

Fases e Atividades do Desenvolvimento de Software

Fases Fundamentais (Ciclo de Vida)

- Fase de definição
 - Análise e Especificação de Requisitos
 - Estudo de Viabilidade
 - Estimativas Planejamento
- Fase de desenvolvimento (ou construção)
 - Design
 - Implementação e integração
 - Verificação e Validação
- Fase de operação
 - Distribuição, Instalação e Configuração
 - Utilização e administração
 - Manutenção – corretiva, evolutiva e adaptativa
- Fase de retirada
 - Migração, reengenharia, engenharia reversa

A Fase de Definição

- Deve-se levantar e analisar os requisitos funcionais e não funcionais, e
- analisar recursos e restrições para
 - apresentar soluções,
 - estudar a viabilidade,
 - planejar e gerenciar o desenvolvimento
 - a partir de estimativas e análise de riscos

A Fase de Definição

- Algumas atividades
 - Levantar requisitos funcionais e não funcionais
 - Documentar os requisitos
 - Construir glossário (ao longo da fase)
 - Definir modelo conceitual inicial (análise inicial)
 - Construir protótipos (se necessário)
 - Priorizar as funcionalidade e distribuí-la entre as iterações (planejamento)

Fase de Desenvolvimento

- Análise
 - Refinamento dos requisitos, refinamento do modelo conceitual, refinamento da arquitetura.
- Design de Software
 - Design detalhado, design da interface de usuário, design de algoritmos e estruturas de dados
- Implementação e integração
 - Codificação, compilação, integração e verificação de programas (testes, inspeção, depuração)
- Verificação da qualidade
 - Testes beta, avaliação de usabilidade, avaliação de desempenho, etc.

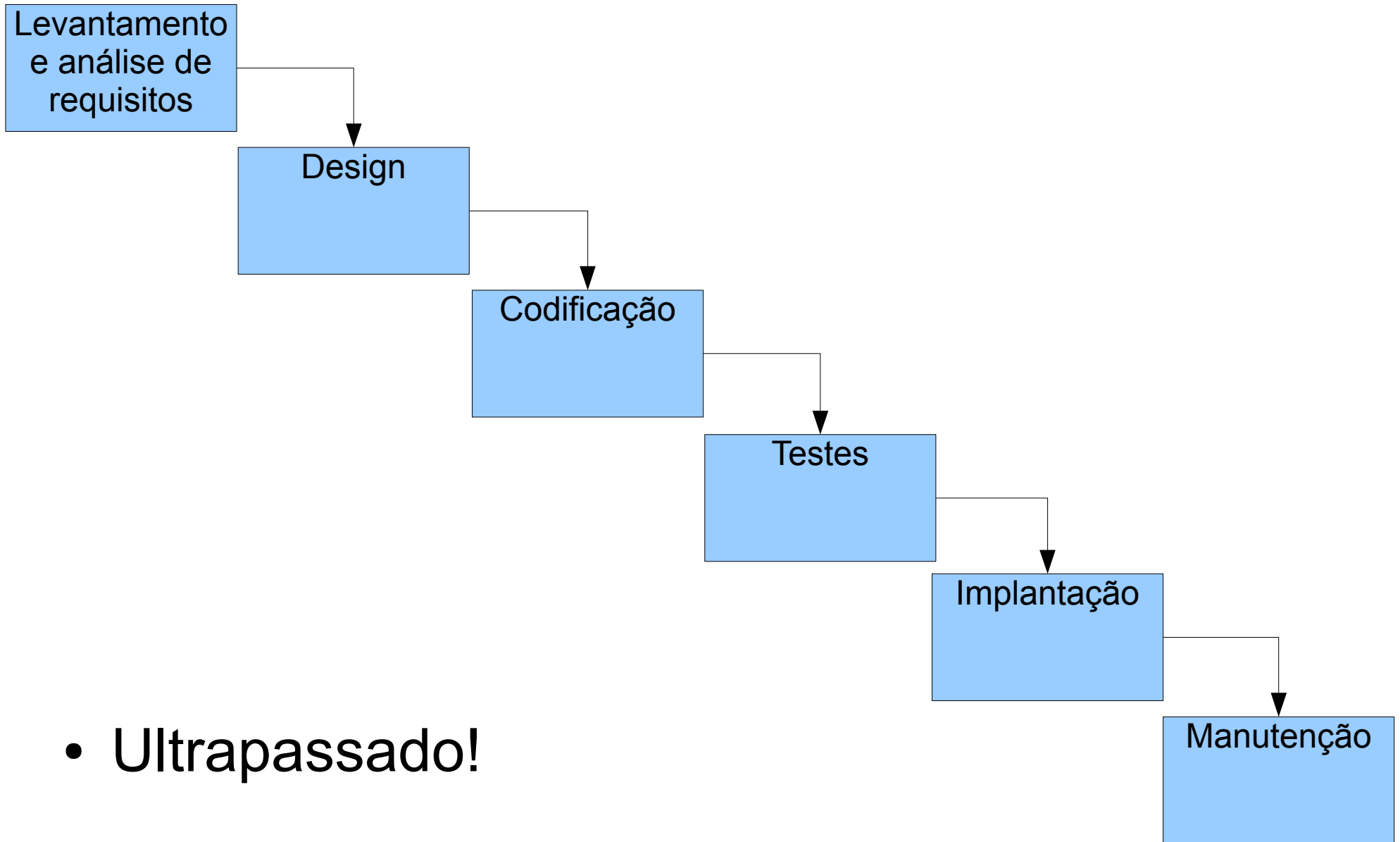
Fase de Operação

- Distribuição e entrega
- Instalação e configuração
- Treinamento de usuários
- Utilização
- Manutenção
 - Corretiva – correção de erros
 - Evolutiva ou adaptativa – novas versões
 - Novos requisitos
 - novas situações de operação – hardware, sistemas operacionais

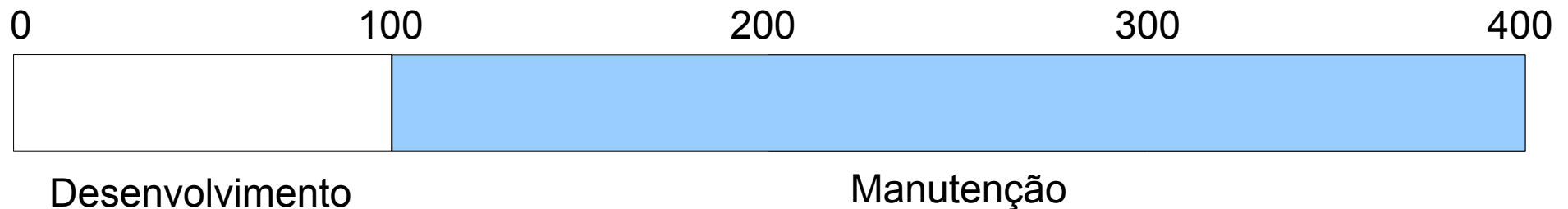
Principais Atividades

- Especificação e análise de requisitos
- Design
- Codificação ou programação ou implementação
- Testes
- Inspeções
- Implantação
- Manutenção

Modelo em Cascata



Custos do Desenvolvimento



Desenvolvimento = Especificação de requisitos, design, codificação, testes e implantação

Manutenção = Correções, adaptações e evoluções

Desenvolvimento Iterativo

- Hoje, é considerado errado ter um processo que gere o software todo de uma vez.
 - Não se deve ter o software inteiro funcionando por inteiro no primeiro release
 - O risco é grande demais!
- Um processo de desenvolvimento deve ser:
 - Iterativo (ter várias iterações no tempo)
 - Incremental (gerar novas versões incrementadas a cada iteração)
 - Uma iteração dura entre 2 semanas e 2 meses

Desenvolvimento Iterativo

- Motivos:
 - Sempre tem algo para entregar para o cliente apressado (a última iteração)
 - Os requisitos mudam com tempo e um processo iterativo mantém freqüentes contatos com o cliente o que ajuda a manter os requisitos sincronizados
 - Altamente motivador para a equipe de desenvolvimento (e o cliente) ver o software funcionando cedo
 - Ajudar a evitar isso: (próximo slide)

Desenvolvimento Iterativo



Como o cliente explicou...



Como o líder de projeto entendeu...



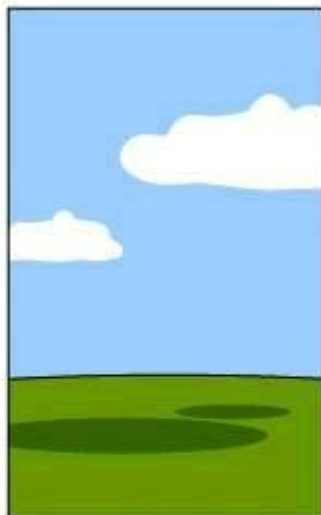
Como o analista projetou...



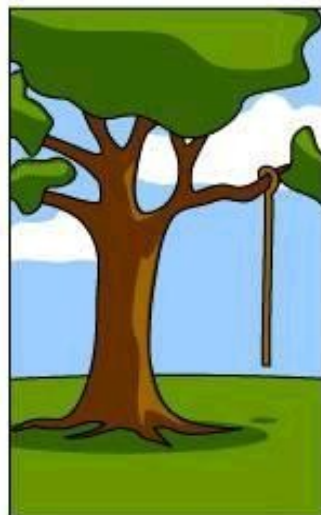
Como o programador construiu...



Como o Consultor de Negócios descreveu...



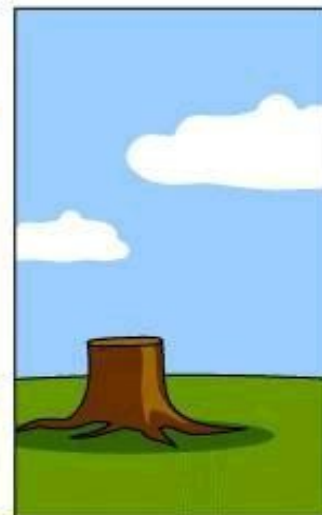
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...

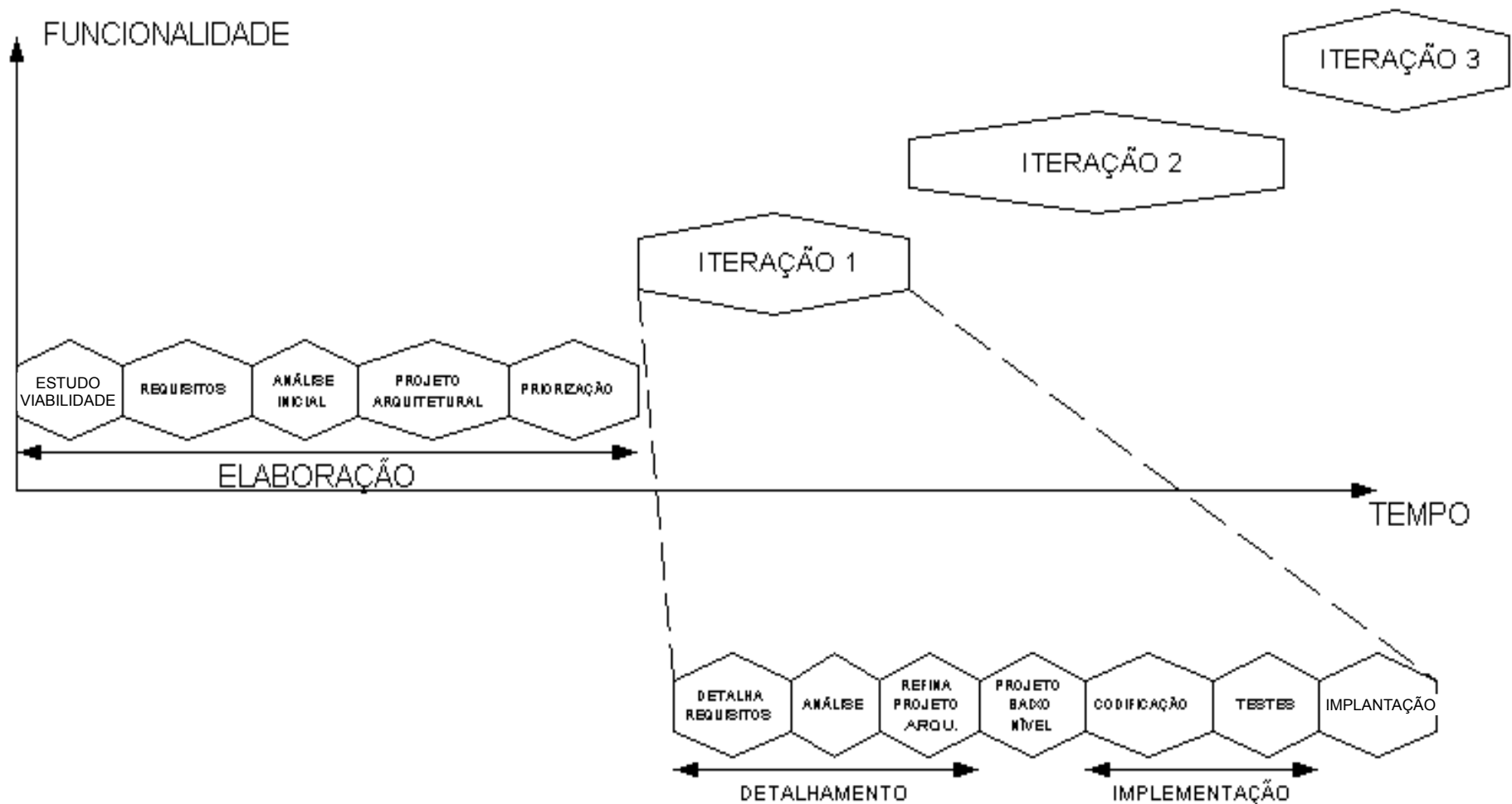


O que o cliente realmente queria...

Desenvolvimento Iterativo

- O que é feito a cada iteração?
 - Análise (refinamento de requisitos, refinamento do modelo conceitual)
 - Design (refinamento do design arquitetural, design detalhado)
 - Codificação
 - Testes
 - Implantação

Desenvolvimento Iterativo



Detalhes sobre o levantamento de requisitos

- Entendimento do que o usuário quer
- O resultado é uma promessa para o cliente
- Não se fala "como" as coisas serão feitas
- "Use cases" descrevem cenários de funcionalidade desejada
 - Também chamados de "User Stories", pois é o usuário que decide o que deve ser feito

Detalhes sobre o levantamento de requisitos

- Não só requisitos funcionais, mas também:
 - Facilidade de uso necessária
 - Quem utilizará o produto
 - Hardware e software alvo para o produto
 - Questões de qualidade
 - Desempenho e Segurança, por exemplo
 - Compatibilidade com outros produtos/versões e necessidades de migração
 - Necessidades de internacionalização do produto
 - Suporte
 - Documentação necessária
 - Aspectos legais
 - Integração com outros produtos
 - etc.

Detalhes sobre a Análise

- A análise gera um modelo para entender o domínio do problema
- Análise também trata em alto nível de como uma solução possível pode ser montada para atender aos requisitos
- Acaba gerando uma especificação, mas sempre do ponto de vista do usuário e tratando apenas do domínio do problema
- Não trata de detalhes de implementação
- Objetos tratados são sempre do domínio do problema (objetos de negócio ou business objects)

Detalhes sobre a Análise

- O modelo é para o cliente e não para o programador
- Atividades típicas durante a análise
 - Refinar use cases
 - Refinar modelo conceitual
 - Refinar glossário
 - Definir diagramas de seqüência (opcional)
 - Definir diagramas de estado (opcional)

Detalhes sobre o design (projeto)

- O projeto é uma extensão do modelo de análise visando sua implementação num computador.
- Novos objetos aparecem, mas não são do domínio do problema.
- O resultado é para o programador ver, não o cliente.
- Objetos da análise são (geralmente) mantidos e são embutidos numa infra-estrutura técnica.

Detalhes sobre o design

- As classes técnicas ajudam os objetos de negócio a, por exemplo:
 - Serem persistentes
 - Estarem distribuídos em diferentes computadores
 - Se apresentarem na interface do usuário
 - Terem desempenho aceitável
 - usando caches ou threads, por exemplo

Detalhes sobre o design

- As atividades de design incluem:
 - Refinamento da arquitetura (high-level design)
 - Definição de pacotes (módulos), interfaces entre pacotes
 - Decisão sobre uso/criação de bibliotecas e/ou frameworks
 - etc.

Detalhes sobre o design

- As atividades de design incluem:
 - Design detalhado
 - Construção/refinamento de diagramas de classes
 - Construção de diagramas de interação (opcional)
 - Levantamento de necessidades de concorrência
 - Considerações de tratamento de falhas
 - Detalhamento do formato de saída (interface com usuário, relatórios, transações enviadas para outros sistemas, ...)
 - Definição do esquema do BD
 - Mapeamento de objetos para tabelas se o BD for relacional

Detalhes sobre a Codificação

- Escrita do código
- Programadores devem normalmente seguir regras de codificação da empresa
- Atividades incluem revisão de código (code reviews)
- Poucos novos diagramas nesta fase

Detalhes sobre os testes

- Inclui várias fases de testes
- Testes feitos pelo próprio programador durante a programação
 - Unit test: teste de classes individuais (ou de grupos de classes relacionadas)
 - Functional test: teste de funções inteiras (item de menu, p. ex.)
- Testes feitos por equipes independentes de teste
 - Teste de integração ou Teste do Sistema: testa a integração entre todos os componentes do software
 - Alpha test: teste do software inteiro dentro de casa
 - Beta test: teste do software inteiro fora de casa
- Testes devem ser automatizados