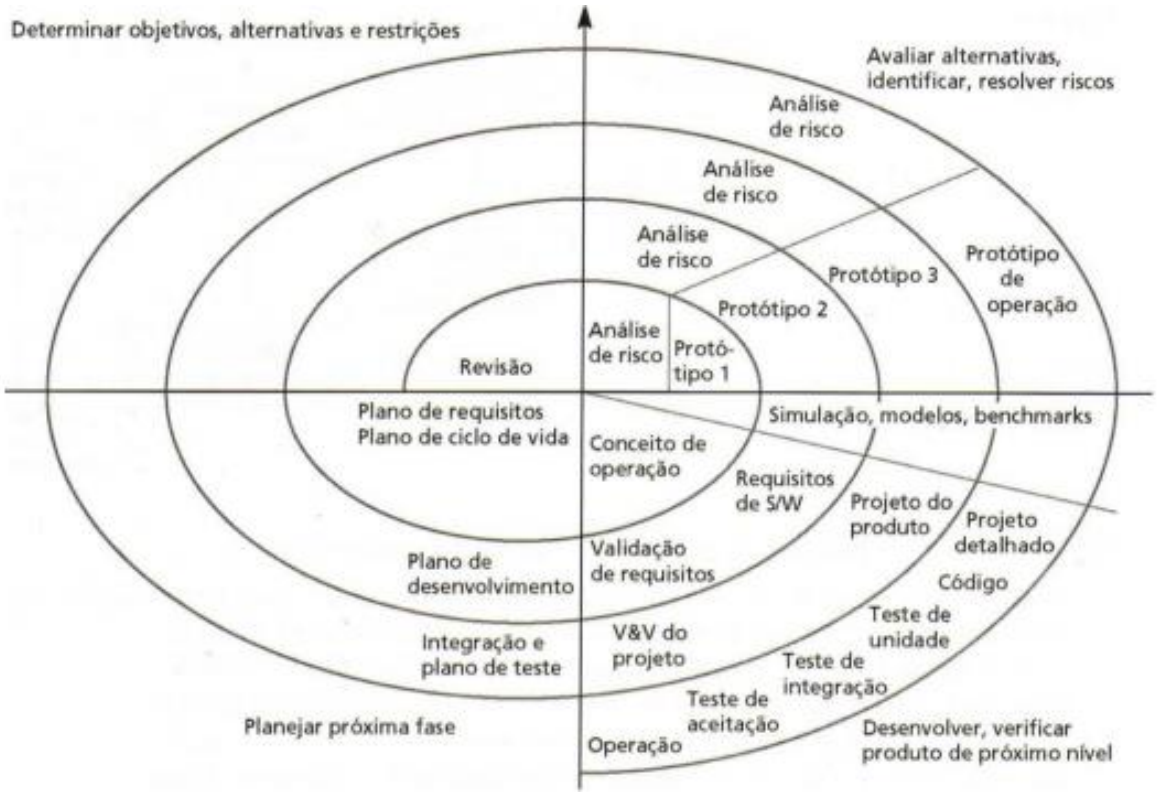
- **Gerenciamento**

– Ações para redução dos riscos

– Ações, em caso da ocorrência, para minimizar o impacto

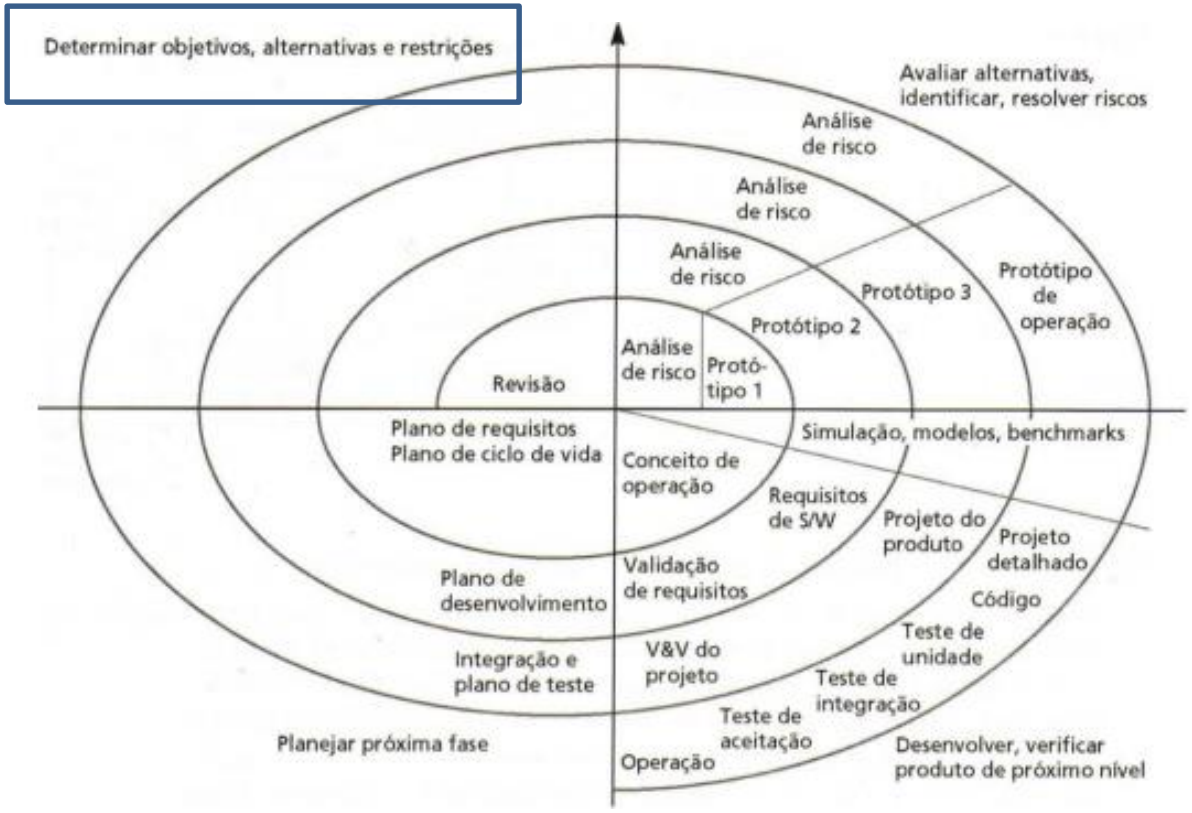
- **Revisões**

# Modelo evolutivo: Espiral




- Cada ciclo na espiral representa uma fase do processo de software.
- O **ciclo mais interno** está concentrado nas **possibilidades do sistema**.
- O **próximo ciclo** está concentrado na **definição dos requisitos do sistema**.
- O **ciclo um pouco mais externo** está concentrado no **projeto do sistema**.
- Um **ciclo ainda mais externo** está concentrado na **construção do sistema**

# Modelo evolutivo: Espiral



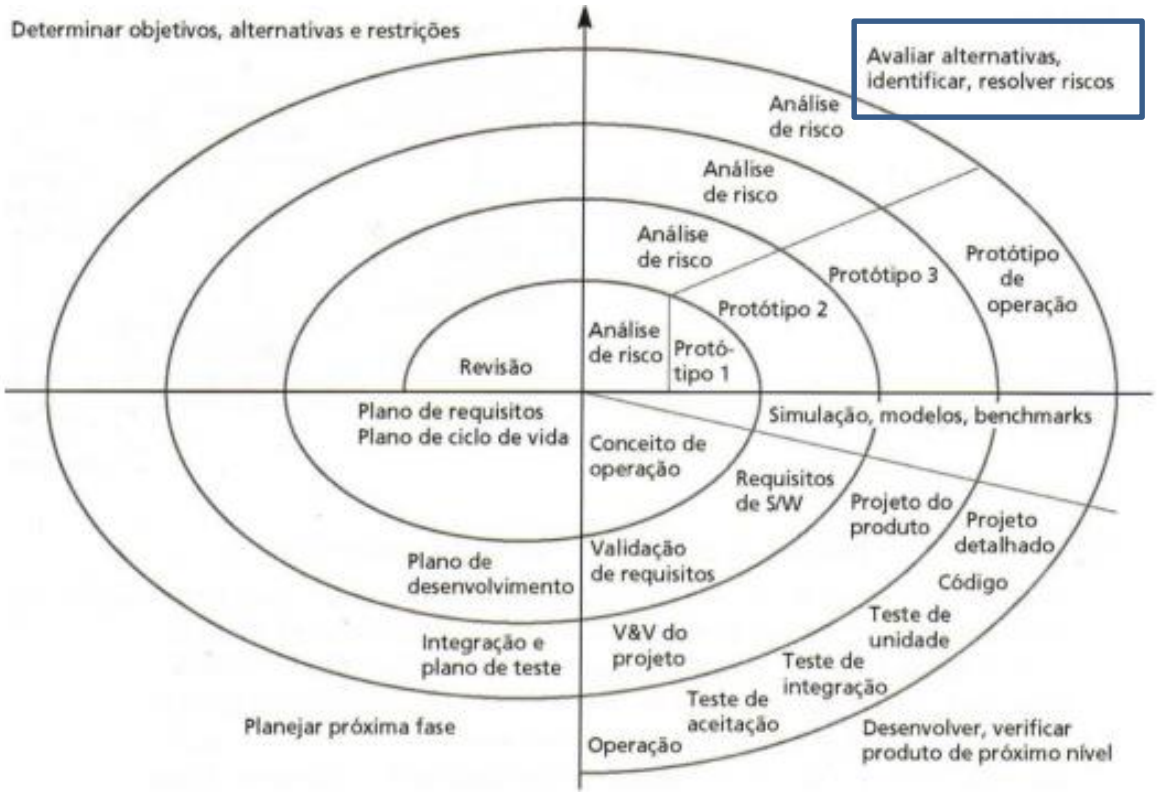

- Cada "loop" do espiral é dividido em 4 setores

**ESTABELECIMENTO DE OBJETIVOS**

São definidos objetivos específicos para a fase do projeto e identificadas restrições sobre o processo e o produto .

É projetado um plano de gerenciamento detalhado. São identificados riscos do projeto, dependendo dos riscos, estratégias alternativas podem ser planejadas.

# Modelo evolutivo: Espiral

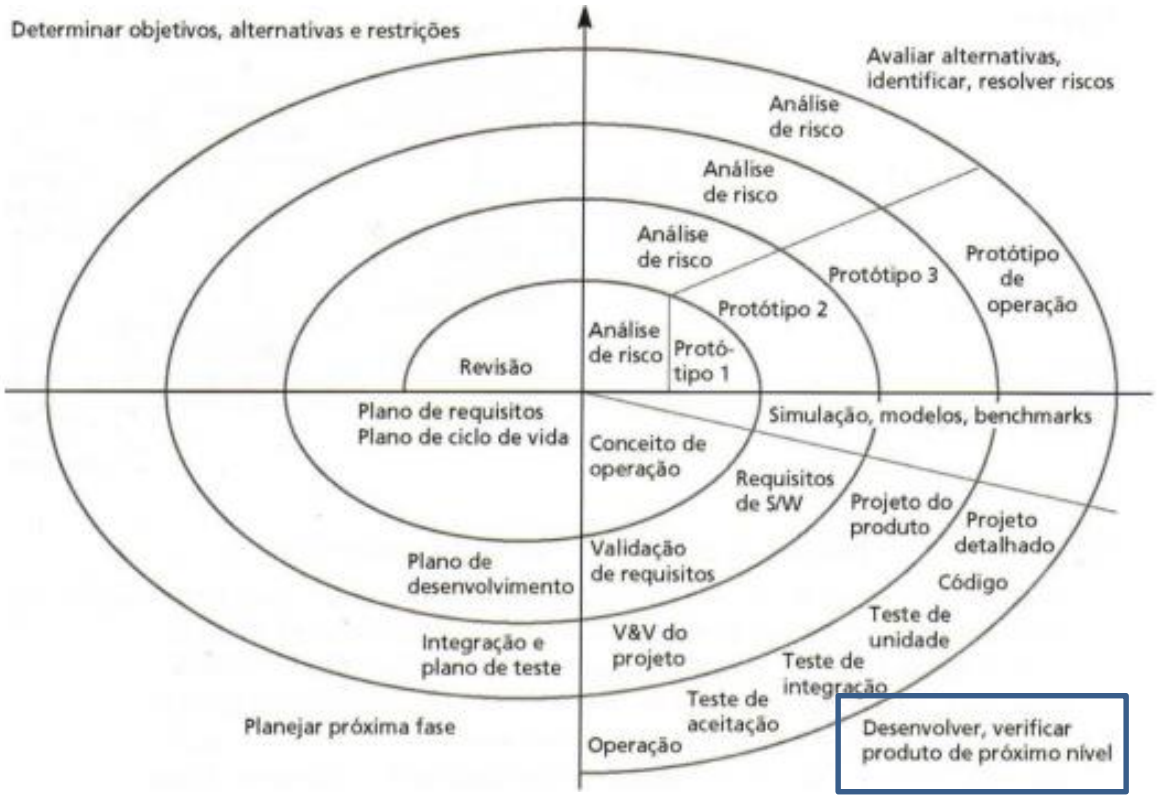



- Cada "loop" do espiral é dividido em 4 setores

**AVALIAÇÃO E REDUÇÃO DE RISCOS**

Para cada um dos riscos identificados, uma análise detalhada é executada. Passos são tomados para reduzir o risco

# Modelo evolutivo: Espiral

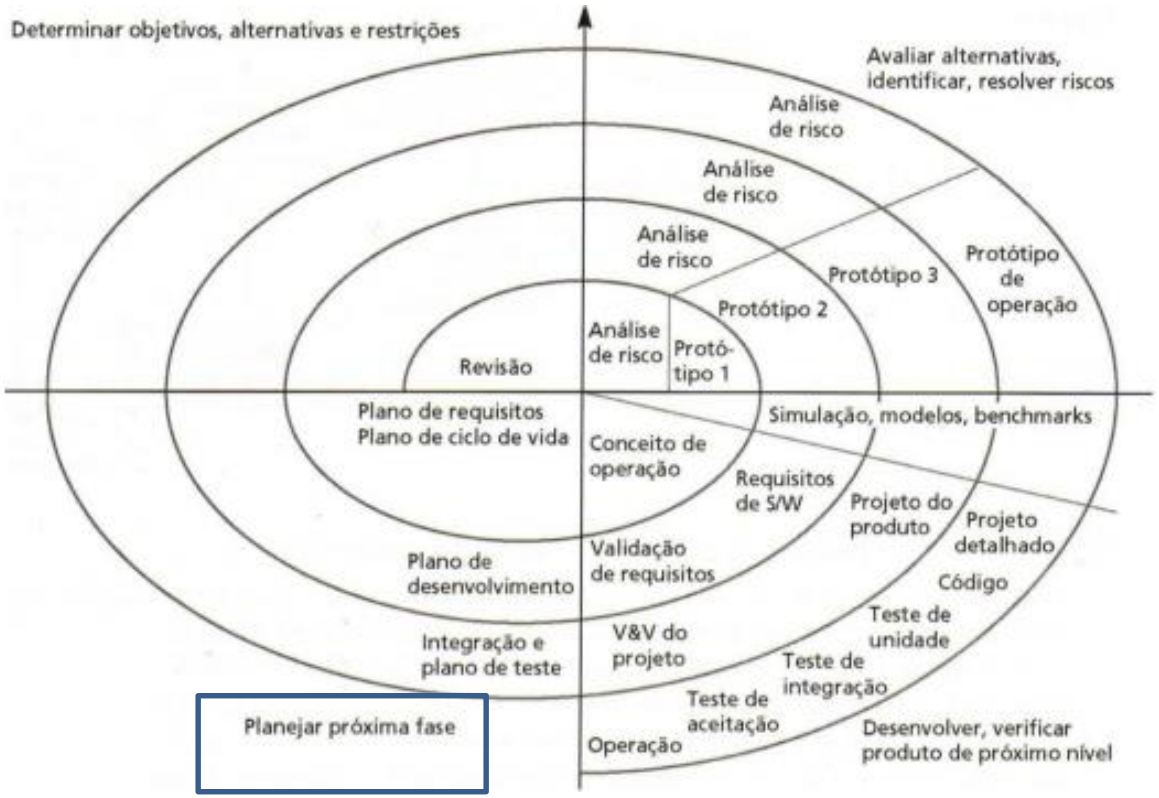



- Cada "loop" do espiral é dividido em 4 setores

**DESENVOLVIMENTO E VALIDAÇÃO**

Depois da avaliação do risco, um modelo de desenvolvimento é escolhido para o sistema

# Modelo evolutivo: Espiral




- Cada "loop" do espiral é dividido em 4 setores

**PLANEJAMENTO**

O projeto é revisto e é tomada uma decisão de continuidade se é decidido continuar.

São projetados planos para a próxima fase do projeto (próximo "loop" )

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

# Modelo evolutivo: Espiral

- Ao terminar uma iteração se comprova que o desenvolvido **cumpre com os requisitos estabelecidos**, e **se verifica se funciona corretamente**.
- O próprio cliente avalia o produto.

## Donde é adequado utilizar este modelo?

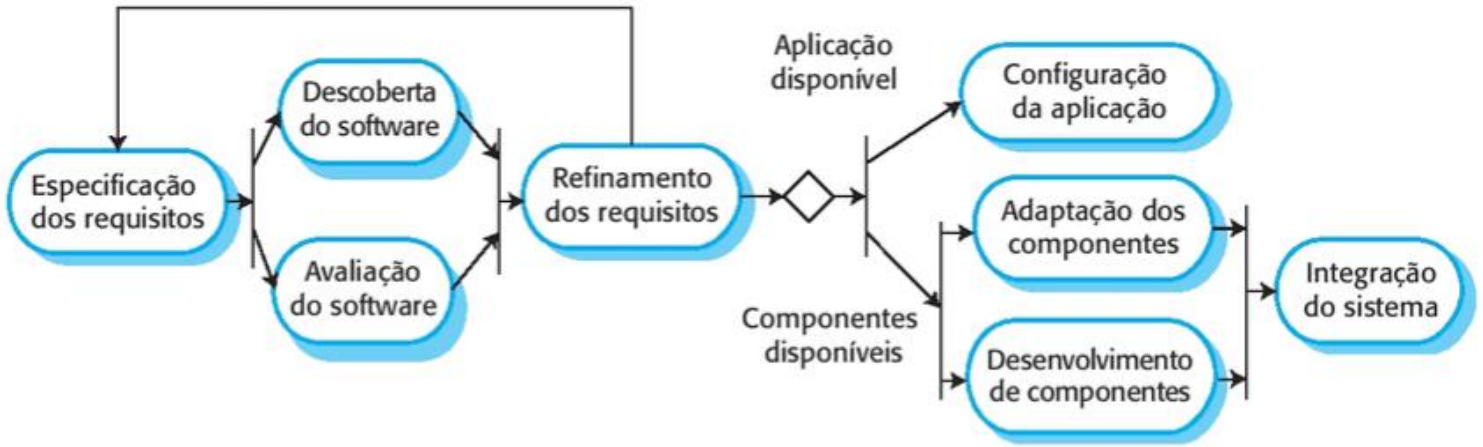
- Sistemas de grandes dimensões
- Projetos onde seja importante o fator de risco
- Quando não seja possível definir ao principio todos os requisitos.

- **Modelo evolutivo: baseado em componentes (reuso)**
  - Os sistemas são baseado em **componentes já existentes**, semelhantes ao desenvolvimento de hardware.
  - Utiliza tecnologias orientadas a objeto, quando projetadas e implementadas apropriadamente as classes orientadas a objeto são reutilizáveis em diferentes aplicações e arquiteturas de sistema.
  - O modelo baseado em componentes incorpora muitas das características do modelo espiral.
  - Modelo que vem crescendo bastante nos últimos tempos.

- **Modelo evolutivo: baseado em componentes (reuso)**
- Fases do processo:
  - Especificação de requisitos.
  - Análise do componente.
  - Modificação dos requisitos.
  - Projeto do sistema com reuso.
  - Desenvolvimento e integração.
  - Validação de sistema.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Modelo evolutivo: baseado em componentes (reuso)




A engenharia de software baseada no reúso, articulada em torno da configuração e da integração, tem a vantagem óbvia de reduzir a quantidade de software a ser desenvolvido, diminuindo custos e riscos. Normalmente, isso também leva a uma entrega mais rápida do software. Entretanto, concessões quanto aos requisitos são inevitáveis, o que pode resultar em um sistema que não satisfaz as necessidades reais dos usuários. Além disso, parte do controle sobre a evolução do sistema se perde, já que novas versões dos componentes reusáveis não estão sob o controle da organização que os utiliza.

## Modelo evolutivo: baseado em componentes (reuso)

- A reusabilidade fornece uma série de benefícios:
  - redução de até 70% no tempo de desenvolvimento
  - redução de até 84% no custo do projeto
  - índice de produtividade de até 26.2 (normal da indústria é de 16.9)
- Esses resultados dependem da robustez da biblioteca de componentes.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

- **Modelo evolutivo:baseado em componentes (reuso)**

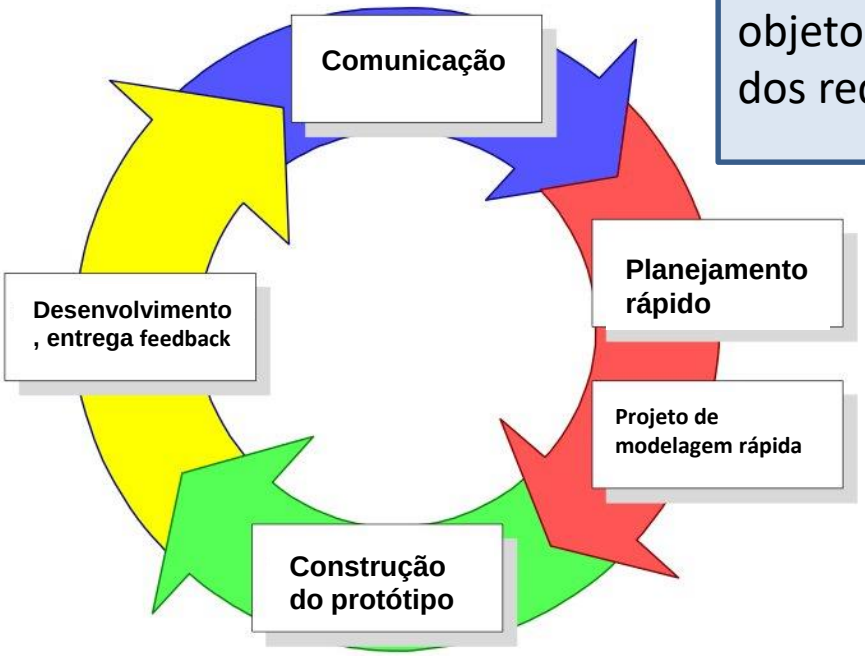
- Aspectos a analisar:
  - Sistemas legados: os sistemas antigos devem ser mantidos e atualizados.
  - Heterogeneidade: sistemas são uma combinação de hardware e software.
  - Prazos de entrega: pressão para um menor prazo de entrega.

- **Prototipação**
  - O objetivo é entender os requisitos do sistema através da construção de protótipos para avaliação e aprovação do cliente.

Um **protótipo** é uma versão inicial de um sistema de software, que é utilizada para mostrar conceitos, experimentar opções de projeto e, em geral, para conhecer mais sobre os problemas e suas possíveis soluções. O desenvolvimento rápido de um protótipo é essencial para que os custos sejam controlados e os usuários possam fazer experiências com o protótipo no início do processo de software.

# Engenharia de software

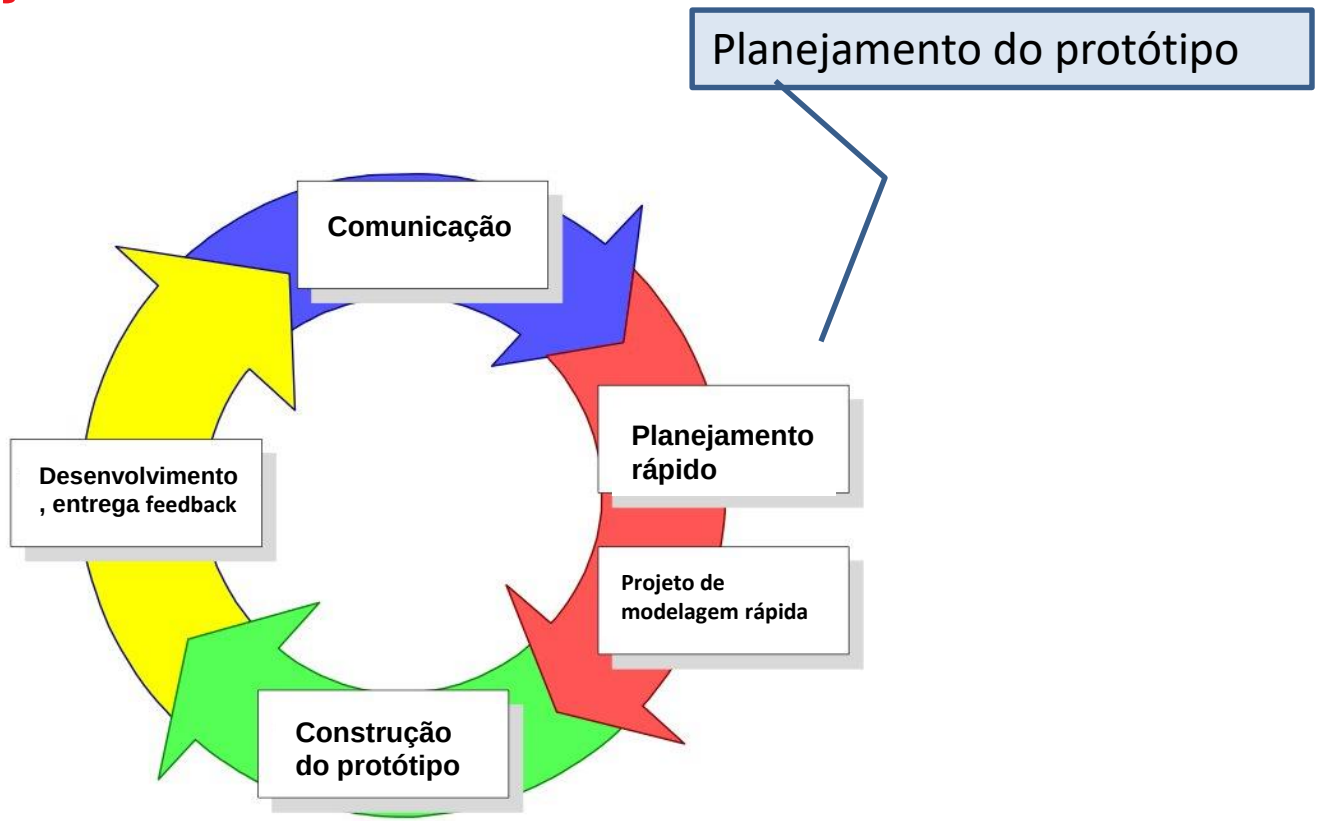
- **Prototipação**




Reuniões com as partes interessadas para definição dos objetos do projeto e levantamento dos requisitos já conhecidos

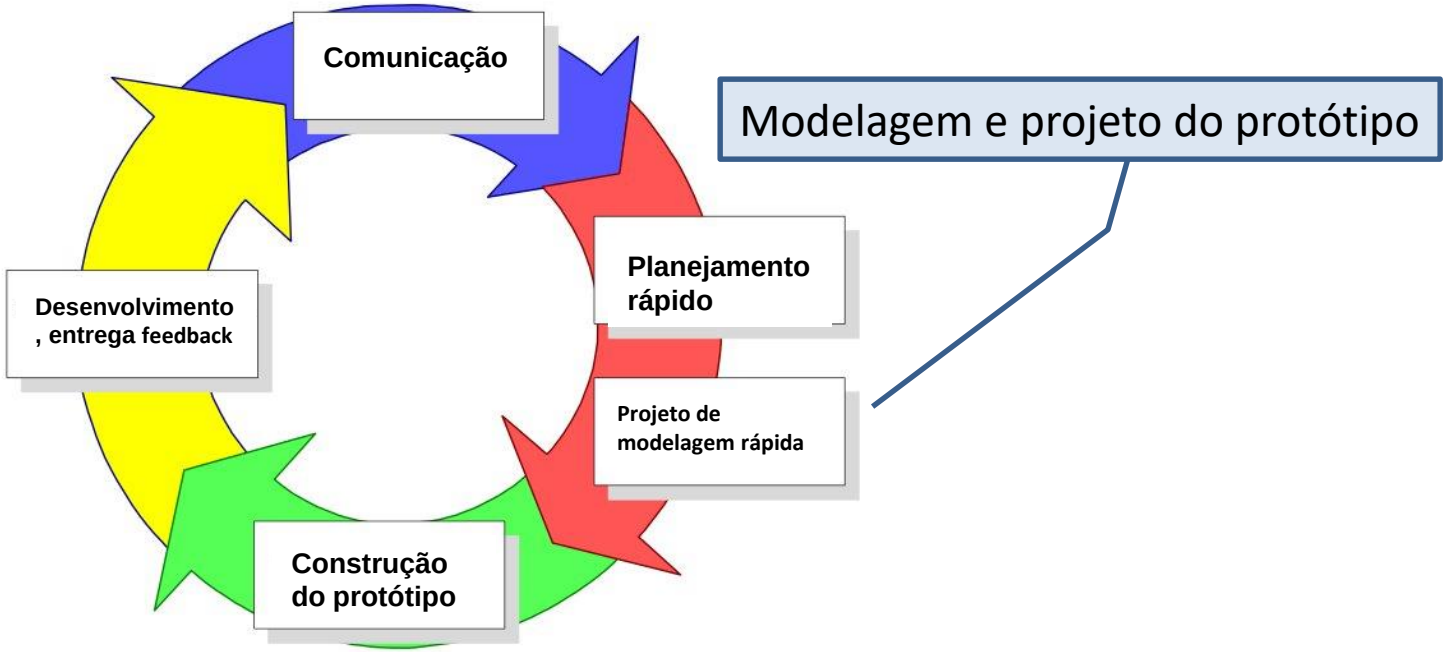
# Engenharia de software

- **Prototipação**



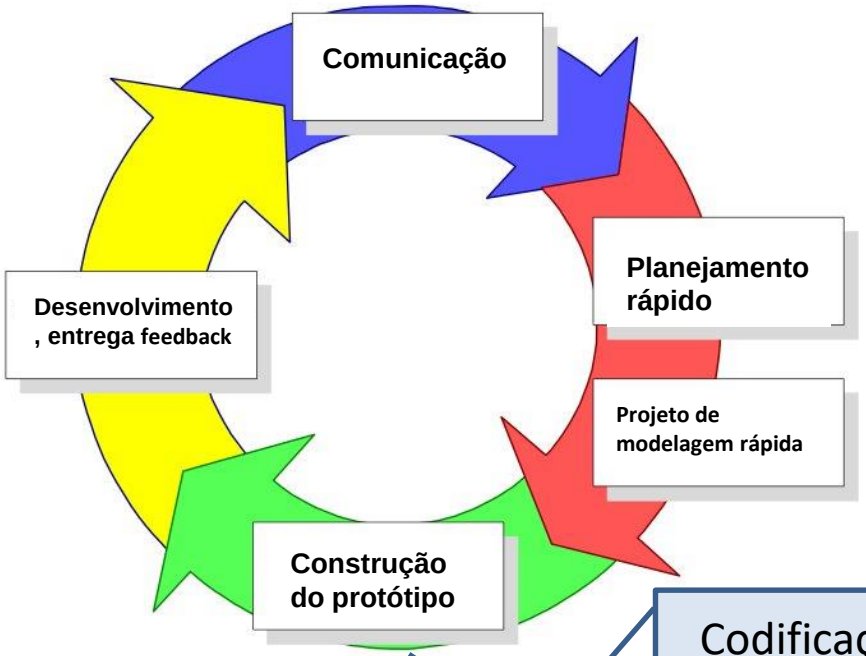
# Engenharia de software

- **Prototipação**



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

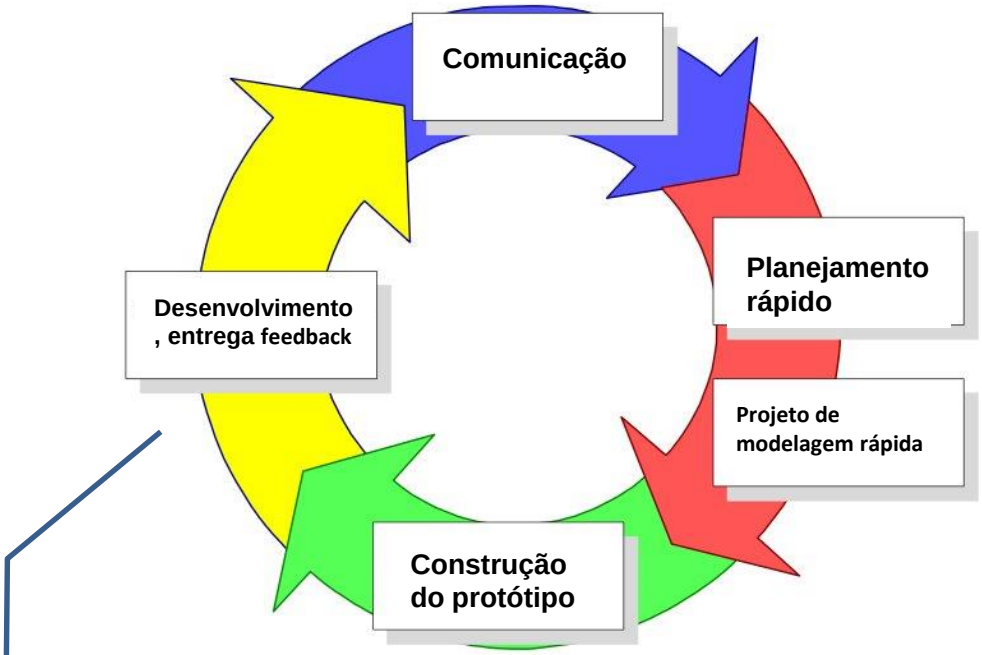
- **Prototipação**




Codificação do protótipo.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Prototipação



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Prototipação

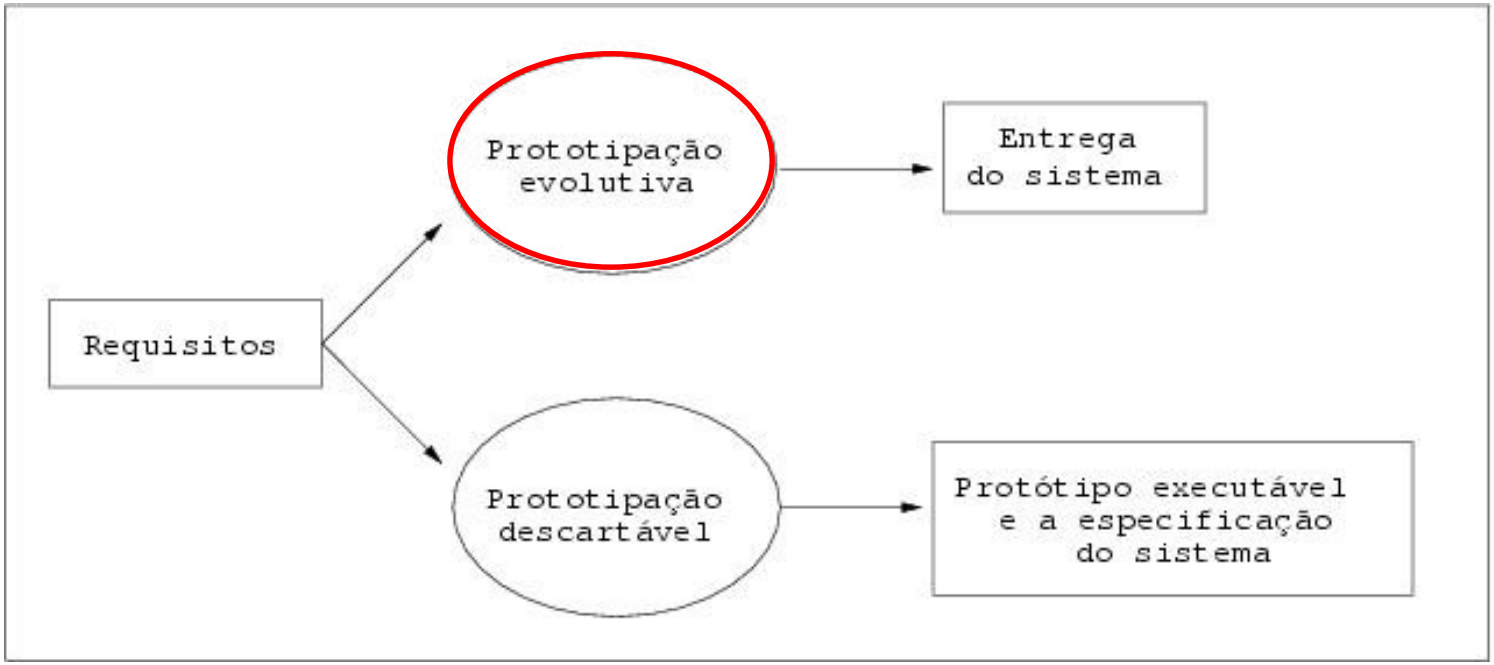
O protótipo pode ser oferecido ao cliente em diferentes formas:

- protótipo em papel;
- modelo executável em PC apresentando a interface ao cliente para compreender a forma de interagir com o software;
- protótipo de trabalho que implemente um subconjunto dos requisitos indicados;
- programa existente (pacote) que permita representar todas ou parte das funções desejadas do software a construir.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

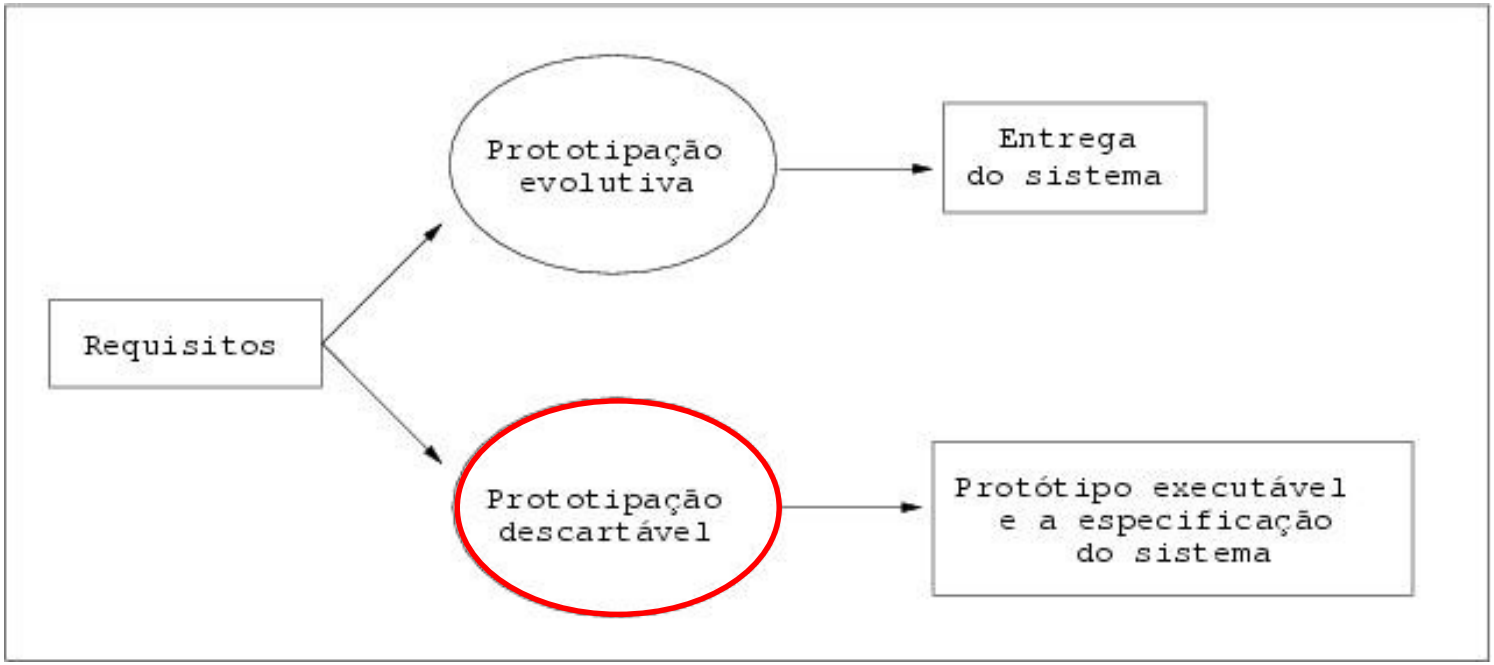
## Modelos de Prototipação

- **Prototipação evolutiva**: é produzido um protótipo inicial e refinado através de vários estágios até atingir o sistema final.



## Modelos de Prototipação

- **Prototipação descartavel**: é produzido para ajudar a levantar os problemas com os requisitos e depois descartado.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

- **Prototipação**

**Vantagens**

- Sistema atual melhora a percepção do usuário em relação software;
- O desenvolvedor constrói algo imediatamente;

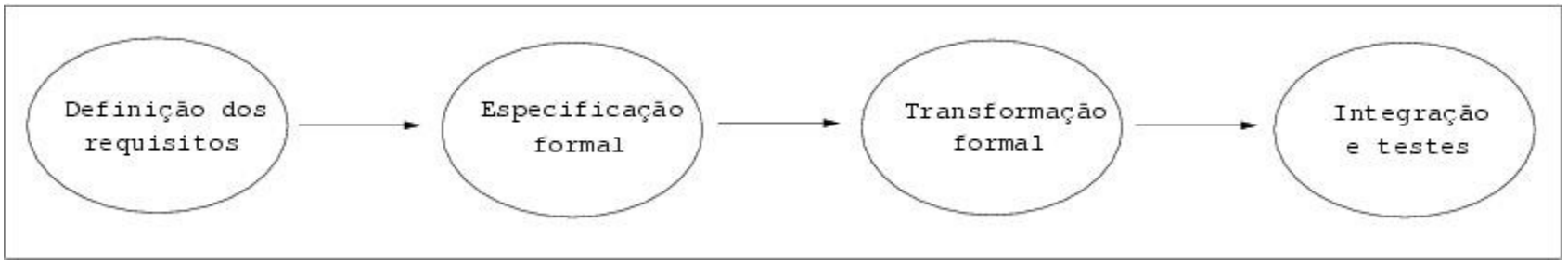
**Desvantagens** (pode levar ao descarte do protótipo)

- Pode haver muitos ajustes no protótipo final, para aumento da qualidade;
- O desenvolvedor pode esquecer estruturas inapropriadas no protótipo;

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

## Desenvolvimento Formal de Sistemas

- Uma **especificação formal** (definição matemática, não ambígua) **do software** é desenvolvida e posteriormente "transformada" em um programa através de regras que preservam a corretude da especificação
- Consegue alcançar os requisitos da especificação mais facilmente.



- **Desenvolvimento Formal de Sistemas**
- Problemas:
  - Dificuldade em encontrar profissionais especializados.
  - Dificuldade em especificar determinados aspectos como a interface do usuário.
- Aplicabilidade:
  - Principalmente para sistemas críticos, onde **não são toleradas falhas.**

# Engenharia de software

Em gestão de projetos as **metodologias tradicionais** funcionavam de maneira limitada porque:

- Não conseguiam antecipar necessidades que futuramente precisarão ser supridas;
- Perdiam muito tempo no planejamento e demoravam mais para produzir;
- Tanto a empresa quanto o cliente precisam ter uma noção muito específica do tempo de desenvolvimento;
- Qualquer alteração faz com que seja necessário um grande retrabalho e gera atrasos;
- Grande parte dos erros no projeto passa despercebida e são notados apenas pelo cliente final.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Campinas

- **Bibliografia**


- Capitulo 1, Pressman, Roger S. *Engenharia de Software. 6ª* edição. McGraw- Hill,2006.
- Capitulo 4 e 5, Sommerville, Ian. *Engenharia de Software. 8ª* edição. Pearson Education, 2007.