

## Linha de Produtos de Software

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>  
[reuso.software@gmail.com](mailto:reuso.software@gmail.com)

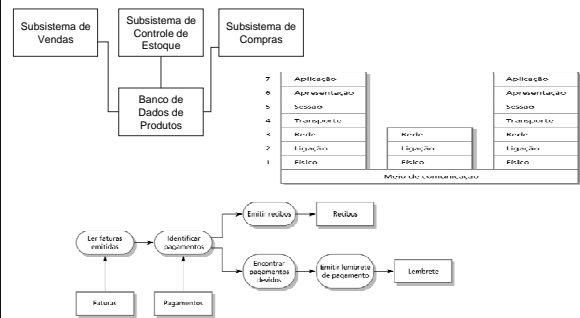
19 Março 2012

## Agenda da Aula

- Padrões arquiteturais
  - Cliente-Servidor
  - Modelo-Visão-Controlê
  - Microkernel
- Linha de produtos de software
  - Motivação e conceitos principais
  - Extração de uma linha de produtos
  - Modelo de características

## Padrões Arquiteturais

## Resumo dos Padrões (Aula 4)



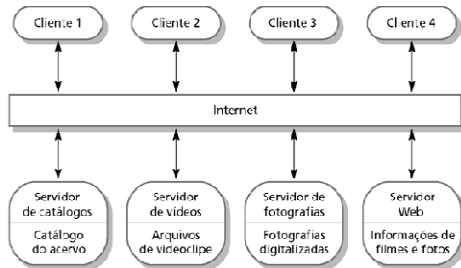
## Arquitetura Cliente-Servidor

- Requer uma estrutura de rede para clientes acessarem os serviços
- Clientes sabem quais os serviços e servidores estão disponíveis
- Servidores não sabem quem são os clientes

## Arquitetura Cliente-Servidor

- Essencialmente, é usado o protocolo *request-reply*
  1. Um cliente faz um pedido ao servidor e espera pela resposta
  2. O servidor executa o serviço e responde ao cliente

## Exemplo de Cliente-Servidor



## Vantagens

- A distribuição de dados é fácil e direta
- Faz uso efetivo dos recursos em rede
  - Possibilita hardware mais barato
- É fácil adicionar novos servidores ou atualizar servidores existentes

## Desvantagens

- Não prevê modelo de dados compartilhado
  - Subsistemas usam diferentes organizações de dados
- Pode haver redundância de serviços em diferentes servidores
- Não prevê registro central de serviços
  - Difícil descobrir quais serviços estão disponíveis

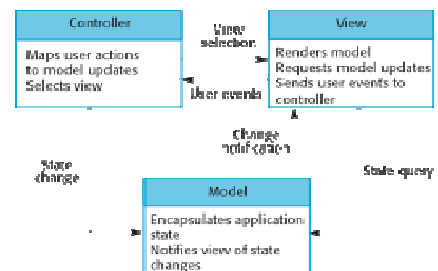
## Padrão MVC

- Largamente utilizado em aplicações interativas Web
- Organiza o sistema em três componentes
  - **Modelo**: contém as funcionalidades e dados principais
  - **Visão**: responsável por apresentar os dados ao usuário
  - **Controlador**: trata os eventos de entrada

## Descrição do MVC

- Separa a apresentação e a interação dos dados do sistema
  - Os três componentes tem responsabilidades distintas mas interagem entre si
- Quando é recomendado?
  - Quando existem várias maneiras de visualizar e interagir com os dados
  - São desconhecidos (ou são voláteis) os requisitos de interação com os dados

## Representação dos Papéis



## Vantagens

- Permite que os dados sejam alterados de forma independente de sua representação (e vice versa)
  - Apóia a apresentação dos mesmos dados de maneiras diferentes
- Facilita a distribuição do componente de visão
  - Os dados são mantidos centralizados e protegidos

## Desvantagens

- Complexidade excessiva quando o modelo de dados e de interações é muito simples
  - A estrutura do padrão pode impor código adicional desnecessário

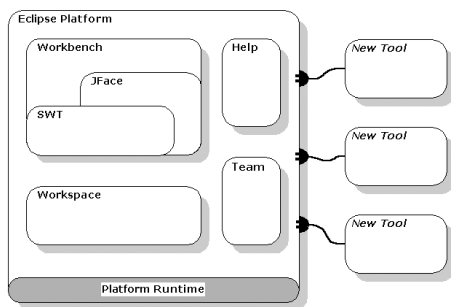
## Microkernel

- Destinado a domínios de sistemas que possuem requisitos muito voláteis
  - Sistemas precisam ser capazes de se adaptar aos requisitos voláteis
- Este padrão de arquitetura separa a funcionalidade mínima em um núcleo
  - Novas funcionalidades são agregadas por extensões na forma de plug-in

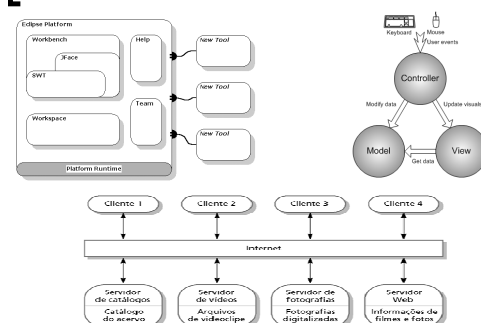
## Responsabilidades

- Microkernel
  - Provê as funcionalidades básicas
  - Oferece o meio de comunicação entre as extensões
  - Gerencia os recursos
- Extensões
  - Inclui novas funcionalidades
  - Fornece uma interface para interação com o núcleo e com outras extensões

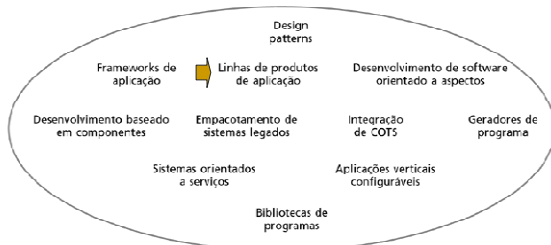
## Exemplo de Microkernel



## Resumo dos Padrões



## Abordagens de Reuso



## Linhas de Produtos de Software (LPS)

## Histórico: Linha de Produção

- Produtos em geral eram feitos manualmente
- Com o crescimento do consumo, foi necessário automatizar a produção
  - No domínio de automóveis, a Ford criou o conceito de **linha de produção**
- Vantagens
  - Produção em massa
  - Baixo custo

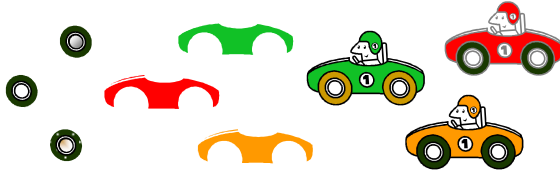
## O problema: diversidade

- No início, a produção em massa atendeu os anseios das pessoas
- Problema
  - Pouca diversidade de produtos



## Customização em Massa

- Para minimizar o problema, surgiu o conceito de customização em massa
  - Alguns itens passaram a ser customizáveis



## Novas Opções de Customização

- A medida que a linha de produção evolui, novos produtos são possíveis



## Vencer a Concorrência

- Empresas de automóveis que trabalhavam com produção e customização em massa
  - Acréscimo de até 35% nas vendas
- Empresas de automóveis que mantinham linhas de produtos separados
  - Perdas de até 7% nas vendas
- O mesmo padrão se repetiu em outras indústrias, com câmeras fotográficas

## Definições

- LPS é um conjunto de aplicações definidas sobre uma arquitetura comum e que compartilham componentes
- SEI
- Engenharia de LPS é um processo de desenvolvimento de aplicações de software usando uma arquitetura comum e customizações

Klaus Pohl

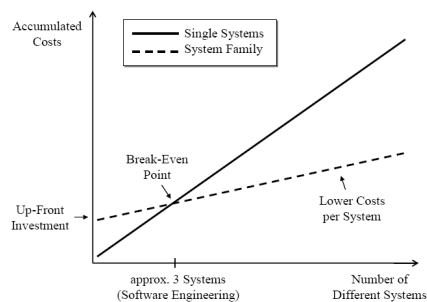
## Propriedades de uma LPS

- LPS permite uma forma mais efetiva de reuso em larga escala
- É geralmente criada a partir de várias aplicações desenvolvidas sobre o mesmo domínio
- Usa outras técnicas de reuso
  - Frameworks, componentes, padrões, etc.

## Motivação

- A principal razão para se criar uma linha de produtos é a redução de custos
  - Entretanto, antes de começar os ganhos de reuso, é necessário um investimento inicial
- Estudos indicam que são necessários de três a quatro produtos de software para se recuperar o investimento

## Gráfico de Custos vs. Produtos



## Melhoria da Qualidade

- Os artefatos de uma LPS são revisados e testados em muitos produtos
  - Estes artefatos tendem a atingir elevado grau de confiabilidade (poucas falhas)
- Artefatos confiáveis elevam a qualidade de todos os produtos da LPS

## Extração de LPS

## Extração da Linha

- Processo de extração de uma linha de produtos
  1. Cria uma aplicação
  2. Cria outra aplicação no mesmo domínio
  3. Reusa algo da primeira aplicação na segunda
  4. Cria uma terceira aplicação e reusa... e assim por diante

## Processo de Desenvolvimento

1. Levantar os requisitos dos usuários
2. Escolher elementos da linha de produtos que atendem de forma aproximada os requisitos
3. Negociar os requisitos com o cliente para minimizar alterações
4. Adaptar o sistema para atender todos os requisitos
5. Integrar o novo membro a "família"
  - Cada nova aplicação é um novo membro da linha de produtos

## Pré-requisitos

- Os principais pré-requisitos para implantação de uma linha de produtos de software são:
  - Tecnologia adequada
  - Maturidade do processo
  - Qualificação da equipe no domínio

## Tecnologias para LPS

- Cada vez mais tecnologias estão maduras o suficiente para apoiar LPS
- Algumas tecnologias importantes
  - Programação OO ainda é a principal forma de se desenvolver LPS
  - Frameworks e componentes ganha cada vez mais espaço por permitirem reuso de partes maiores de software

## Maturidade do Processo

- Processos mais maduros têm maior chance de sucesso ao implantar LPS
- Por muito tempo, processos de desenvolvimento de software foram desestruturados, pouco definidos e pouco entendidos
  - Os modelos de avaliação da capacitação, e.g., CMMI, estão mudando este panorama

## Qualificação da Equipe

- Para se implantar uma linha de produtos, é preciso ter conhecimento suficiente no domínio
  - A experiência da equipe é fundamental
- Exemplo: deve-se saber quais os pontos de variação
  - Criar variabilidade desnecessária leva a custos adicionais de desenvolvimento e gerência

## Modelos de Variabilidade

## Componentes de uma Linha

- Elementos mandatórios
  - Aquelas que são encontradas em todos os membros da linha de produto
- Elementos variáveis
  - **Opcionais:** um produto pode ou não contê-los
  - **Alternativos:** um produto deve conter uma das alternativas

## O conceito de característica

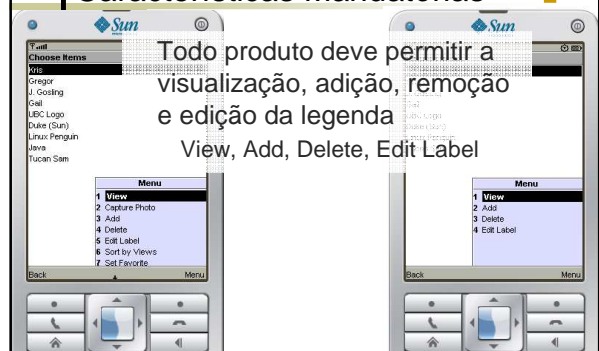
- Característica (*feature*) é um propriedade importante e observável do sistema
  - Um **modelo de característica** permite expressar a configurabilidade da LPS
- Uma característica deve ter um nome conciso
  - Facilita a comunicação entre desenvolvedores

## Exemplo de LPS: MobileMedia

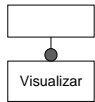


## Características Mandatórias

Todo produto deve permitir a visualização, adição, remoção e edição da legenda  
View, Add, Delete, Edit Label

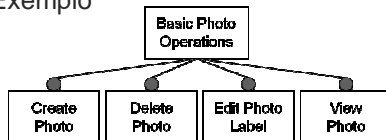


## Representação

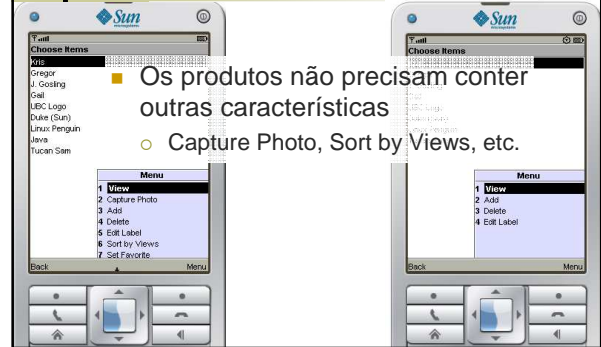


- Características mandatórias
  - Representadas por círculos preenchidos

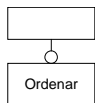
### Exemplo



## Características Opcionais

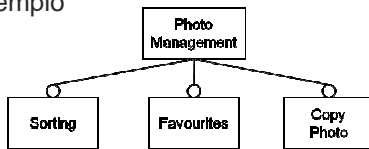


## Representação



- Características opcionais
  - Representadas por círculos vazios

### Exemplo



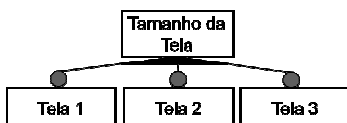
## Características Alternativas



## Representação

- Alternativas OR
  - Representadas por um arco preenchido entre as alternativas

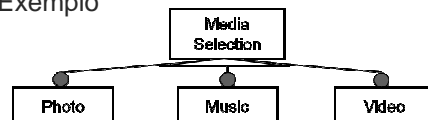
### Exemplo



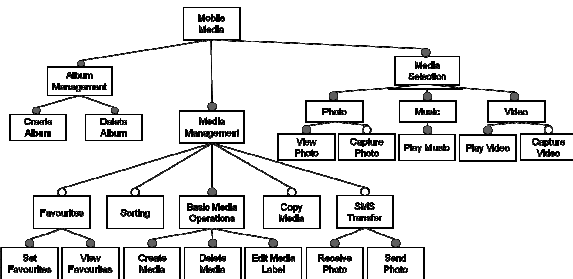
## Representação

- Alternativas não exclusivas
  - Representadas por um arco vazio entre as alternativas

### Exemplo



## Um Modelo de Características



## Bibliografia Principal

- K. Pohl, G. Bockle, F. Linden. **Software Product Line Engineering: Foundations, Principles and Techniques**, 1st edition, Springer, 2005.
  - Capítulos 1 e 2

## Próxima Aula

- **Atenção!** Não haverá aula na próxima quarta (21/03)
- Próxima segunda
  - Ferramentas para linha de produtos de software