Engenharia de Software II

Aula 3

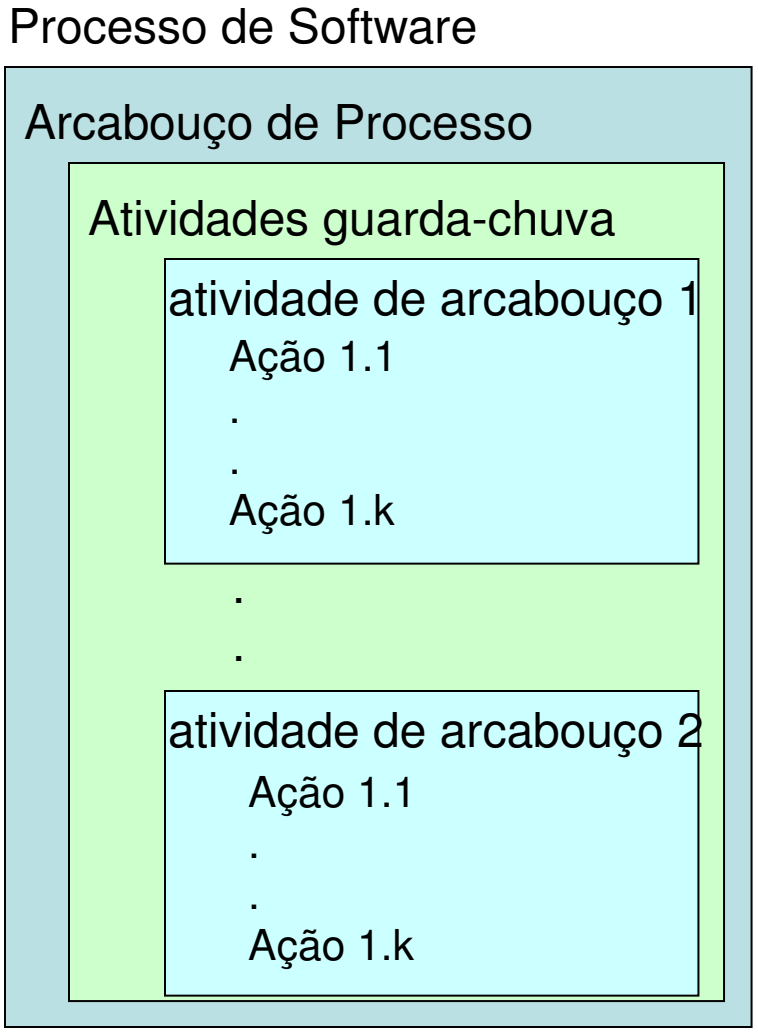
http://www.ic.uff.br/~bianca/engsoft2/

Monitoria

- Marina Albuquerque
- E-mail: monitoriaes2@yahoo.com.br
- Horário de Atendimento:
  - Terça e quinta de 09:00 às 11:00 hrs
  - Quarta de 14:00 às 16:00 hrs

Revisão: arcabouço de processo

- É o **alicerce** ou esqueleto de um processo de software completo.
- Contém as **atividades de arcabouço** que são aplicáveis a todos os projetos de software.
- Engloba um conjunto de **atividades guarda-chuva** que são exercidas durante todo o processo.



Revisão: atividades genéricas

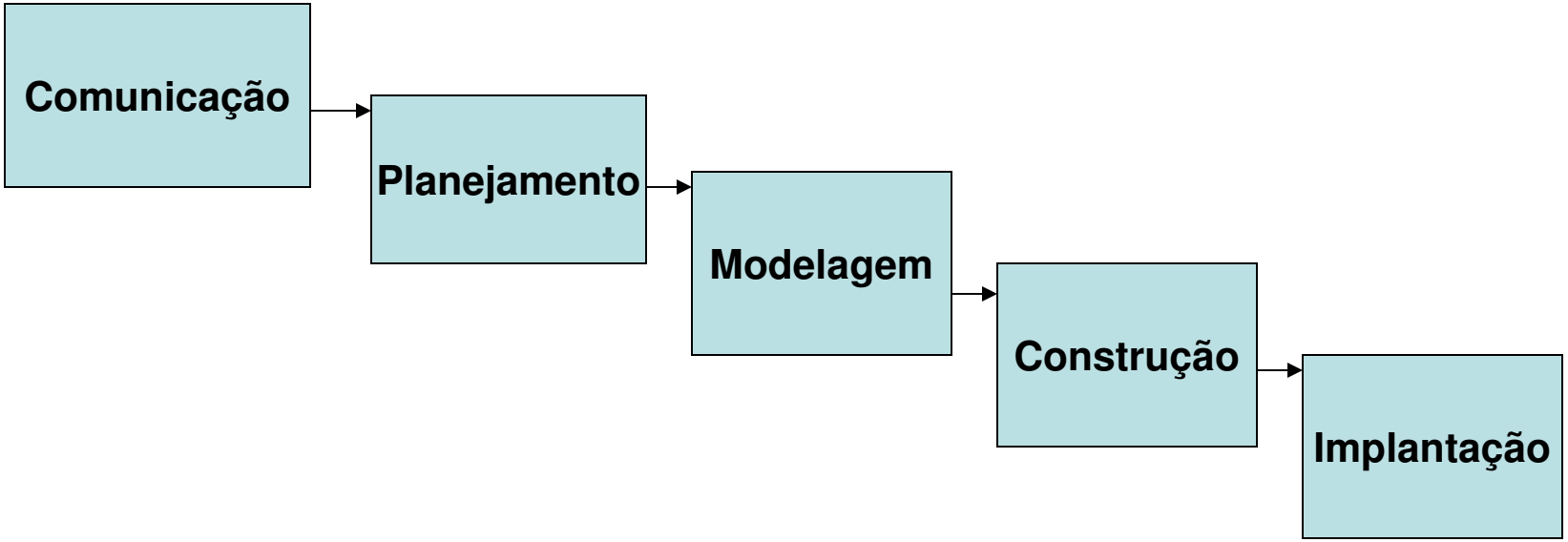
- Quais são as **atividades de arcabouço** aplicáveis à maioria dos projetos de software?
  1. **Comunicação**: levantamento de requisitos em colaboração com o cliente.
  2. **Planejamento:** descreve as tarefas, os riscos, os recursos, os produtos e um cronograma.
  3. **Modelagem:** criação de modelos que permitam ao desenvolvedor entender melhor o projeto e seus requisitos. Ações:
     - Análise – modelos de especificação de requisitos.
     - Projeto – modelos de especificação de projeto.
  4. **Construção:** geração de código e testes.
  5. **Implantação:** entrega do software ao cliente.

Modelos Prescritivos de Processo

- Um modelo de prescritivo de processo *preenche* o arcabouço de processo com conjuntos explícitos de tarefas.
- Cada modelo prescritivo de processo também prescreve um *fluxo de trabalho* = maneira como os elementos se inter-relacionam.
- Todos os modelos acomodam as atividades genéricas de arcabouço, mas diferem na *ênfase* e no fluxo.

Modelo em Cascata

- Também chamado de *ciclo de vida clássico*.
- Sugere uma abordagem *sistemática* e *seqüencial* para o desenvovimento de software.

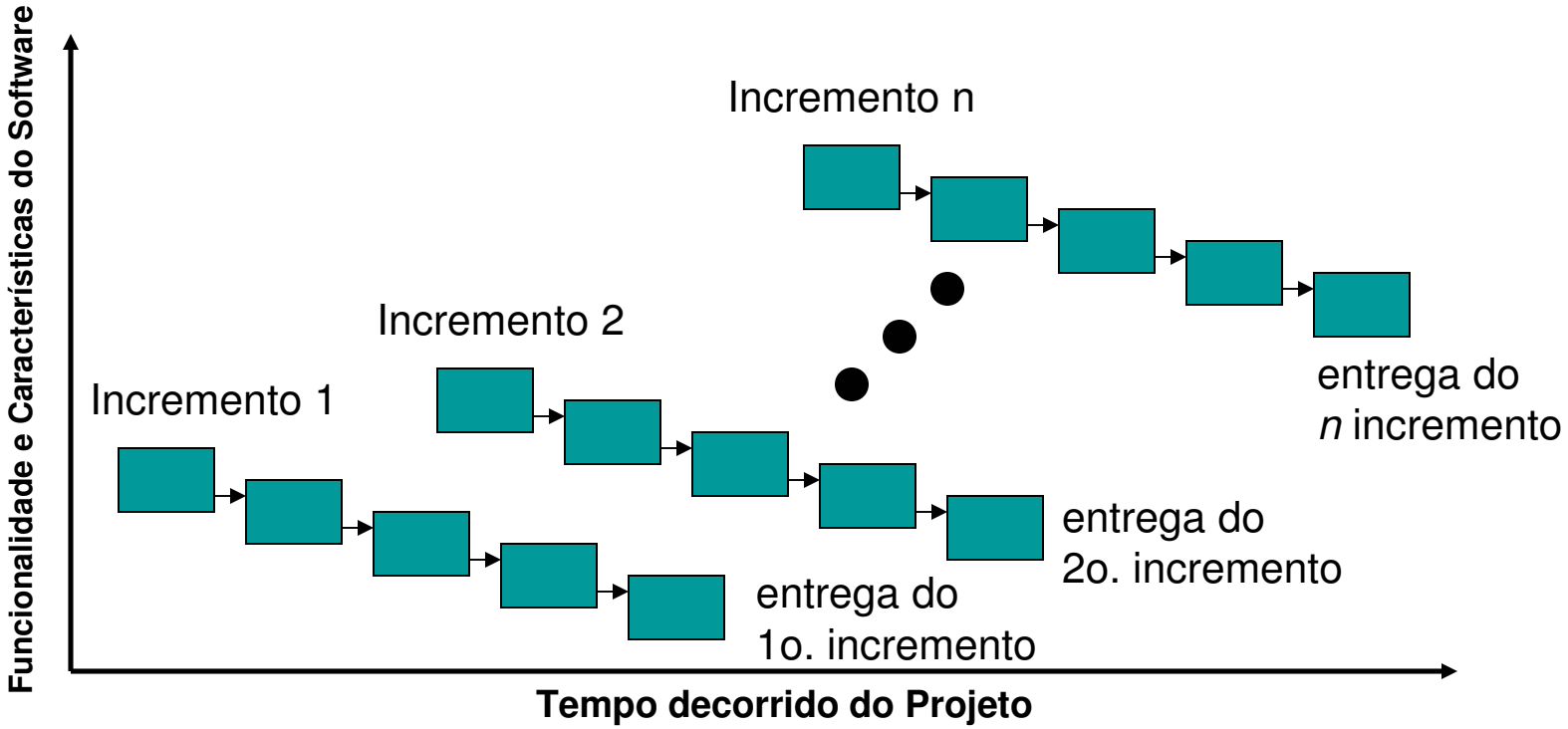


Modelo em Cascata

- É o paradigma mais antigo da engenharia de software.
- Nas últimas duas décadas, têm surgido críticas e questionamentos sobre a sua eficácia.
- Por que o modelo de cascata freqüentemente falha?
  - Projetos reais raramente seguem o fluxo seqüencial e modificações podem causar confusão.
  - É difícil para estabelecer todos os requisitos inicialmente.
  - O cliente precisa ter paciência porque uma versão executável do programa só ficará disponível no final do processo.
  - O modelo leva a “estados de bloqueio”, nos quais membros da equipe ficam esperando outros membros terminar a sua parte.
- O modelo em cascata é adequado quando os requisitos são bem compreedidos, como em aperfeiçoamentos de um sistema existente.

Modelo Incremental

- Combina elementos do modelo em cascata aplicado de maneira iterativa.



Modelo Incremental

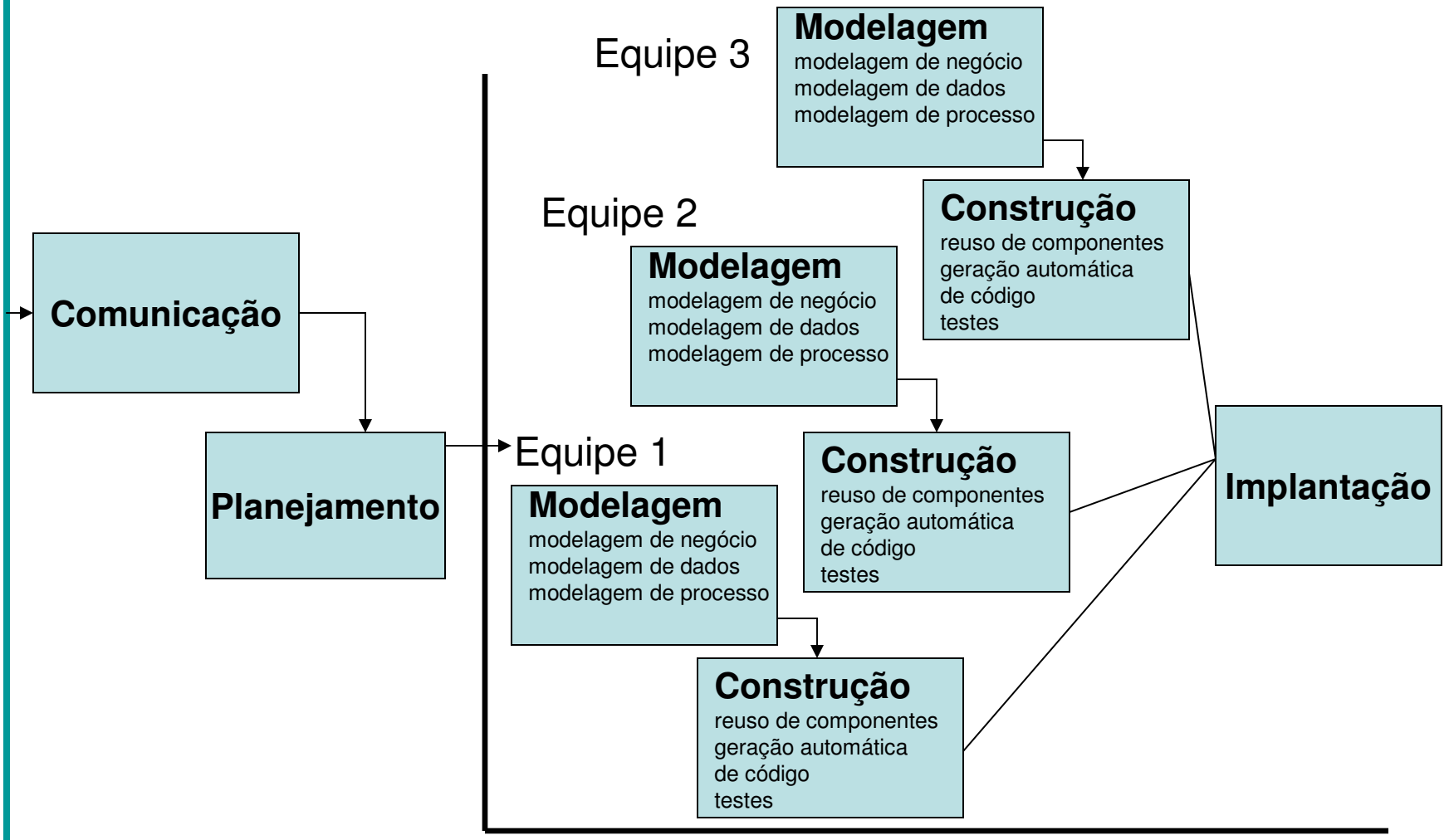
- Cada seqüência produz incrementos do software passíveis de serem entregues, que fornecem progressivamente mais funcionalidade.
- O primeiro incremento é chamado de *núcleo do produto*.
  - Os requisitos básicos são satisfeitos.
  - Um plano é desenvolvido para o próximo incremento como resultado do seu uso/avaliação.
- O modelo incremental é particularmente útil quando não há mão-de-obra/recursos disponíveis para uma implementação completa.

Modelo RAD (*Rapid Application Development*)

- Enfatiza um ciclo de desenvolvimento curto, com o uso de uma abordagem de construção baseada em *componentes*.
- O planejamento é essencial, porque equipes trabalham *em paralelo* em diferentes funções do sistema.
- A modelagem abrange três fases:
  - Modelagem de negócio
  - Modelagem de dados
  - Modelagem de processo

  e estabelece representações de projeto que servem como base para a atividade de construção.
- A implantação estabele a base para iterações subsequentes, se necessárias.

Modelo RAD


Equipe 3
Modelagem
modelagem de negócio
modelagem de dados
modelagem de processo
Construção
reuso de componentes
geração automática
de código
testes
Equipe 2
Modelagem
modelagem de negócio
modelagem de dados
modelagem de processo
Comunicação
Planejamento
Equipe 1
Construção
reuso de componentes
geração automática
de código
testes
Implantação
Modelagem
modelagem de negócio
modelagem de dados
modelagem de processo
Construção
reuso de componentes
geração automática
de código
testes
60-90 dias

Modelo RAD

- Recomendável quando uma aplicação pode ser modularizada de maneira que a função principal possa ser implementada em menos de 3 meses.
- Quais são as desvantagens do modelo RAD?
  - Exige pessoal suficientes para criar um número de equipes RAD.
  - Desenvolvedores e clientes têm que estar comprometidos com as atividades rápidas.
  - Exige que o sistema seja modularizável.
  - Não é adequado quando os riscos técnicos são altos.

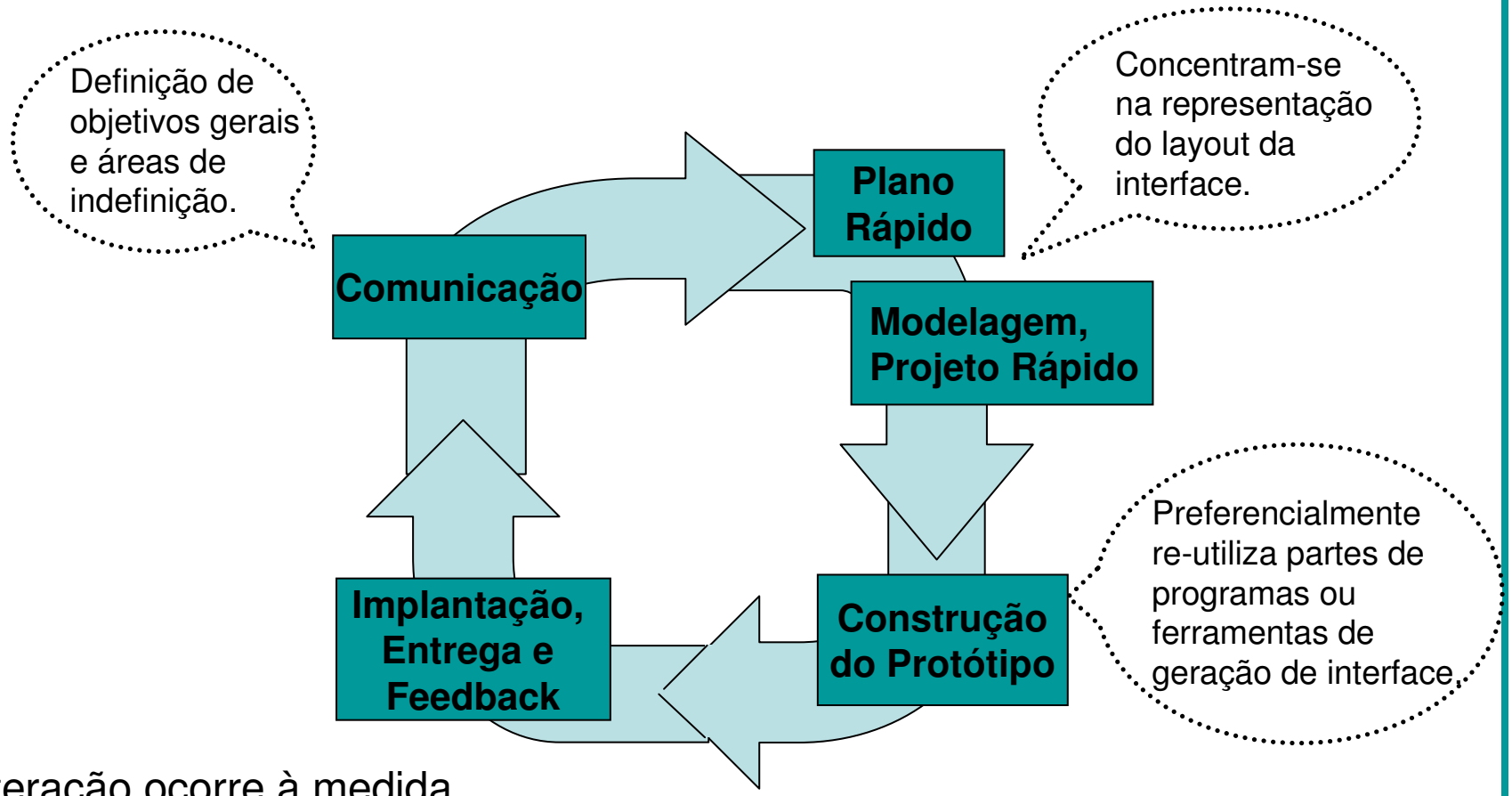
Modelos Evolucionários

- São explicitamente projetados para acomodar um produto que *evolui* com o tempo.
- A cada iteração, produzem uma versão cada vez mais completa do software.
- Exemplos:
  - Modelo de Prototipagem
  - Modelo Espiral
  - Modelo de Desenvolvimento Concorrente

Modelo de Prototipagem

- Auxilia o engenheiro de software e o cliente a entenderem melhor o que deve ser construído quando os requisitos estão confusos.
- Mais comumente usado como uma técnica que pode ser implementada dentro do contexto de qualquer outro modelo.

Modelo de prototipagem



A iteração ocorre à medida que o protótipo é ajustado para satisfazer às necessidades do cliente.

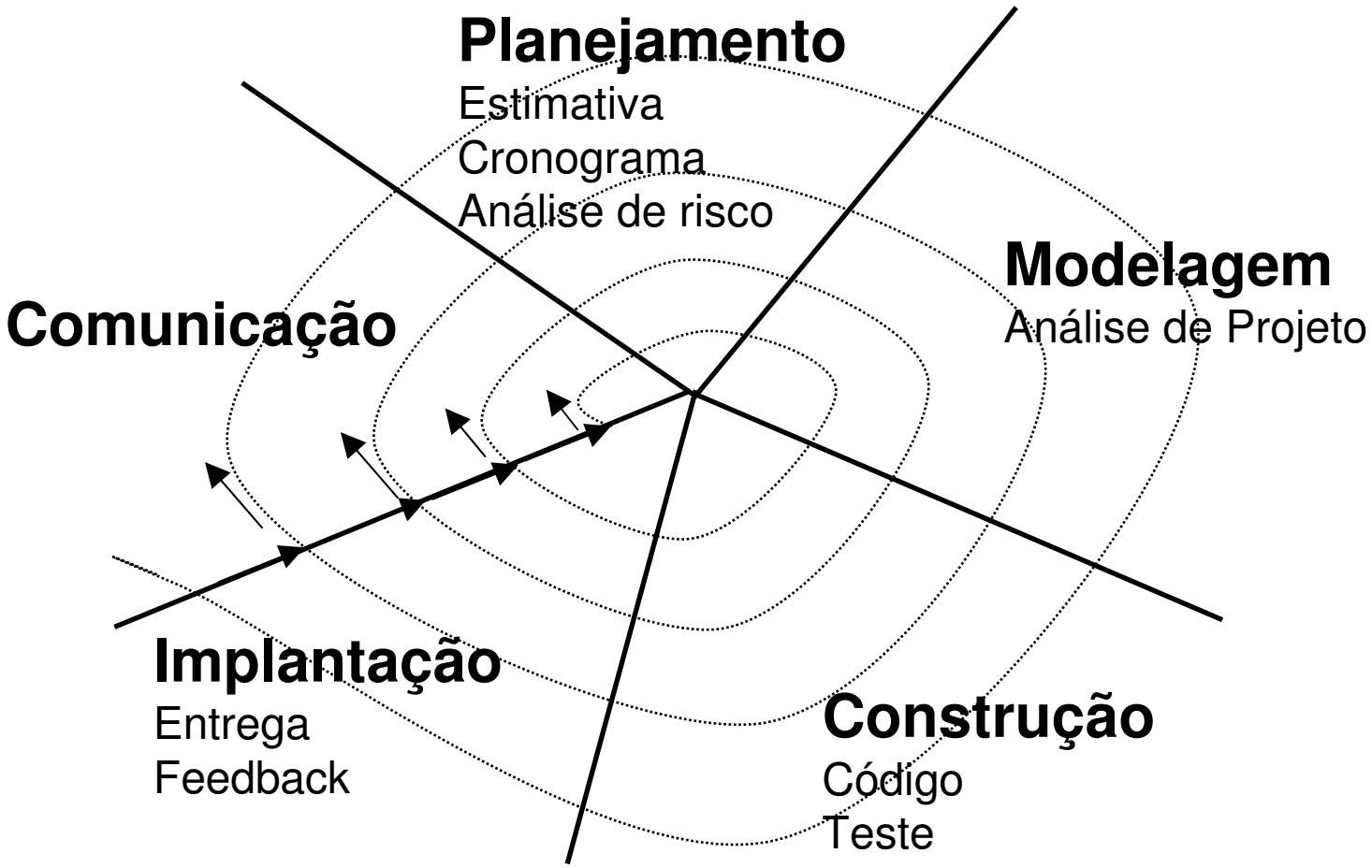
Modelo de Prototipagem

- Problemas:
  - Pode haver pressão do cliente para transformar um protótipo malfeito em produto final, resultando em baixa qualidade.
  - Concessões na implementação podem fazer com que o desenvolvedor fique familiarizado com escolhas não ideais.
- O cliente tem que concordar que o protótipo será usado apenas para levantamento de requisitos e que o software real será submetido à engenharia com olho na qualidade.

Modelo Espiral

- Combina a natureza iterativa da prototipagem com os aspectos controlados do modelo em cascata.
- O software é produzido numa série de versões evolucionárias.
  - Primeiras versões no só papel ou protótipo.
- É uma abordagem cíclica que aumenta incrementalmente o grau de definição, enquanto diminui o risco.
- O modelo pode ser aplicado ao longo de todo ciclo de vida de uma aplicação.

Modelo Espiral



Modelo Espiral

- É uma abordagem realista do desenvolvimento de software.
- Problemas:
  - Pode ser difícil convencer os clientes que a abordagem é controlável.
  - A gerência pode exigir um orçamento fixo, o que não é compatível com o modelo espiral.
  - Exige competência na avaliação de riscos.