

## *Engenharia de Software*

### **Tema da Aula**

### Conceitos Introdutórios de Engenharia de Software

*Prof. Cristiano R R Portella*

portella@widesoft.com.br



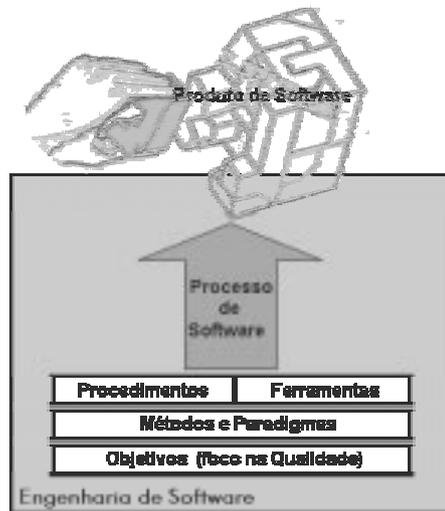
## Conceitos Introdutórios Camadas da ESw



ESw é uma tecnologia em camadas.

Como todo enfoque de Engenharia, tem um comprometimento com a Qualidade e a eficiência dos processos.

O cerne da ESw é a camada de processo. O processo é a cola que une as demais camadas e permite um desenvolvimento racional de Sw.



### **Metodologia**

Conjunto de métodos (técnicas) e processos.

Por exemplo: Metodologia Estruturada, Metodologia Orientada a Objeto etc

**Método**

Caminho para se atingir um objetivo (modo de proceder).

Por exemplo:

Método de Análise Estruturada, Método de Projeto Estruturado, etc.

**Processo**

Maneira pela qual se realiza uma operação, segundo determinado método ou norma.

Por exemplo:

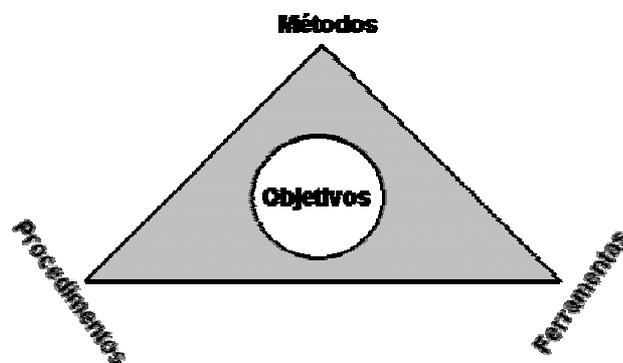
Processo de detecção do fluxo da informação

**Técnica**

Aplicação prática do processo;

Por exemplo:

Conhecer o processo alvo do sistema e levantar as transformações da informação



**Objetivos:**

A exigência de sólidos princípios de engenharia pressupõe como objetivos:

- Rigor no Processo (qualidade e eficiência)
- Medição do processo
- Repetibilidade
- Otimização de recursos
- Base científica

**Métodos:**

Detalhamento de “como fazer”, utilizando-se os princípios de engenharia.

Métodos envolvem amplo conjunto de tarefas que incluem:

- Planejamento
- Codificação
- Estimativa de Projeto
- Testes
- Análise de Requisitos
- Manutenção etc .....

Métodos geralmente introduzem uma notação gráfica, uma terminologia especial e um conjunto de critério para sua aplicação.

**Ferramentas:**

Proporcionam apoio automatizado ou semi-automatizado à aplicação dos métodos.

Quando integradas (o produto gerado por uma ferramenta serve como input para a ferramenta seguinte), obtém-se um ambiente de suporte ao desenvolvimento de software, chamado de “Engenharia de Software Auxiliada por Computador” (CASE-Computer-Aided Software Engineering).

**Procedimentos:**

Constituem o elo de ligação entre Métodos e Ferramenta, permitindo o desenvolvimento racional de software.

Procedimentos definem a seqüência em que os métodos serão aplicados, os produtos que devem ser gerados, os controles a serem estabelecidos e os marcos de referência (milestone) que permitem a avaliação do processo.

**Paradigmas:**

São abordagens (modelos) preconizados pela EWs.

Paradigma: (sm) Modelo, padrão (Aurélio-2a Ed)

Não existe “uma” abordagem em particular, que seja a melhor para qualquer problema de software.

O desenvolvedor deve escolher um paradigma ou combinar vários deles, de forma a obter a potencialidade de cada um.

**Paradigmas:**

Escolher métodos abrangentes para todas as fases de desenvolvimento, as melhores ferramentas para automatizar os métodos, blocos de construção mais poderosos, melhores técnicas para a garantia da qualidade e uma filosofia de coordenação predominante, controle e administração - essa “disciplina” (rigor) para desenvolvimento de software chama-se **Engenharia de Software**.

**Paradigmas:**

Um determinado paradigma é escolhido baseado em:

- Natureza do projeto
- Área de aplicação
- Métodos e ferramentas disponíveis
- Controles necessários
- Produtos gerados em cada fase

**Paradigmas:**

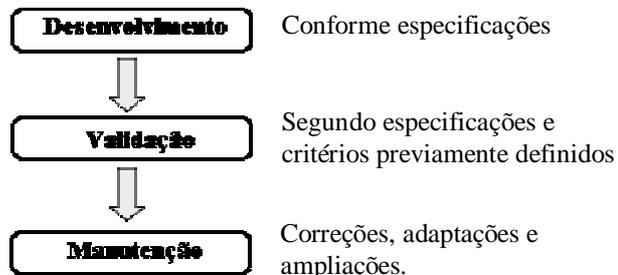
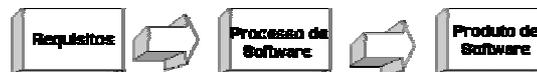
- ✓ Modelo de ciclo de vida clássico (cascata ou linear)
- ✓ Modelo em Prototipação
- ✓ Modelo em espiral
- ✓ Modelo RAD (Rapid Application Development)
- ✓ Modelo incremental
- ✓ Modelo em espiral WINWIN
- ✓ Modelo dos métodos formais
- ✓ Desenvolvimento baseado em componentes
- ✓ Técnicas de 4ª geração
- ✓ etc

### Processo de Software:

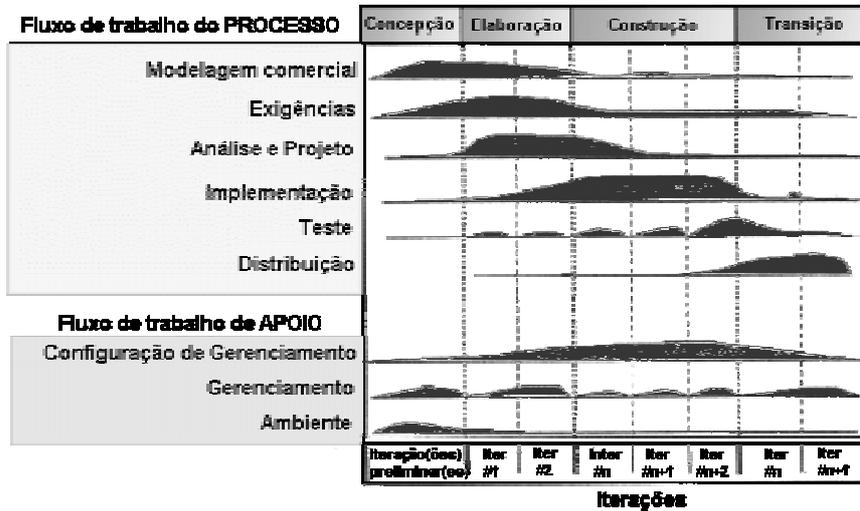
Consiste em atividades, métodos e práticas úteis no desenvolvimento do “produto de software”.



### Processo de Software:



Processo de Software (RUP - Rational Unified Process):



**Produto de Software** (NORMA NBR ISO 9000-3 - Introdução)

Com o progresso da tecnologia da informação, a quantidade de software vem crescendo e tornando essencial a gestão da qualidade de produtos de software.

...

**Entretanto, o processo de desenvolvimento e manutenção de software é diferente da maioria dos demais tipos de produtos industriais,** tornando necessário prover, nesse campo da tecnologia de desenvolvimento tão rápido, orientações adicionais para o estabelecimento de sistemas da qualidade onde estejam envolvidos os produtos de software, levando-se em conta o estágio atual da tecnologia.

A natureza do desenvolvimento de software é tal, que algumas atividades estão relacionadas às fases específicas do processo de desenvolvimento, enquanto outras podem ser aplicadas ao longo de todo o processo

**Produto de Software:**

Um produto de software é um programa de computador combinado com os itens que o tornam inteligível, utilizável e extensível. (Brooks, 1975)

Software é um elemento de sistema lógico, e não físico. Consistem de Instruções, estruturas de dados e documentação (Pressman, 92-2000)

**Produto de Software:**

Um produto de software é constituído de itens executáveis em máquina e itens não executáveis em máquina (Pressman)

- Código
- Estruturas de dados
- Documentos (técnicos, de instrução, normativos etc)

( pessoal habilitado)

**Produto de Software:**

- Genéricos (pacote)
  - Custo
  - Adequação
  - Prazo para uso (disponível)
- Específicos (taylor-made)
  - Custo
  - Adequação
  - Prazo para uso

**Produto de Software (segundo sua aplicação):**

Software básico:

Utilitários, ferramentas de desenvolvimento,  
gerenciadores de métodos de acesso

Software de tempo real:

Monitora/analisa/controla eventos do mundo real.

- Dispositivos de coleta de dados
- Análise (lógica e set points)
- Dispositivos de atuação (saída - feedback)
- Monitor (coordenação e controle do tempo-1ms)

**Produto de Software (segundo sua aplicação):**

Software de processamento comercial:

Sistemas de informação vulgarmente chamados de “sistemas” ou aplicativos. Foco na informação e na automação de rotinas administrativas.

Software científico e de Engenharia:

Aplicações com algoritmos voltados ao processamento de cálculos de média/alta complexidade (processamento superescalar, vetorial, computação gráfica)

**Produto de Software (segundo sua aplicação):**

Software embarcado/embutido (embedded):

Produtos para controle de processos de baixa complexidade (funções limitadas) como autotrônica, eletrodomésticos, até alta complexidade como sistemas de aviãoica. Geralmente são residentes em memórias read-only e executados em processador dedicado.

Software de Inteligência Artificial:

Algoritmos não-numéricos que trabalham na resolução de problemas complexos que não são favoráveis a análise direta (algoritmo prévio) e lógica booleana.

**Produto de Software (segundo sua aplicação):**

Software de Inteligência Artificial:

- Sistemas especialistas  
(bases de conhecimento e máquina de inferência)
- Redes neurais artificiais
- Lógica fuzzy (fuzzi sets – lógica nebulosa)

**NBR ISO 9000-3 Definições:**

3.1 Software: criação intelectual compreendendo os programas, procedimentos, regras e qualquer documentação correlata à operação de um sistema de processamento de dados.

3.2 Produto de Software: Conjunto completo de programas de computador, procedimentos e documentação correlata, assim como dados designados para entrega a um usuário.

3.3 Item de Software: Qualquer parte identificável de um produto de software em etapa intermediária ou na etapa final do desenvolvimento.

3.4 Desenvolvimento: Todas as atividades a serem realizadas para a criação de um produto de software.

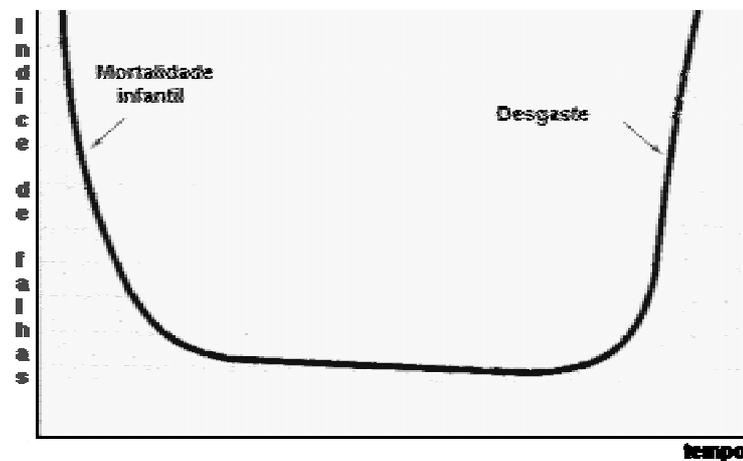
Características do software (Pressman):

Software é desenvolvido e não manufaturado (no sentido estrito de manufatura).

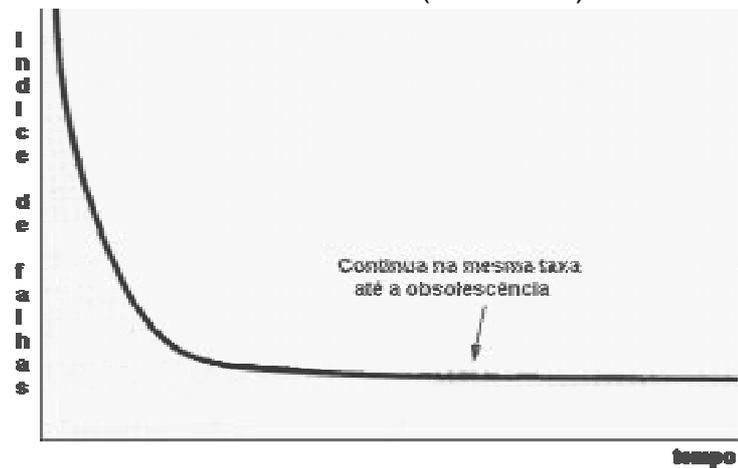
Software não se desgasta, porém fica obsoleto.

Apesar da indústria estar se movendo no sentido de montar software através de componentes, a maior parte dos softwares continuam sendo feito customizados (sob medida).

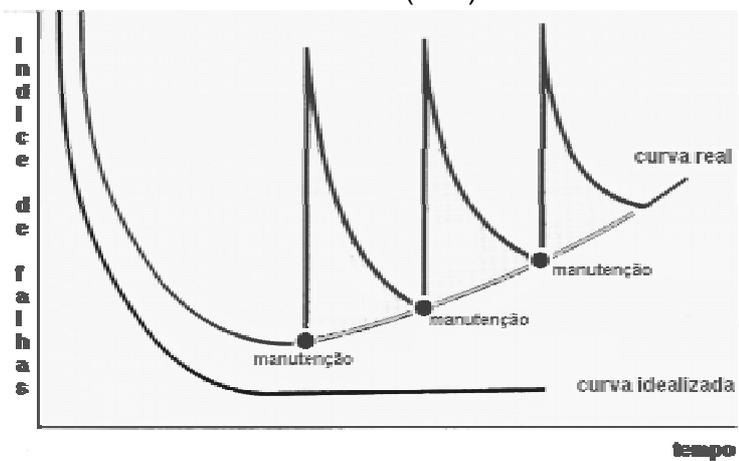
Curva de falhas para o hardware



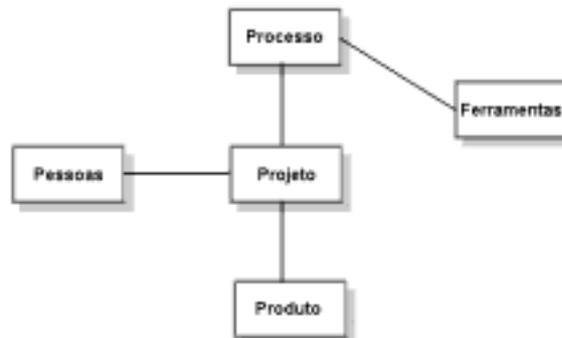
Curva de falhas do software (idealizada)



Curva de falhas do software (real).



### Os quatro P's do desenvolvimento de software



Descrevem algumas propriedades gerais e referem-se tanto ao produto como ao processo de software.

A aplicação dos princípios é feita através de métodos e ferramentas. Os princípios podem ajudar na escolha de metodologias e ferramentas apropriadas para o produto que se deseja construir.

Princípios:

- Formalidade
- Abstração
- Decomposição
- Generalização
- Flexibilização

Desenv.SW é uma atividade criativa e com tal tende a “seguir a inspiração do momento”.

Um enfoque formal pode gerar SW mais confiável e exercer controle sobre seu custo.

O nível de formalidade não deve restringir a criatividade e deve ser adequado à dificuldade conceitual de cada desenvolvimento.

A formalidade estará contida no projeto (descrição formal), na programação (pgms. são componentes formais), nas rotinas de teste, nos procedimentos da instalação etc.

- (Exemplo da Microsoft e a informalidade)

É o processo de identificar os aspectos importantes do produto/processo, ignorando-se os detalhes (Princípio de Pareto).

- Modelos são abstração da realidade.
- O sistema de informação é uma abstração do processo real.
- Linguagens de programação abstraem ao programador, detalhes da máquina e da solução (algoritmo em baixo nível)
- O encapsulamento é uma técnica de abstração para diminuir a complexidade e melhorar a reusabilidade dos objetos.

A decomposição é uma técnica utilizada para permitir que lidemos com a complexidade.

Dividir para conquistar ou o limite da habilidade humana de entender a complexidade ( $5 \pm 2$ ).

- Decomposição do processo
  - Atividades de controle da qualidade
  - Atividades de gerenciamento do cronograma
  - Atividades de gerenciamento de fornecedores externos
  - Atividades de teste
  - Atividades de documentação

Decomposição do produto

- Programas
- Módulos ou rotinas
- Componentes

## Conceitos Introdutórios Princípio da Decomposição

- Programação procedural ⇔ refinamento sucessivo;
- Programação OO ⇔ Programa é um conjunto de objetos que se intercomunicam prestando/solicitando serviços;
- Decomposição do Processo em Sub-Processos;
- Decomposição estrutural do Sistema em módulos/programas;
- Decomposição funcional do Sistema em entradas e correspondentes saídas;

## Conceitos Introdutórios Princípio da Generalização

A generalização é também uma forma de abstração, buscando-se as características comuns e esquecendo-se as características específicas dos itens a serem generalizados.

Uma solução mais genérica tem maior potencialidade de ser reutilizada (reusabilidade e generalização dos componentes).

A generalização é o processo inverso da decomposição (generalização e especialização em OO).

Custo do desenvolvimento voltado à generalização versus benefício da reutilização.

A flexibilização irá conferir ao produto de software a facilidade de adaptação a novos ambientes, a mudanças ocorridas no ambiente, a casos de uso não implementados e às manutenções que se fizerem necessárias.

- ✓ Flexibilização do produto de software:
  - Customização das saídas do sistema;
  - Parametrização do processo e inclusão de variações do algoritmo original;
  - Facilidade em agregar novos módulos/funções
  - Facilidade em porta-lo para outras plataformas

Flexibilização do processo de software:  
Procedimentos e técnicas alternativas no processo de software visando:

- Corrigir erros
- Corrigir atrasos
- Adaptar técnicas de outros paradigmas
- Atender às constantes mudanças de requisitos