



CEFET/RJ – Uned Nova Friburgo
Introdução a Engenharia de SW

INTRODUÇÃO A ENGENHARIA DE SOFTWARE



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Apresentação Pessoal:

- Dacy Câmara Lobosco;
- Email: dacy.lobosco@cefet-rj.br;
- Graduação em Informática na Estácio Friburgo;
- Pós-Graduação em Análise de Sistema e Gerência de Projetos na PUC-RJ;
- Mestrado na UERJ – Modelagem Computacional;
- 15 anos de experiência em desenvolvimento de sistemas WEB.

Minha Experiência em C++ :

- Graduação
- Mestrado: em implementações de trabalhos Acadêmicos.
- Ensino de métodos estocásticos em C++ no mestrado.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- O que é Software ?



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- O que é Software?
 - Programas de computador
 - Entidade abstrata.
 - Ferramentas (mecanismos) pelas quais:
 - exploramos os recursos do hardware.
 - executamos determinadas tarefas.
 - resolvemos problemas.
 - interagimos com a máquina.
 - tornamos o computador operacional e útil.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

O que é software?

Conceito mais amplo que inclui também:

- Instruções que executam uma função desejada.
- Estrutura de dados para manipular informação.
- Documentos para desenvolver, operar e manter os programas.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

O que é software?

Software como um produto!

Incorporado em todas as atividades humanas cotidianas:
cultura, comércio, lazer, etc.



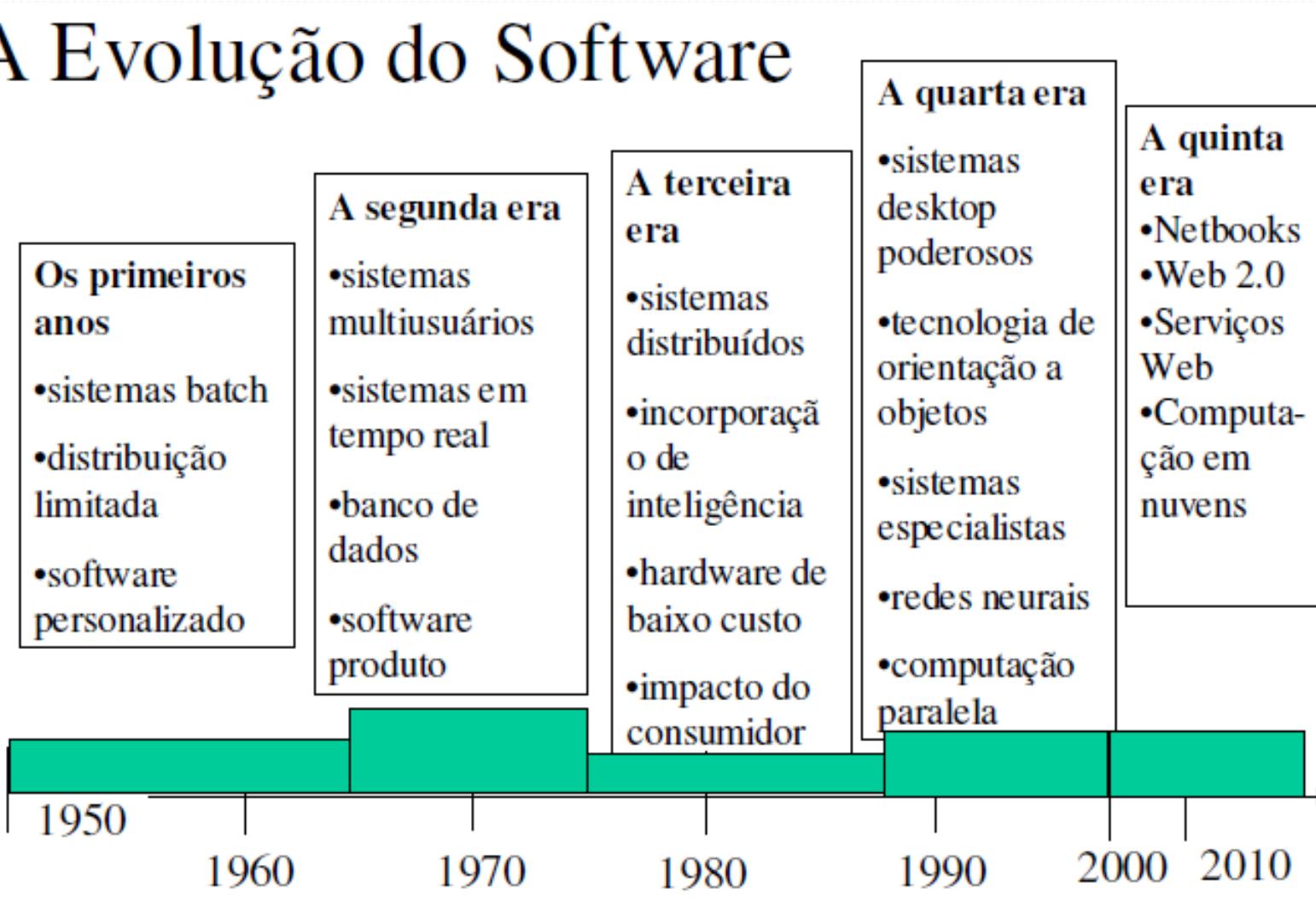
CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Tipos de Sistemas de Software:

- Software básico
- Software stand-alone – aplicativos para PC que não precisam estar conectados à rede.
- Software para sistema em tempo real
- Software baseado em transações: comércio eletrônico
- Software comercial e sistemas financeiros
- Software para engenharia e aplicações científicas
- Software embarcado (ex. microwave, celulares)
- Software baseado em inteligência artificial
- Software de entretenimento: jogos, cinema

A Evolução do Software





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Dificuldades para desenvolver software

- Saber o que o software deve fazer : quais os requisitos;
- Ferramentas; linguagem; SO – tecnologias diferentes
- Tempo e custos elevados de desenvolvimento.
- Prever falhas (antes de entregar).
- Tratar manutenção e versões.
- Produtividade não cresce com a demanda de serviços.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Falso ou verdadeiro?

- Já temos um manual repleto de padrões e procedimentos para a construção de sw. Neles está tudo o que a equipe precisa saber para produzir com qualidade.
- Meu pessoal tem ferramentas de última geração, isso é suficiente.
- Se nós estamos atrasados nos prazos podemos adicionar mais programadores e isso irá resolver o problema com o cronograma.
- Podemos terceirizar o projeto de software. Assim a responsabilidade será da empresa contratada.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Falso ou verdadeiro?

- Assim que colocarmos o programa em funcionamento posso ficar tranquilo porque meu trabalho estará finalizado.
- A engenharia de software irá nos fazer criar documentação volumosa e desnecessária e, invariavelmente nos retardar.
- Só poderemos avaliar a qualidade do software quando tivermos o programa funcionando.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Falso ou verdadeiro?

- Uma definição geral dos objetivos é suficiente para começar a escrever programas, podemos adicionar detalhes depois.
- Os requisitos de software mudam constantemente mas mudanças podem ser facilmente assimiladas pois o software é flexível.
- Eu, cliente, sei o que preciso.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Mitos da Engenharia de Software

Mitos atuais:

- O teste de sw ou sua verificação formal pode remover todos os erros.
- Para garantir qualidade é só aumentar a confiabilidade do software.
- O reúso de componentes prontos me traz segurança com relação à qualidade.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Crise de Software
problemas para desenvolver software:

- estimativas de prazo e de custo - Alto
- produtividade das pessoas - Baixo
- qualidade de software - Baixo
- software difícil de manter - Alto



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Problemas no Desenvolvimento do Software:

- Software inadequado.
- Cronogramas e custos imprecisos - dificuldades em prever o progresso durante o desenvolvimento.
- Inexistência de dados históricos sobre o processo de desenvolvimento.
- Comunicação deficiente - insatisfação de usuários.
- Carência de conceitos quantitativos sobre confiabilidade, qualidade, reusabilidade.
- Software existente é de difícil manutenção.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Crise de Software

Solução:

- Combinar métodos para as fases de desenvolvimento.
- Ferramentas para automatizar esses métodos.
- Técnicas para assegurar qualidade.
- => Disciplina: Engenharia de Software.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Software

- Produto projetado e construído pelos Engenheiros de Software.
- Abrangem programas de computador, documentos (tangíveis e eletrônicos), dados e modelos (representação de conceitos através de figuras).



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Engenharia de Software





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Engenharia de Software

- Processo definido e gerenciado de forma sistemática em todas as fases: Análise, desenvolvimento, operação, e manutenção do software.
- Confiança e desempenho, o software deve se comportar conforme o esperado, sem falhas e estar sempre disponível quando requisitado.
- Gerenciar a especificação dos requisitos, para produzir e manter softwares úteis ao cliente dentro de prazos e custos estimados.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Ética na Engenharia de Software

- Confidencialidade: você deve respeitar naturalmente a confidencialidade das informações dos seus empregadores e clientes, independente de ter sido assinado um acordo formal
- Competência: Você não deve aceitar um trabalho que está fora da sua competência.
- Direitos de Propriedade Intelectual: Você deve ter conhecimento das leis locais a respeito da propriedade intelectual, como patentes e copyrights.
- Mau uso do Computador: Você não deve fazer mau uso das habilidades técnicas para utilizar computadores de outras pessoas sem o seu conhecimento, como disseminar vírus, ataques, jogar games na conta de outro usuário, etc..



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Engenharia de Software

Alguns Conceitos:

- Formalidade: Documentação reduz inconsistências e erros de comunicação.
- Abstração (o que): Visão do todo, especificação dos requisitos.
- Decomposição (como): lidar com a complexidade da abstração, visão das partes.
- Generalização: reutilização, diminuir custos de desenvolvimento.
- Flexibilização: Aberto a mudanças, fácil de adaptar.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo de Software



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo (Aurélio):
- ...
- Maneira pela qual se realiza uma operação, segundo determinadas normas;
- Método, técnica;
- Sequência de estados de um sistema que se transforma;
- Evolução;
- ...



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software

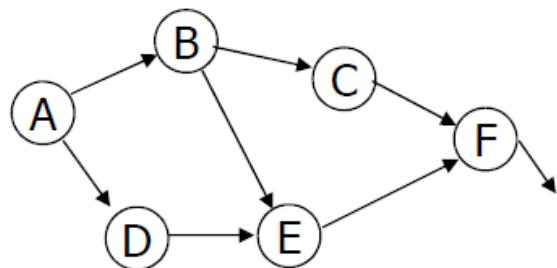
- Um conjunto estruturado de atividades requeridas para desenvolver um produto de software
- Envolve: fases, atividades, pessoas (responsáveis, participantes, papéis), entradas e saídas (artefatos), recursos, passos para execução, regras, procedimentos.

CEFET/RJ – Uned Nova Friburgo

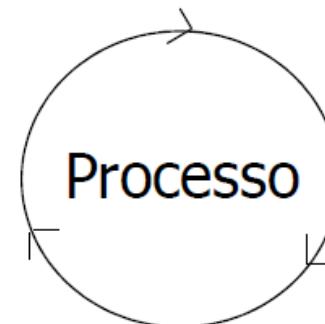
Introdução a Engenharia de SW

Processos envolvem:

Pessoas com habilidades, treinamento e motivação



Procedimentos e métodos definindo o relacionamento entre as tarefas



Ferramentas e equipamentos adequados



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software

- Fases: grandes divisões (marcos) dos processos, períodos de tempo no qual algumas atividades são executadas.
- Atividades: possui entradas e saídas bem definidas, pessoas (responsáveis, participantes, papéis), entradas e saídas.
- Artefatos: documentos, programas, uma mudança. Algo mensurável como saída de uma atividade.
- Recursos: Pessoas, hardware, software, material consumível ou não.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software

- Existem muitos tipos de processos, mas todos devem contemplar as seguintes atividades:
 - Especificação do Software: Definição das funcionalidades do software e as restrições e premissas do seu funcionamento.
 - Projeto e implementação do Software: O projeto precisa ser produzido de acordo com as especificações.
 - Validação do Software
 - Evolução e/ou Manutenção do Software



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Especificação do Software
 - Atividades Principais:
 - Estudo de Viabilidade
 - Elicitação e análise de Requisitos
 - Especificação de requisitos
 - Validação de Requisitos
 - Documentação de Requisitos



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Projeto e implementação do Software
 - Principais Atividades de Projeto
 - Definição do Projeto de Arquitetura;
 - Definição do Projeto de interface;
 - Definição dos Componentes do Sistema;
 - Definição do Projeto de Banco de Dados;
 - Principais Atividades de Implementação
 - Definição das classes (atributos e métodos);
 - Desenvolvimento das interfaces;



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Validação do Software
 - Principais Atividades:
 - Teste de Desenvolvimento: Os componentes e Classes são testados pelos programadores que desenvolveram.
 - Teste de Sistema: Testes entre as integrações das camadas e módulos do sistema, ou seja, testa-se as trocas de mensagens entre objetos desde a interface até as classes de persistência de uma determinada funcionalidade.
 - Teste de aceitação: O sistema é testado com um todo com dados fornecidos pelo cliente. Neste teste pode-se aferir se os requisitos foram completamente atendidos e simular dados do universo do cliente, não viciados pelo programador.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Evolução e Manutenção do Software
 - Principais Atividades:
 - Analisar e avaliar novas tecnologias;
 - Alterar requisitos existente ou definir novos requisitos para o Sistema;
 - Propor mudanças no sistema;
 - Avaliar impactos da mudança;
 - Modificar Sistema existente;



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

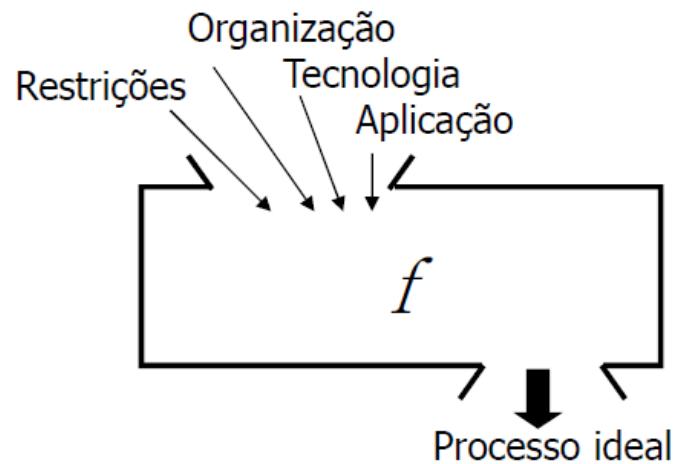
Processo de Software:

- Assim como nos métodos de Engenharia, processos de software se baseiam:
 - Em ações sistemáticas;
 - Na não improvisação.

Processo de Software

Nenhum processo é perfeito para:

- Qualquer organização;
- Qualquer tecnologia;
- Qualquer aplicação.





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software

- Processos se modificam:
 - Devem ser complementados e personalizados para passarem à prática nas organizações produtoras de software;
 - Devem ser adicionados padrões que cubram aspectos específicos:
 - das aplicações;
 - da tecnologia;
 - dos métodos gerenciais;
 - da cultura da organização.
- Processos evoluem:
 - Pode-se também aplicar as lições aprendidas a um processo (mesmo um processo comercial).
 - Modelos de processos se mesclam.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

- Personalizar um processo de desenvolvimento, a partir de um processo padrão, significa, portanto, adaptá-lo afim de se considerar:
 - as particularidades da organização,
 - as particularidades da equipe de desenvolvimento
 - as características funcionais e não funcionais do produto a ser gerado.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

- Mesmo os processos comerciais requerem esforço considerável de adaptação;
- O time de Engenharia de Processos de uma organização trata as questões relativas aos processos da organização.
 - Padroniza, desenvolve, discute, documenta,
 - divulga, afere, adapta, ...



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

- Processos (de uma maneira geral) evoluem conforme o grau de maturidade das organizações;
- Processos de software evoluem conforme o grau de maturidade com que as organizações gerenciam os artefatos associados ao software e o grau de maturidade com que as organizações gerenciam as diversas atividades desenvolvidas durante o ciclo de vida do software.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

- Como dissemos, processos evoluem.
- A evolução pode ser medida de acordo com vários critérios (métricas).
 - Duas delas:
 - SW-CMM (*)
 - Mps.BR (**)

(*) CMM: Modelo de Maturidade da Capacidade

(**) Melhoria de Processo do S/W Brasileiro



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

No SW-CMM:

- A maturidade é medida em níveis;
- Um nível de maturidade é um patamar definido de evolução de processo;
- Os níveis de maturidade estabelecem o estágio atual e as etapas necessárias para melhoria dos processos de software



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

No SW-CMM:

- Os níveis são patamares bem definidos conduzem a processos mais maduros de software.
- Cada patamar compreende um conjunto de objetivos e comprometimentos (KPAs – Key Process Areas) que devem ser satisfeitos.



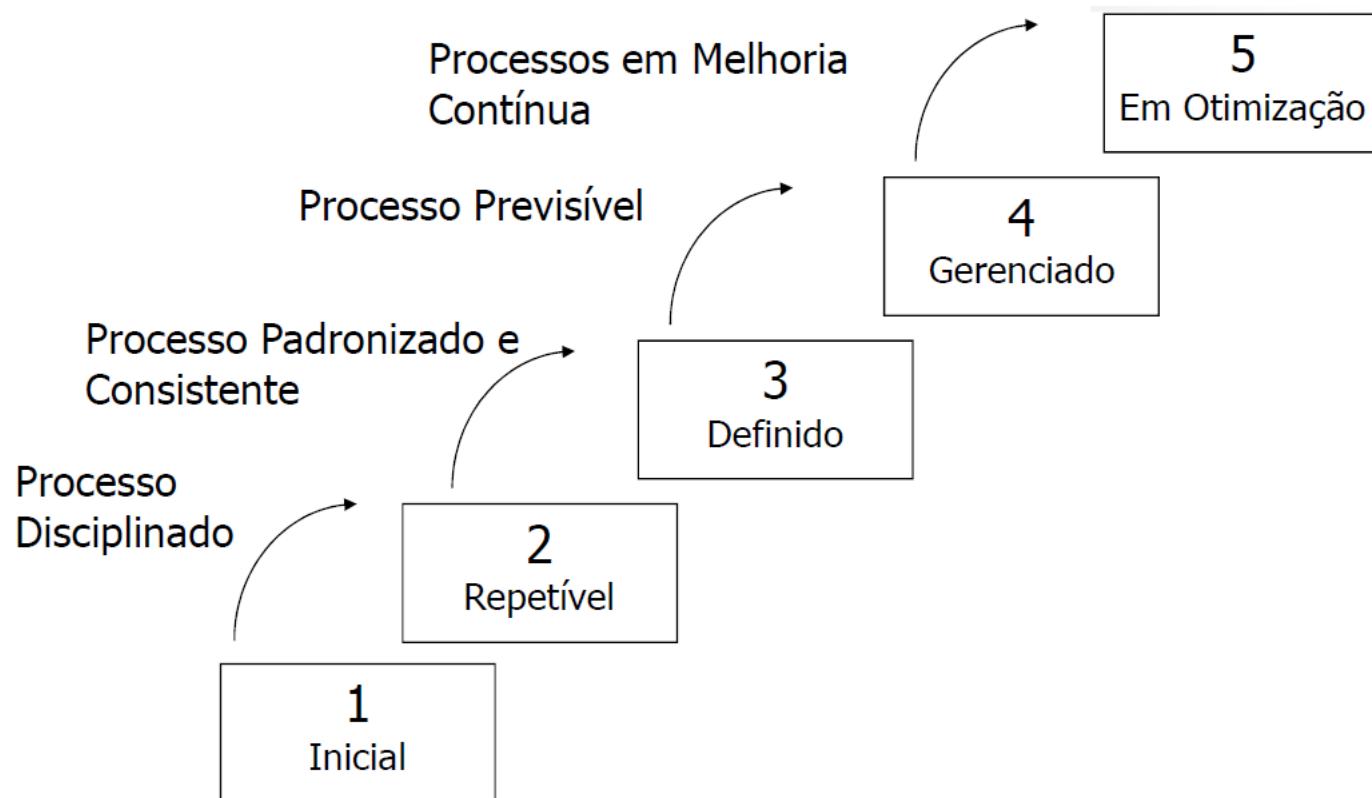
CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

- Organização imatura: o processo de software é improvisado. Mesmo que o processo seja especificado, ele não é seguido. São organizações reacionárias.
- Organização madura: possui capacidade organizada para gerenciar o desenvolvimento e manutenção de software.

Processo de Software: No SW-CMM:





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

No SW-CMM:

- Níveis:

- 1. Inicial:**

- Processo ad-hoc, ocasional e até caótico. Poucos ou nenhum processo está definido. Sucesso depende do esforço individual.

- 2. Repetível:**

- Os processos básicos de gerência estão estabelecidos para acompanhamento de custo, prazos e funcionalidade. Existe a capacidade de repetir processos bem sucedidos em projetos anteriores.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

No SW-CMM:

- Níveis:

3. Definido:

- O processo de software para as atividades de gerência e engenharia estão documentados, padronizados e integrados no processo de desenvolvimento de software da organização.
- Todos os projetos usam uma versão documentada e aprovada do processo da organização para desenvolvimento e manutenção. Inclui o nível 2.

4. Gerenciado:

- Os processos básicos de gerência estão estabelecidos para acompanhamento de custo, prazos e funcionalidade. Existe a capacidade de repetir processos bem sucedidos em projetos anteriores. Medições detalhadas do processo e do produto são coletadas. Ambos são quantitativamente compreendidos e controlados através de métricas minuciosas. Inclui o nível 3.

5. Em Otimização (ou Otimizante):

- Um processo contínuo de melhoria baseado nos resultados quantitativos de outros projetos e em testes de novas idéias e tecnologias está estabelecido. Inclui o nível 4.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Processo de Software:

Hacking (codifica-remenda):

- Prática frequente para o desenvolvimento de s/w;
- “Atropela” as diversas fases para a solução do problema;
- Consiste em sentar-se em frente ao computador e começar a escrever código, sem nenhum projeto.
- Não é um processo de software... Muito menos um modelo de processo.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelos de Processos:
 - Cascata (Clássico)
 - Prototipação
 - RAD
 - Espiral (incremental)



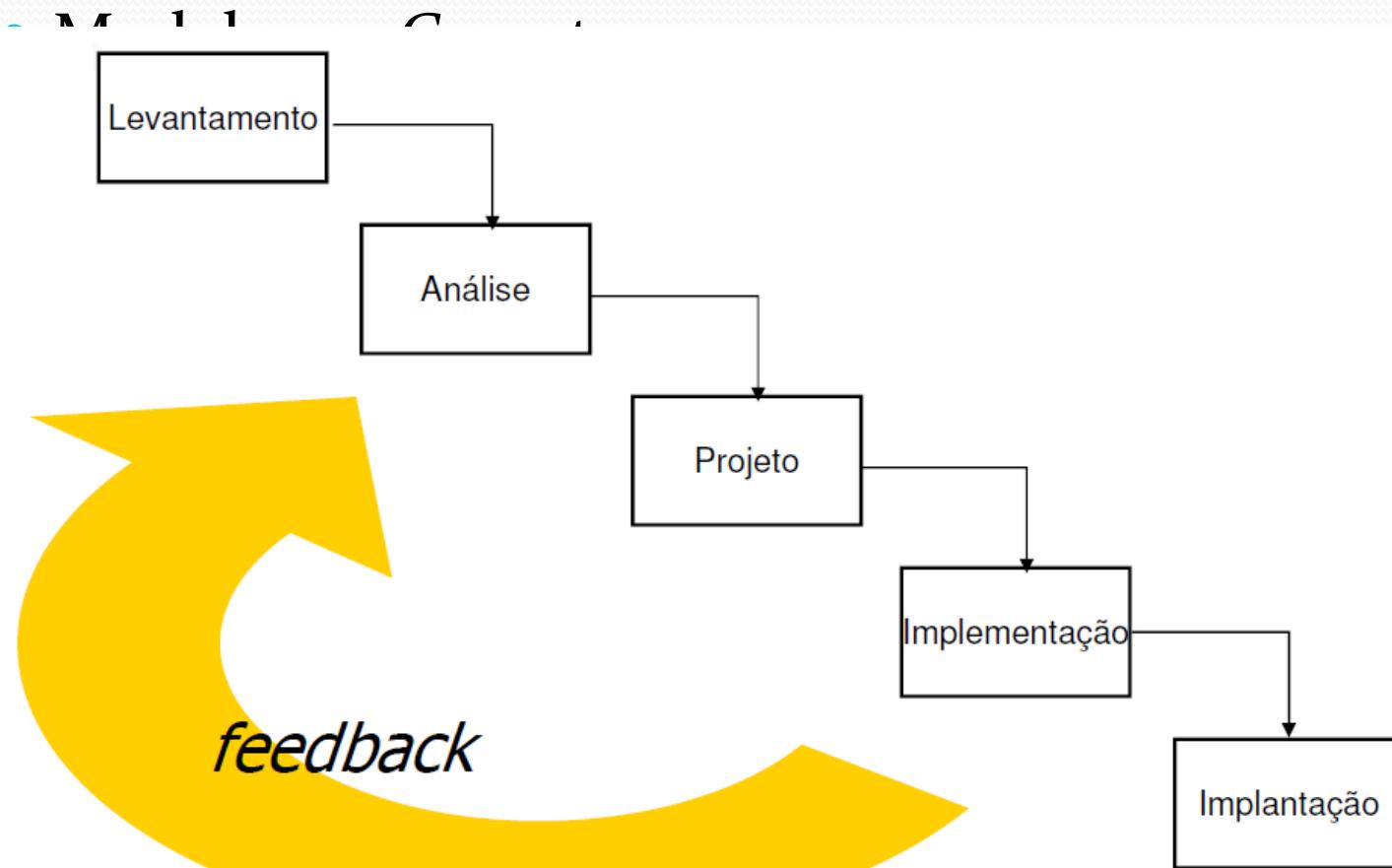
CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo em Cascata:
 - Modelo linear e sequencial;
 - Pode usar feedback ou não;
 - Possui, em geral, um ciclo longo de desenvolvimento;
 - Aplicável no desenvolvimento de sistemas pouco susceptíveis a mudanças de requisitos e em organizações estáveis;
 - Serve como base para outros modelos de processo.

CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW



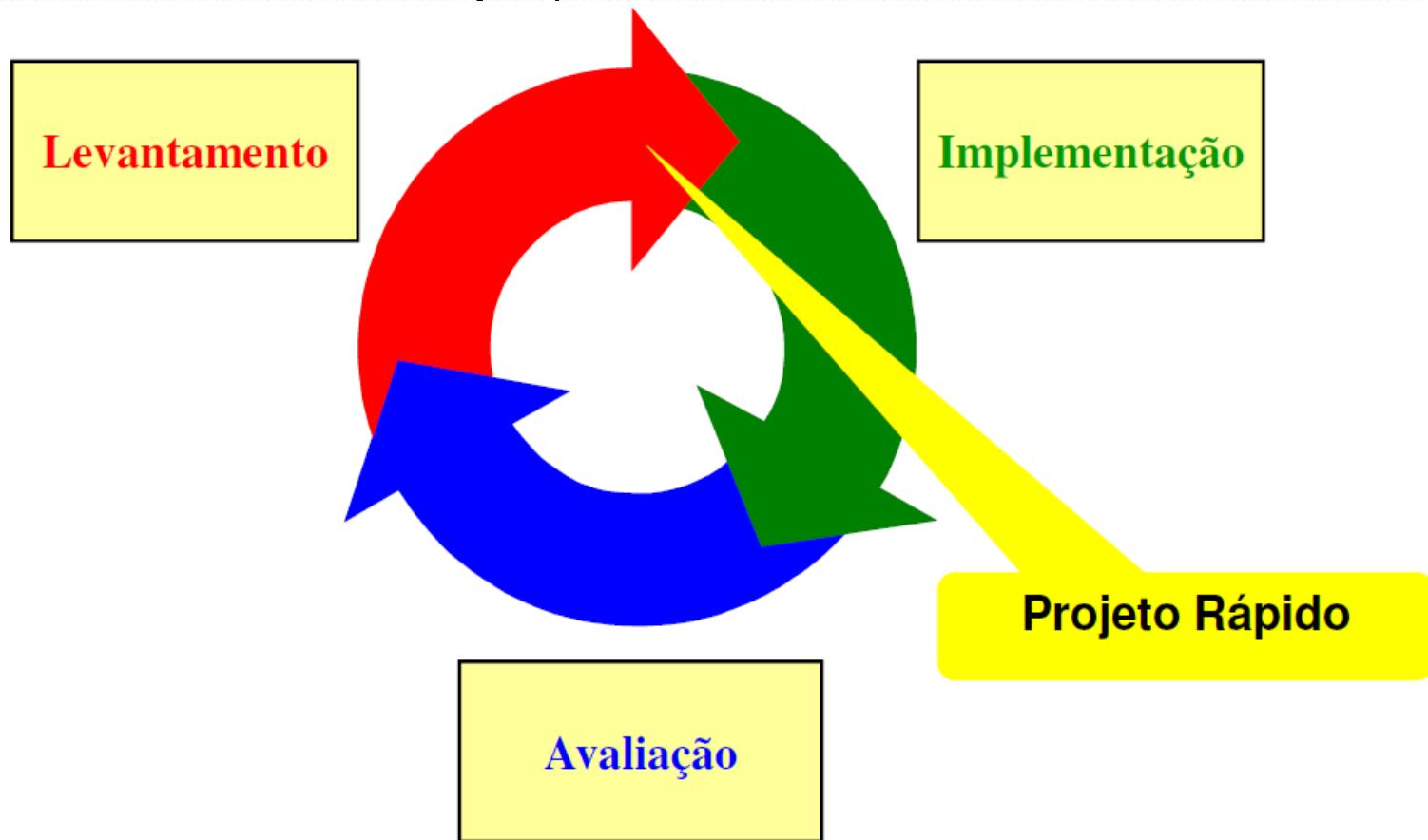


CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo de Prototipação:
 - Construção de protótipos baseado nas informações do cliente;
 - Adequado para quando o “negócio” não é bem conhecido ou o cliente não sabe exatamente do que precisa;
 - Idealmente serve p/ identificar requisitos;
 - Protótipo = “Sistema” (alguns autores recomendam que o joguemos fora).

- Modelo de Prototipação





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo de Prototipação:
 - Problemas:
 - O usuário não entende que o software desenvolvido sacrifica a qualidade para obter velocidade no desenvolvimento e não pode ser considerado como um produto que possa entrar em produção.
 - O desenvolvedor muitas vezes toma decisões ineficientes de projeto, para facilitar o desenvolvimento, e acaba se acostumando com tais decisões, esquecendo o motivo que o levou a tomá-las a versão em produção permanece com “bolsões” de baixa qualidade.



CEFET/RJ – Uned Nova Friburgo

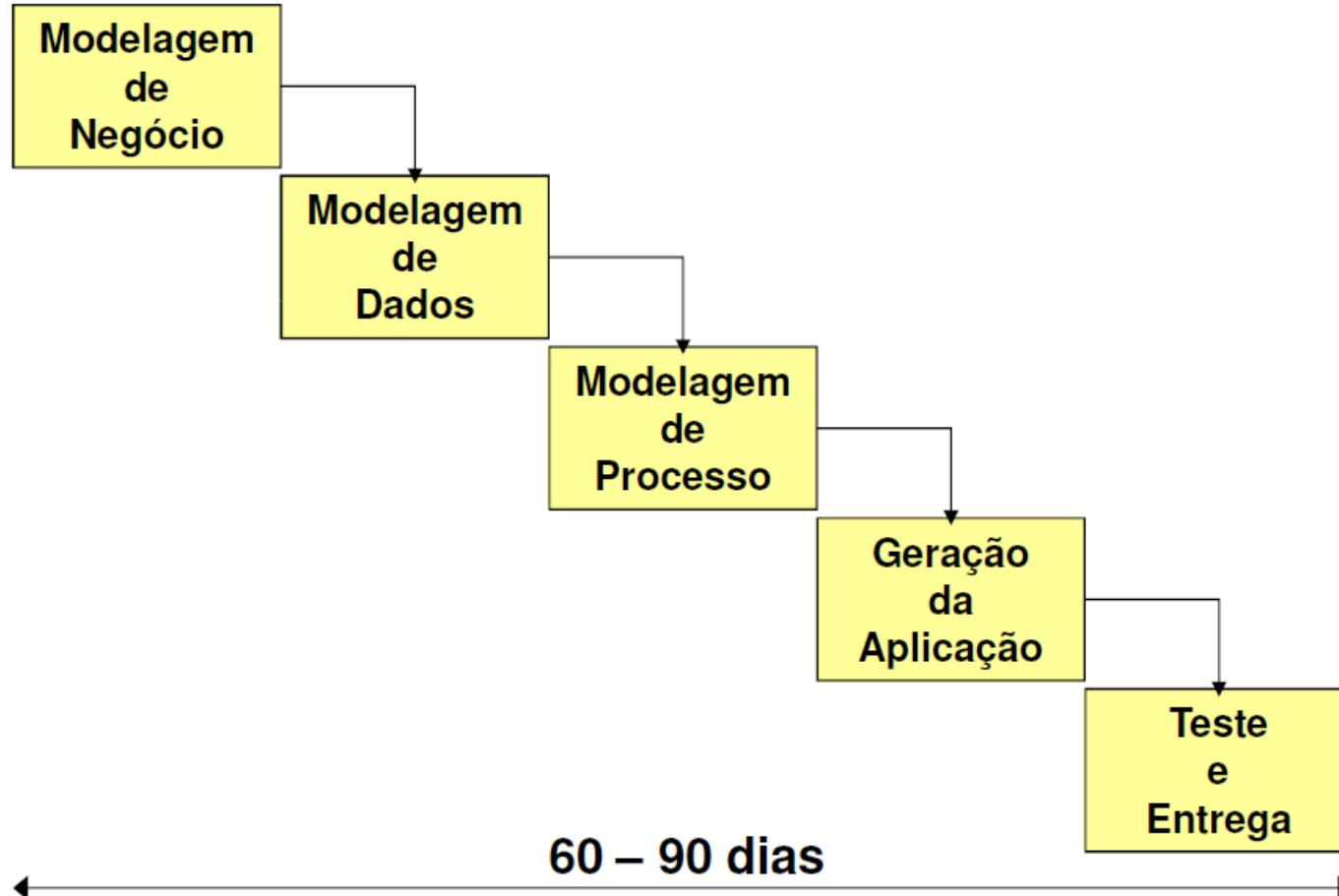
Introdução a Engenharia de SW

- Modelo RAD = Rapid Application Development
 - Como o modelo cascata com um ciclo de desenvolvimento muito curto;
 - Uso de componentes para acelerar a construção;
 - Razoável para projetos de escopo controlado e reduzido;
 - Aplicado a grandes projetos, se for possível sua divisão em módulos independentes;
 - Pouco disciplinado;
 - Normalmente utilizado em desenvolvimento de sistemas de informação (escopo controlado).

CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo RAD





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo RAD
 - Problemas:
 - Em projetos grandes, que podem ser divididos em subprojetos onde o RAD pode ser aplicado, demanda recursos humanos de forma que se permita montar uma série de equipes (para desenvolvimento em paralelo);
 - Usuários e desenvolvedores devem estar comprometidos a tomar decisões e agir rapidamente;
 - Inadequado quando se precisa de grande performance (necessitando, p. ex., de algoritmos sofisticados/complexos) ou quando existem riscos técnicos.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo Espiral
 - Inicialmente publicado por Barry Boehm(*);
 - Reduz sensivelmente o risco de insucesso de um projeto;
 - Permite uma maior interação com o cliente;
 - Adequado à maioria dos tipos de projetos/sistemas;

(*) "A Spiral Model for Software Development and Enhancement", 1988



CEFET/RJ – Uned Nova Friburgo

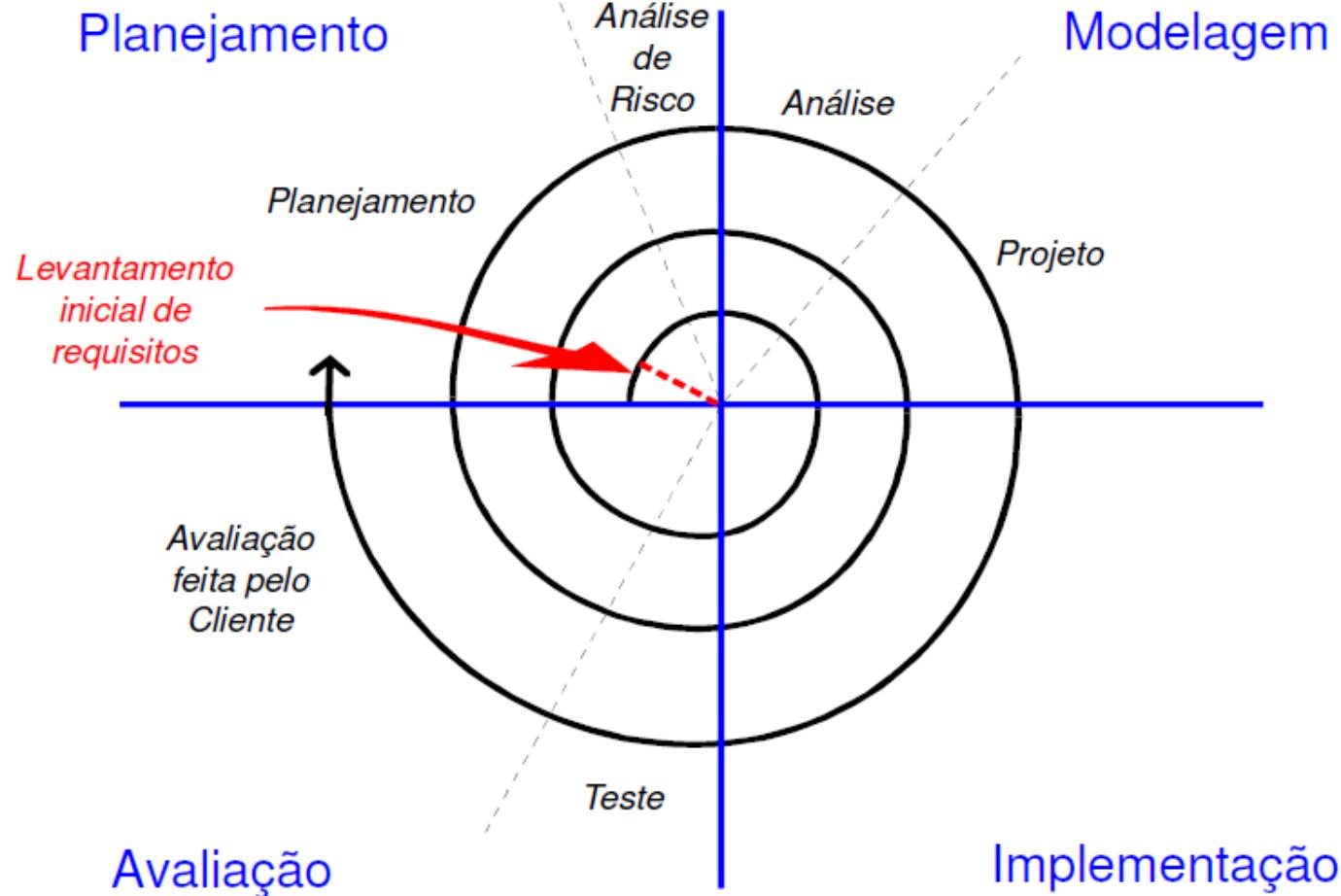
Introdução a Engenharia de SW

- Modelo Espiral
 - Incremental, evolutivo, iterativo;
 - Espiral é dividida em uma série de regiões;
 - Cada região contempla uma série de tarefas que são adaptadas às características do projeto a ser conduzido;
 - Um ciclo da espiral pode produzir, tanto uma especificação, como versões do software;
 - Pode ser adaptado para toda a vida do software.

CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Modelo Espiral**





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo Espiral

- Vantagens:
 - Adaptabilidade a mudanças de requisitos
 - (cuidado com a convergência para uma solução final, no Tempo e Custos definidos);
 - Redução dos riscos;
 - Acúmulo gradativo de experiência;
 - Permite a homogeneização da equipe.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo Espiral:
 - Problemas:
 - Dificuldade para convencer o cliente de que o processo evolucionário pode ser controlável;
 - Depende da capacidade de análise de riscos;
 - Requer gestão sofisticada para ser previsível e confiável;



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo RUP
 - RUP = Rational Unified Process (Processo Unificado da Rational);
 - Oferece uma abordagem por disciplinas;
 - Atribui tarefas e responsabilidades aos integrantes da equipe de desenvolvimento de software.
 - é uma base de conhecimento, contendo orientações para o processo de desenvolvimento;



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

● Modelo RUP

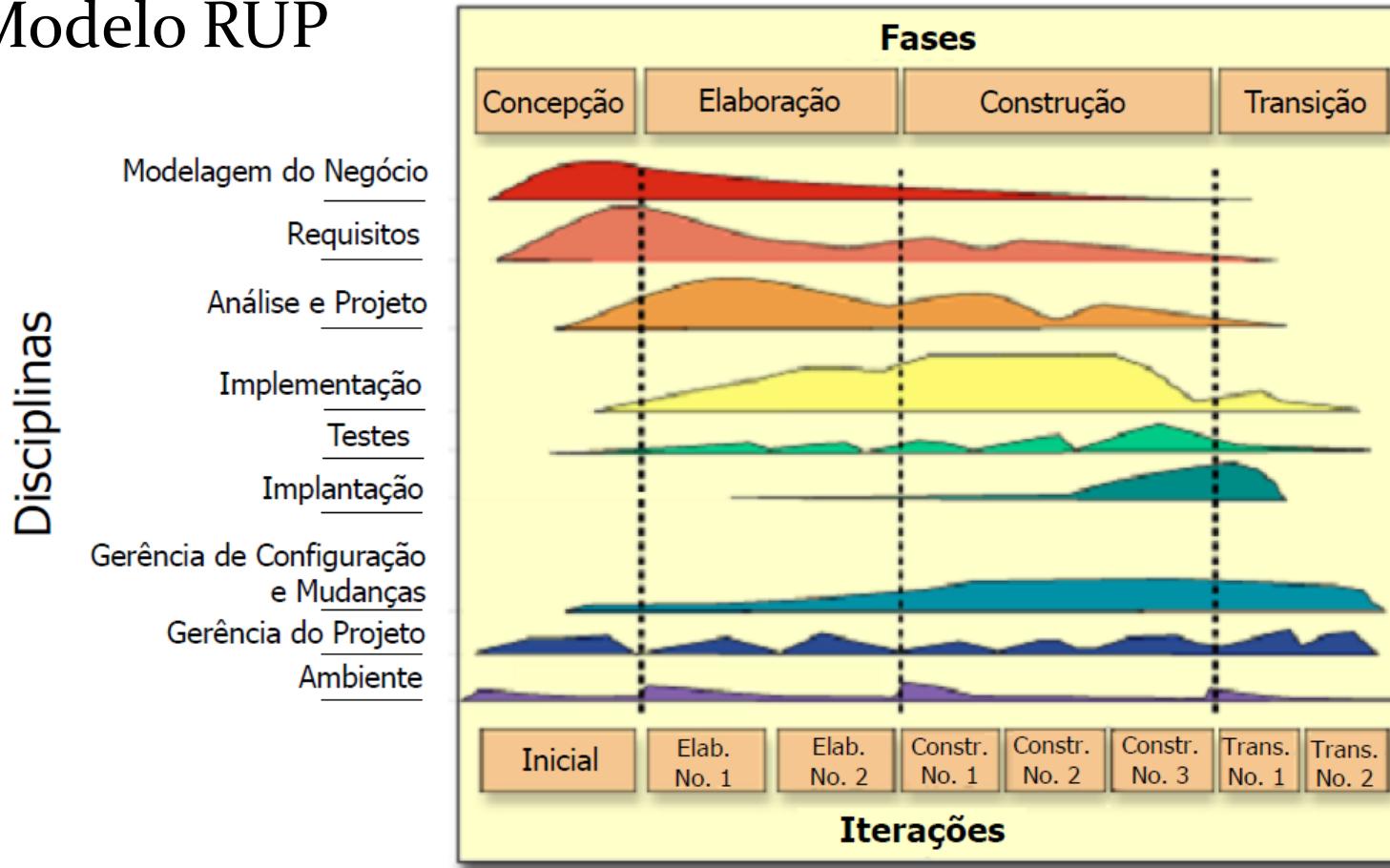
Divisões gerenciais

Fase	Objetivo
Concepção	Justificar o projeto do ponto de vista do negócio
Elaboração	Detalhar o produto visando ao planejamento preciso
Construção	Producir de uma versão operacional do produto
Transição	Colocar o produto à disposição dos usuários

Atividades técnicas

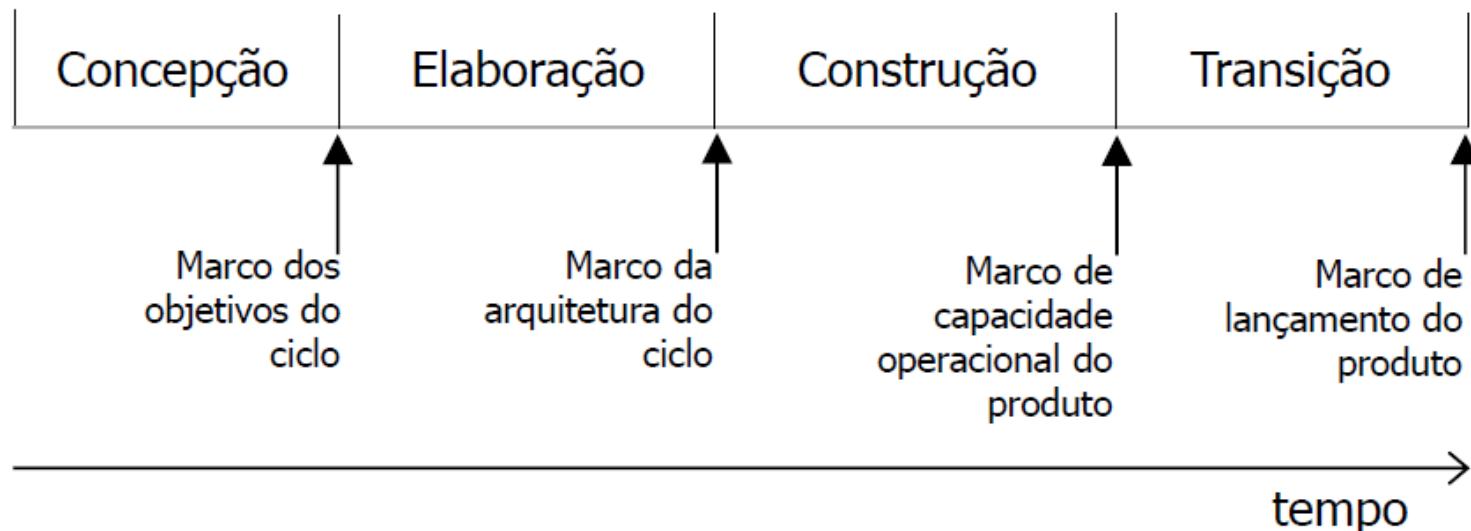
Fluxo	Objetivo
Requisitos	Definir as necessidades/funcionalidades
Análise	Detalhar, estruturar e validar requisitos
Projeto	Formular um modelo completo com vistas à implementação
Implementação	Construir o produto propriamente dito
Teste	Verificar os resultados da implementação

• Modelo RUP



- Modelo RUP

- A dimensão horizontal é expressa em termos de fases, iterações e marcos.



- Uma passagem pelas quatro fases compõe um ciclo de desenvolvimento



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Modelo RUP
 - Cada ciclo (passagem pelas quatro fases) produz uma geração do software;
 - O produto vai se desenvolvendo de geração em geração, repetindo a mesma seqüência: Concepção, Elaboração, Construção e Transição...



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil (Desenvolvimento Ágil)
 - Agilidade refere-se à
 - Velocidade das operações que ocorrem nas organizações;
 - Velocidade nas respostas aos clientes;
 - Tempo reduzido de ciclo.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil - Motivação
 - Contexto típico de desenvolvimento de S/W:
 - Negócio não é bem conhecido e/ou é dinâmico (necessidade constante de manutenção evolutiva e adaptativa);
 - Documentação externa permanece desatualizada em boa parte do ciclo de desenvolvimento;
 - As manutenções são feitas com base no código (ninguém consulta/atualiza a documentação externa);
 - Necessidade de disponibilidade rápida de código;



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil - Motivação
 - Contexto típico de desenvolvimento de S/W (cont.):
 - Desenvolvimento centrado no código;
 - Demanda por maiores níveis de qualidade do produto (qualidades externa e interna);
 - Se há processo, ele é rígido e burocratizado;



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil - Motivação
 - Contexto típico de desenvolvimento de S/W (cont.):
 - Assunções clássicas:
 - Usuários especificam exatamente o que querem;
 - Desenvolvedores entregam o que os usuários especificaram.
 - A situação real típica:
 - Usuários não sabem o que querem;
 - Desenvolvedores não entregam o que prometeram.

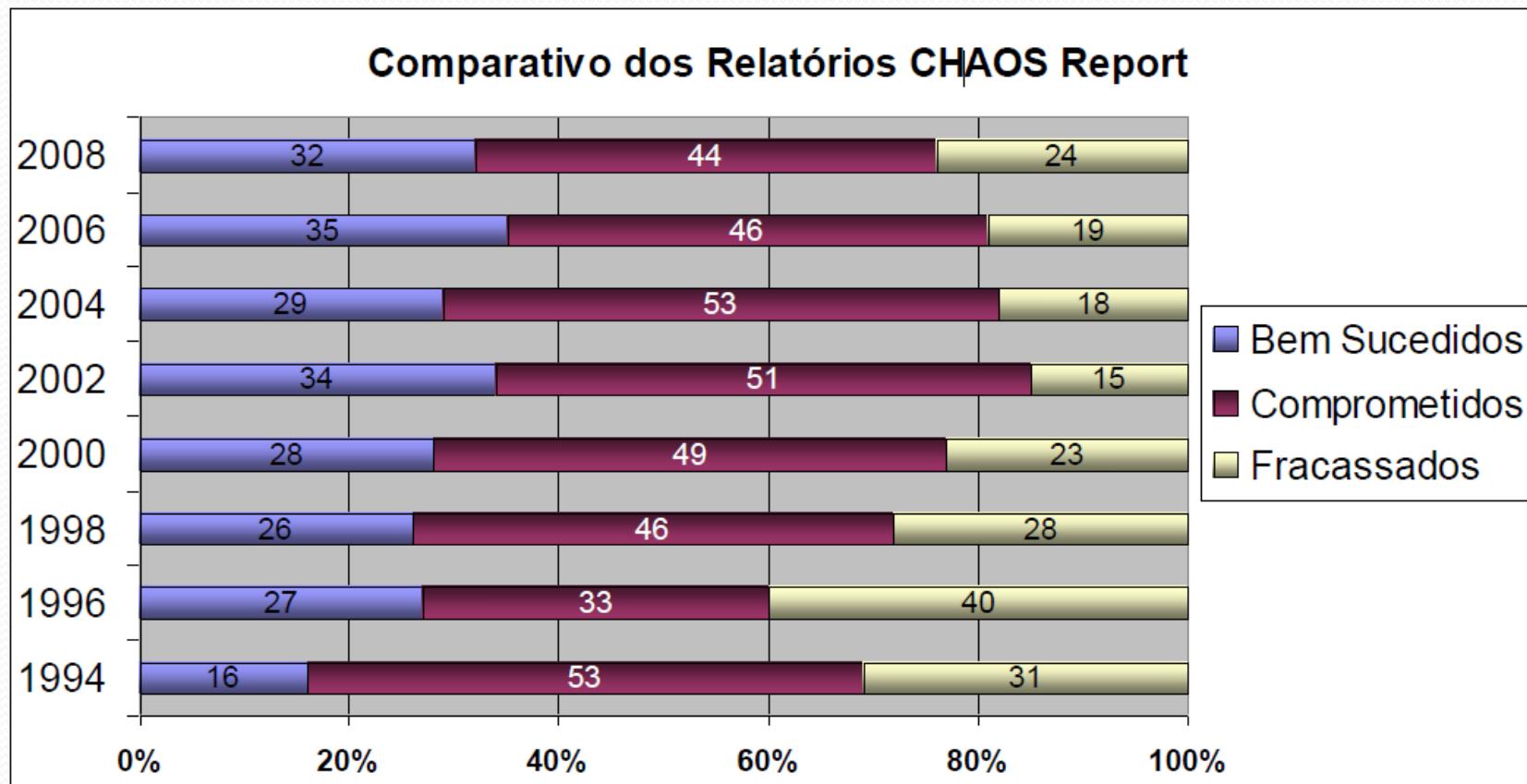


CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Motivação
 - Algumas constatações importantes:
 - A entrega de *software em prazos e custos estabelecidos quase nunca é conseguida*;
 - Excesso de formalismo nos modelos de processo propostos nos últimos 30 anos;
 - Desenvolvimento de *software é arriscado e difícil de ser gerenciado*.

- Processo Ágil – Motivação





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Motivação
 - Algumas constatações importantes (cont.):
 - Planejamento das etapas é fundamental para a execução/construção de algo;
 - Uso de métodos conhecidos e aprovados também é uma medida efetiva;
 - Idem com relação ao controle constante do produto e do processo em cada etapa do projeto.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Motivação
 - Necessita-se de novas metodologias que, dentre outras coisas:
 - Tratem adequadamente requisitos vagos e “mutantes”;
 - Mantenham a liberdade necessária para que os programadores trabalhem de forma efetiva e prazerosa;
 - Produzam software de qualidade;
 - Diminuam os encargos da equipe com documentação;
 - Mantenham as expectativas dos usuários atendidas e em níveis realizáveis;
 - Mantenham os projetos gerenciáveis;
 - Tenham base científica.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Metodologias ágeis focam em:
Redução dos custos de mudanças nos requisitos;
Entrega do sistema em pequenas liberações:
O sistema não fica pronto só no final do prazo...
... O sistema está “pronto” ao longo do prazo.
 - Metodologias ágeis propõem:
desenvolvimento de software de forma mais rápida, porém com qualidade.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Metodologias Ágeis:
 - Movimento iniciado por programadores experientes e consultores em desenvolvimento de software;
 - Questionam e se opõe a uma série de mitos/práticas adotadas em abordagens tradicionais de Engenharia de Software e Gerência de Projetos.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Manifesto Ágil :
 - Assinado por 17 desenvolvedores (Kent Beck et al) em Utah – EUA - em fevereiro de 2001.
 - Descreve a essência de um conjunto de abordagens para desenvolvimento de software criadas ao longo da última década.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Manifesto Ágil – Assunções:
 - Indivíduos e interações são mais importantes que processos e ferramentas;
 - Software funcionando é mais importante que documentação completa e detalhada;
 - Colaboração com o cliente é mais importante que (re)negociação de contratos;
 - Adaptação a mudanças é mais importante que seguir o plano inicial.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Manifesto Ágil - Objetivos:
 - Satisfazer o cliente, entregando, rapidamente e com freqüência, sistemas com algum valor.
 - Entregar versões funcionais em prazos curtos;
 - Estar preparado para mudanças de requisitos;
 - Fazer com que o pessoal de negócios e desenvolvedores trabalhem juntos;
 - Trocar informações através de conversas diretas, face-to-face.

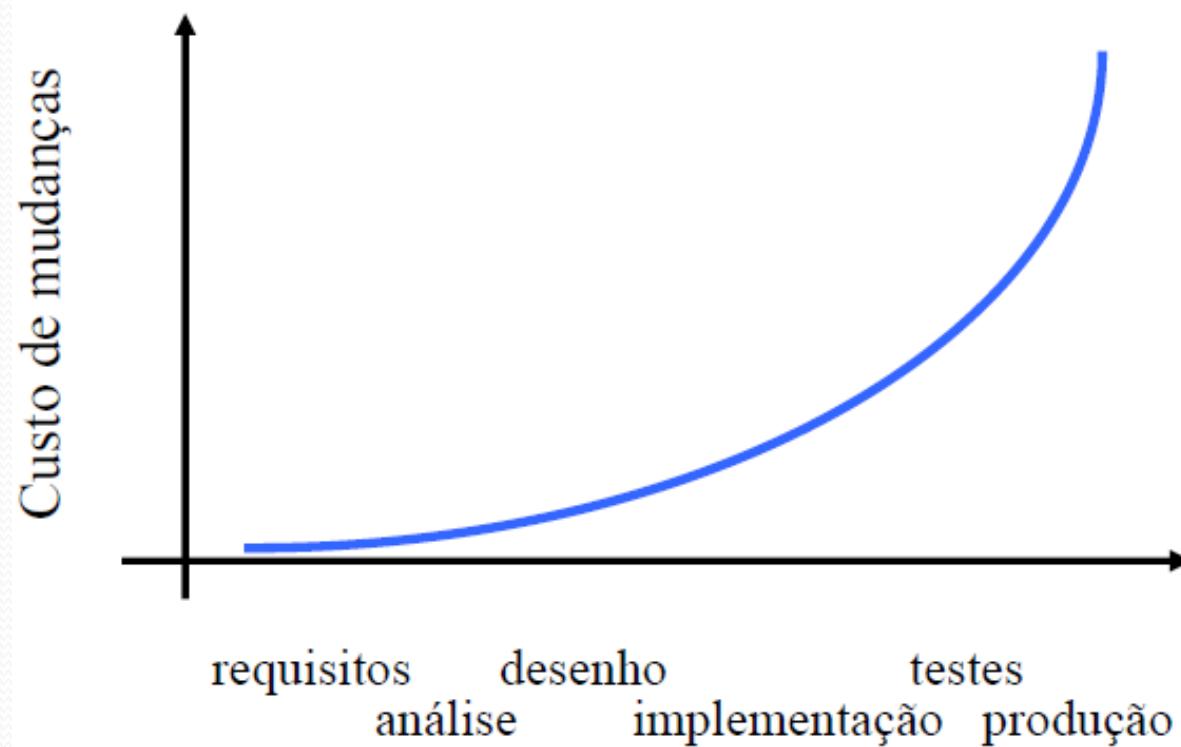


CEFET/RJ – Uned Nova Friburgo

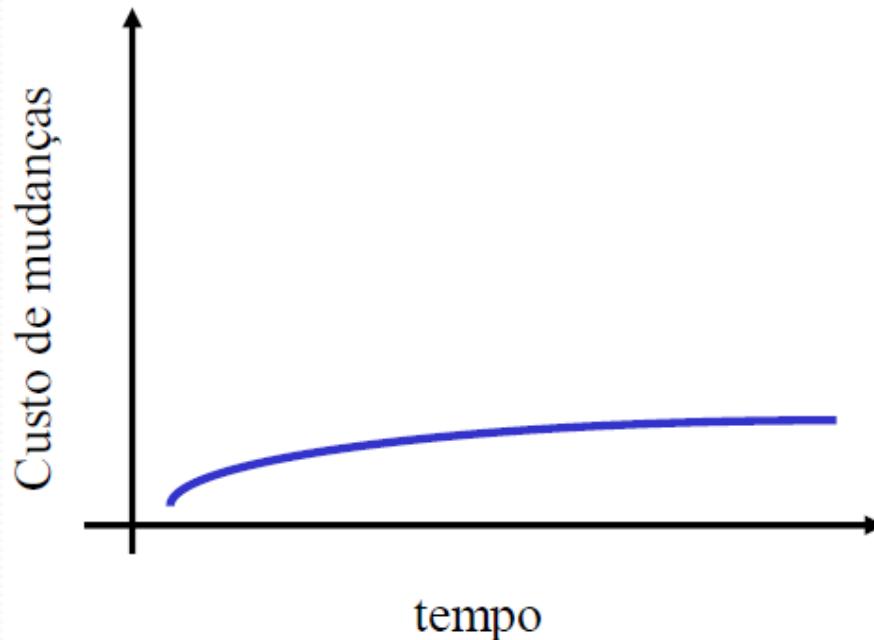
Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - O custo das modificações
 - Premissa clássica:
 - Em processos tradicionais, os requisitos devem ser conhecidos a priori, já que segundo o entendimento tradicional, mudanças de requisitos têm um impacto no custo cada vez maior quanto mais tarde, no ciclo de desenvolvimento, as mesmas ocorrerem.

- Processo Ágil – Fundamentos
 - O custo das modificações



- Processo Ágil – Fundamentos
 - O objetivo principal é diminuir os custos de mudanças. Toda mudança é bem vinda.





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - E quais são essas tecnologias e práticas?
 - Objetos são a tecnologia chave
 - Encapsulamento provê manutenibilidade;
 - Modificações de comportamento, sem alteração de código existente, podem ser (mais facilmente) implementadas através de mudanças nas trocas de mensagens entre os objetos.
 - Projeto simples, sem elementos extras (antecipação de necessidades, flexibilidades não solicitadas, etc.);
 - Testes automatizados aplicados logo após as implementações das modificações;
 - Experiência prática desenvolvida na aplicação de modificações em projetos, provendo auto-confiança na hora em que elas são necessárias.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Nesse contexto, as atitudes dos desenvolvedores ágeis são:
 - Implementam agora somente o que precisam agora;
 - Tomam as grandes decisões o mais tarde possível (quem sabe não será preciso tomá-las, de fato?);
 - Não implementam flexibilidade desnecessária num dado momento (não antecipam necessidades – quem sabe se serão mesmo necessárias?).



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Fundamentos
 - Constações:
 - O que muda em um software?
 - Os requisitos;
 - A tecnologia;
 - O projeto;
 - O time;
 - ...
 - Mudanças são inevitáveis. É importante saber lidar com elas;
 - Pequenas (às vezes grandes) e constantes mudanças são (quase inevitavelmente) necessárias



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Processo Ágil – Scrum
 - Características
 - Papéis
 - Práticas
 - O Processo



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
 - Características:
 - Desenvolvido por Jeff Suttherland (1) e Ken Schwaber (2) - outros dois signatários do Manifesto Ágil - a partir de práticas em empresas como DuPont e Honda;
 - É um processo ágil e leve que pode ser usado para controlar o desenvolvimento de um produto, gerenciar qualquer trabalho ou desenvolver software usando práticas iterativas e incrementais;
 - É um framework para gerência de projetos



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
 - Características (cont.):
 - Muito bom para projetos urgentes e críticos;
 - Usa práticas do RUP e XP (pode co-existir com os dois);
 - Escalável (aplicável desde um único projeto a toda a organização);
 - Utiliza reuniões diárias de coordenação do processo.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
 - Os papéis no Scrum :
 - O dono do produto (Product Owner)
 - Representa os interesses do cliente;
 - Detém a definição do sucesso;
 - Gerencia o produto, sprint a sprint, com vistas a maximizar o ROI (Return On Investment), definindo prioridades e planos de liberações.
 - O master (ou o facilitador) do Scrum (Scrum Master)
 - É o gerente do projeto;
 - Ensina ao dono do produto e ao time (interface entre eles);
 - É o responsável pelo sucesso do projeto;
 - Protege o time das interrupções externas.
 - O time
 - 5 a 9 membros;
 - Gerencia a si mesmo;
 - É quem produz e quem aplica os métodos de engenharia;
 - Tem poderes para realizar todo e qualquer esforço para realizar os objetivos do sprint.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
 - O backlog:
 - Especificado para realizar a “visão” que os clientes têm do sistema;
 - Lista de requisitos funcionais e não funcionais para o novo sistema;
 - É priorizado de tal forma que os itens que agregam mais valor têm prioridades mais altas;
 - Também organiza o produto em releases;
 - Organizado dinamicamente pelo dono do produto, refletindo mudanças no negócio e a performance do time;
 - O master pode ajudar o dono na escolha de um backlog de maior sucesso.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum:
 - As práticas no Scrum:
 - A reunião de planejamento do sprint
 - O dono do produto descreve para o time as funcionalidades de maiores prioridades (isso não demora mais do que quatro horas);
 - Nas quatro horas seguintes, o time decide o que dá para tornar produto até o final do sprint;
 - Reunião não demora mais do que oito horas;
 - O time se compromete a fazer o máximo possível e traça o plano de trabalho;
 - O resultado do planejamento é um backlog do sprint.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum:
 - As práticas no Scrum (cont.):
 - O sprint
 - Iteração no Scrum;
 - Estritamente enquadrado em um período de 30 dias; (personalizado)
 - Não gananciosa: melhor errar por menos do que por mais.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum:
 - As práticas no Scrum (cont.):
 - O scrum diário
 - Limitado a 15 minutos;
 - Presentes, apenas, o time e o master do Scrum;
 - Cada membro do time responde a três perguntas:
 - O que você fez desde ontem?
 - O que será feito hoje?
 - Algo está atrapalhando?

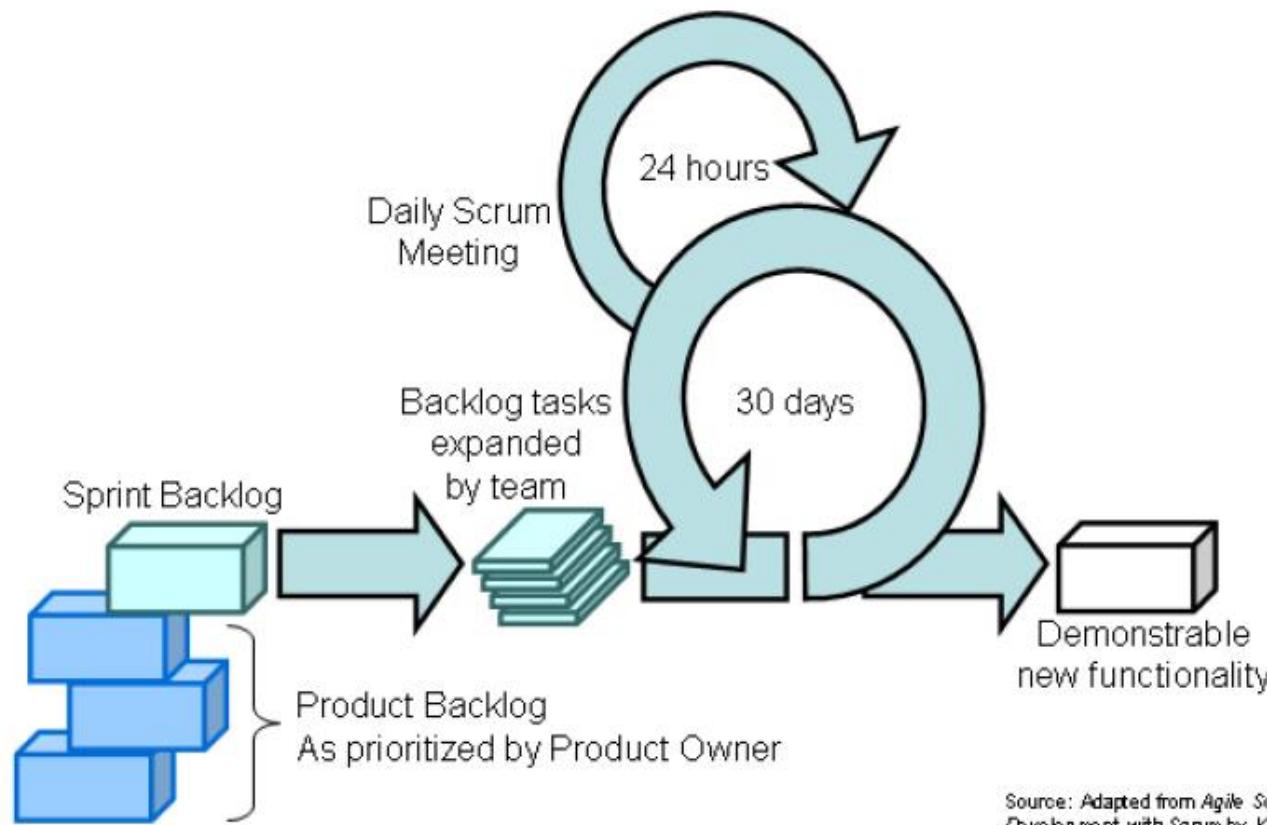


CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum:
 - As práticas no Scrum (cont.):
 - A reunião de revisão do sprint
 - Presentes: todos!
 - Limitada a 1 hora de preparação e 4 horas de apresentação;
 - O time demonstra a realização do que foi proposto;
 - Informalidade é melhor do que “Power Points”.
 - A retrospectiva do sprint
 - Presentes: o master e o time;
 - Revisão conjunta do último sprint (lições aprendidas);
 - 3 horas para isso.

- Scrum:



Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
- Os artefatos:
 - O backlog do produto:
 - Tabela contendo os itens que compõem cada sprint;
 - Cada item tem uma estimativa, um fator de ajuste e a estimativa ajustada, além de campos para a registro dos tempos necessários para cumprimento das tarefas em cada momento.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

• Scrum

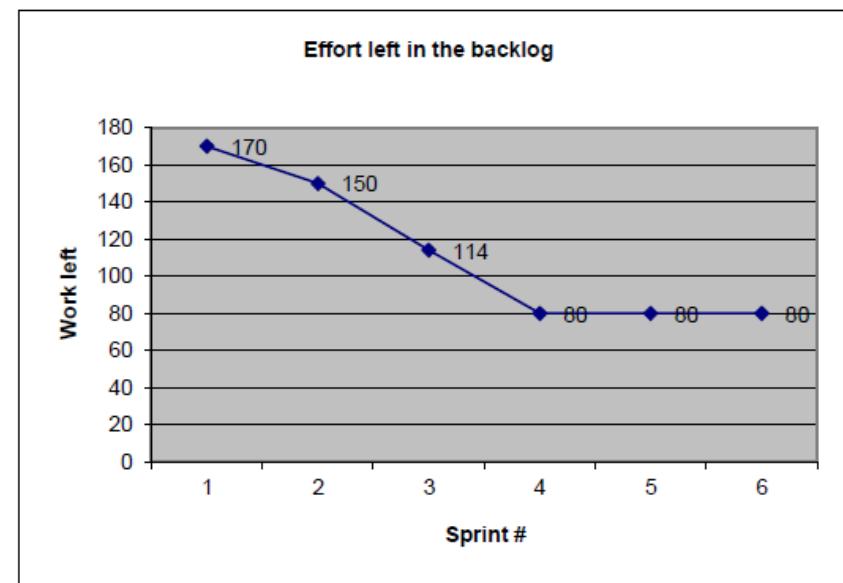
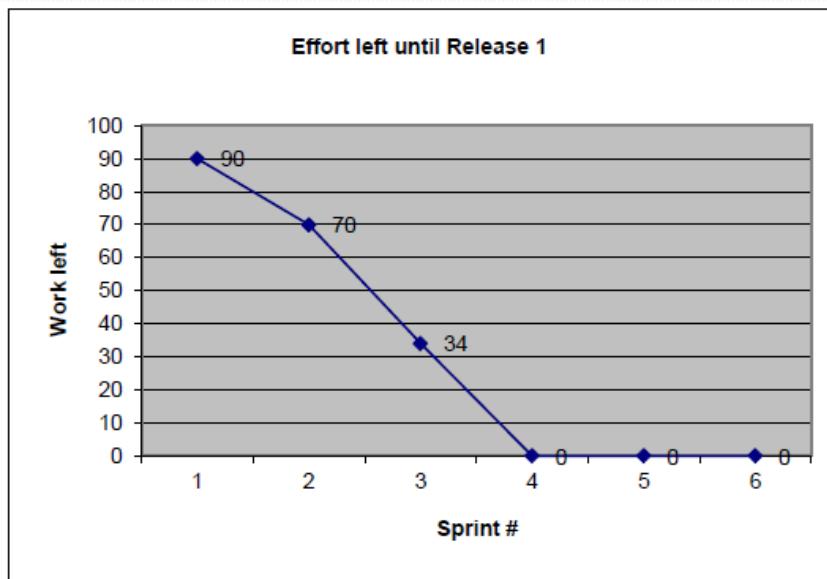


CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
 - Os artefatos:
 - O gráfico de desempenho (burndown chart):
 - Exibe o trabalho que falta em cada momento no tempo.
 - Permite realizarmos estimativas da conclusão do projeto.

- Scrum





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

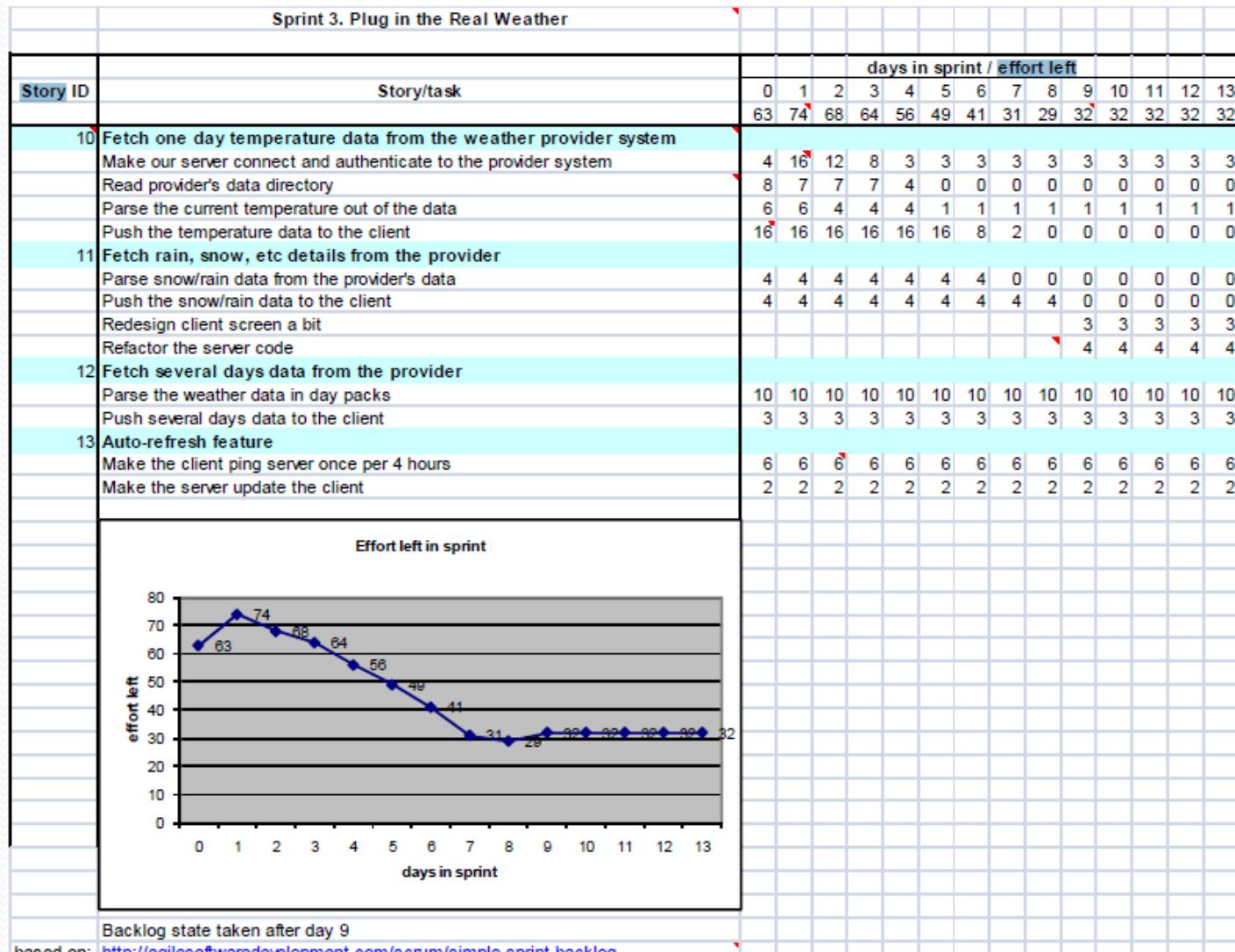
- Scrum
 - Os artefatos (cont.):
 - O backlog do sprint;
 - Especifica as tarefas que o time entende como necessárias para a realização do sprint;
 - Preparada nas quatro últimas horas da reunião de planejamento do sprint;
 - As tarefas são divididas de forma a demandarem de 4 a 16 horas de trabalho;
 - Só o time pode alterar o backlog;
 - É uma tabela onde as linhas relacionam as tarefas e as colunas relacionam os executores e as horas restantes para realizá-las em cada um dos 30 dias do sprint.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Scrum
 - O status “pronto” de um incremento (resultado de um sprint) significa que:
 - Ele foi suficientemente testado;
 - O código está bem estruturado e bem escrito;
 - O código executável foi gerado;
 - A documentação do usuário (help e/ou doc. externo) está pronta.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos:

Sommerville, Engenharia de Requisitos é o processo de **descobrir, analisar, documentar e verificar as funções e restrições do sistema.**

Um requisito de sistema descreve o que é requerido para que o sistema cumpra o seu objetivo.

Foco no “O QUE” será feito



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos
 - Compreender os conceitos dos requisitos do usuário e dos requisitos de sistema;
 - Compreender as diferenças entre requisitos funcionais e não funcionais;
 - Compreender como os requisitos podem ser organizados em um documento de requisitos de software.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos
 - **Fatores que contribuem para o fracasso de um projeto [Fonte: Standish Group, 1994] :**
 1. requisitos incompletos (13,1%)
 2. falta de envolvimento por parte do usuário (12,4%)
 3. falta de recursos (10,6%)
 4. expectativas não realistas (9,9%)
 5. falta de apoio dos executivos (9,3%)
 6. modificações nos requisitos e nas especificações (8,7%)
 7. falta de planejamento (8,1%)
 8. o sistema não era mais necessário (7,5%)



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Requisito (Definição - IEEE):

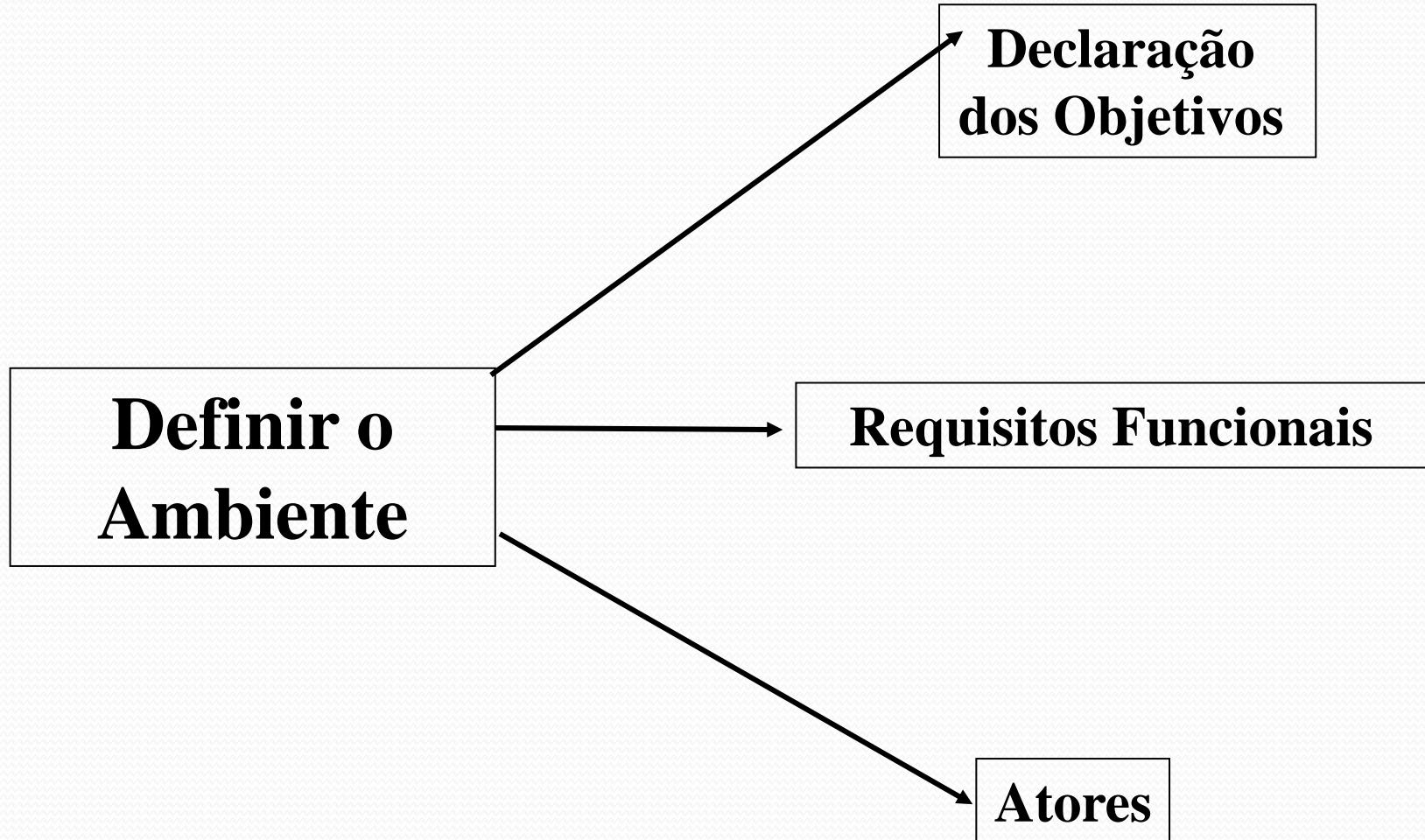
- Uma condição ou uma funcionalidade necessária a um usuário para resolver um problema.
- Uma condição ou funcionalidade que deve ser atingida ou influenciada por um componente de sistema para satisfazer um contrato, padrão, especificação, ou outro documento formalmente definido.

- Engenharia de Requisitos



- O limite do contexto de um sistema, separa a parte relevante do ambiente do sistema a ser desenvolvido da parte irrelevante (*).
- O contexto do sistema é levantado na fase de requisitos

(*) Veremos que não é tão irrelevante assim.





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Análise do Contexto:

- Qual a finalidade do sistema?
- A que ele se propõe?
- Que problemas ele deverá desenvolver?
- Que requisitos devem ser atendidos?
- Algum sistema será por ele substituído?



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Análise do Contexto:

- **Analisar Aspectos Cognitivos:** Um sistema ao entrar em produção muda a forma das pessoas trabalharem, e por consequência a forma da empresa trabalhar como um todo.
 - Análise da Dificuldade de adaptação das pessoas.
 - Interfaces que levem símbolos que dizem algo para as pessoas daquela empresa.
 - Levar em consideração a acessibilidade das pessoas (problemas físicos e visuais).
 - Interfaces preparadas e direcionadas para o público alvo da empresa que vai usar.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Análise do Contexto:

- **Analizar Aspectos Cognitivos:**
 - Projetar sistemas capazes de se adaptarem às necessidades dos operadores.
 - Analisar o processo de cultura e aspectos psicológicos das pessoas na interação com o sistema.
 - Desenvolver sistemas inteligentes capazes de interagir da melhor forma possível com os seus usuários. Nem sempre 100% de automação é desejável do ponto de vista da eficiência.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Objetivos da Engenharia de Requisitos:

- Estabelecer e manter concordância com os clientes e desenvolvedores;
- Registrar e acompanhar requisitos ao longo de todo o desenvolvimento;
- Definir as fronteiras do sistema;
- Fornecer a base para o cronograma/custo de desenvolvimento do sistema.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Tipos de Fontes de Requisitos:

- **Stakeholders:** São pessoas ou organizações que influenciam no desenvolvimento do sistema. Exemplos de Stakeholders: Usuários do futuro sistema, operadores do sistema, desenvolvedores, arquitetos, clientes e testadores.
- **Documentos:** muitas vezes contém informações importantes que fornecem requisitos. Normas da empresa, documentos legais, relatórios da organização e documentos dos sistemas existentes na empresa.
- **Sistemas em operação:** Sistemas legados, existentes, concorrentes, tudo o que os stakeholders podem usar para criticar ou fornecer necessidades que são necessárias para o futuro sistema.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Papel do cliente e do desenvolvedor:

- **Cliente:** formular (de modo concreto) as necessidades em termos de funções e desempenho.
- **Desenvolvedor:** atua como indagador, consultor e solucionador de problemas.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Foco no Usuário:

- Identificar **Objetivos de Negócio** (**Por que** desenvolver algo?)
- Identificar **Stakeholders** (**Quem** está envolvido?)
- Obter diferentes **Pontos de Vista** (**Com** que os stakeholders estão preocupados? Existem conflitos?)
- Resolver **Conflitos**
- Identificar **Cenários** (**Quais resultados** as pessoas desejam? Sob **que circunstâncias**?)



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos
- **Requisitos do usuário** → Declarações em linguagem natural e também em diagramas, sobre as funções que o sistema deve fornecer e as restrições sob os quais deve operar.
- **Requisitos de sistema** → Um documento estruturado com descrições detalhadas dos serviços de sistemas. Escrito como um contrato entre o cliente e contratante .



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Tipos dos Requisitos

- Requisitos funcionais
- Requisitos não funcionais



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Requisitos funcionais

- Declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações.
- Exemplo: o sistema deve prever um relatório de notas do aluno.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos

Requisitos Não funcionais

- Expressam qualidade e restrições sobre os serviços ou as funções oferecidas pelo sistema.
 - Ex. restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, etc.
- Surgem conforme a necessidade dos usuários, em razão de restrições de orçamento etc.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

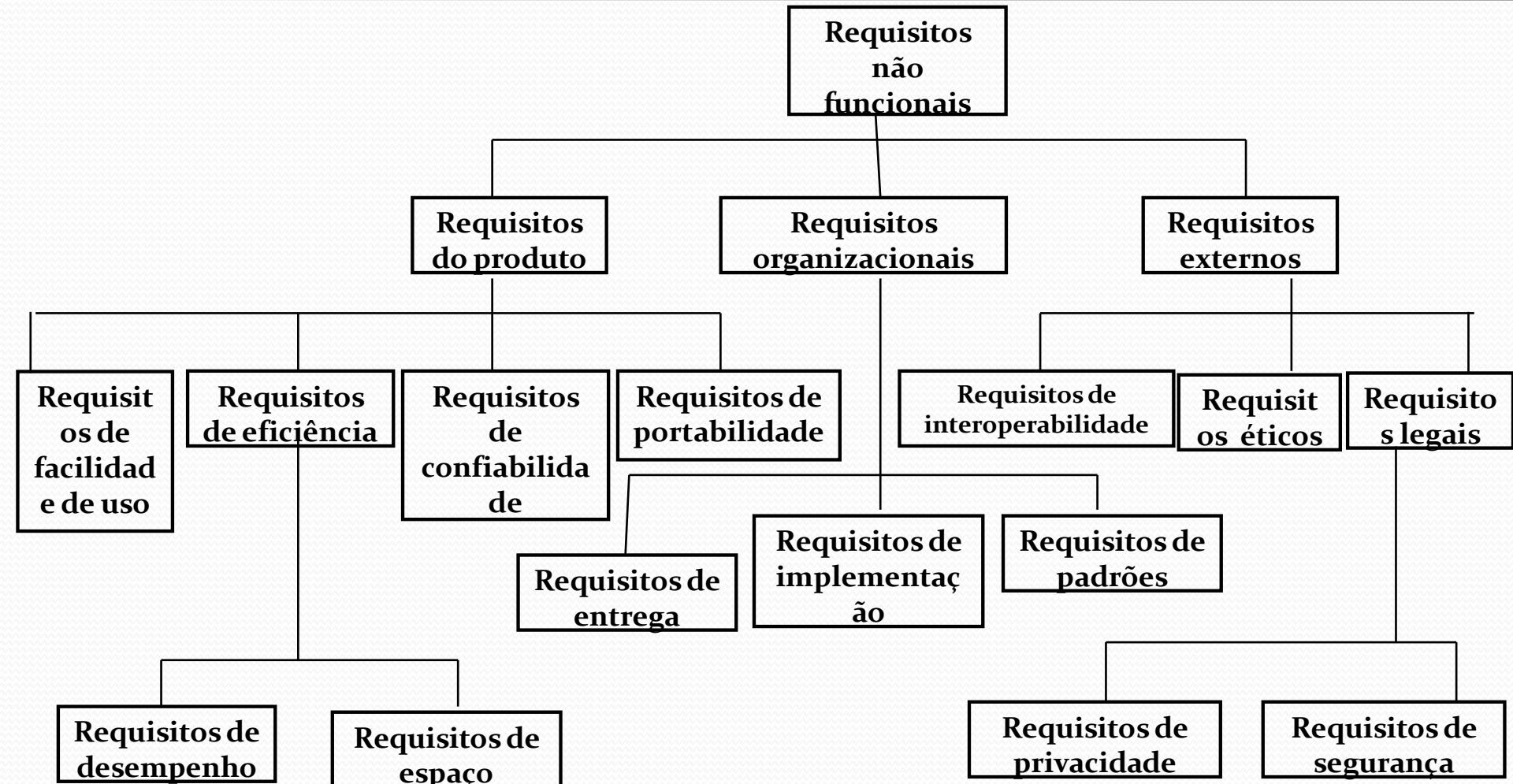
- Engenharia de Requisitos

Requisitos Não funcionais (cont.)

- Podem estar relacionados propriedades de confiabilidade, tempo de resposta e espaço em disco.
- A falha de não cumprir com um requisito não funcional de sistema pode tornar todo o sistema inútil. (ex. requisito confiabilidade num sistema de aviação).

CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Requisitos não funcionais:

- **Requisitos de produtos** → Requisitos que especificam o comportamento do produto.
Ex. portabilidade; velocidade de execução; confiabilidade, etc.
- **Requisitos da organização** → Requisitos decorrentes de políticas e procedimentos organizacionais. Ex. padrões, infra-estrutura, etc.
- **Requisitos externos** → Requisitos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento.
Ex. requisitos de interoperabilidade, legislação, etc.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Requisitos não funcionais

- **Requisitos de facilidade de uso** → Ex. usuários devem ser capazes de usar as funções do sistema após duas horas de treinamento.
- **Requisitos de eficiência** → Ex. o sistema deve processar n requisições por segundo.
- **Requisitos de confiabilidade** → Ex. o sistema deve estar disponível 99% das vezes.
- **Requisitos de portabilidade** → Ex. o sistema deve rodar nas plataformas X e Y.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Requisitos não Funcionais

- **Requisitos de entrega** → Ex. um relatório de progresso deve ser entregue a cada duas semanas.
- **Requisitos de implementação** → Ex. o sistema deve ser implementado na linguagem Java.
- **Requisitos de padrões** → Ex. uso de métodos orientados a objetos; desenvolvimento utilizando a ferramenta X.



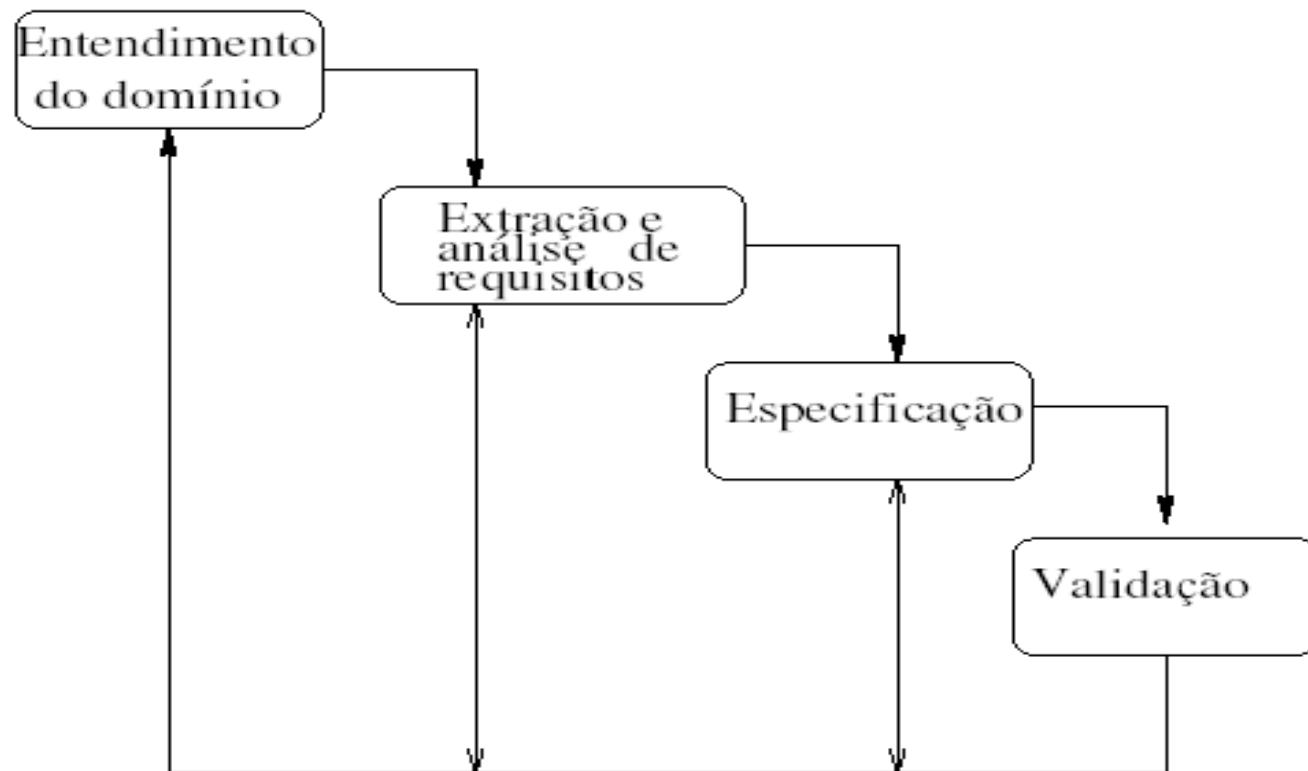
CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

Requisitos não funcionais

- **Requisitos de interoperabilidade** → Ex. o sistema deve interagir com os sistemas X e Y.
- **Requisitos éticos** → Ex. o sistema não deverá revelar aos operadores nenhuma informação pessoal dos clientes.
- **Requisitos legais** → Ex. o sistema deverá armazenar as informações de acordo com a Lei XXYY de ZZ.

- Passos para o processo de extração de requisitos de software





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Passos para o processo de extração de requisitos de software**
 - **Entendimento do domínio:** os desenvolvedores devem entender o domínio da aplicação (documentos, livros, pessoas).
 - **Extração e análise de requisitos:** acontece a descoberta, revelação e entendimento dos requisitos, através de interação com o(s) usuário(s); e desenvolvedores envolvendo:
 - Classificação e organização dos requisitos;
 - Determinação de suas prioridades;
 - Resolução de inconsistência;
 - Conflitos e a descoberta de omissões.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Passos para o processo de extração de requisitos de software**
 - **Especificação dos requisitos:** armazenamento dos requisitos em uma ou mais formas, incluindo linguagem formal ou semiformal, representações simbólicas ou gráficas.
 - **Validação dos requisitos:** verificação dos requisitos, visando determinar se estão completos com as necessidades do usuário.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Passos para o processo de extração de requisitos de software**
 - Mudanças nos requisitos acontecem na maioria dos sistemas complexos (mudanças das necessidades dos usuários, outras advêm da interpretação incorreta dos requisitos do produto a ser desenvolvido).
 - Requisitos incompletos, incorretos ou mal entendidos são as causas mais frequentes da baixa qualidade, ultrapassagem dos custos previstos e atraso na entrega do produto de software.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Dificuldades no processo de extração de requisitos

Falta de conhecimento do usuário

- O usuário não conhece sua real necessidade.
- Desenvolvedores não conhecem o domínio do problema.
- Diferenças entre o que os usuários querem e o que precisam.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Procedimento de extração de requisitos**
 - **Perguntar:** identificar a pessoa apropriada.
 - **Observar e inferir:** observar o comportamento dos usuários e inferir suas necessidades.
 - **Discutir e formular:** discutir com os usuários suas necessidades e, juntamente com eles, formular um entendimento comum dos requisitos.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Procedimento de extração de requisitos**
 - **Negociar:** a partir de um conjunto-padrão de requisitos, negociar com os usuários quais dessas características serão incluídas, excluídas ou modificadas.
 - **Identificar problemas:** investigar os problemas para identificar os requisitos que podem melhorar o produto.
 - **Supor:** quando não existe usuário, ou para a criação de um produto inexistente é preciso usar intuição.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Técnicas de extração de requisitos**
- **Técnicas informais** – baseada em comunicação estruturada e interação com o usuário.
 - **Entrevistas**
 - **Questionário**
 - **Técnica dos 5 W's**
 - **Brainstorming**
 - **Observação**
- **Técnicas formais** – construção de um modelo conceitual do problema sendo analisado, ou de um protótipo de um produto de software a ser construído.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Técnicas de extração de requisitos
- Identifique os principais pontos através das questões:
 - o que (What?)
 - quando (When?)
 - onde (Where?)
 - por que (Why?)
 - quem (Who?)
 - e ainda pode acrescentar a pergunta como (How?).



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **O documento de requisitos de software**
 - O documento de requisitos é a declaração oficial do que é exigido dos desenvolvedores de sistemas
 - Deve incluir uma definição e uma especificação detalhada dos requisitos.
 - Não é um documento de projeto. Até onde possível, deve fixar o que o sistema deve fazer em lugar de **COMO** deve fazer.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **O documento de requisitos de software**
 - Especificar o comportamento externo do sistema.
 - Especificar as restrições à implementação.
 - Ser fácil de modificar.
 - Servir como uma ferramenta de referência para os responsáveis pela manutenção do sistema.
 - Registrar a estratégia sobre o ciclo de vida.
 - Caracterizar respostas aceitáveis para eventos indesejáveis.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **O documento de requisitos de software**
 - O Documento de Requisito de Software é geralmente documentado em linguagem natural. Onde se produz um texto com todos as prerrogativas encontradas no momento da elucidação dos requisitos.
 - Hoje em dia além da linguagem natural usa-se modelos da UML para documentar os requisitos , como caso de uso e o digrama de classes.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Requisitos
 - Validando os Requisitos:
 - Prototipação;
 - Revisando o documento com os stakeholders;
 - Revisitando os modelos;
 - Formalizando o documento e uma assinatura de compromisso com todos os interessados.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Projeto de Software



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Projeto de Software
 - Arquitetura de Software
 - Patterns e Frameworks
 - Introdução
 - Padrões de projeto
 - Frameworks
 - MDA (Model-Driven Architecture)



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software
 - Durante o desenvolvimento de um sistema
 - Parte das decisões está ligada ao ambiente onde o software irá operar...
 - ... e parte diz respeito a quais serão os componentes do software, como eles se comunicarão e quais serão suas funcionalidades e restrições de uso.
 - Essas questões dizem respeito à arquitetura do sistema sendo desenvolvido.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software

- Por que a nossa preocupação com a arquitetura?
 - Porque arquiteturas diferentes envolvem custos diferentes;
 - Arquiteturas diferentes impõem qualificações diferentes da equipe de desenvolvimento;
 - Arquiteturas diferentes impõem custos diferentes, inclusive de implantação e de infra-estrutura.
 - Porque arquiteturas diferentes envolvem projetos diferentes.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software

- A opção por uma arquitetura tem de ser feita, portanto, antes do início do projeto para que:
 - Os custos globais sejam mais precisamente definidos e, portanto, possam ser perseguidos;
 - Idem com relação aos prazos.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Engenharia de Software envolve Arquitetura de Software.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software
 - Arquitetura de software ganhou força com a OO e componentes;
 - Já é tratada como uma área de especialização;
 - Principal referência: Mary Shaw & David Garland, “Software Architecture – Perspectives on an Emerging Discipline”, 1996.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software:
 - Descrição, de alto nível, da estrutura do sistema como um todo. Inclui:
 - Organização dos componentes do sistema;
 - Estruturas globais de controle;
 - Protocolos;
 - Funcionalidades dos componentes;
 - Distribuição física dos componentes;
 - Capacidade de evolução ...



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de software
 - Descrever a arquitetura de um sistema é descrever
 - Os componentes
 - Os conectores
 - As restrições
 - Normalmente utiliza-se diagramas, onde os componentes são representados por blocos e as interações por linhas (diagramas de classes, de componentes, de implantação).



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software

Componentes:

- Matéria prima para descrição da arquitetura de um sistema. Exemplos:
 - Clientes
 - Servidores
 - Filtros
 - Bancos de Dados
- Camadas de Software (Layers)



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software

Conectores:

- Provêem a interação entre os componentes, para que o sistema funcione. Exemplos:
 - Chamadas de rotinas
 - Variáveis compartilhadas
 - Protocolos Cliente-Servidor
 - Broadcasting
 - Eventos
 - ...



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software:
 - Estilos:
 - Existem diferentes estilos para a organização do software -> Estilos de Arquitetura;
 - Compreendem conjuntos de componentes, conectores e formas (regras) de como combiná-los.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software
 - Principais Estilos:
 - Sistemas em Camadas
 - Pipes and Filters
 - Invocação Implícita
 - Repositórios
 - Sistemas Distribuídos

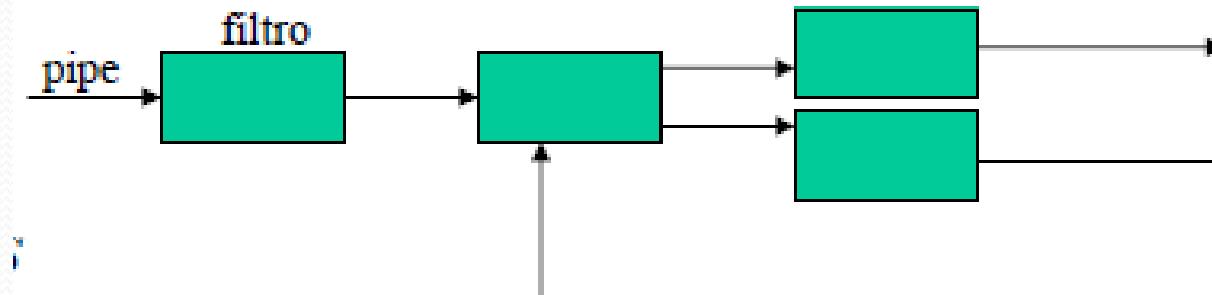


CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software
 - Estilo Pipe and Filters
 - Tipicamente divide a tarefa de um sistema em vários passos de processamento sequencial
 - Componentes:
 - São chamados *Filtros*
 - Tem um conjunto de entradas e um conjunto de saídas
 - Realiza o processamento de um stream de dados
 - Conectores:
 - São chamados *Pipes*
 - Servem como condutores, transmitindo as saídas de um filtro para as entradas de outro.

- Arquitetura de Software
 - Estilo Pipe and Filters





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Softwares
 - Estilo Pipe and Filters - Implementação
 - Divilda as tarefas do sistema em uma sequênciade estágios de processamento:
 - Cada estágio deve depender apenas da saída do seu predecessor direto
 - Todos os estágios são conectados pelo fluxo de dados
 - Defina o formato de dados a ser passado ao longo de cada pipe
 - Decida como implementar cada conexão com pipe
 - Implica em decidir se os filtros são componentes ativos ou passivos



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

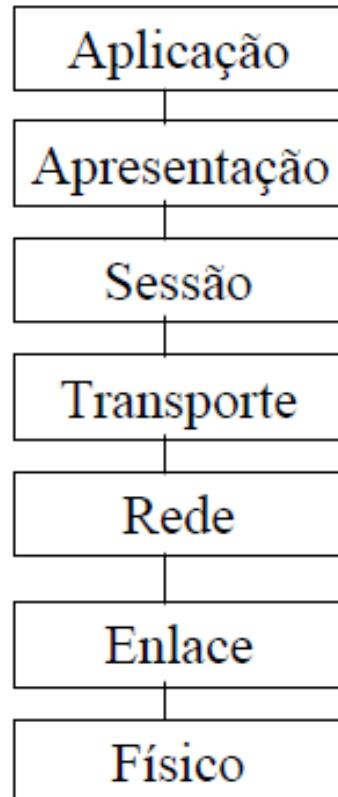
- Arquitetura de Software
 - Estilo em Camadas:
 - Componentes
 - Camadas
 - Conectores
 - Protocolos que definem as interações entre as camadas

CEFET/RJ – Uned Nova Friburgo

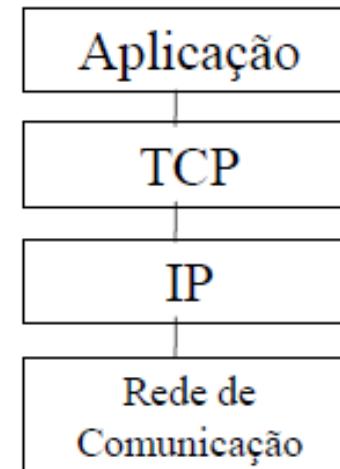
Introdução a Engenharia de SW

- Modelo de Estilo em Camadas - Exemplos

Modelo OSI da ISO

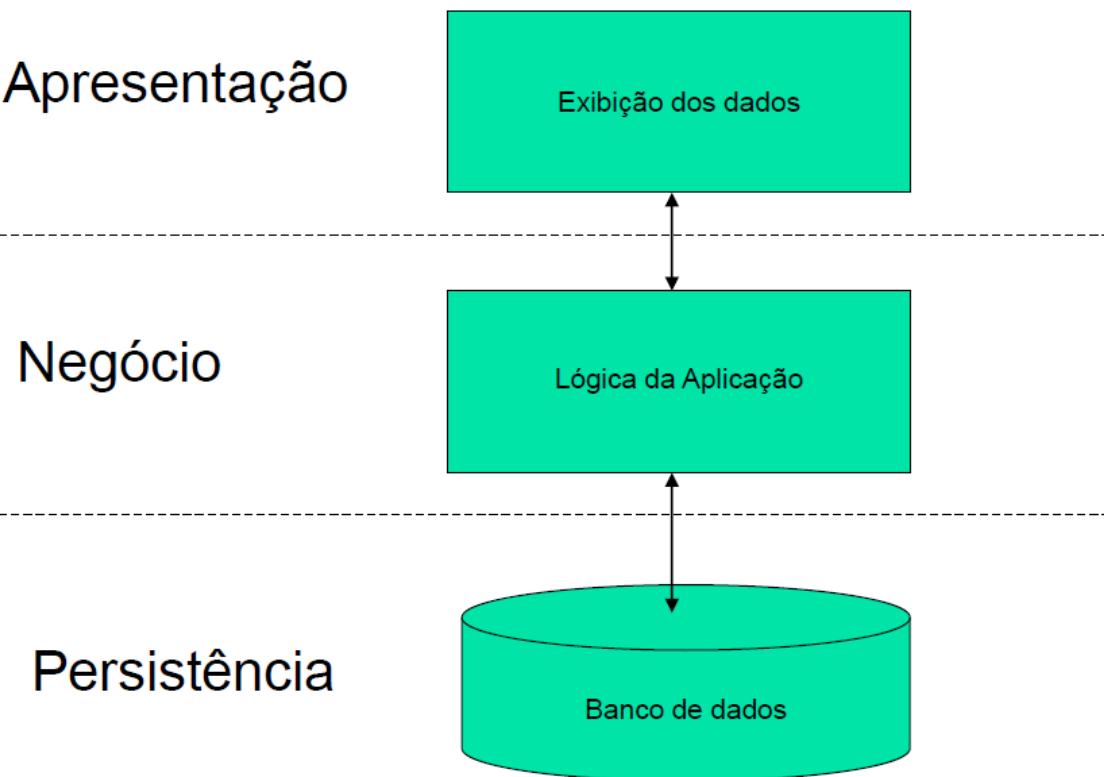


Modelo TCP/IP



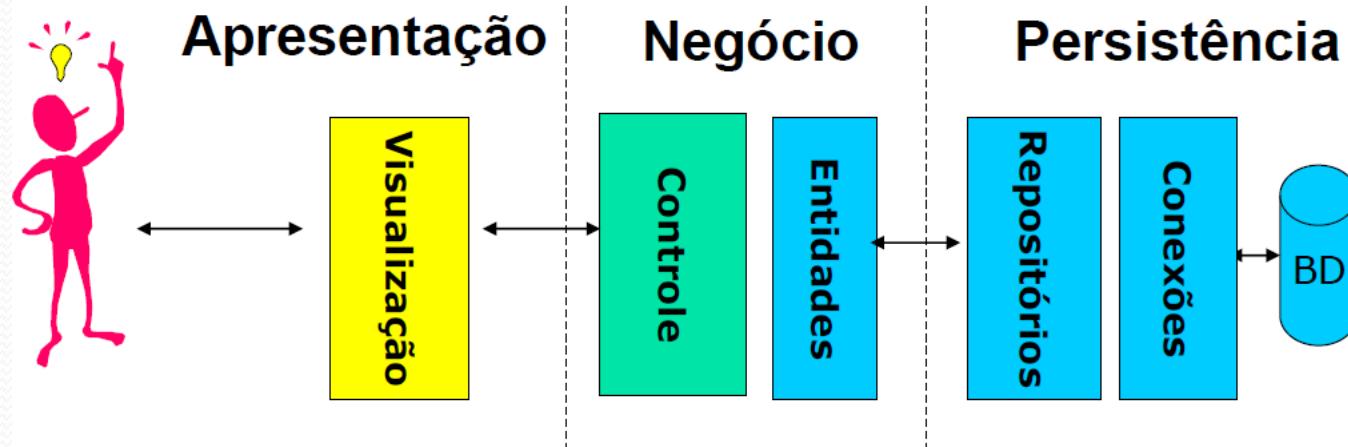
- Sistemas em Camadas - Exemplo

A arquitetura clássica de três camadas



- Sistemas em Camadas - Exemplo

3-tier & MVC



■ Modelo
■ Visão
■ Controle



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Sistemas em Camadas
 - Vantagens :
 - Suportam diferentes níveis de abstração;
 - Permitem, também, a organização da aplicação em camadas com funcionalidades/conceitos coesos (e.g. MVC, 3-tier)
 - Permitem a evolução (retira-se uma camada e coloca-se outra no lugar, bastando manter a API).



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Sistemas em Camadas
 - Desvantagens
 - A estruturação em camadas pode ser impossível em alguns sistemas (por exemplo, por conta da degradação da performance devido a necessidade de pelo menos uma chamada para atravessar cada camada de software em sistemas de controle em tempo real);
 - Dificuldade em se determinar os níveis corretos de abstração durante o projeto.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Arquitetura de Software
 - Estilo de Sistemas Distribuídos
 - Componentes:
 - Módulos distintos e especializados que interagem de forma colaborativa, eventualmente processados em nós diferentes
 - Conectores :
 - Protocolos de requisição de serviços e provimento de respostas
 - Aplicações :
 - Sistemas tolerantes a falhas, p. exemplo



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Sistemas Distribuídos
 - Vantagens
 - Distribuição da carga de processamento (cada nó processa um pouquinho)
 - Reuso
 - Especialização
 - Desvantagens
 - Performance/confiabilidade sujeitos aos critérios da distribuição (qualidade da infraestrutura de comunicação, segurança física também é distribuída -> custo)



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Patterns e Frameworks
 - Atual busca por
 - Economia de recursos (tempo, “fosfato” e \$\$\$);
 - Qualidade.
 - Aliada ao fato de que a similaridade de problemas pressupõe a similaridade de soluções;
 - **Reúso** de técnicas, artefatos, processos, ferramentas, ..., software, ..., que já se provaram corretos, eficazes, eficientes, etc.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Testes de Softwares



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Verificação e Validação (V & V)**
 - Objetivo: assegurar que o software
 - cumpra as suas especificações e
 - atenda às necessidades dos usuários e clientes.
 - Verificação:
 - “Estamos construindo certo o produto?”
 - O software deve está de acordo com a suas especificações técnicas.
 - Conjunto de Atividades que garante que o software está sendo construído de forma correta.
 - Validação:
 - “Estamos construindo o produto certo?”
 - O software deve atender às necessidades dos usuários.
 - Ocorrem em todo o ciclo de vida completo
 - Revisões de requisitos, revisões de design, testes de código



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **O que deve ser verificado e validado (V & V)?**
 - Fatores de Qualidade Operacionais
 - Correção
 - Eficiência ou desempenho
 - Robustez
 - Confiabilidade
 - Usabilidade
 - Utilidade e validade
 - Fatores de Qualidade de Revisão
 - relacionados com a manutenção, evolução e avaliação do software
 - Fatores de Qualidade de Transição
 - relacionados com a instalação, reutilização e interação com outros produtos



CEFET/RJ – Uned Nova Friburgo

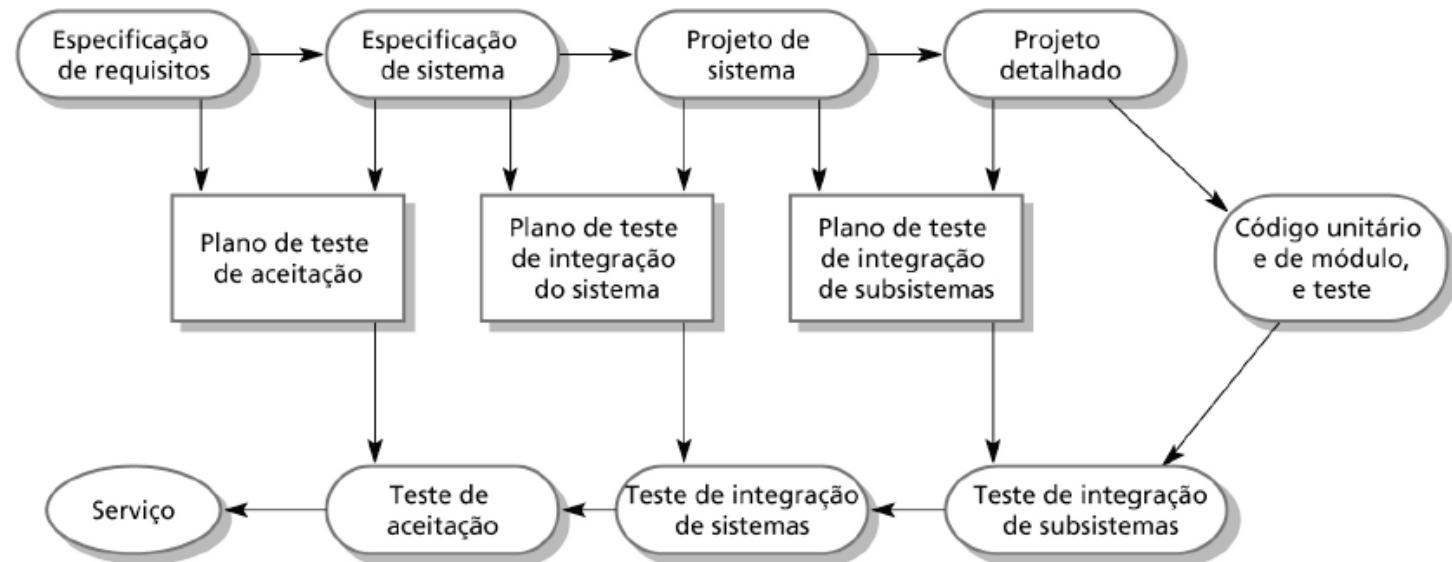
Introdução a Engenharia de SW

- **Técnicas de V & V (Sommerville)**
 - Inspeções de software (V & V estática)
 - Análise da documentação e código fonte do software
 - Pode ser auxiliado por ferramentas de depuração
 - Testes de software (V & V dinâmica)
 - O programa ou um protótipo devem ser executados
 - Casos de testes deve ser elaborados: dados de entrada e comportamento esperado.

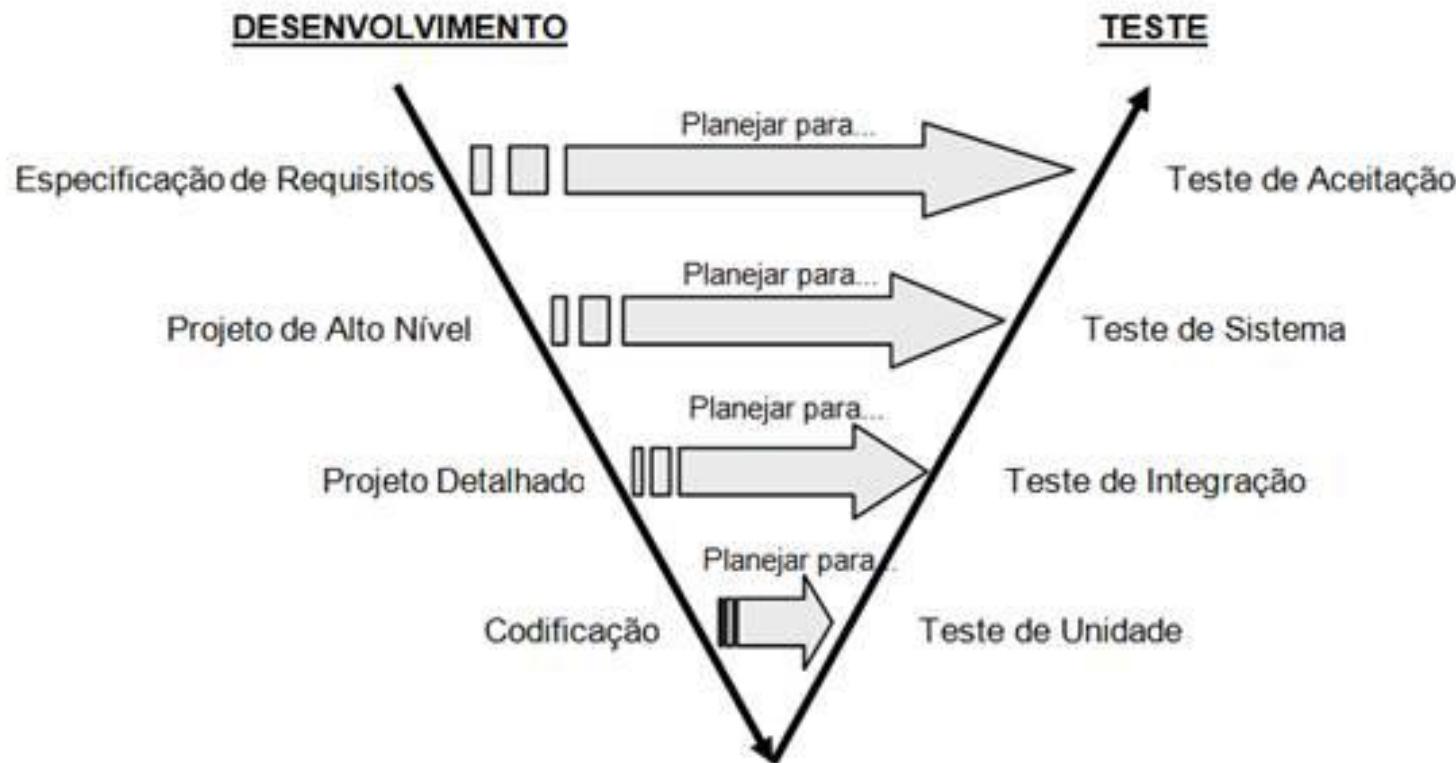
• Plano de V&V

- O processo de V&V ocorre durante todo o ciclo de vida
- Precisa ser planejado em conjunto com outras atividades do processo de software

Figura 22.3 Plano de teste como ligação entre o desenvolvimento e os testes.



- Plano de V & V





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

● Inspeções de Software

- Características

- Técnica preventiva – permite a V & V antes do software ser codificado
 - Mais barata
 - Baseada na experiência do inspetor
 - Mais aplicada a fatores de revisão e transição
 - Pouco eficaz para fatores operacionais
- Aplicações mais comuns
 - Inspeção de programa fonte (estática e dinâmica)
 - Inspeção de documentos e modelos



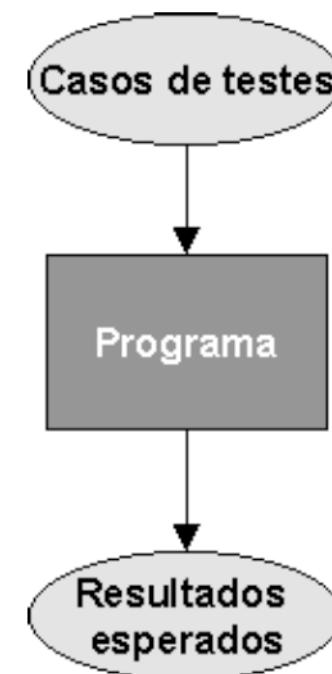
CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Abordagens para verificação da correção**
 - Inspeções analíticas
 - Estática
 - Rastreamento do código fonte (walkthrough) – percorre-se o código executando-o mentalmente.
 - Automatizada (dinâmica)
 - Depuração – execução passo-a-passo e visualização de variáveis do programa
 - Testes de correção de programas
 - Testes de unidade
 - Testes de integração
 - Prova Formal de Programas
 - Utiliza métodos formais de desenvolvimento de software – técnicas de especificação, transformação e prova formal

- **Testes de software**

- Elaboração de casos de testes baseados na especificação funcional
 - Dados de entradas
 - Comportamento esperado
- Podem ser classificados
 - Quanto ao objetivo
 - Quanto ao escopo
 - Quanto ao método
- Aplicações em fatores operacionais
 - Correção
 - Usabilidade
 - Desempenho
 - Robustez





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Tipos de testes quanto ao objetivo**

- Testes de Defeitos

- Tem por objetivo encontrar defeitos – inconsistências entre o programa e a sua especificação.

- Verifica a correção – conhecido também por **testes de correção**

- Normalmente realizados com protótipos funcionais

- Testes de Validação

- Utilizado para demonstrar ao desenvolvedor e ao cliente do sistema que o software atende aos seus requisitos.

- Um teste bem sucedido visa mostrar que o sistema opera conforme especificado pelo cliente.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Tipos de testes quanto ao método**
 - Testes Controlados
 - Realizados em laboratório ou em condições operacionais sob a supervisão de um testador
 - Baseados na geração de **casos de testes**
 - Podem ser realizados com protótipos funcionais
 - Testes Estatísticos
 - Utilizados para verificar como o software se comporta nas condições operacionais de utilização (versão beta)
 - Avalia desempenho, robustez, confiabilidade, etc
 - Utiliza programa de monitoramento e “logs” com ocorrências operacionais.
 - Exemplos de medições:
 - Número de falhas observadas
 - Tempos de resposta
 - Tempos de execução



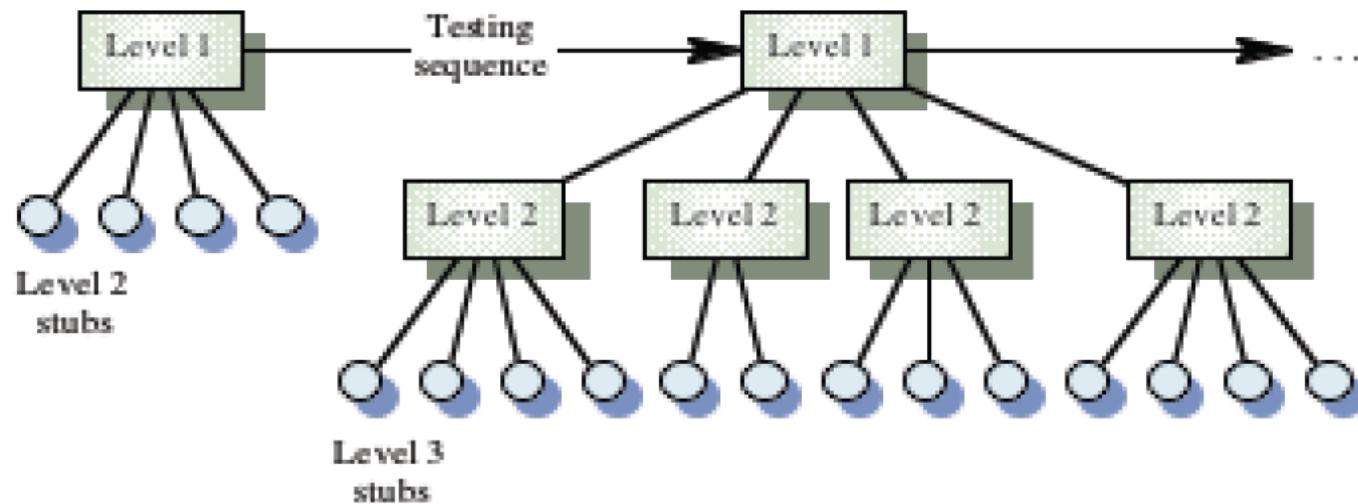
CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

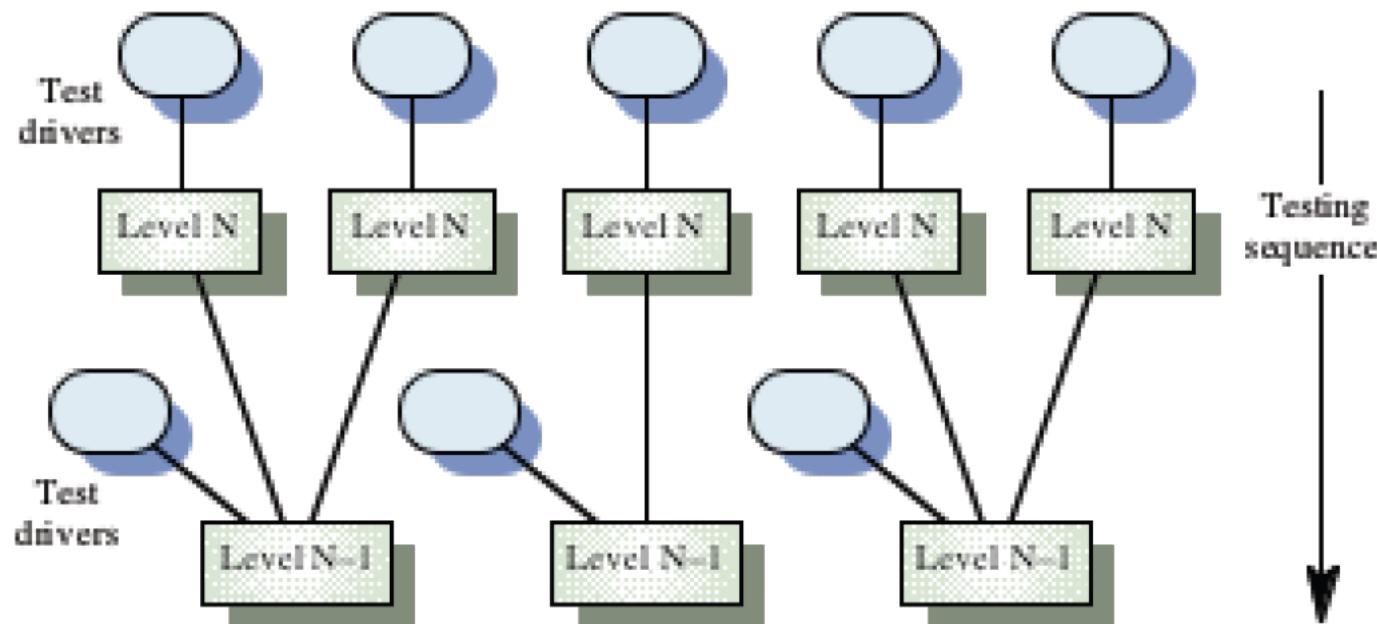
- **Tipos de testes quanto ao escopo**
 - Testes de Unidade (componente)
 - Conhecidos como testes em ponto pequeno
 - São testadas as unidades individuais – funções, objetos, componentes.
 - Testes de Integração (sistema)
 - Conhecidos como testes em ponto grande
 - Os componentes são integrados e o conjunto maior é testado – módulo e sub-sistemas.

- **Testes de integração Top-down**

- Começa pelos componentes de alto nível e vai descendo na hierarquia de componentes, de acordo com a arquitetura do software.
- Os sub-componentes são representados por *stubs*.
- Stubs tem a mesma interface, mas não precisa ter funcionalidade. Basta retornar valores esperados.



- **Testes de integração Bottom-up**
 - Integra unidades já testadas em módulos de mais alto nível, de acordo com a arquitetura.





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Manutenção de Software



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Manutenção de Software**

- Alterações efetuadas no software depois de sua liberação.
- As alterações ocorrem por diversas razões.
- As razões para as alterações determinam a categoria de manutenção.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

• **Categorias de Manutenção**

- Identificar e Corrigir Erros (*Manutenção Corretiva*).
- Adaptar o Software ao Ambiente (*Manutenção Adaptativa*).
- Atender Pedidos do Usuário para Modificar Funções Existentes, Incluir Novas Funções e Efetuar Melhoramentos Gerais (*Manutenção Perfectiva*).
- Melhorar a manutenibilidade ou confiabilidade futuras e fornecer uma base melhor para futuros melhoramentos (*Manutenção Preventiva*)



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **A Fase de Manutenção de Software**
 - Fase mais problemática do Ciclo de Vida de Software
 - Pode despende mais de 70% de todo esforço de uma Organização
 - Esses sistemas devem continuar rodando e as alterações são inevitáveis



CEFET/RJ – Uned Nova Friburgo

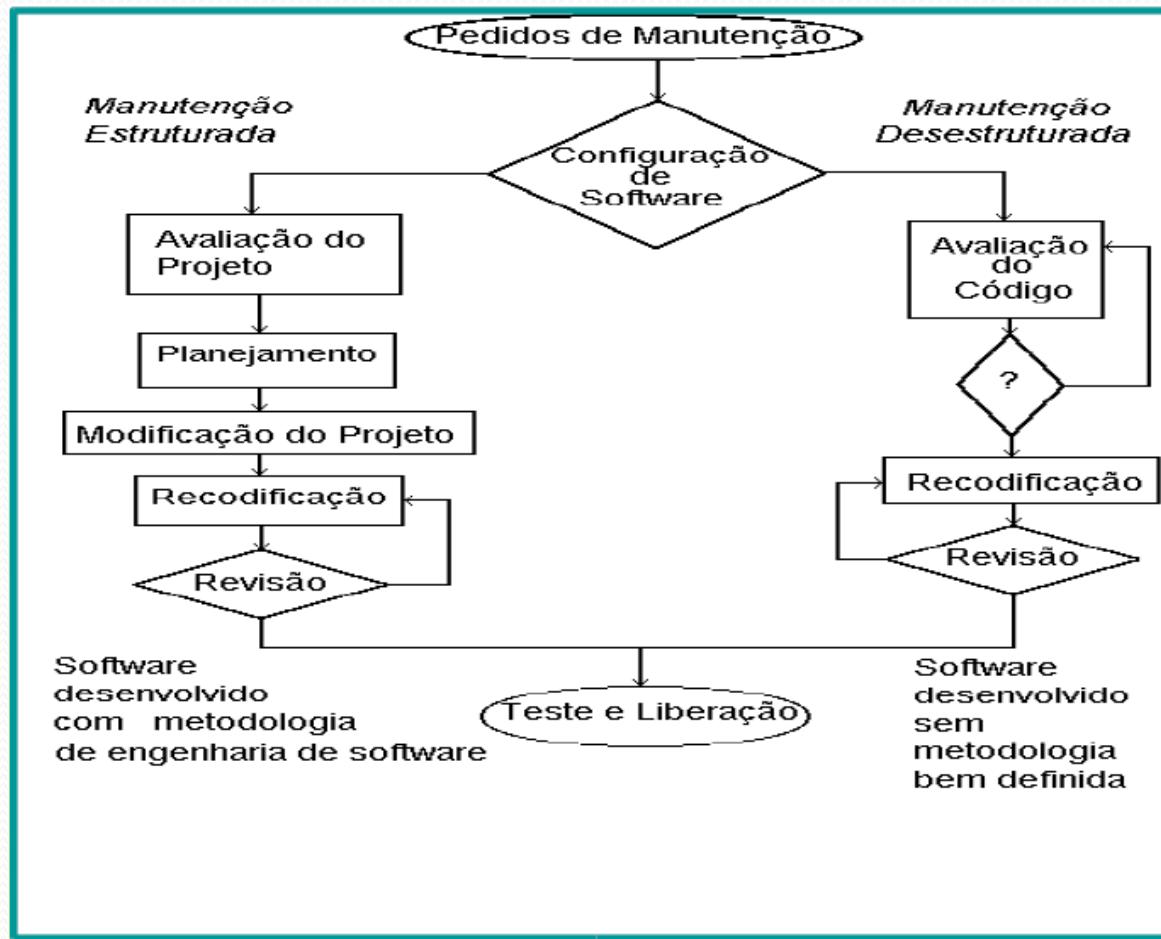
Introdução a Engenharia de SW

- **A Fase de Manutenção de Software**
 - Por que é exigida tanta Manutenção e por que é despendido tanto Esforço nessa atividade?
 - Melhoramentos para atender novas necessidades
 - Nenhuma preocupação com a arquitetura global
 - Codificação, lógica e documentação ruins

CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- Manutenção Estruturada x Não-Estruturada





CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Custo de Manutenção**

- Adiamento de oportunidades de novos desenvolvimentos
- Redução da qualidade global do software
- Insatisfação do cliente
- Insatisfação do pessoal de manutenção
- Diminuição dramática na produtividade.
- tentar entender o que o software faz
- interpretar as estruturas de dados, as características de interface e limites de desempenho
- analisar, avaliar, projetar, codificar e testar as modificações



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

• Problemas da Manutenção

- A maioria dos problemas com a manutenção do software é causada por deficiências na maneira como o software foi planejado, desenvolvido e documentado.
- É difícil ou impossível traçar a evolução do software através das várias versões. As alterações não são adequadamente documentadas.
- É difícil ou impossível rastrear o processo através do qual o software foi criado.



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

• Problemas da Manutenção

- É muito difícil entender programas "de outras pessoas".
- "As outras pessoas" frequentemente não estão presentes para explicar.
- A documentação não existe, é incompreensível ou está desatualizada.
- A maioria dos softwares não foram projetados para apoiar alterações.
- A manutenção não é vista como um trabalho "glamoroso".



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Manutenibilidade**
 - A Manutenibilidade pode ser definida qualitativamente como a facilidade com que o software pode ser entendido, corrigido, adaptado e ou melhorado.
 - A manutenibilidade é afetada por muitos fatores:
 - *cuidado inadequado com o projeto, codificação e teste*
 - *configuração de software ruim*
 - *disponibilidade de pessoal qualificado de software*



CEFET/RJ – Uned Nova Friburgo

Introdução a Engenharia de SW

- **Medidas Quantitativas de Manutenibilidade**

- *tempo de reconhecimento do problema*
- *tempo de demora administrativa*
- *tempo de coleta de ferramentas de manutenção*
- *tempo de análise do problema*
- *tempo de especificação da alteração*
- *tempo de correção ou modificação*
- *tempo de teste local e global*
- *tempo de revisão da manutenção*