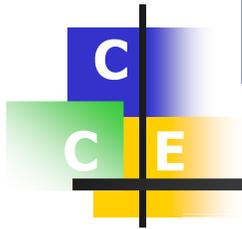
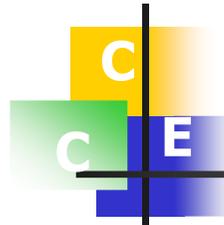


Tecnologias Atuais de Desenvolvimento de *Software*



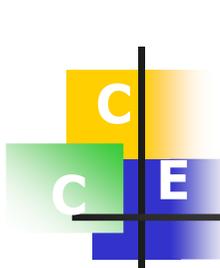
Evolução dos Processos de Desenvolvimento de *Software*

Prof. Luiz Antônio
lpereira@uninet.com.br



Agenda

- Processo de *Software*
 - Conceitos
 - Evolução dos Processos de *Software*
 - Modelos de Processos



Processo de *Software*

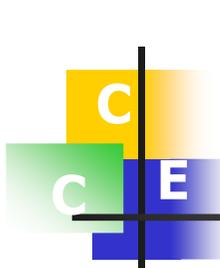
Conceitos

- *Software*
 - Produtos projetados e construídos pelos Engenheiros de *Software*.
 - Abrangem programas de computador, documentos (tangíveis e eletrônicos), dados e modelos (representação de conceitos através de figuras).

Processo de *Software*

Conceitos

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



Processo de *Software*

Conceitos

- Processo (re-arrumando as palavras):
 - É a execução de sequências* de operações visando à evolução de algo, ou seja, à transformação de algo em algo melhor ou mais útil por meio da aplicação de métodos e técnicas.

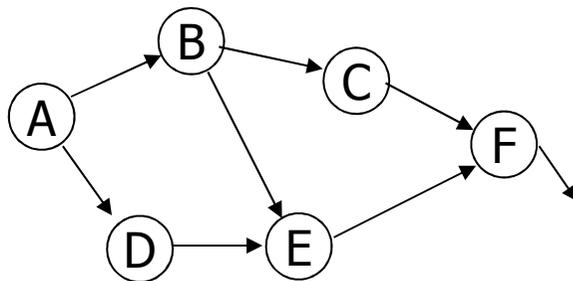
* sequências podem compreender execução de operações parcialmente ordenadas (*grafo* de execução)

Processo de *Software*

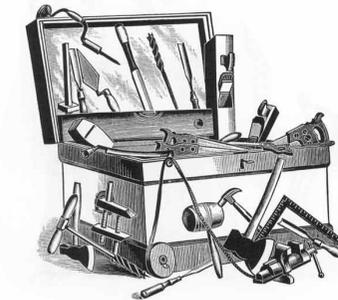
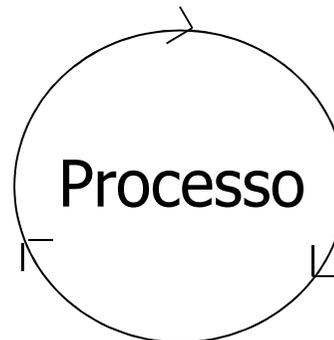
Conceitos

Processos envolvem:

Pessoas com habilidades, treinamento e motivação



Procedimentos e métodos definindo o relacionamento entre as tarefas



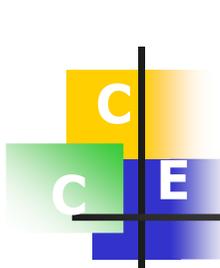
Ferramentas e equipamentos adequados

Processo de *Software*

Conceitos

de construção

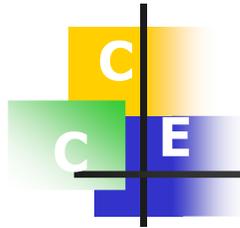
- Processo de *Software*:
 - Um conjunto de atividades, métodos, práticas e transformações que pessoas empregam para definir, desenvolver e manter *software* (lembrar que estão incluídos o plano do projeto, documentos do projeto, código, casos de teste, manuais do usuário ...)



Processo de *Software*

Conceitos

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



Processo de *Software*

Conceitos

- Um processo de *software* possui uma série de tarefas definidas executadas de forma definida.
 - Tarefas diferentes são organizadas de formas diferentes, de acordo com o *modelo de processo* adotado;
 - Tarefas demandam marcos para documentação e garantia da qualidade.

Processo de *Software*

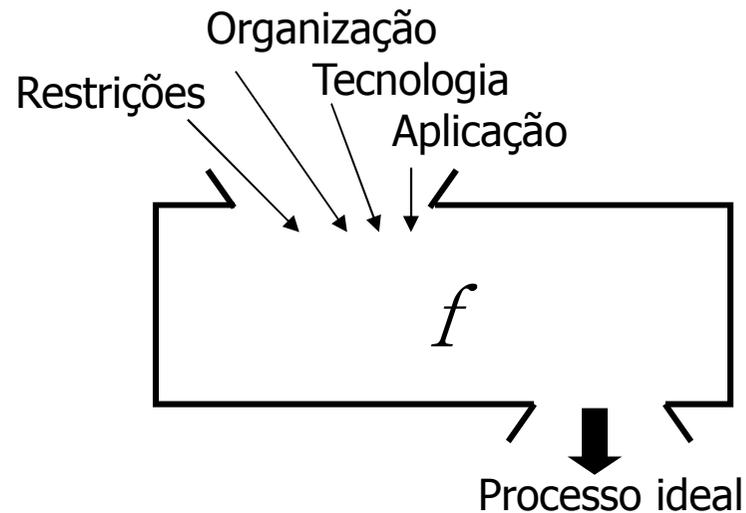
Conceitos

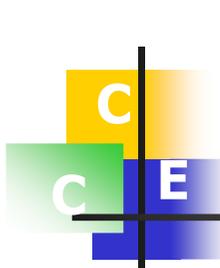
- Qualquer modelo de processo (ou, simplesmente, processo) envolve as fases genéricas:
 - Definição: **o que** fazer para atender às necessidades dos **clientes e usuários**;
 - Desenvolvimento: **como** fazer;
 - Manutenção: modificar
 - corrigir,
 - adaptar,
 - evoluir,
 - prevenir.

Processo de *Software*

Conceitos

- Nenhum processo é perfeito para:
 - Qualquer organização;
 - Qualquer tecnologia;
 - Qualquer aplicação.

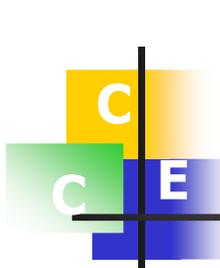




Processo de *Software*

Conceitos

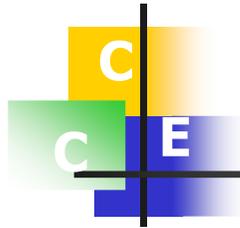
- Processos se modificam:
 - Devem ser complementados e *personalizados* para passarem à prática nas organizações produtoras de *software*;
 - Devem ser adicionados padrões que cubram aspectos específicos:
 - das aplicações;
 - da tecnologia;
 - dos métodos gerenciais;
 - da cultura da organização.
-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



Processo de *Software*

Conceitos

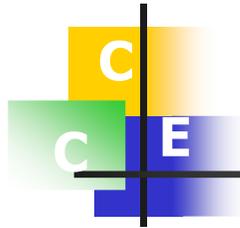
- Personalizar um processo de desenvolvimento, a partir de um processo padrão, significa, portanto, adaptá-lo afim de se considerar:
 - as particularidades da organização,
 - as particularidades da equipe de desenvolvimento e
 - as características funcionais e não funcionais do produto a ser gerado.



Processo de *Software*

Conceitos

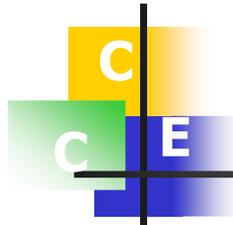
- Mesmo os processos comerciais requerem esforço considerável de adaptação;
- O time de *Engenharia de Processos* de uma organização trata as questões relativas aos processos da organização.
 - Padroniza, desenvolve, discute, documenta, divulga, afere, adapta, ...



Processo de *Software*

Evolução dos Processos de *Software*

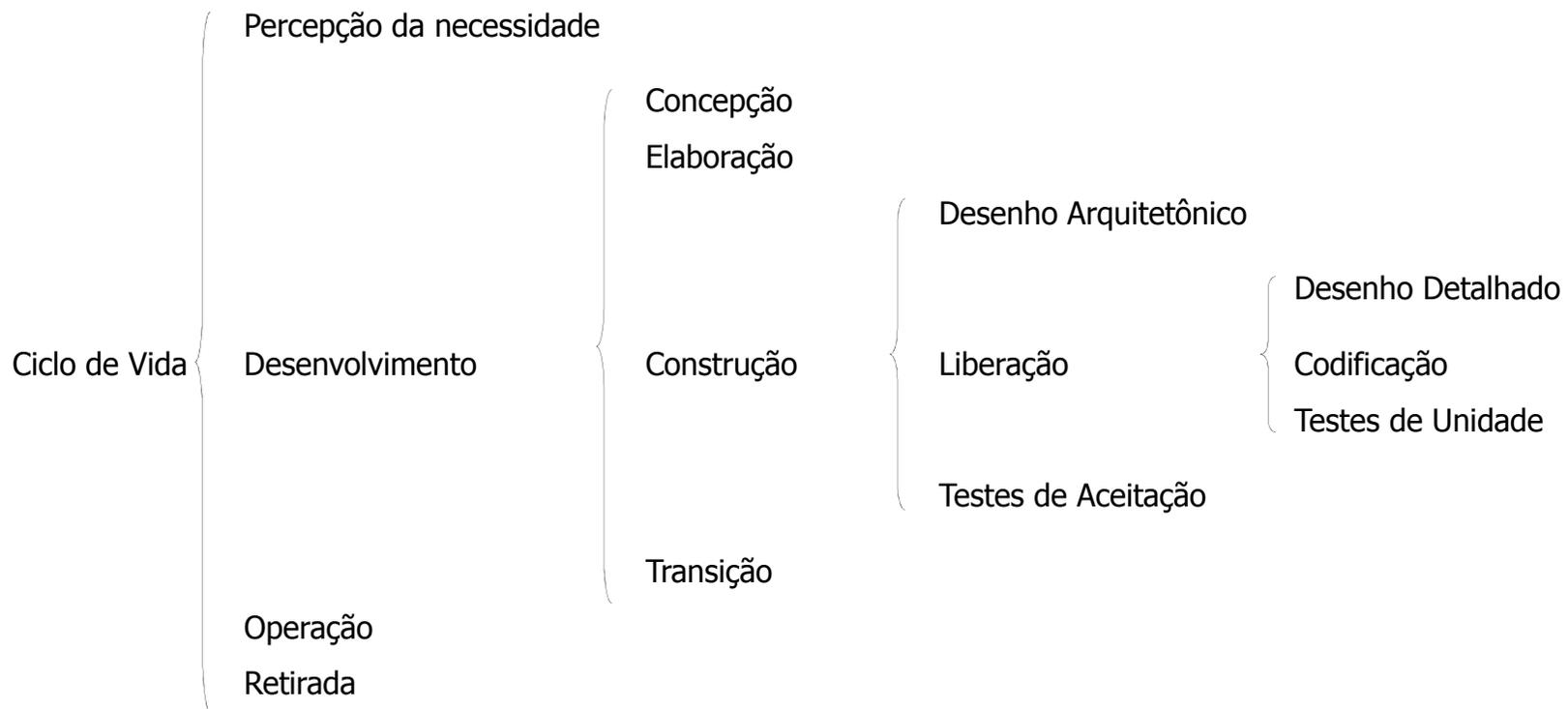
- Processos (de uma maneira geral) evoluem conforme o grau de maturidade das organizações;
- Processos de software evoluem conforme
 - o grau de maturidade com que as organizações gerenciam os artefatos associados ao software e
 - o grau de maturidade com que as organizações gerenciam as diversas atividades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

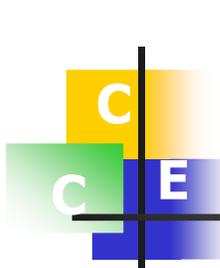


Processo de *Software*

Evolução dos Processos de *Software*

Ciclo de vida (só um parêntesis):





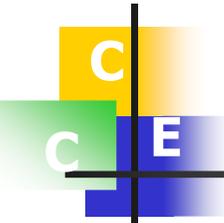
Processo de *Software*

Evolução dos Processos de *Software*

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

(*) CMM: Modelo de Maturidade da Capacidade

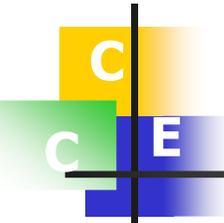
(**) Melhoria de Processo do S/W Brasileiro



Processo de *Software*

Evolução dos Processos 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*.



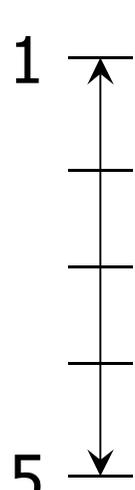
Processo de *Software*

Evolução dos Processos de *Software*

- No SW-CMM (cont.):
 - Os níveis são patamares bem definidos conduzem a processos mais maduros de *software*.
 - Cada patamar compreende um conjunto de objetivos e compromentimentos (KPAs – *Key Process Areas*) que devem ser satisfeitos.

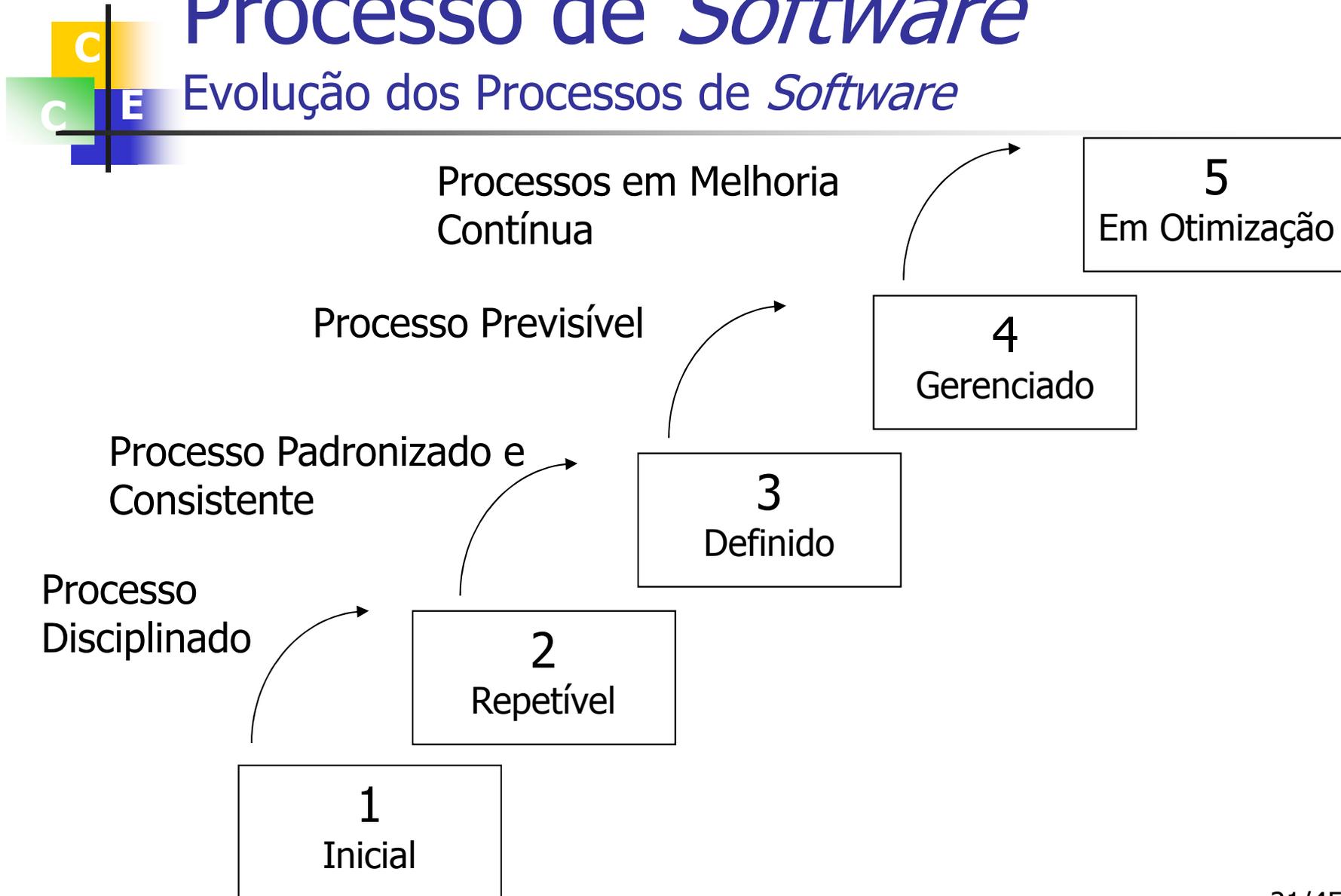
Processo de *Software*

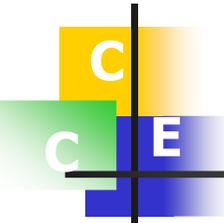
Evolução dos Processos de *Software*

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

Processo de *Software*

Evolução dos Processos de *Software*





Processo de *Software*

Evolução dos Processos de *Software*

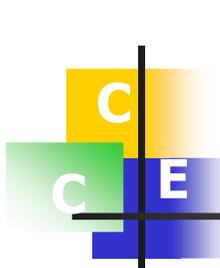
- Níveis:

- 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.



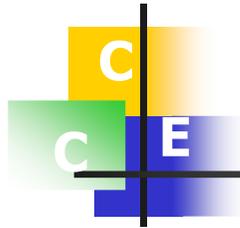
Processo de *Software*

Evolução dos Processos de *Software*

- Níveis (cont.):

- 3 - Definido

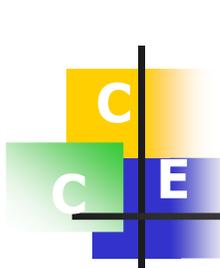
- O processo de software para as atividades de gerência e engenharia estão documentados, padronizados e integrados no processo de desenvolvimento de software da organização.
 - Todos os projetos usam uma versão documentada e aprovada do processo da organização para desenvolvimento e manutenção. Inclui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



Processo de *Software*

Evolução dos Processos de *Software*

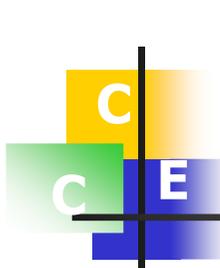
- Níveis (cont.):
 - 4 - Gerenciado
 - Medições detalhadas do processo e do produto são coletadas. Ambos são quantitativamente compreendidos e controlados através de métricas minuciosas. Inclui o nível 3.
 - 5 - Em Otimização (ou Otimizante)
 - Um processo contínuo de melhoria baseado nos resultados quantitativos de outros projetos e em testes de novas idéias e tecnologias está estabelecido. Inclui o nível 4.



Processo de *Software*

Evolução dos Processos de *Software*

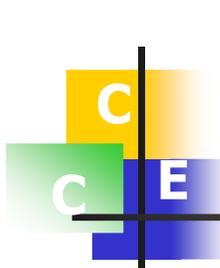
- Exemplo:
 - KPAs para nível 2:
 - Gerência dos Requisitos
 - Planejamento do Projeto de Desenvolvimento
 - Controle do Projeto de Desenvolvimento
 - Gerência da Aquisição de Software
 - Garantia da Qualidade
 - Gerência de Configuração



Processo de *Software*

Evolução dos Processos de *Software*

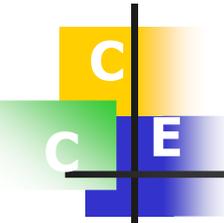
- No mps.BR, em linhas gerais:
 - Foco na micro, pequena e média empresas de desenvolvimento de s/w.
 - Modelo tem base em normas ISO e no CMMI → também se baseia no conceito de maturidade de processo.



Processo de *Software*

Evolução dos Processos de *Software*

- O modelo MPS está descrito por meio de documentos em formato de guias:
 - Guia Geral: contém a descrição geral do modelo MPS.
 - Guia de Aquisição: descreve um processo de aquisição de software e serviços correlatos.
 - Guia de Avaliação: descreve o processo e o método de avaliação e os requisitos para avaliadores e instituições avaliadoras.
 - Guia de Implementação: série de 10 documentos que fornecem orientações para implementar nas organizações os níveis de maturidade descritos no Modelo de Referência MR-MPS.



Processo de *Software*

Evolução dos Processos de *Software*

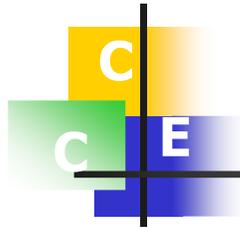
- O MR-MPS define **sete** níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado).
- O alcance de um determinado nível se obtêm quando são atendidos os propósitos, os resultados esperados dos processos e os resultados estabelecidos para o nível.

Processo de *Software*

Modelos de Processos

Hacking (codifica-remenda):

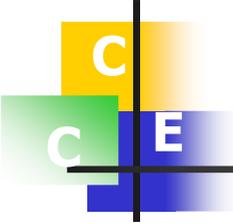
- Prática frequente para o desenvolvimento de s/w (☹);
- “Atropela” as diversas fases para a solução do problema;
- Consiste em sentar-se em frente ao computador e começar a escrever código, sem nenhum projeto.
- **Não é um processo de *software*...** muito menos um modelo de processo.



Processo de *Software*

Modelos de Processos

- Serão vistos rapidamente:
 - Cascata (Clássico)
 - Prototipação
 - RAD
 - Espiral
- Em maiores detalhes, nas próximas aulas:
 - RUP
 - Processos ágeis



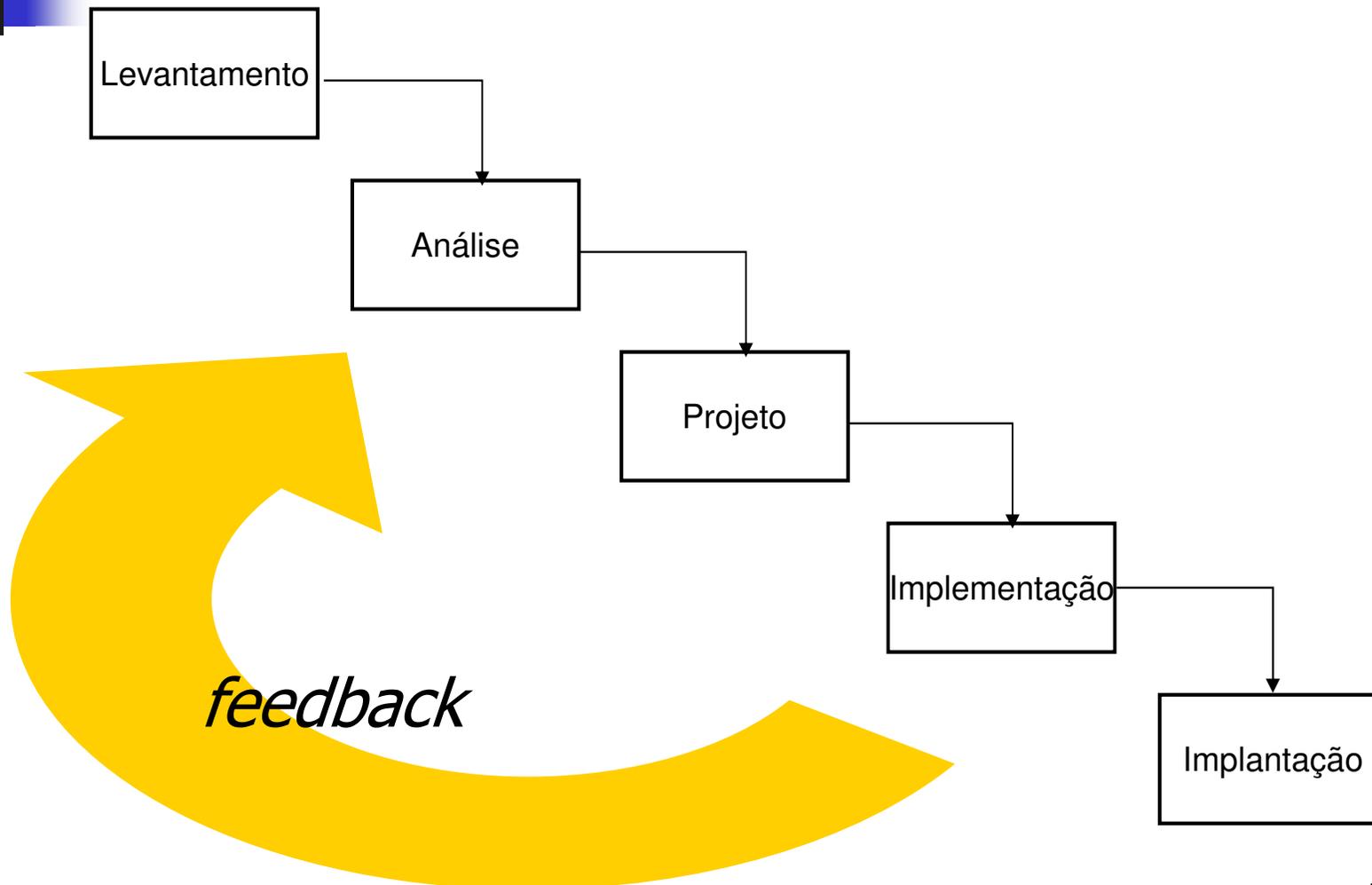
Processo de *Software*

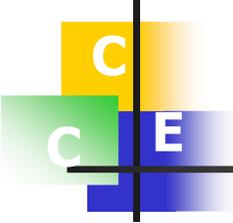
Modelos de Processos - Cascata

- Principais características:
 - 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.

Processo de *Software*

Modelos de Processos – Fases do Cascata





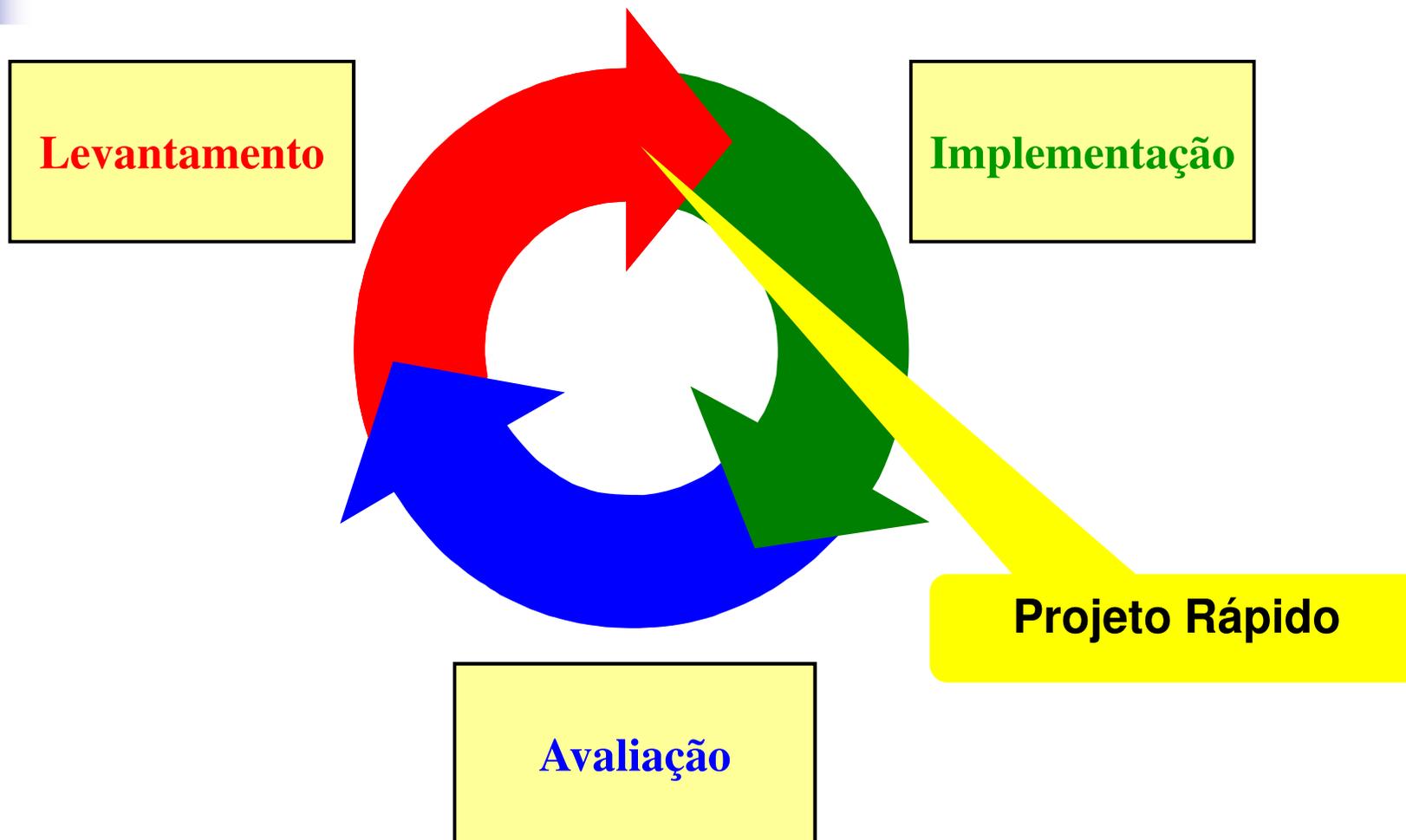
Processo de *Software*

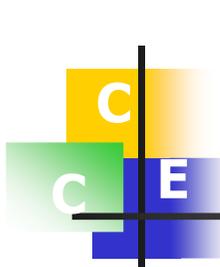
Modelos de Processos - Prototipação

- Principais características:
 - 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 = “1o. Sistema” (alguns autores recomendam que o joguemos fora).

Processo de *Software*

Modelos de Processos - Prototipação

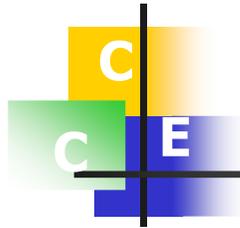




Processo de *Software*

Modelos de Processos – 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.



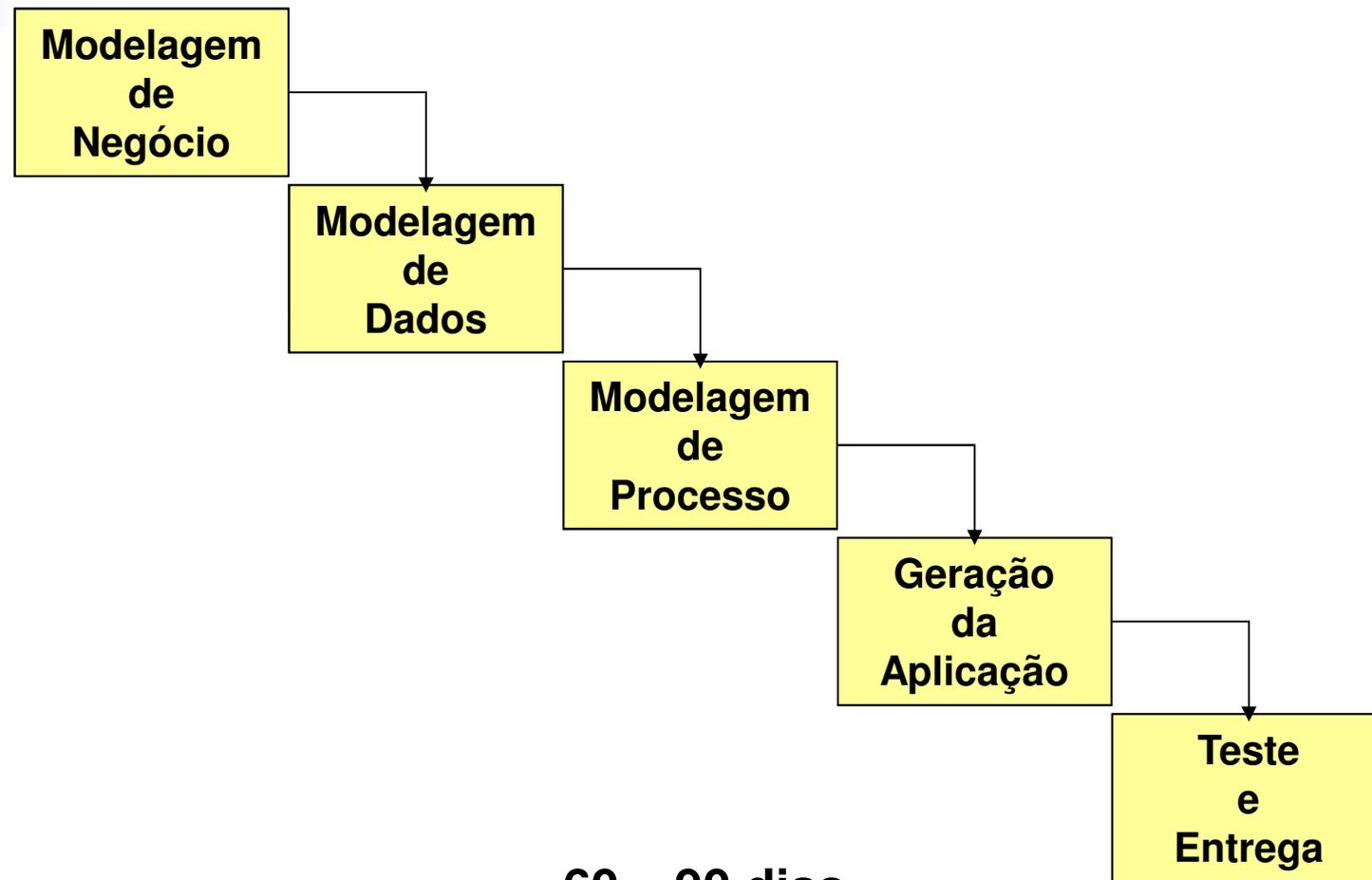
Processo de *Software*

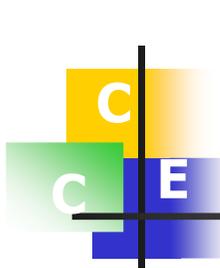
Modelos de Processos - RAD

- RAD = Rapid Application Development
- Principais características:
 - 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).

Processo de *Software*

Modelos de Processos – Fases do RAD

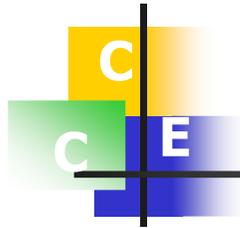




Processo de *Software*

Modelos de Processos – 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.



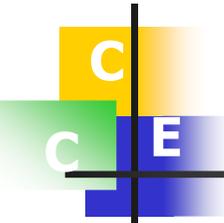
Processo de *Software*

Modelos de Processos - Espiral

- Características:

- 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;

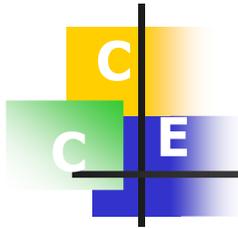
(*)"A Spiral Model for Software Development and Enhancement",
1988



Processo de *Software*

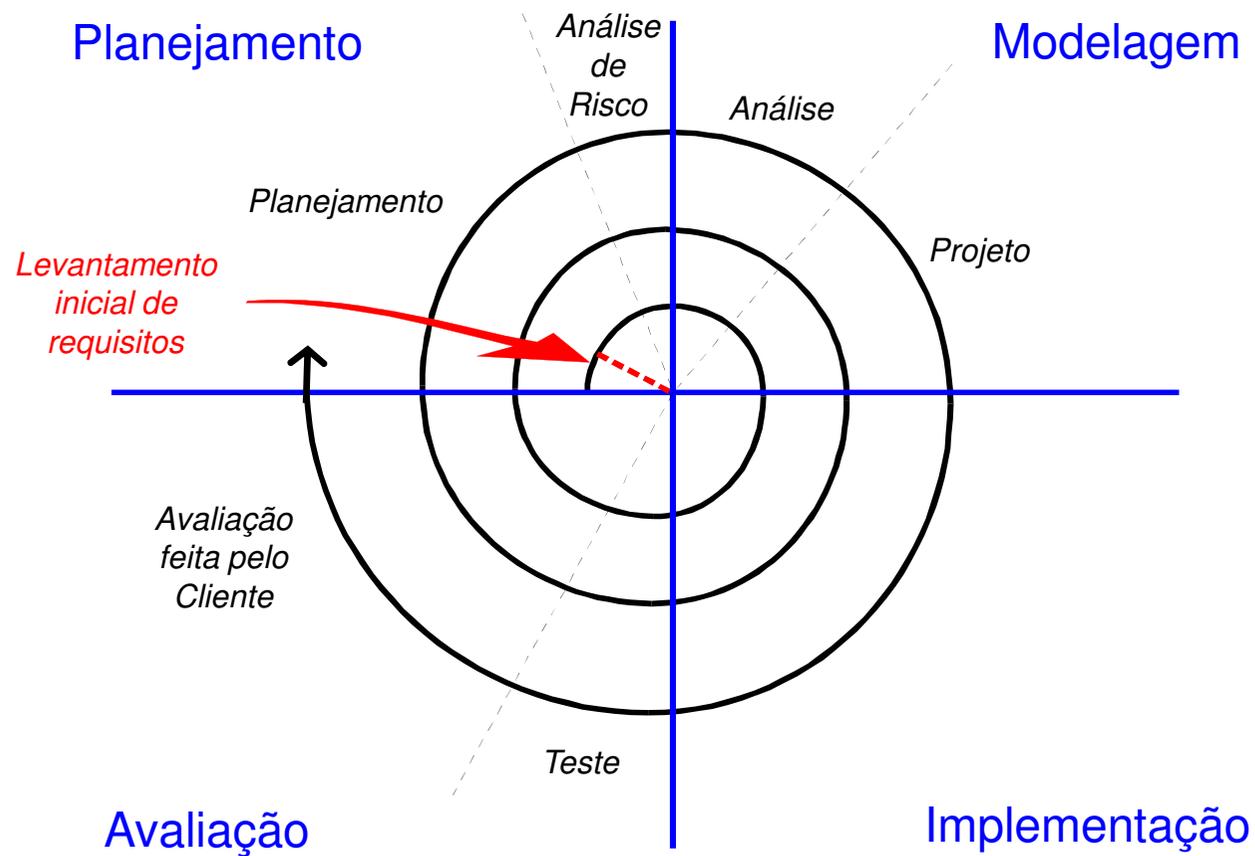
Modelos de Processos - Espiral

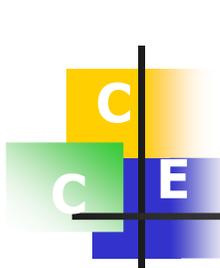
- Características (cont.):
 - Incremental, evolutivo, iterativo;
 - Espiral é dividida em uma série de regiões;
 - Cada região contempla uma série de tarefas que são adaptadas às características do projeto a ser conduzido;
 - Um ciclo da espiral pode produzir, tanto uma especificação, como versões do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



Processo de *Software*

Modelos de Processos - Espiral



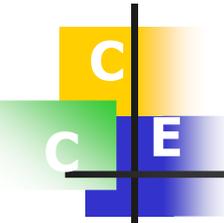


Processo de *Software*

Modelos de Processos - Espiral

- **Vantagens:**

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



Processo de *Software*

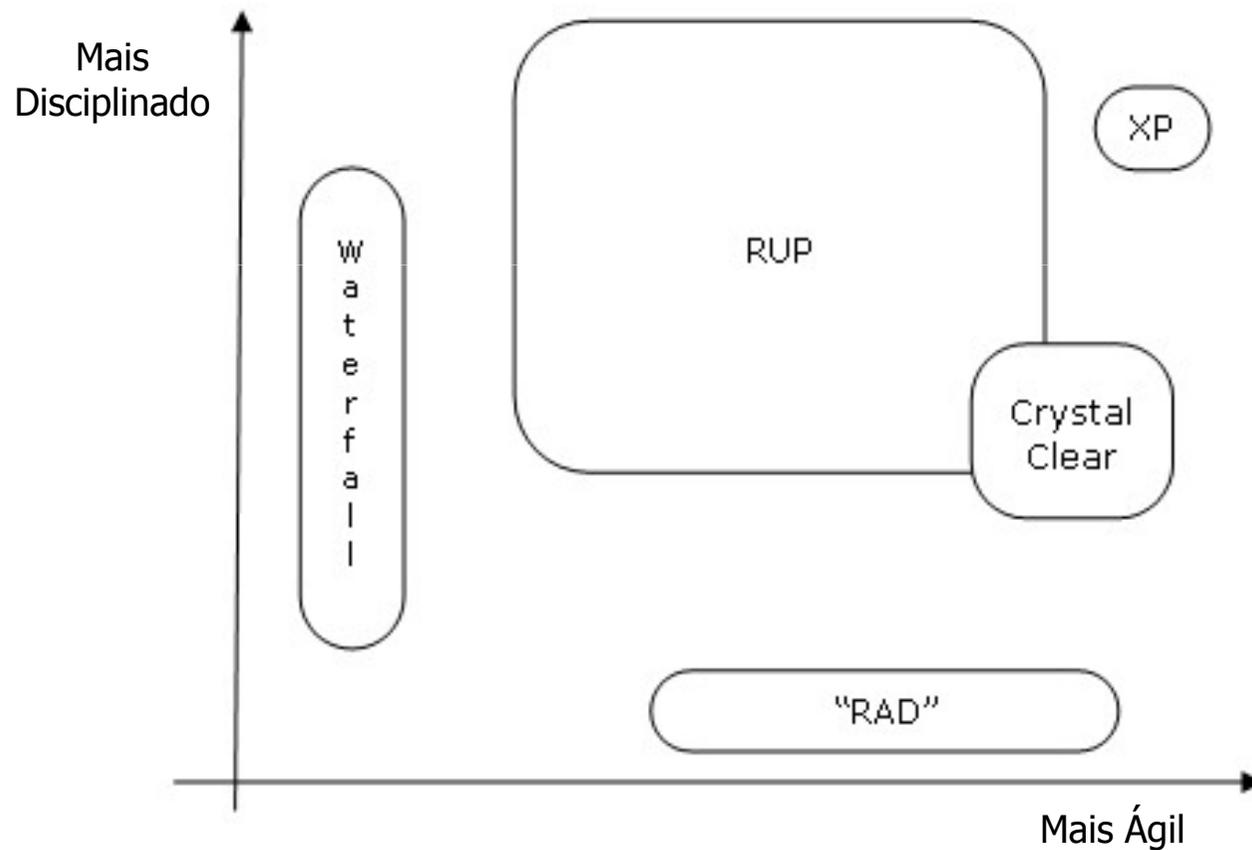
Modelos de Processos - Espiral

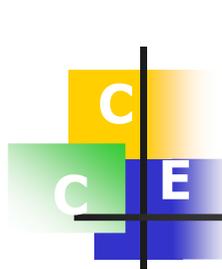
- Problemas:

- 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;
- Ainda não foi tanto utilizado.

Processo de *Software*

Modelos de Processos – Alguns Processos Comparados





Processo de *Software*

Lembrete

Próxima aula: UP/RUP