



# MODELOS DE DESENVOLVIMENTO DE SOFTWARE

Espiral e Prototipagem

# Processo



Conjunto de:

Atividades  
Ações  
Tarefas



Realizadas quando  
algun produto é  
necessário ser criado

# Processos

Atividades são focadas em atingir um objetivo amplo e são utilizadas independente do domínio, tamanho, complexidade do projeto.

- Por exemplo: comunicação com o stakeholder

Ações são um conjunto de tarefas que produz algo

- Normalmente é grande
- Por exemplo: Um modelo arquitetônico

Tarefas focam em objetivos pequenos mas bem definidos

- Por exemplo: escrever um teste unitário

# Modelo de processo

- Um genérico para engenharia de software define cinco atividades:
  - Comunicação
  - Planejamento
  - Modelagem
  - Construção
  - *Deployment*

# Atividades extras



Acompanhamento e controle de projeto de software

Mede o progresso e faz correções necessárias



Gestão de risco

Verifica os riscos que podem afetar o projeto



Garantia de qualidade de software

Define e conduz atividades que são requeridas para garantir a qualidade do software



Revisões técnicas

Verifica os artefatos gerados focando em descobrir erros e impedir sua propagação



Métricas

Coleta métricas para ajudar o time a entregar software



Gestão de configuração de software

Gere efeitos das mudanças durante o processo de software



Gestão de reusabilidade

Define critérios e mecanismos para reutilizar código ou componentes



Produção e preparação de artefatos

Atividades que são necessárias para criar artefatos

# Modelo genérico de processo de software

- Um modelo de processo para engenharia de software define as cinco atividades descritas anteriormente mais as atividades extras

## Software process

### Process framework

#### Umbrella activities

##### Framework activity #1

software engineering action #1.1

Task sets

work tasks  
work products  
quality assurance points  
project milestones

⋮

Software engineering action #1.k

Task sets

work tasks  
work products  
quality assurance points  
project milestones

⋮

##### Framework activity #n

software engineering action #n.1

Task sets

work tasks  
work products  
quality assurance points  
project milestones

⋮

Software engineering action #n.m

Task sets

work tasks  
work products

# Fluxo do processo



Para software, mais importante que o modelo em si, é o seu fluxo

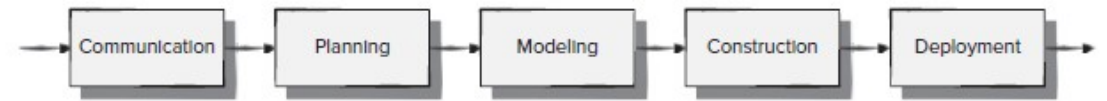


Define como as ações, atividades e tarefas ocorrem dentro do modelo

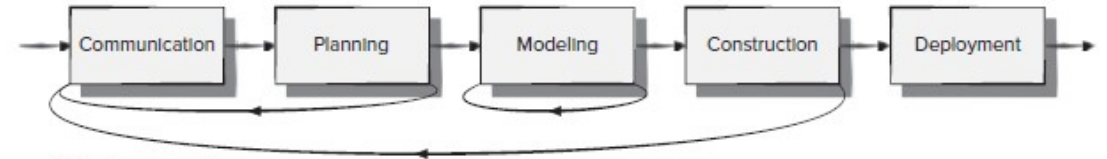


Tem como guia o tempo e a sequencia

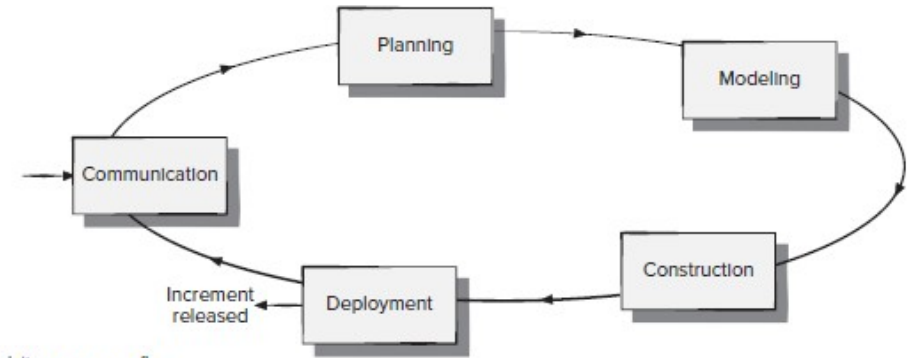
# FLUXO DE PROCESSOS



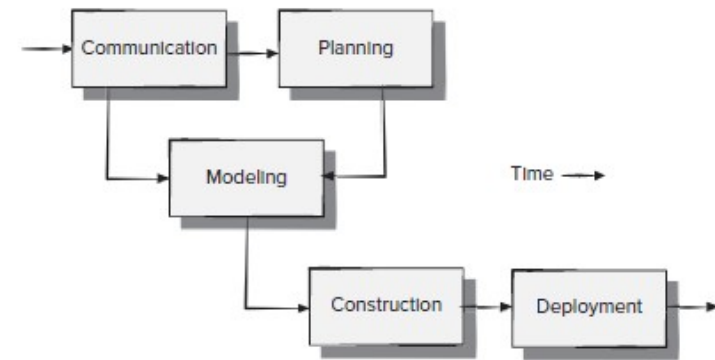
(a) Linear process flow



(b) Iterative process flow

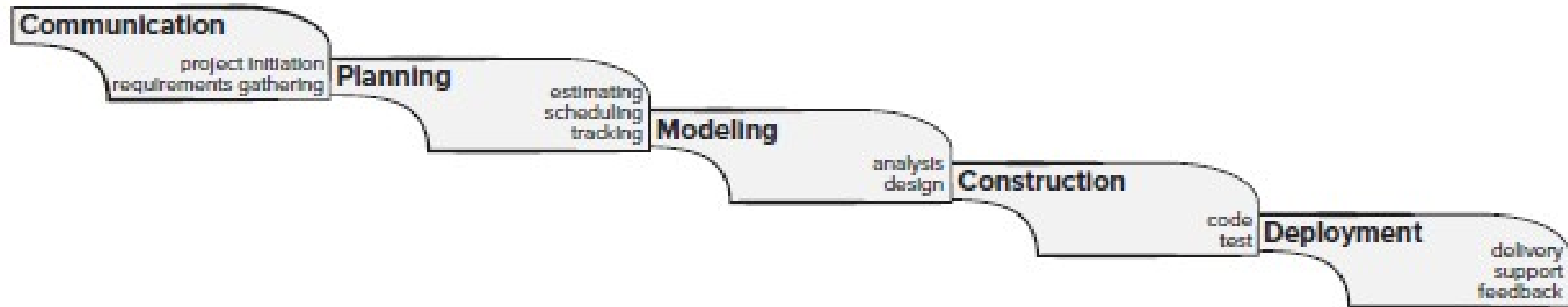


(c) Evolutionary process flow



(d) Parallel process flow





# CASCATA

# Processos Evolutivos de Software

Software evolui conforme a passagem do tempo

Regras de negócios e produtos mudam conforme o desenvolvimento continua

Tratar o desenvolvimento de software como algo linear está errado

Uma versão completa é impossível de entregar

- Uma versão “limitada” é necessária para aliviar a pressão de negócios
- Engenheiros de Software precisam de um processo que seja preparado para acomodar mudanças com o tempo

# Modelo Espiral



**Modelo evolucionário**

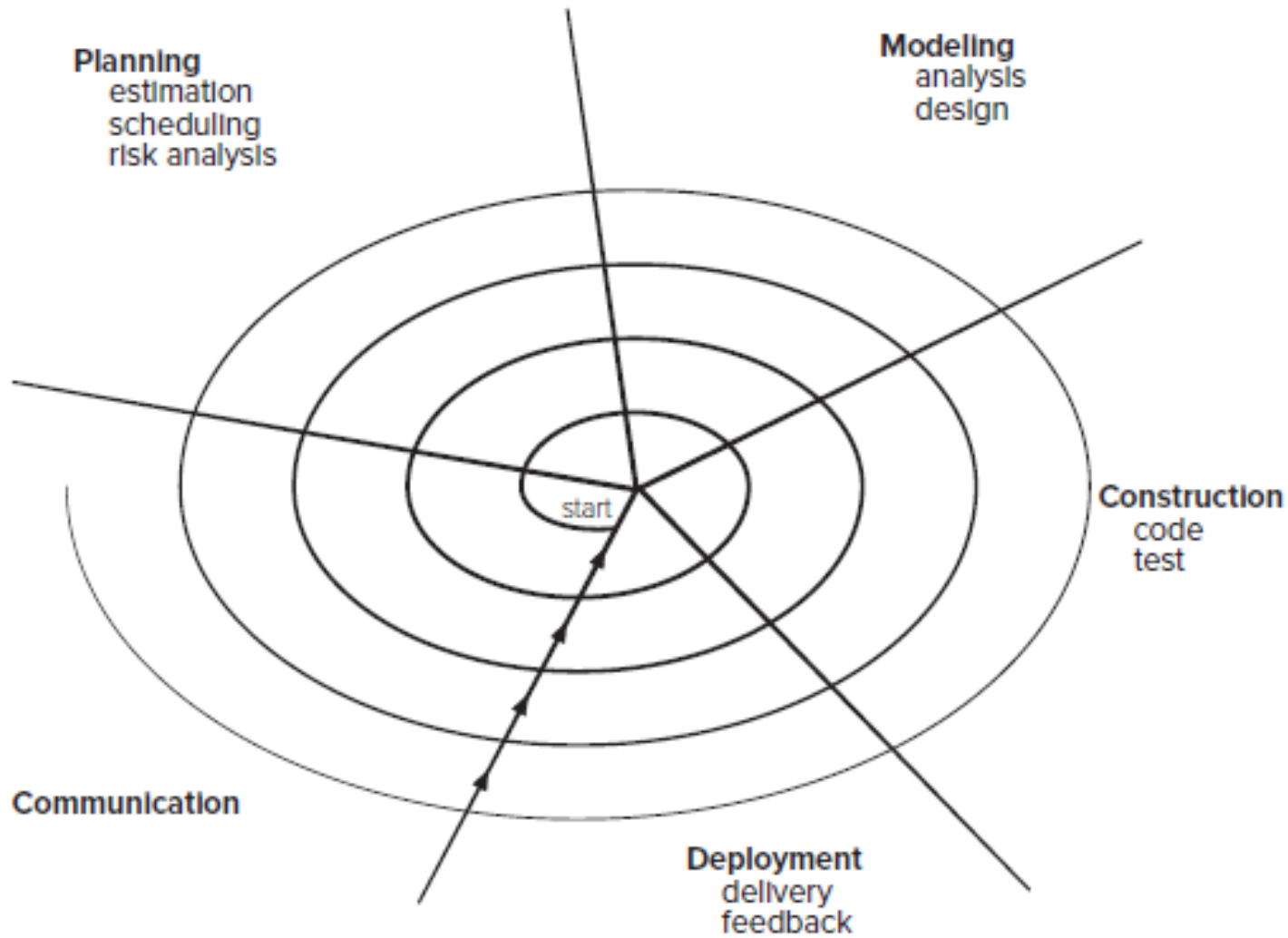


**Combina aspectos da  
prototipação com o cascata**



**Software é desenvolvido numa  
série de releases incrementais**

Nas primeiras é possível a criação de um  
"modelo de papel" ou protótipo



# MODELO ESPIRAL

# Prototipação

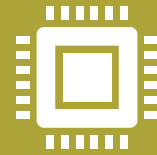
É normal que um cliente defina um conjunto de objetivos geral

Não existe a definição em detalhes dos requisitos

É possível que o desenvolvedor esteja inseguro quanto algumas características do software

Nesses casos a utilização da prototipação é ideal

# Prototipação



Pode ser  
utilizado como  
um processo

Normalmente  
é utilizado  
juntamente  
com algum  
outro processo  
de software



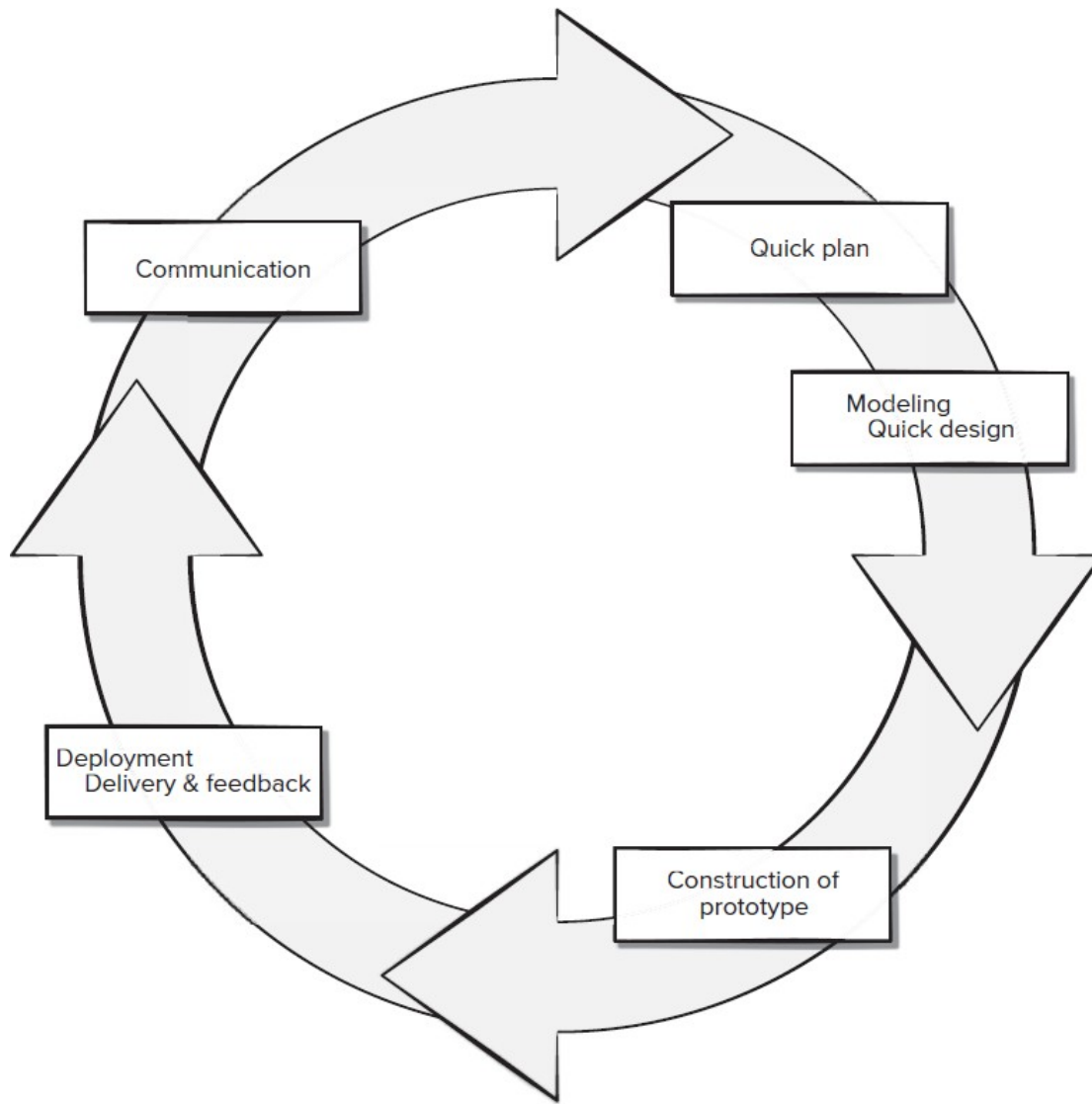
Indiferente de quando ele  
sempre é utilizado quando  
existe dúvidas quanto aos  
requisitos

# Prototipação

O protótipo serve como um mecanismo para identificar os requisitos

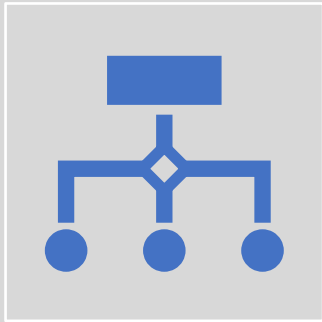
Se um protótipo funcional deve ser construído é possível usar fragmentos de outros programas ou ferramentas no-code para sua rápida geração

# Prototipação



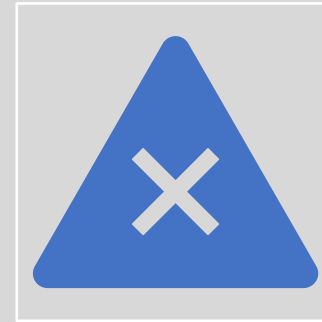


# Riscos



**Stakeholders podem achar que o protótipo é uma versão funcional do sistema. Eles não terão a compreensão que a arquitetura também está evoluindo**

Desenvolvedores podem não ter levado em consideração a qualidade técnica e nem a manutenibilidade do software



**Você pode introduzir trechos de código que são problemáticos e que, se não houver cuidados apropriados, serão integrados ao produto final**

# Comparando

## Espiral

- Combina elementos do cascata com modelos evolutivos
- Foca em análise de risco em cada ciclo
- Adequado para projetos grandes e complexos
- Permite refinamento contínuo dos requisitos

## Prototipação

- Cria protótipos rápidos para validar requisitos
- Fica em interface e funcionalidade visíveis ao usuário
- Ciclos curtos de desenvolvimento e feedback
- Bom para projetos com requisitos incertos
- Risco de foco excessivo no protótipo

# Abordagem e escopo

## **Espiral**

- Abrangente e sistemático
- Cobre todo o ciclo de vida do software
- É adequado para projetos grandes e complexos

## **Prototipação**

- Focado e ágil
- Focado na interface com usuário e nas funcionalidades visíveis
- Ideal para projetos menores ou com requisitos pouco claros

# Gestão de risco

## **Espiral**

- É especialmente focado na gestão de risco
- Possui ações dentro da atividade de planejamento para isso

## **Prototipação**

- Não tem uma abordagem formal
- Riscos são tratados de forma reativa

# Interação com o cliente

## **Espiral**

- Envolve o cliente em pontos específicos
- Geralmente no final de cada ciclo

## **Prototipação**

- Interage frequentemente com o cliente
- Recebe feedback constante sobre os protótipos

# Flexibilidade e mudanças

## **Espiral**

- Permite mudanças, sempre de forma controlada e planejada
- Normalmente incorporadas no próximo ciclo

## **Prototipação**

- Altamente flexível
- Protótipos são rapidamente alterados com base no feedback

# Documentação

## **Espiral**

- Requer documentação extensa em cada fase do processo

## **Prototipação**

- Tem baixa ênfase em documentação formal
- Foca no desenvolvimento rápido de protótipos funcionais

# Tempo de desenvolvimento

## **Espiral**

- Normalmente possui tempo mais longo
- Isso devido sua natureza sistemática e análise de riscos

## **Prototipação**

- Tempo inicial mais rápido
- Resultados visíveis rapidamente



# Adequação a diferentes tipos de projetos

## **Espiral**

- Projetos grandes, complexos e de alto risco

## **Prototipação**

- Projetos onde os requisitos são incertos ou quando é necessário um feedback rápido do usuário

# Aplicando...

- Uma empresa de eventos chamada "EventoPro" deseja desenvolver um novo sistema de software para gerenciar seus serviços. O sistema deve permitir o planejamento, organização e execução de eventos de diversos tipos e tamanhos, desde pequenas reuniões corporativas até grandes conferências e festivais.
  - Requisitos principais do sistema:
    1. Cadastro e gerenciamento de clientes
    2. Criação e acompanhamento de orçamentos
    3. Planejamento de cronogramas de eventos
    4. Gestão de fornecedores e prestadores de serviços
    5. Controle de estoque de equipamentos e materiais
    6. Venda e controle de ingressos (para eventos aplicáveis)
    7. Geração de relatórios financeiros e de desempenho
  - A empresa está indecisa entre utilizar o modelo espiral ou o modelo de prototipação para o desenvolvimento deste software.
- Você deve:
    - Analisar as vantagens e desvantagens de cada modelo (espiral e prototipação) para este projeto específico.
    - Escolher o modelo que você acredita ser mais adequado para este projeto e justifique sua escolha.
    - Descrever como você aplicaria o modelo escolhido no desenvolvimento deste sistema, detalhando as fases ou iterações principais.
    - Identificar os principais riscos associados ao desenvolvimento deste sistema e explique como o modelo escolhido ajudaria a mitigá-los.
    - Propor um cronograma inicial para o desenvolvimento do sistema, considerando as fases do modelo escolhido.
    - Discutir como você garantiria o envolvimento adequado dos stakeholders (clientes, usuários finais, equipe de desenvolvimento) ao longo do processo de desenvolvimento.