

Engenharia de Software – ESOF

Modelos de Processo

Prof. Leonardo Vieira Barcelos

Modelos de Processo de Software

- Cada modelo prescreve um fluxo de trabalho, isto é, a maneira como os elementos de processo se inter-relacionam uns com os outros.
- Independentemente do modelo de processo que seja selecionado, os engenheiros de software têm tradicionalmente escolhido uma estrutura de processo genérico que inclui as seguintes atividades de estrutura: comunicação, planejamento, modelagem, construção e implantação.
- Existem vários modelos de processo de software, cada um representa uma tentativa de colocar ordem no caos do desenvolvimento de software.
- Pode-se citar os seguintes modelos de processo de software.

Modelos de Processo de Software

1. O Modelo Cascata

2. Modelos Incrementais de Processo

2.1 O Modelo Incremental

2.2 O Modelo RAD (Desenvolvimento Rápido de Aplicação)

3. Modelos Evolucionários de Processo de Software

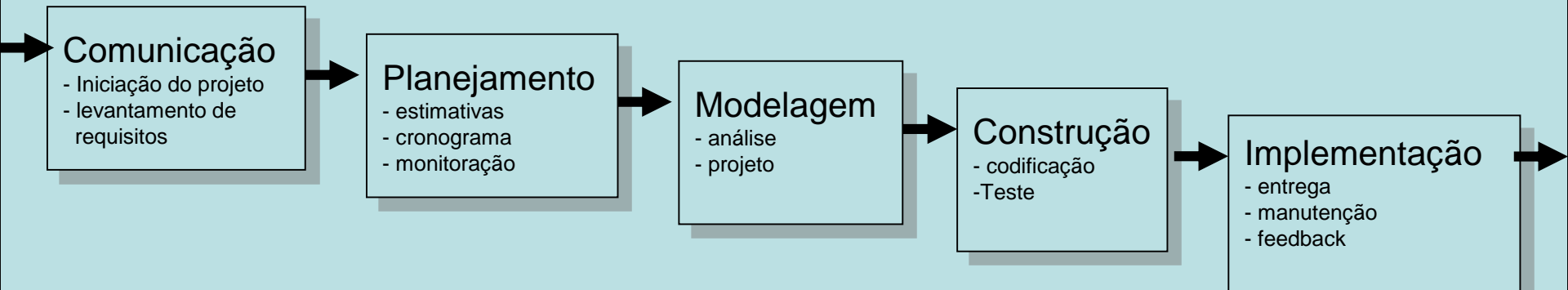
3.1 Prototipagem

3.2 O Modelo Espiral

1. O Modelo Cascata

- O modelo em cascata, algumas vezes chamado de ciclo de vida clássico, sugere uma abordagem sistemática e seqüencial para o desenvolvimento de softwares que começa com a especificação dos requisitos pelo cliente e progride ao longo do planejamento, modelagem, construção e implantação, culminando na manutenção progressiva do software acabado.
- O modelo mais antigo da engenharia de software.

O Modelo em Cascata



Problemas com o Modelo Cascata

- 💣 Projetos reais raramente seguem o fluxo seqüencial que o modelo propõe;
- 💣 Em geral, é difícil para o cliente estabelecer todos os requisitos explicitamente. No começo dos projetos sempre existe uma incerteza natural;
- 💣 O cliente deve ter paciência. Uma versão executável do software não ficará disponível até o período final do intervalo de tempo do projeto.

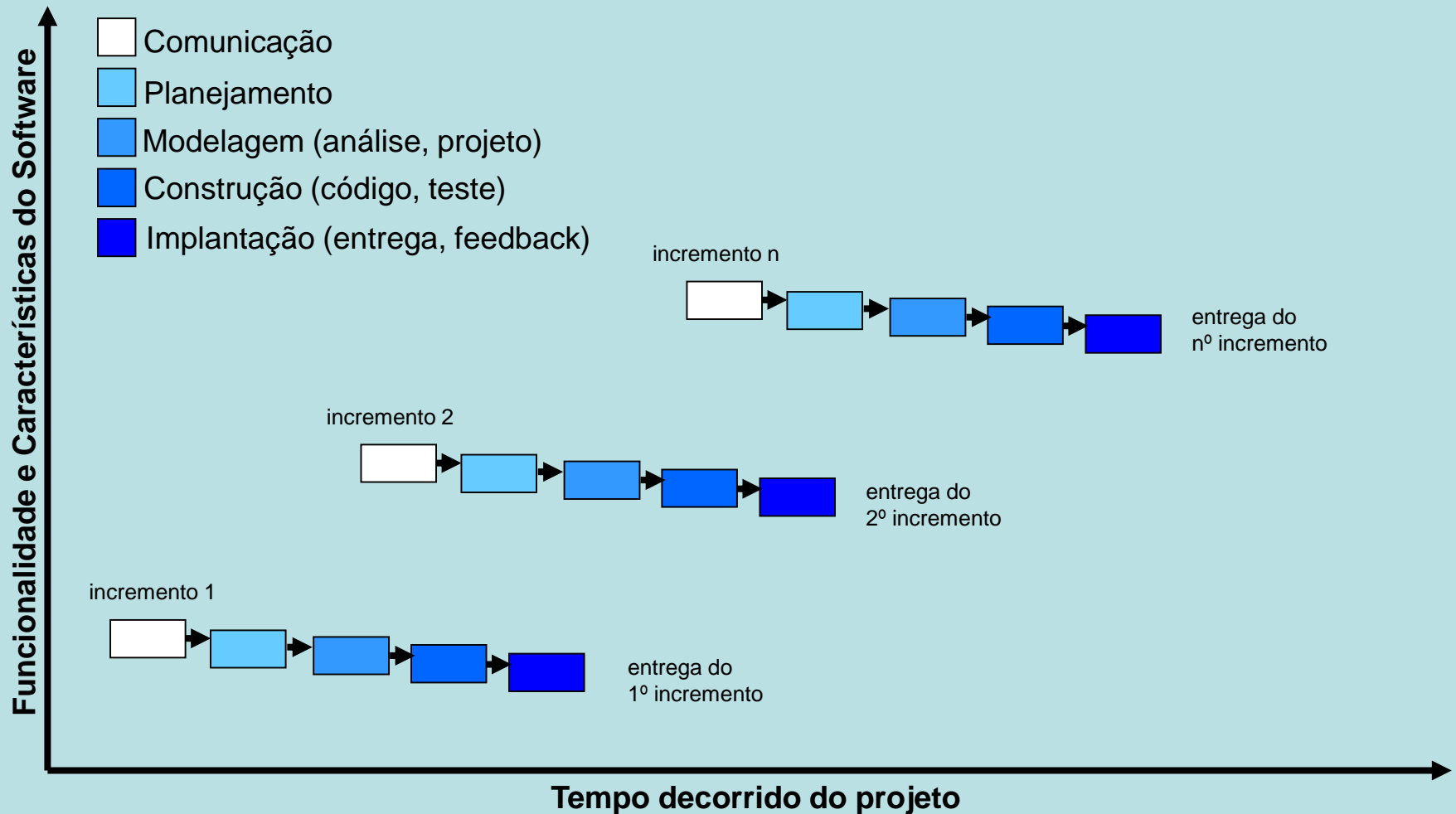
Embora o Modelo Cascata tenha fragilidades, ele é significativamente melhor do que uma abordagem casual ao desenvolvimento de software

2. Modelos Incrementais de Processo

O Modelo Incremental

- O modelo incremental combina elementos do modelo cascata aplicado repetidamente;
- O objetivo é trabalhar junto do usuário para descobrir seus requisitos, de maneira incremental, até que o produto final seja obtido;
- A versão inicial é frequentemente o **núcleo do produto** (a parte mais importante)
 - o desenvolvimento começa com as partes do produto que são mais bem entendidas;
 - a evolução acontece quando novas características são adicionadas à medida que são sugeridas pelo usuário.
- O desenvolvimento incremental é importante quando é difícil, ou mesmo impossível, estabelecer a especificação detalhada dos requisitos;
- As primeiras versões podem ser implementadas com poucas pessoas
 - se o núcleo do produto for bem recebido, pessoas adicionais podem ser colocadas para a implementação da nova versão.

O Modelo Incremental



O Modelo Incremental

- As novas versões podem ser planejadas de modo que os riscos técnicos possam ser administrados.
Exemplo:
 - O sistema pode exigir a disponibilidade de um hardware que está em desenvolvimento e cuja data de liberação é incerta;
 - Pode-se planejar pequenas versões de modo a evitar o uso desse hardware;
 - O software com funcionalidade parcial pode ser liberada para o cliente.
- O modelo incremental é mais apropriado para sistemas pequenos
 - Os problemas de mudança podem ser contornados através da reimplementação do sistema quando mudanças substanciais se tornam necessárias.
- No modelo incremental os testes podem ser mais efetivos
 - é mais fácil testar cada versão do sistema do que o sistema todo no final.

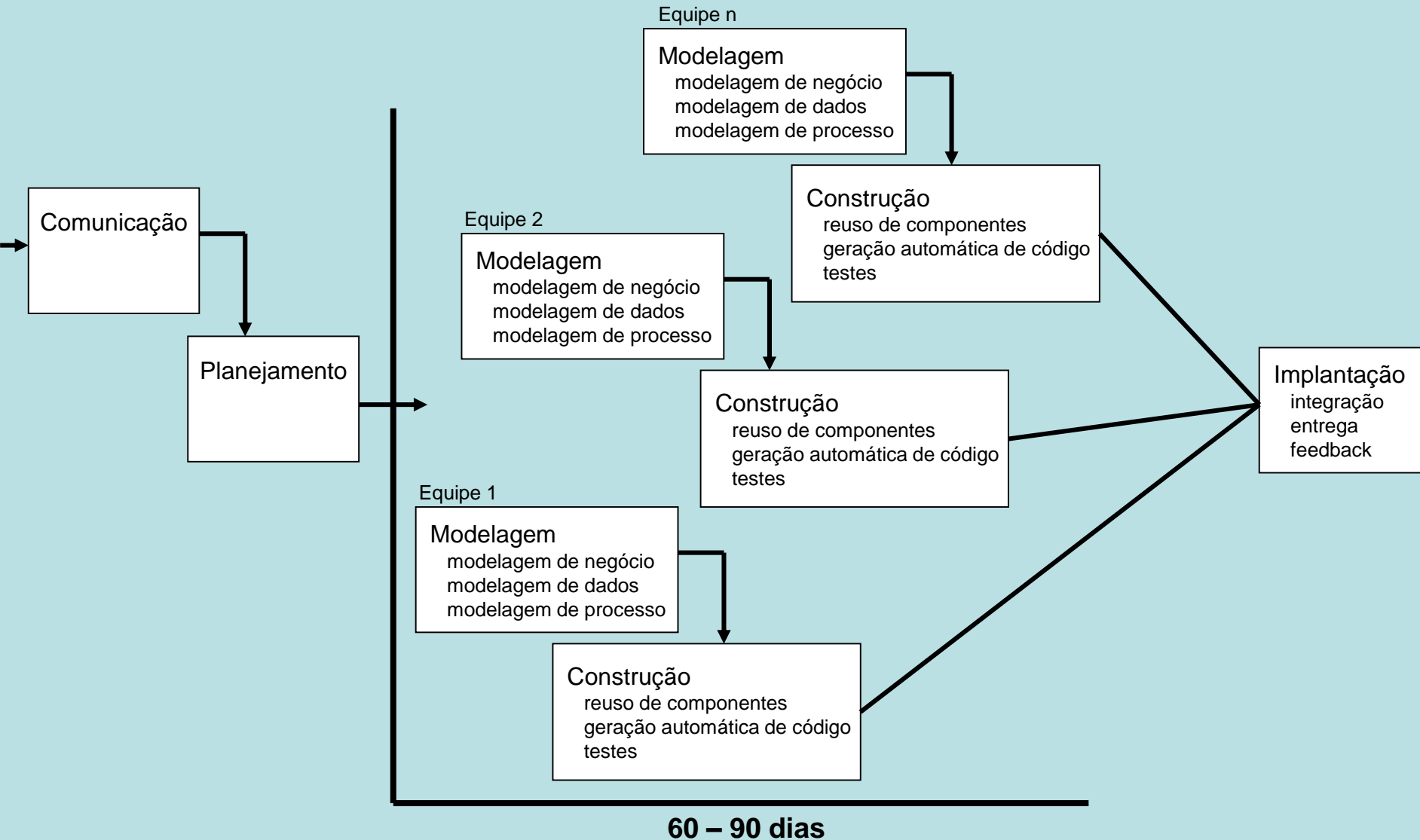
O Modelo RAD

- RAD (*Rapid Application Development*) é um modelo de processo de software incremental que enfatiza um ciclo de desenvolvimento curto.
- O modelo RAD é uma adaptação de “alta velocidade” do modelo em cascata no qual o desenvolvimento rápido é obtido usando uma abordagem de construção baseada em componentes.
- Se os requisitos forem bem entendidos e o objetivo do projeto for restrito, o processo *RAD* permite a uma equipe de desenvolvimento criar um “sistema plenamente funcional”, dentro de um período de tempo muito curto (exemplo: 60 a 90 dias).

O Modelo RAD

- Como outros modelos de processo, a abordagem RAD se enquadra nas atividades genéricas de estrutura apresentadas anteriormente.
 - A *comunicação* trabalha para entender os problemas do negócio e as características de informação que o software precisa acomodar.
 - O *planejamento* é essencial, porque várias equipes de software trabalham em paralelo em diferentes funções do sistema.
 - A *modelagem* estabelece representações de projeto que servem como base para a atividade de construção do RAD e abrange três das principais fases:
 - modelagem de negócio;
 - modelagem dos dados;
 - modelagem de processos.
 - A construção enfatiza o uso de componentes de softwares preexistentes e a aplicação da geração automática de código.
 - Finalmente, a *implantação* estabelece a base para iterações subsequentes se necessárias.

O Modelo RAD



Cada função principal pode ser direcionada para uma equipe RAD separada e então integrada para formar o todo.

O Modelo RAD

- Desvantagens:
 - Para projetos grandes, mas passíveis de sofrer aumento, o RAD exige recursos humanos suficientes para criar o número adequado de equipes RAD.
 - *RAD* exige desenvolvedores e clientes que estejam comprometidos com as atividades continuamente rápidas, necessárias para completar o sistema em um curtíssimo espaço de tempo - se faltar comprometimento o projeto *RAD* irá falhar
 - Nem todos os tipos de aplicação são apropriadas para o RAD:
 - Se o sistema não puder ser modularizado apropriadamente, a construção dos componentes necessários para o RAD será problemática
 - RAD não é apropriado quando os riscos técnicos são altos.
 - quando uma nova aplicação faz uso de novas tecnologias

3. Modelos Evolutivos de Processo de Software

- O software, como todo sistema complexo, evolui com o tempo. Os requisitos do negócio e do produto mudam freqüentemente à medida que o desenvolvimento prossegue, dificultando um caminho direto para um produto de software final; prazos reduzidos de mercado tornam impossível completar um produto de software abrangente, mas uma versão reduzida pode ser elaborada para fazer à competitividade ou às pressões do negócio.
- Neste caso, e em situação semelhantes, os engenheiros de software precisam de um modelo de processo que tenha sido explicitamente projetado para acomodar um produto que evolui com o tempo.

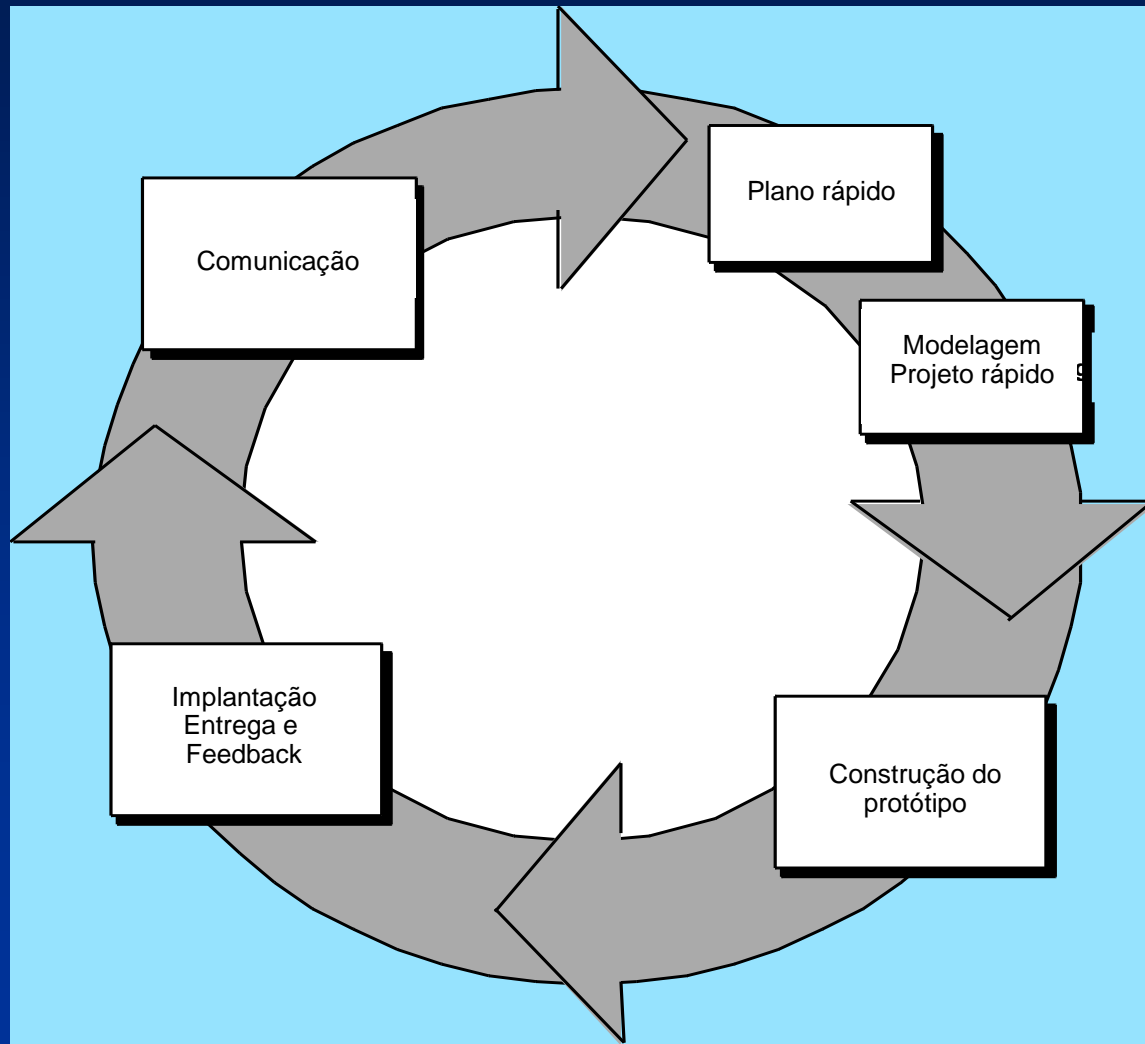
Prototipagem

- O cliente, freqüentemente, define um conjunto de objetivos gerais para o software, mas não identifica detalhadamente requisitos de entrada, processamento ou saída. Em outros casos, o desenvolvedor pode estar inseguro da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação homem/máquina deve assumir. Nessas e muitas outras situações, um paradigma de prototipagem pode oferecer a melhor abordagem.

Prototipagem

- O paradigma de prototipagem começa com a comunicação. O engenheiro de software e o cliente encontram-se e definem os objetivos gerais do software, identificam as necessidades conhecidas e delineiam áreas que necessitam de mais definições. Uma iteração de prototipagem é planejada rapidamente e a modelagem (na forma de um “projeto rápido” ocorre.
- O projeto rápido concentra-se na representação daqueles aspectos do software que estarão visíveis para o cliente/usuário (por exemplo, layout da interface humana ou formatos de saída de tela). O projeto leva à construção de um protótipo, que é implantado e depois avaliado pelo cliente/usuário. O feedback é usado para refinar os requisitos do software.

O Modelo de Prototipagem



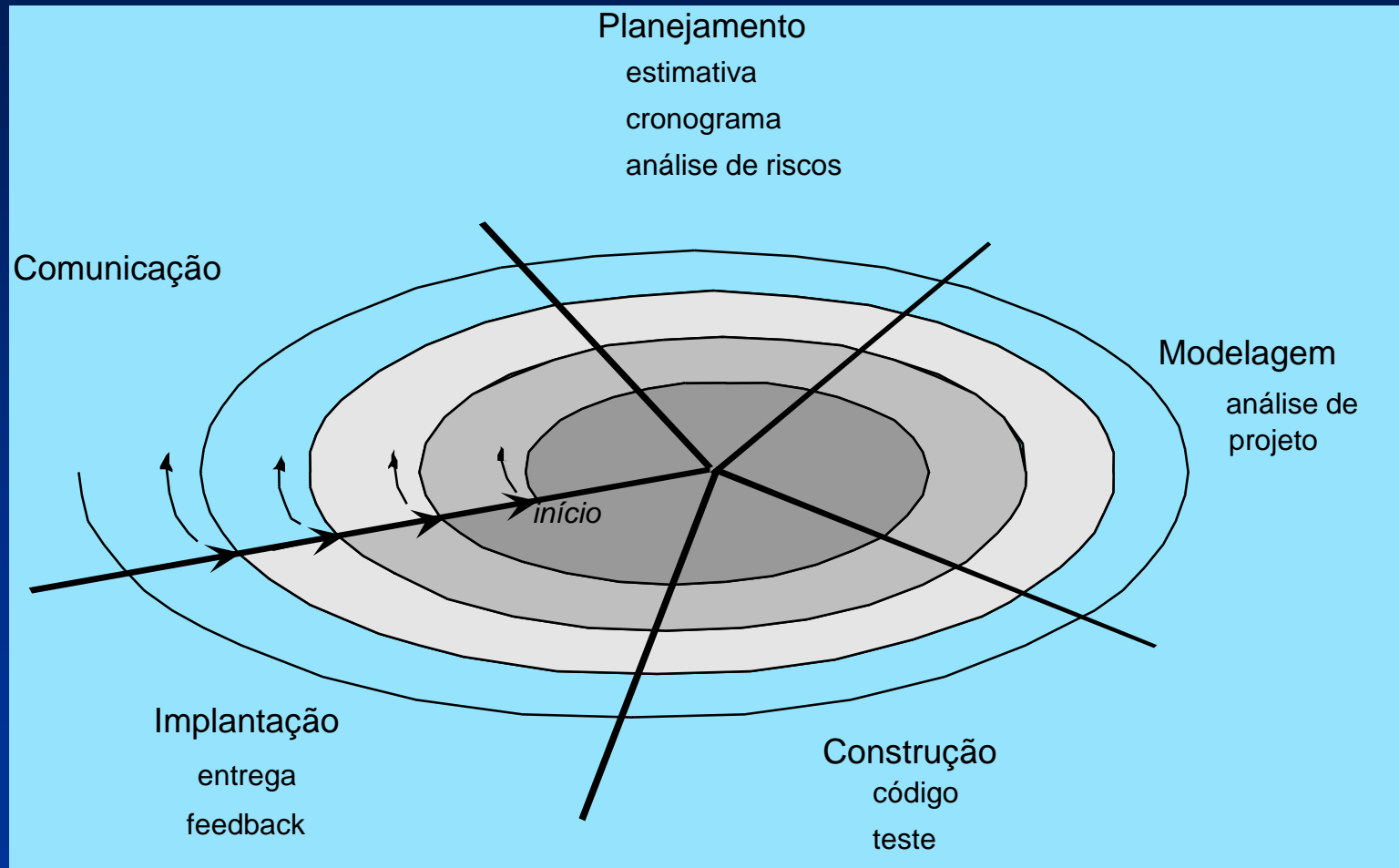
Problemas com a Prototipagem

- Cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo;
- Desenvolvedor freqüentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo.
- O cliente e o desenvolvedor devem ambos concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos.

O Modelo Espiral

- O modelo espiral acopla a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos do modelo cascata.

Modelo espiral típico



O Modelo Espiral

- O primeiro círculo em torno do espiral poderia resultar no desenvolvimento da especificação de um produto;
- Passagens subseqüentes em torno da espiral poderiam ser usadas para desenvolver um protótipo e depois, progressivamente, versões mais sofisticadas do software.
- Cada passagem pela região de planejamento resulta em ajustes do plano de projeto. O custo e o cronograma são ajustados com base no feedback derivado do cliente após a entrega.

O Modelo Espiral

- O modelo espiral é uma abordagem realística para o desenvolvimento de software de grande porte.
- Como o software evolui à medida que o processo avança, o desenvolvedor e o cliente entendem melhor e reagem aos riscos de cada nível evolucionário.