

Análise e Projeto de Sistemas

Universidade Federal do Ceará – UFC

Campus de Quixadá

Curso de Sistemas de Informação

Prof. Marcos Antonio de Oliveira (deoliveira.ma@gmail.com)

"Sendo um sociólogo, constatei que [o desenvolvimento] de software é um processo social altamente cooperativo." (JORG STRUBING)

O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Esses slides são uma adaptação das notas de aula do professor Eduardo Bezerra autor do livro Princípios de Análise e Projeto de Sistemas com UML

Índice

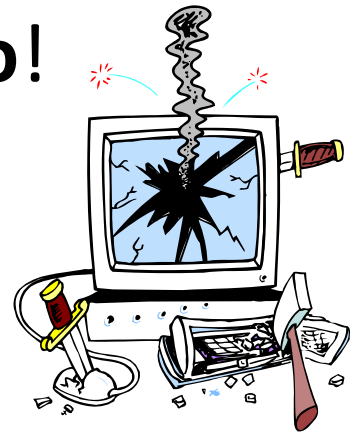
- Introdução
- Atividades típicas do desenvolvimento de software
- O componente humano (participantes do processo)
- Modelos de ciclo de vida
- UML no processo I&I, prototipagem e ferramentas CASE

INTRODUÇÃO

“Software is hard...”

- Porcentagem de projetos que...
 - ...terminam dentro do prazo estimado: 10%
 - ...são descontinuados antes do fim: 25%
 - ...acima do custo esperado: 60%
- Atraso médio nos projetos é de **um ano!**

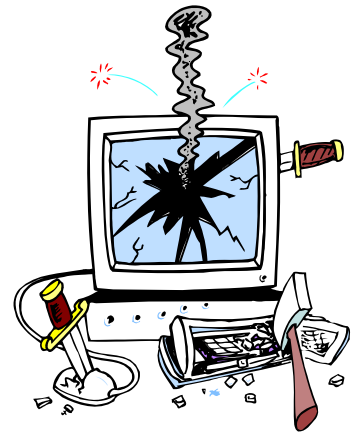
Chaos Report (1994)



“Software is hard...”

- Porcentagem de software...
 - ...pago mas não entregue: 29.7%
 - ...que pode ser usado quando entregue: 2%
 - ...entregue mas nunca usado: 47%
 - ...usado mas posteriormente modificado ou abandonado: 19%
 - ...que podia ser usado após mudanças: 3%

GAO Survey (1992)



O que é Software?

“Programas de computador e documentação associada, tais como requisitos, modelos de projetos e manuais de usuário.”

Lembrem-se, existem artefatos que não “rodam”!

Processo de Desenvolvimento

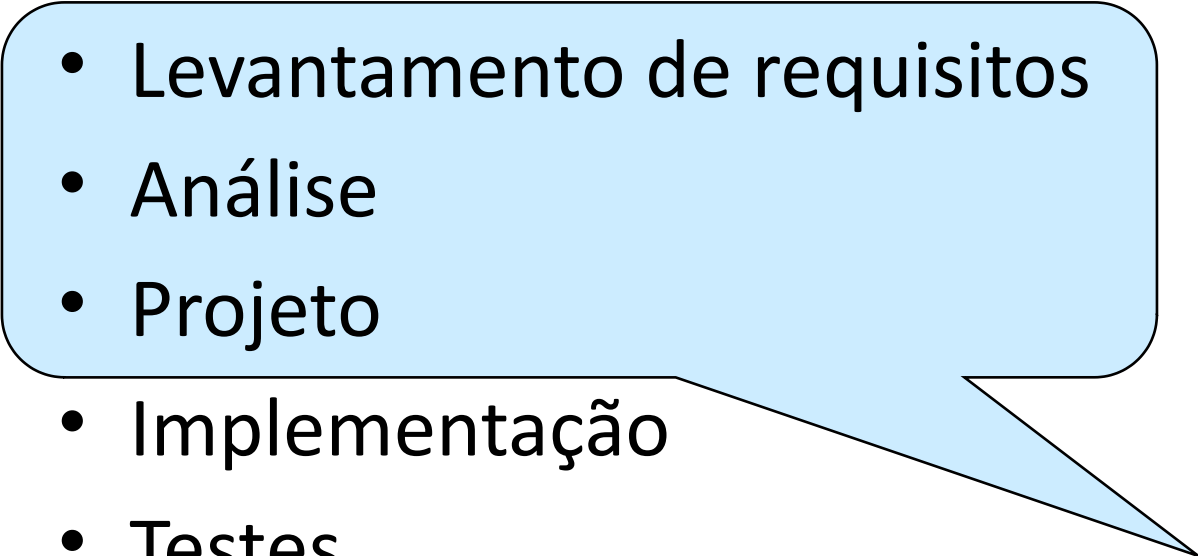
- Compreende as atividades necessárias para definir, desenvolver, testar e manter um produto (sistema) de software
- Tentativas de...
 - ...lidar com a complexidade e minimizar os problemas envolvidos no desenvolvimento de software

Objetivos de um Processo de Desenvolvimento de Software

- Definir ***quais*** as atividades a serem executadas ao longo do projeto
- ***Quando, como*** e por ***quem*** tais atividades serão executadas
- Prover pontos de ***controle*** para verificar o andamento do desenvolvimento
- ***Padronizar*** a forma de desenvolver software em uma organização

ATIVIDADES TÍPICAS DO DESENVOLVIMENTO DE SOFTWARE

Atividades

- 
- Levantamento de requisitos
 - Análise
 - Projeto
 - Implementação
 - Testes
 - Implantação
 - Manutenção

Foco da Disciplina

Levantamento de Requisitos

- Conhecida como *elicitação de requisitos*
- É a etapa relacionada a compreensão do problema aplicado ao desenvolvimento
- O principal objetivo é garantir que o cliente e os desenvolvedores tenha a mesma visão do problema a ser solucionado

Levantamento de Requisitos

- Nessa etapa são levantadas e definidas as necessidades (***requisitos***) dos futuros usuários

“Um ***requisito*** é uma condição ou capacidade que deve ser alcançada ou possuída por um sistema ou componente deste para satisfazer um contrato, especificação ou outros documentos formalmente impostos.” (Maciaszek, 2000)

Levantamento de Requisitos

- Os requisitos
 - ...são identificados a partir de um **domínio**. O entendimento do domínio demanda **especialistas**
 - ...são registrados no **documento de requisitos**
 - ...são a base para a definição do **escopo** (o que faz parte e o que não faz parte do sistema)
- Um documento de requisitos deve possuir a descrição de: requisitos funcionais, requisitos não-funcionais e requisitos normativos

Levantamento de Requisitos

- Requisitos funcionais
 - Definem as funcionalidades do sistema
 - Exemplo
 - “O sistema deve permitir que cada professor realize o lançamento de notas das turmas nos quais lecionou.”
 - “O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas em um semestre letivo.”

Levantamento de Requisitos

- Requisitos não-funcionais
 - Declaram as características de *qualidade* que o sistema deve possuir e que estão relacionadas às suas funcionalidades
 - Exemplos
 - Confiabilidade, desempenho, portabilidade, segurança e usabilidade

Levantamento de Requisitos

- Requisitos normativos
 - Declaram **restrições** impostas ao desenvolvimento do sistema. Essas restrições, dentre outras coisas, definem
 - Adequação a custos e prazos; a plataforma tecnológica; aspectos legais; limitações de interface com usuário; componentes de hardware e software; etc...

Regras de Negócio: São restrições ou políticas de funcionamento específicas do domínio do problema.

Levantamento de Requisitos

- Uma das forma de se medir a qualidade de um software é pela sua utilidade
 - Em geral, um sistema é útil para seus usuários se atender aos requisitos
- Requisitos...
 - ...**devem** ser entendidos para então serem **verificados** e comunicados a leitores **técnicos** e **não-técnicos**
 - ...**não** devem possuir informações técnicas para a resolução do problema
 - ...são **voláteis** (na maioria dos)
 - ...**tem** prioridades uns sobre os outros (*adição de valor*)

Análise

- *Engenharia de Requisitos* = Levantamento de Requisitos + Análise (**de Requisitos**)

“O termo ***análise*** corresponde a “quebra” do sistema em seus componentes e estudar como tais componentes interagem com o objetivo de entender como esse sistema funciona.”

Análise

- Nessa atividade são construídos *modelos* a partir do estudo dos analistas sobre os requisitos levantados
- A atividade de análise
 - ...**não** leva em consideração *soluções tecnológicas* para o sistema
 - ...tem como objetivo entender **o que** o sistema deve fazer ao invés de **como**

Análise

- Modelos construídos no desenvolvimento devem ser *validados e verificados*
 - Validação (análise)
 - Tem como objetivo assegurar que as necessidades do cliente estão sendo atendidas
 - Será que estamos construindo o software certo?
 - Verificação (projeto)
 - Tem como objetivo analisar se os modelos construídos estão em conformidade com os requisitos definidos
 - Será que estamos construindo corretamente o software?

Projeto

- A fase de projeto...
 - ...também conhecida como atividade de **Desenho**
 - ...determina **como** o sistema funcionará para atender aos requisitos levando em consideração restrições tecnológicas (ex: linguagem de programação e SGBD)
 - ...produz uma descrição *computacional* do que o software deve fazer coerente com a descrição feita na análise

Projeto

- A fase de projeto...
 - ... consiste em duas atividades principais
 - Projeto de arquitetura (ou projeto de alto nível)
 - Projeto detalhado (ou projeto de baixo nível)

Projeto

- Projeto de arquitetura
 - Foca no agrupamento das classes relacionadas do sistema em subsistemas e componentes
 - Faz também a distribuição dos subsistemas e componentes sobre os elementos de hardware disponíveis

Projeto

- Projeto detalhado
 - São modeladas as relações entre as classes de cada módulo com o objetivo de realizar as funcionalidades deste módulo
 - São realizados o projeto de interface com usuário, o projeto do banco de dados, avaliação da concorrência e distribuição do sistema
 - O projeto dos algoritmos a serem utilizados no sistema

Demais Atividades

- Implementação
 - Fase responsável pela *codificação* do sistema
 - Em projetos OO usa-se linguagens OO (ex: Java)
- Testes
 - Fase responsável por *verificar e validar* o sistema
 - O principal produto é o relatório de testes que traz informações sobre os erros existentes

Demais Atividades

- Implantação
 - Fase responsável pelo empacotamento, distribuição e instalação no ambiente do usuário
 - Também são escritos os manuais e os funcionários são treinados
- Manutenção
 - Fase responsável por acompanhar a evolução do sistema até a descontinuação

O COMPONENTE HUMANO (PARTICIPANTES DO PROCESSO)

Os Participantes do Processo

- Gerentes de projeto
- Analistas
- Projetistas
- Arquitetos de software
- Programadores
- Especialistas do domínio
- Avaliadores de qualidade



Gerentes de Projeto

- São os profissionais responsáveis
 - ...pela gerencia ou coordenação do projeto
 - ...por fazer o orçamento do projeto
 - Estimar o tempo necessário para o desenvolvimento
 - Definir qual o processo de desenvolvimento
 - Cronograma de atividades
 - Alocação de pessoal
 - Solicitar recursos de hardware e software

Analistas

- São os profissionais responsáveis
 - ...por conhecer o *domínio de negócio*
 - ...por entender os problemas do domínio de negócio para definir os requisitos
 - ...pela comunicação com os especialistas de domínio para obter conhecimento
 - ...entender as necessidades dos clientes e repassar para a equipe (*“a ponte”*)

Projetistas

- São os profissionais responsáveis
 - ...por avaliar as alternativas de solução do problema resultante da análise
 - ...gerar a especificação de uma solução computacional detalhada
- Existem diversos tipos de projetistas
 - Projetistas de interface, de redes, de banco de dados, etc...

Arquitetos de Software

- São os profissionais responsáveis
...por elaborar a arquitetura do sistema como um todo
- Os arquitetos são encontrados em equipes que desenvolvem sistemas **muito complexos**

Programadores

- São os profissionais responsáveis
...pela codificação do sistema!

Analistas estão focados em entender questões ligadas a tecnologia da informação e ao processo de negócio. Já os Programadores se debruçam sobre os aspectos tecnológicos.



Especialistas do Domínio

- São os profissionais responsáveis
 - ...quem possuem o conhecimento a cerca da área ou do negócio em que o sistema em desenvolvimento estará inserido
- Clientes
 - Clientes usuários (especialistas do domínio)
 - Clientes contratantes

Avaliadores de Qualidade

- São os profissionais responsáveis
 - ...por assegurar a adequação do processo de desenvolvimento e do produto de software sendo desenvolvido aos padrões de qualidade estabelecidos pela organização





Como o cliente explicou...



Como o líder de projeto entendeu...



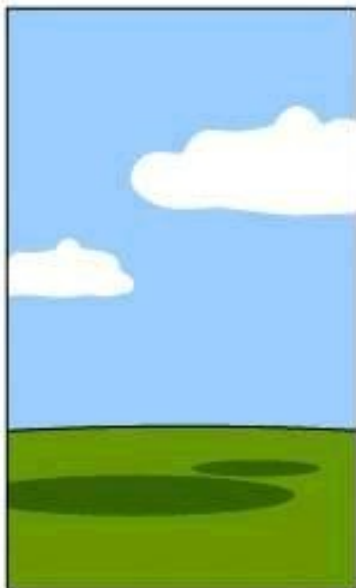
Como o analista projetou...



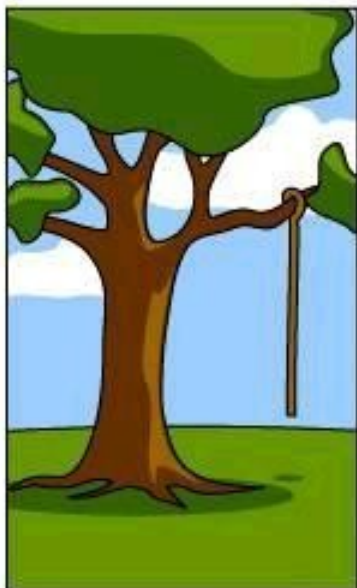
Como o programador construiu...



Como o Consultor de Negócios descreveu...



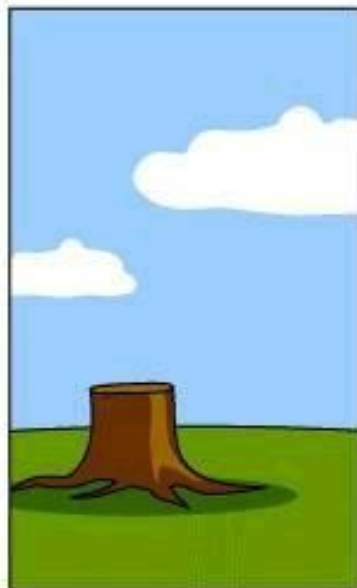
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

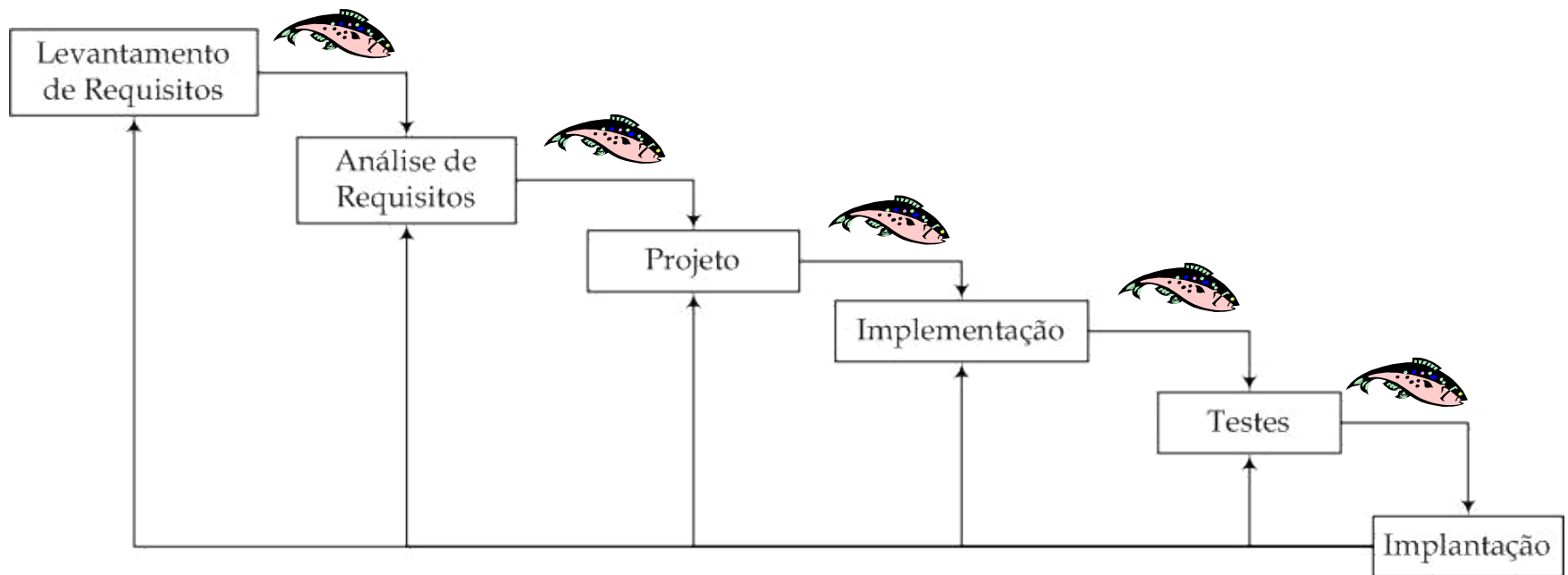
MODELOS DE CICLO DE VIDA

Modelo de Ciclo de Vida

- Um ciclo de vida corresponde a um encadeamento específico das fases para construção de um sistema
- Dois modelos de ciclo de vida
 - *Modelo em cascata*
 - *Modelo iterativo e incremental*

Modelo em Cascata

- Tendência na progressão seqüencial entre uma fase e a seguinte



Modelo em Cascata

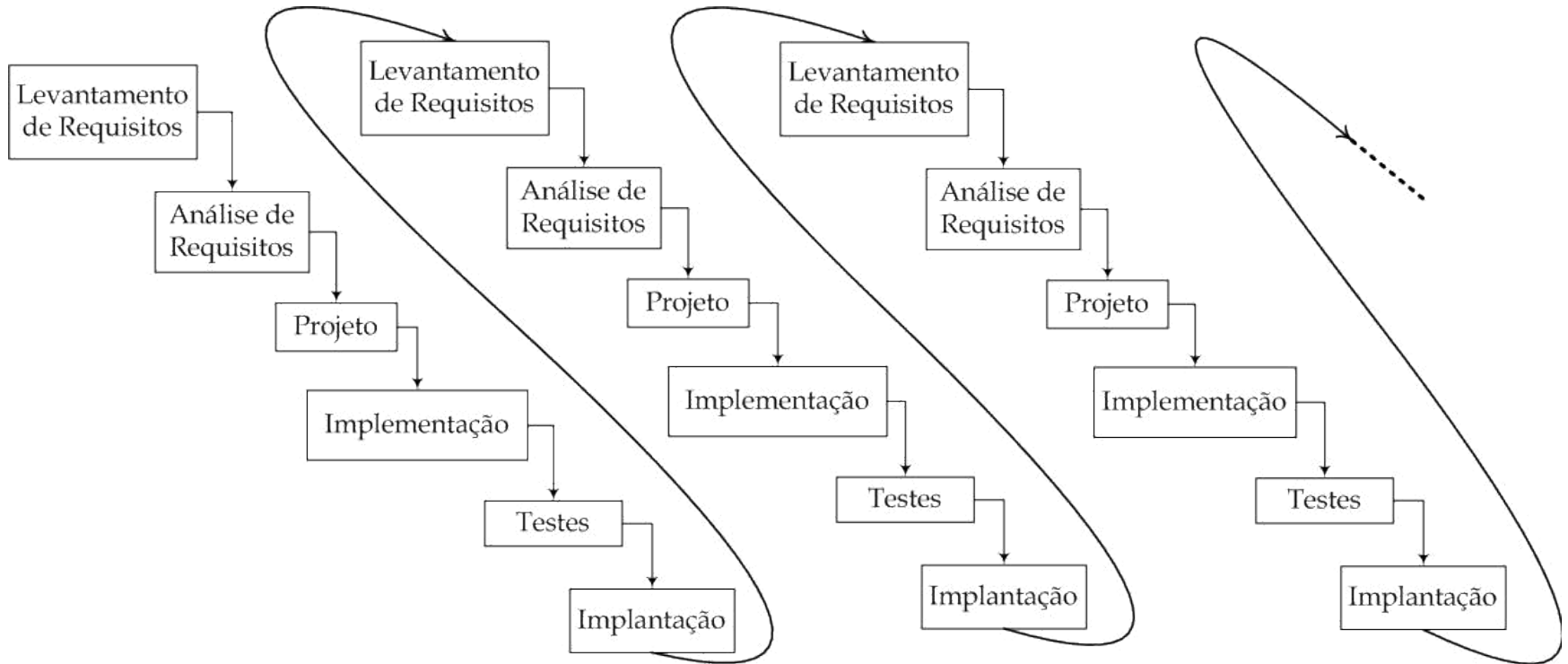
- Projetos reais raramente seguem um fluxo seqüencial
- Assume que é possível declarar detalhadamente todos os requisitos antes do início das demais fases do desenvolvimento
 - Propagação de erros pelas as fases do processo
- Uma versão de produção do sistema não estará pronta até que o ciclo do projeto de desenvolvimento chegue ao final

Modelo Iterativo e Incremental

- Divide o desenvolvimento de um produto de software em *ciclos*
- Em cada ciclo de desenvolvimento, podem ser identificadas as fases de análise, projeto, implementação e testes
- Cada ciclo considera um subconjunto de requisitos
- Esta característica contrasta com a abordagem clássica, na qual as fases são realizadas uma única vez

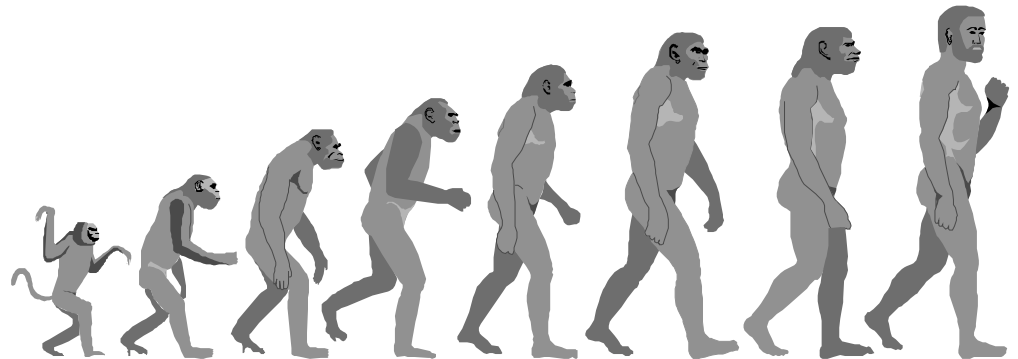
Modelo Iterativo e Incremental

- Desenvolvimento em “mini-cascatas”



Modelo Iterativo e Incremental

- Iterativo
 - O sistema de software é desenvolvido em vários passos similares
- Incremental
 - Em cada passo, o sistema é estendido com mais funcionalidades



Modelo Iterativo e Incremental: Vantagens e Desvantagens

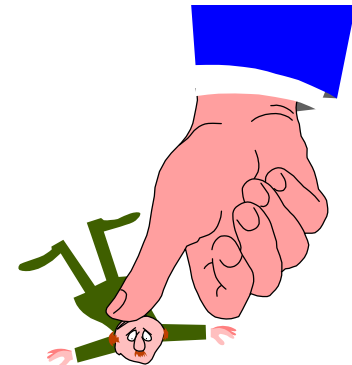
- ✓ Incentiva a participação do usuário
- ✓ Riscos do desenvolvimento podem ser mais bem gerenciados
 - Um risco de projeto é a possibilidade de ocorrência de algum evento que cause prejuízo ao processo de desenvolvimento, juntamente com as conseqüências desse prejuízo.
 - Influências: custos do projeto, cronograma, qualidade do produto, satisfação do cliente, etc.
- ✗ Mais difícil de gerenciar

Ataque aos Riscos

- *“Se você não atacar os riscos ativamente, então estes irão ativamente atacar você.”*
(Tom Gilb)
 - Ou seja, a abordagem incremental e iterativa aconselha que as partes mais arriscadas sejam consideradas inicialmente

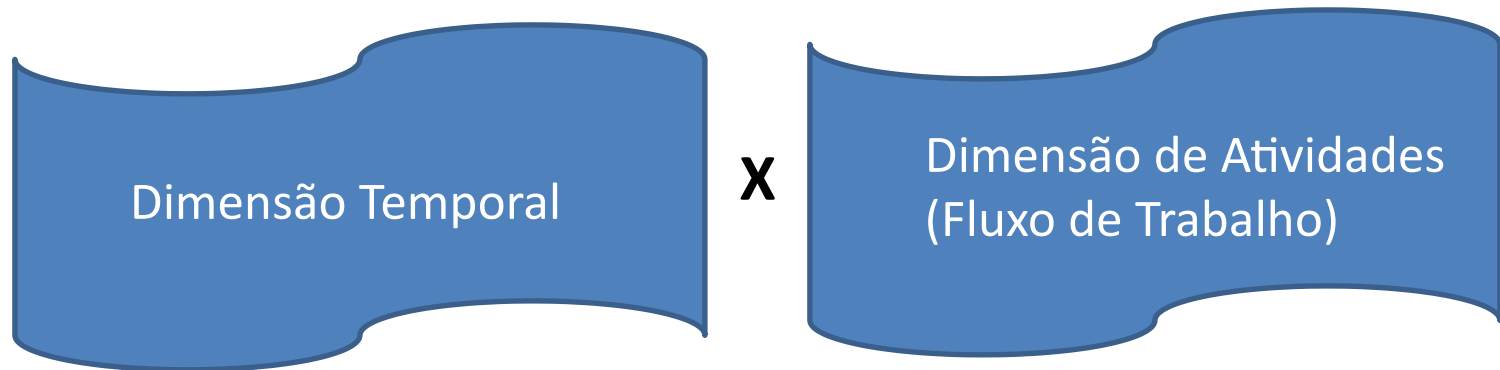


Riscos não Gerenciados



Organização de um Processo Incremental e Iterativo (I&I)

- O ciclo de vida de um processo incremental e iterativo pode ser estudado sob duas dimensões



Organização de um Processo Incremental e Iterativo (I&I)

- Dimensão temporal
 - O processo é estruturado em *fases*
 - Em cada uma das fases, há uma ou mais *iterações*
 - Cada iteração tem uma duração preestabelecida (de duas a seis semanas)
 - Ao final de cada iteração, é produzido um *incremento* (uma parte do sistema final)
 - O incremento pode ser liberado para o cliente ou não

Organização de um Processo Incremental e Iterativo (I&I)

- Dimensão de atividades
 - Compreende às atividades realizadas em cada *fase*
 - Levantamento de requisitos, análise, projeto,...
 - Cada fase gera ou estende (das fases anteriores) um conjunto de *artefatos*
 - Cada fase é concluída com um *marco*
 - Um marco é um ponto de desenvolvimento no qual decisões sobre o projeto são tomadas e importantes objetivos são alcançados

Marcos são úteis para estimar gastos e o andamento do cronograma!

Organização de um Processo Incremental e Iterativo (I&I)

- Fases do Processo Unificado
 - Concepção
 - Elaboração
 - Construção
 - Transição

Organização de um Processo Incremental e Iterativo (I&I)

- Concepção
 - A idéia geral e o escopo do desenvolvimento são definidos
 - Um planejamento de alto nível do desenvolvimento é realizado
 - Os marcos que separam as fases são determinados

Organização de um Processo Incremental e Iterativo (I&I)

- Elaboração
 - O entendimento do sistema deve ser alcançado
 - O planejamento do projeto é completado
 - O domínio do negócio é analisado
 - Os requisitos são ordenados por prioridade e risco
 - As iterações da próxima fase são planejadas
 - Duração da iteração
 - O que será desenvolvido em cada iteração

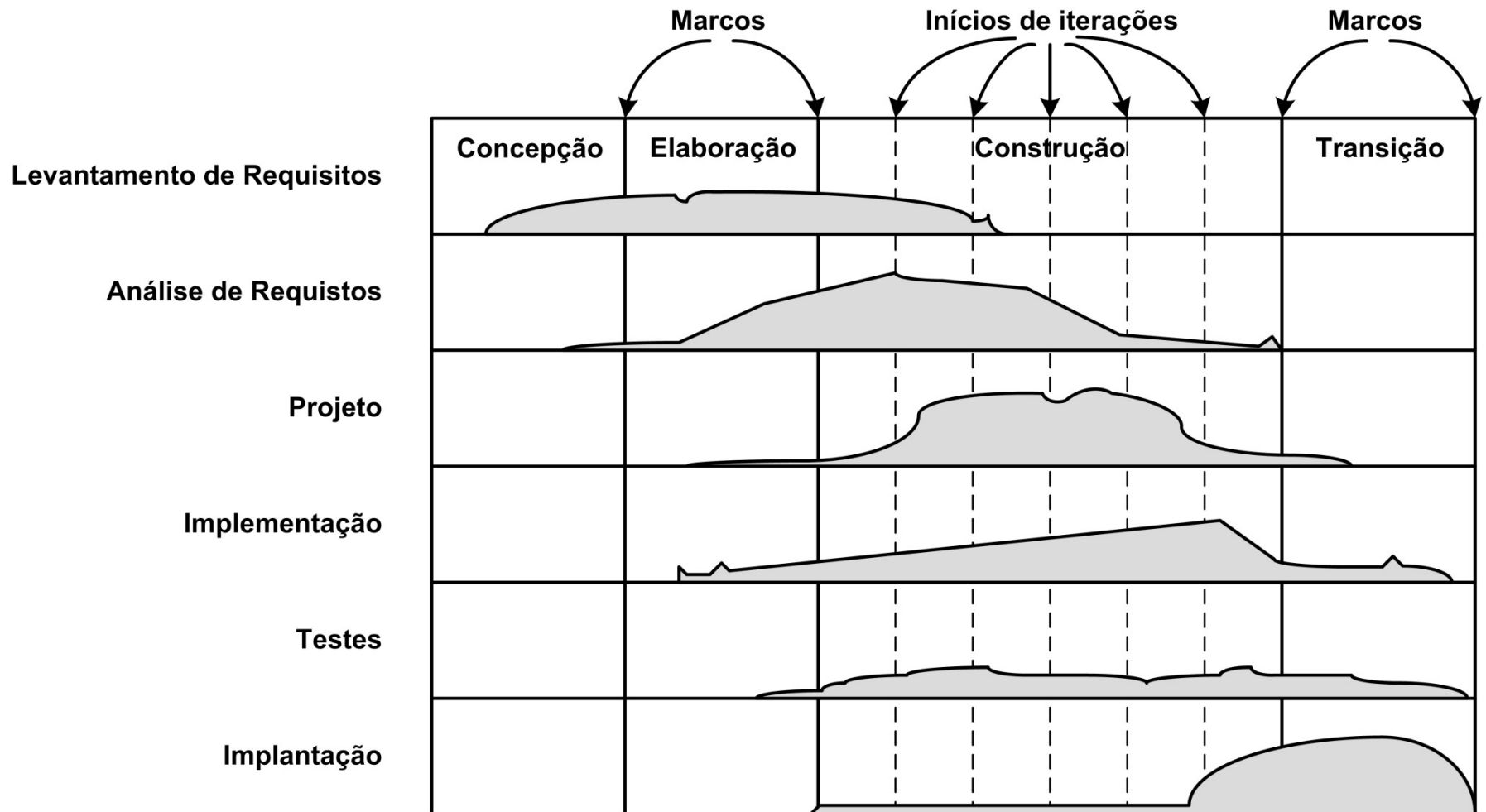
Organização de um Processo Incremental e Iterativo (I&I)

- Construção
 - Atividade de análise e projeto aumentam
 - Ocorre o maior número de incrementos
 - O produto é efetivamente construído
 - A construção do manual do usuário é iniciado

Organização de um Processo Incremental e Iterativo (I&I)

- Transição
 - Os usuários são treinados para utilizar o sistema
 - São tratadas questões de instalação e configuração do sistema
 - O cliente pode pedir “reparos de garantia”
 - Uma nova versão deve ser entregue
 - Um novo ciclo de desenvolvimento pode ser iniciado

Organização de um Processo Incremental e Iterativo (I&I)



UML NO PROCESSO I&I, PROTOTIPAGEM E FERRAMENTAS CASE

Uso da UML em um processo I&I

- A UML é independente do processo de desenvolvimento
 - Vários processos podem utilizar a UML para modelagem de um sistema OO
- Os artefatos de software construídos através da UML evoluem à medida que as iterações são realizadas
 - A cada iteração, novos detalhes são adicionados a esses artefatos
 - Além disso, a construção de um artefato fornece informações para adicionar detalhes a outros

Prototipagem

- Um **protótipo** é um esboço de alguma parte do sistema
- A prototipagem é uma técnica complementar à análise de requisitos que tem como objetivo
 - Assegurar que os requisitos do sistema foram bem entendidos
- A construção de protótipos utiliza ambientes com facilidades para a construção da interface gráfica

Prototipagem

- Técnica freqüentemente aplicada quando
 - Há dificuldades no entendimento dos requisitos do sistema
 - Há requisitos que precisam ser mais bem entendidos
- Após o Levantamento de Requisitos, um protótipo é construído para ser usado na validação

Prototipagem

- Com o protótipo feito
 - Usuários fazem críticas...
 - O protótipo é então corrigido ou refinado...
 - O processo de revisão e refinamento continua até que o protótipo seja aceito pelos usuários...
 - Após a aceitação, o protótipo é descartado ou utilizado como uma versão inicial do sistema

Prototipagem

- A prototipagem NÃO é um substituto à construção de modelos do sistema
 - É uma técnica **complementar** à construção dos modelos do sistema
 - Mesmo com o uso de protótipos, os modelos do sistema devem ser construídos
 - Os erros detectados na validação do protótipo devem ser utilizados para modificar e refinar os modelos do sistema

Ferramentas de Suporte

- O desenvolvimento de um software pode ser facilitado através do uso de ferramentas que auxiliam
 - Na construção de modelos
 - Na integração do trabalho de cada membro da equipe
 - No gerenciamento do andamento do desenvolvimento
- (...)

Ferramentas de Suporte

- Sistemas de software que são utilizados para dar suporte ao desenvolvimento são normalmente chamados de **Ferramentas CASE (Computer Aided Software Engineering)**
- Além das ferramentas CASE, outras ferramentas importantes são as que fornecem suporte ao gerenciamento
 - Desenvolver cronogramas de tarefas,
 - Definir alocações de verbas,
 - Monitorar o progresso e os gastos,
 - Gerar relatórios de gerenciamento(...)

Funcionalidades Desejáveis em Ferramentas de Suporte

- Criação e manutenção da consistência entre diagramas
- *Round-trip engineering*
- Depuração de código fonte
- Relatórios de testes
- Testes automáticos
- Gerenciamento de versões
- Verificação de desempenho
- Verificação de erros em tempo de execução
- Gerenciamento de mudanças nos requisitos

Algumas Ferramentas de Suporte



Exercícios

- Exercícios do livro texto
 - Página 45: 2.1, 2.2, 2.3, 2.4 e 2.5

Referências

- BEZERRA, E. Princípios de Análise e Projeto de Sistemas com UML. 2ª ed. Rio de Janeiro: Elsevier, 2007.
- FOWLER, M. 3. UML Essencial. 3. ed. Porto Alegre: Bookman, 2007.
- PRESSMAN, R. Engenharia de Software. 6ª ed. São Paulo: Mc Graw-Hill, 2007.
- SOMMERVILLE, I. Engenharia de Software. 7ª Ed. São Paulo: Addison-Wesley, 2007.