

# LINHA DE PRODUTO DE SOFTWARE

---

Por Sergio Crespo

UFF Puro

# INTRODUÇÃO

- A engenharia de software “tradicional” tem como base a construção de um elemento de software ou um produto, no prazo definido. Um elemento de software é um bloco de construção da arquitetura de software (“*building block*”) que possui uma implementação interna, interfaces para ligação com outros elementos de software bem como documentação associada.
- A arquitetura de software preocupa-se com a definição dos elementos de software e do seu comportamento, na perspectiva da relação com outros elementos e não tanto com os detalhes da implementação de cada elemento. Neste contexto, se o comportamento de um determinado elemento influenciar a implementação de outros elementos que com ele pretendem relacionar-se, o seu comportamento deve ser parte integrante da arquitetura.

Esta apresentação esta baseada no trabalho  
situado em <http://goo.gl/6WJfba>

# INTEGRANTES DA ARQUITETURA

- **Módulo:** Um módulo é a implementação do código fonte de um elemento de software (ou parte dele), das suas funcionalidades e comportamentos, descritos pelos requisitos funcionais, satisfazendo as decisões de desenho necessárias para satisfazer os atributos de qualidade definidos para o sistema, e.g. uma classe ou pacote (*package*) contendo várias classes;
- **Componente:** Um componente é uma unidade de processamento e armazenamento de dados que um sistema executa. São tipos de componentes: servidor, cliente, filtro, objeto, base de dados. Os componentes expõem interfaces para o exterior, de forma a poderem ser utilizados ou invocados por outros componentes, essas interfaces são descritos através de portos. Um elemento de software poderá conter “n” componentes;
- **Conector:** Um conector é uma ligação para a partilha de dados e controlo entre componentes, e.g. chamadas remotas de procedimentos entre um cliente e um servidor, ou entre dois; mensagens assíncronas, divulgação de mensagens mediante subscrição. Os conectores possuem uma ou mais papeis que descrevem a forma como os componentes deverão estabelecer a comunicação através do conector.

# REUSO

- Reutilização de software é o processo de criação de sistemas de software a partir de software existente em vez de desenvolvê-lo por inteiro.
- A reutilização de software na concepção de produtos com características semelhantes, gerando um menor custo, torna-se um elemento de vantagem competitiva na engenharia de software.
- São exemplos de reutilização de software: compiladores, bibliotecas de classes (em desenvolvimento OO), padrões de projeto, sistemas distribuídos (e.g. COM, CORBA), interfaces gráficas, frameworks de desenvolvimento, sistemas de gestão de bases de dados (SGDB) (em inglês: *Database Management Systems* – DBMS).

# REUSO

- O desenvolvimento de software para promover a reutilização é contudo mais oneroso em termos de esforço dias/homem e prazo de implementação, pois existem um conjunto de princípios e regras que norteiam a equipa de desenvolvimento. Neste sentido, Bosh apresenta uma abordagem para reutilização de software em que os elementos de software a serem reutilizados são projetados à nível arquitetural e desenvolvidos com um grau de abstração adequado.
- Para Reuso é importante: Programar **visando** REUSO, que permitirá Programar **usando** REUSO.
- A **reutilização** de software é uma das **motivações** para a utilização de uma **LPS**, esta ideia é defendida em Bosh [1], CAFÉ [2], ESAPS [3], PuLSE [4].

[1] J. Bosh, Design & Use of Software Architectures - Adopting and Evolving a Product Line Approach, 2000.

[2] CAFÉ, "From Concepts to Application in System-Family Engineering," Eureka ! 2023 Programme, ITEA project ip00004, 2001.

[3] ESAPS, "Engineering Software Architectures, Processes and Platforms for System- Families," I. p. Eureka ! 2023 Programme, Ed., 2001.

[4] J. Bayer, O. Flege, P. Knauber, Roland Laqua, D. Muthig, K. Schmid, T. Widen, and J.- M. DeBaud, "PuLSE: A Methodology to Develop Software Product Lines," 1999.

# VARIABILIDADE

## Conceito

- “A variabilidade é a capacidade de alterar ou adaptar um sistema. Aumentar a variabilidade de um sistema implica torná-lo mais fácil a certos tipos de alterações.”
- **A gestão de variabilidades** no desenvolvimento de software sempre existiu, embora aplicada a cada produto em particular. A implementação das variabilidades (desde a análise, desenho e desenvolvimento) **era realizada produto a produto**, replicando todo o esforço necessário.
- A introdução de mecanismos de variabilidade no desenvolvimento de software tem como principal objetivo **permitir a reutilização dos elementos de software**, abstraindo as variações que estes possam conter em pontos concretos da arquitetura, face às necessidades específicas de cada projeto ou implementação.

O conceito de variabilidade está relacionado às possibilidades de mudar ou personalizar um sistema.

# LPS - SEI

- A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.
- Software product lines are emerging as a viable and important development paradigm allowing companies to realize order-of-magnitude improvements in time to market, cost, productivity, quality, and other business drivers. Software product line engineering can also enable rapid market entry and flexible response, and provide a capability for mass customization.

# LINHA DE PRODUTO DE SOFTWARE - LPS

- “Uma linha de produtos de software é um conjunto de sistemas de software, que partilham um conjunto de funcionalidades comuns e que satisfazem as necessidades de um segmento de mercado particular ou missão e que são desenvolvidos de forma sistemática, a partir de um conjunto de ativos base”.
- A produção e reutilização de elementos de software base, os quais poderão incorporar variantes é específico do desenvolvimento de software suportado por uma LPS.
- Um elemento de software base da LPS poderá ser “**obrigatório**”, “**opcional**” ou “**externo**” (i.e. desenvolvido por uma entidade externa) no contexto da instanciação em novos produtos.
- Um elemento de software base do tipo “obrigatório” ou “opcional” poderá conter um ou mais pontos de variabilidade, contendo variantes.



# LPS - VANTAGENS

- Teorias relacionadas a linhas de produtos de software pode gerar diversos benefícios classificados em três tipos:
- **benefícios organizacionais**, os benefícios de engenharia de software e os benefícios de negócio. Os benefícios organizacionais agrupam vantagens como uma melhor compreensão do domínio, a maior facilidade de treinar pessoas, um produto de maior qualidade e conseqüentemente confiança do cliente.
- **benefícios da engenharia de software** incluem vantagens como a reutilização de requisitos e seus componentes, uma melhor análise de requisitos, uma outra visão sobre os requisitos para o cliente, controle de qualidade de software, estabelecimento de padrões de programação.
- **benefícios comerciais** dizem respeito à redução de manutenção e custos de teste (graças à reutilização entre vários produtos semelhantes). Além disso, as linhas de produtos geram uma melhor eficiência nos processos e a possibilidade de aumentar o orçamento e melhorar o planejamento do tempo por ter maior controle dos componentes que fazem parte do produto final.

# LPS

- LPS favorece o 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.

# LPS

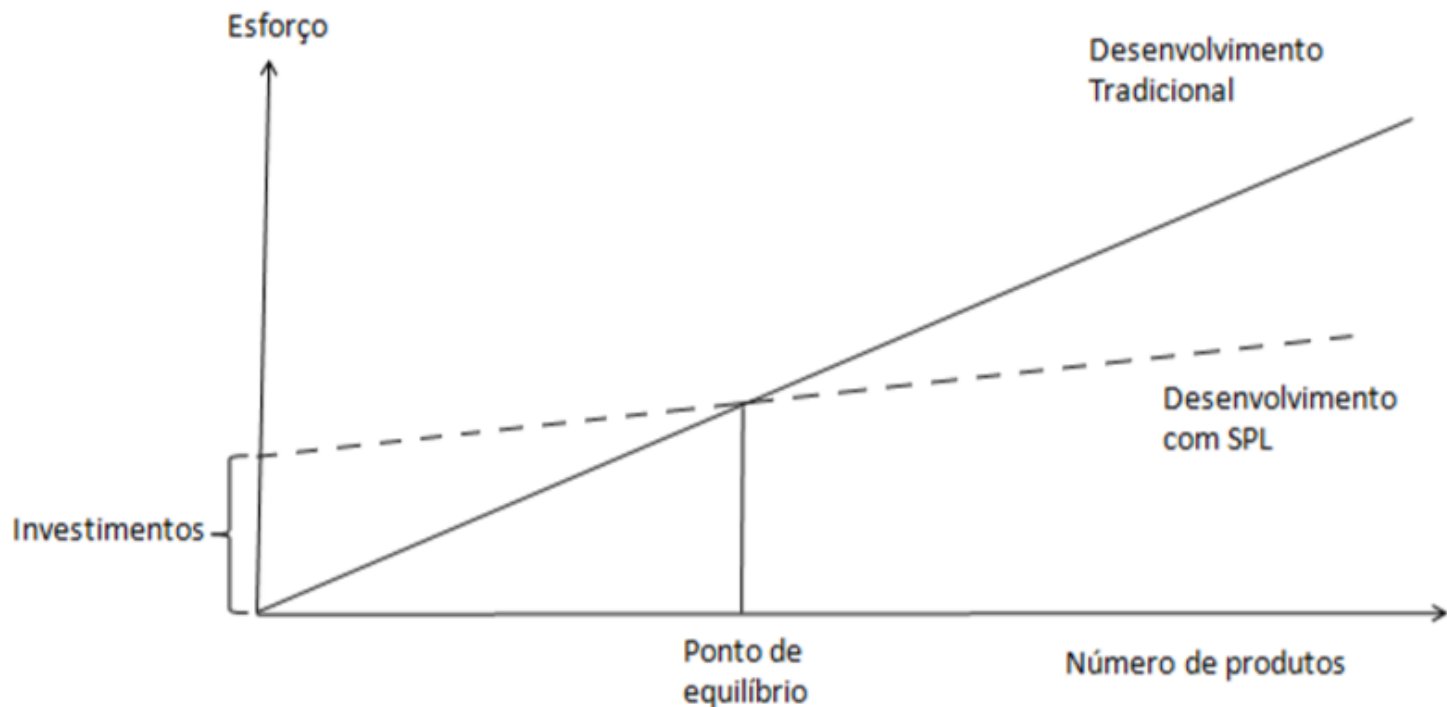
- A principal razão para se criar uma linha de produtos é a redução de custos.
  - Entretanto, antes de começar os ganhos, um investimento inicial é necessário.
  - Estudos indicam que são necessários de três a quatro produtos de software para se recuperar o investimento.
-

# EXEMPLOS DAS VANTAGENS

- 1. Nokia é capaz de produzir 25-30 modelos diferentes de celular por ano por causa da abordagem de linha de produtos.
- 2. Cummins, Inc., foi capaz de reduzir o tempo que leva para produzir o software de um motor diesel de cerca de um ano para cerca de uma semana.
- 3. Motorola observou uma melhoria de produtividade de 400% em uma família de determinado tipo de celular.
- 4. Hewlett-Packard reportou uma redução no *time to market* por um fator de sete e um aumento na produtividade por um fator de seis, em uma família de sistemas de impressoras.

## Comparação de custos entre abordagem tradicional e LPS

	Risco	Porcentagem
Resistencia organizacional		52%
Resistência da gestão		36%
Resistência dos desenvolvedores		32%
Preocupações com grandes investimentos		45%
Falta de pessoal devidamente treinado		29%
Incapacidade de medir o impacto		19%
Preocupação com o tempo de um projeto		18%



# LPS

- 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.

# LPS - VARIABILIDADE

- “A variabilidade é a capacidade de alterar ou adaptar um sistema. Aumentar a variabilidade de um sistema implica torná-lo mais fácil a certos tipos de alterações”.
- Com a abordagem de LPS as variabilidades são modeladas na arquitetura de referência da LPS (através da representação dos Pontos de Variabilidade e Variantes respectivos) e resolvidas antes da instalação dos produtos.

# LPS - VARIABILIDADE

- Um Ponto de Variabilidade corresponde a um aspecto de variação funcional num elemento de software base. O ponto de variabilidade define o conjunto de possíveis variantes, o mecanismo de variabilidade a utilizar para os instanciar e o tempo de ativação dos variantes, e.g. instanciação da arquitetura de software do produto, tempo de compilação, execução.
- Um Variante corresponde a uma opção do conjunto de possíveis instâncias de variação que um ponto de variabilidade poderá originar.



# LPS - VARIABILIDADE

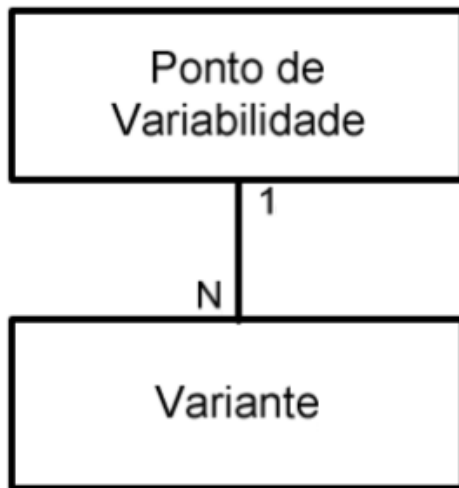


Figura 1 – Conceitos na variabilidade



Figura 2 – Exemplo de ponto de variabilidade e variantes

# LPS - VARIABILIDADE

De forma a podermos identificar variabilidades no software devemos utilizar as seguintes questões:

- Onde variam os vários produtos do mesmo software? A resposta a esta questão permite identificar pontos de variabilidade.
- Como varia? Ou seja qual o conjunto de variantes detectados para um ponto de variabilidade.
- Porque é que varia? A resposta permite aferir das razões para a variação e identificar o variante que melhor a satisfaz.

# LPS PERSPECTIVAS

- Numa LPS, existem duas perspectivas de engenharia de software: engenharia de domínio e engenharia da aplicação.
- Engenharia de Domínio: Este processo é responsável por estabelecer a plataforma de reutilização e, assim definir requisitos/artefatos comuns e a variabilidade da linha de produtos. A plataforma consiste em todos os tipos de artefatos de software(requisitos, design, testes, etc.) também chamados de ativos base.
- Engenharia de Aplicação: Este processo é responsável por derivar aplicações concretas a partir da plataforma estabelecida na engenharia de domínio. Ela explora a variabilidade da linha de produtos e assegura sua correta instanciação de acordo com as necessidades específicas das aplicações finais.
- A vantagem dessa divisão é que há uma separação de objetivos, para construir uma plataforma robusta e para construir aplicações específicas em um curto espaço de tempo. Para ser eficaz, os dois processos devem interagir de uma maneira que seja benéfica para ambos.

# LPS E ARQUITETURA DE SOFTWARE

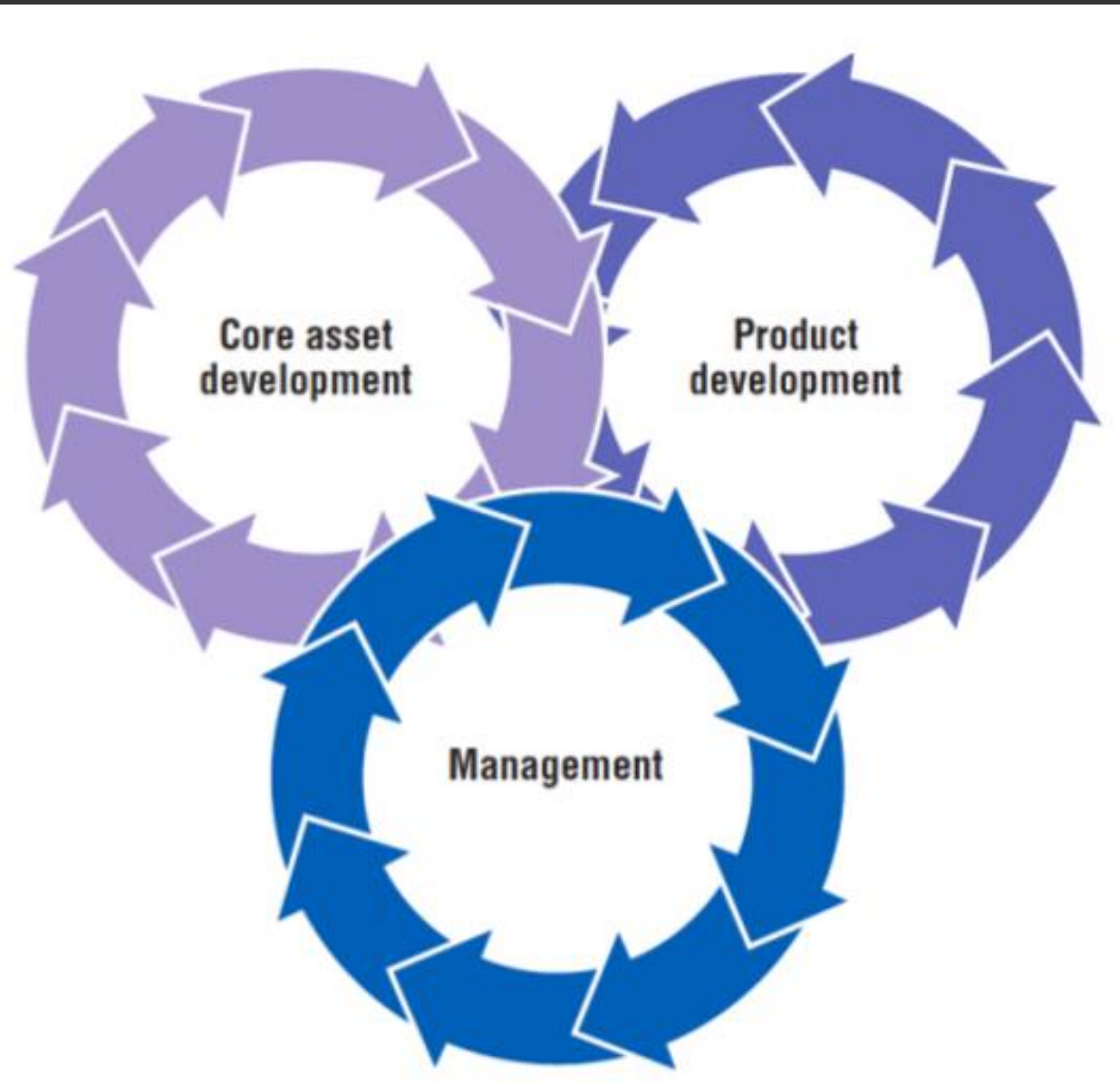
- Cada LPS possui uma arquitetura de software própria (Arquitetura de Software da LPS).
  - A arquitetura de software de uma LPS permite a instanciação de implementações dos elementos de software base da LPS em organizações com realidades específicas.
  - As variabilidades dos produtos são obtidas pela instanciação dos pontos de variabilidade, definidos nos elementos de software base da LPS.
-

# LPS E ARQUITETURA DE SOFTWARE

- “Uma arquitetura de uma linha de produtos de software é uma arquitetura de software que satisfaz as necessidades da linha de produto em geral e os produtos individuais em particular, explicitamente admitindo um conjunto de pontos de variabilidade necessários para suportar o espectro dos produtos no âmbito da linha de produtos. Contém a arquitetura dos produtos e as especificações das interfaces para os componentes que constituirão os ativos nucleares.”

# ENGENHARIA DE LINHA DE PRODUTOS DE SOFTWARE

- Atividades Essenciais em Linha de Produtos de Software Linhas de Produto de Software combinam três atividades essenciais e altamente interativas que se misturam práticas de negócios e tecnologia.
- Em primeiro lugar, atividade de **Core Asset** Development onde o objetivo não é criar o produto final imediatamente e sim visa o desenvolvimento de ativos a serem reutilizados em outras atividades posteriores.
- Em segundo lugar, vem a atividade denominada **Product Development** que parte dos ativos base já desenvolvidos anteriormente reusando-os.
- Finalmente **Management Activity**, que inclui gestão técnica e organizacional.



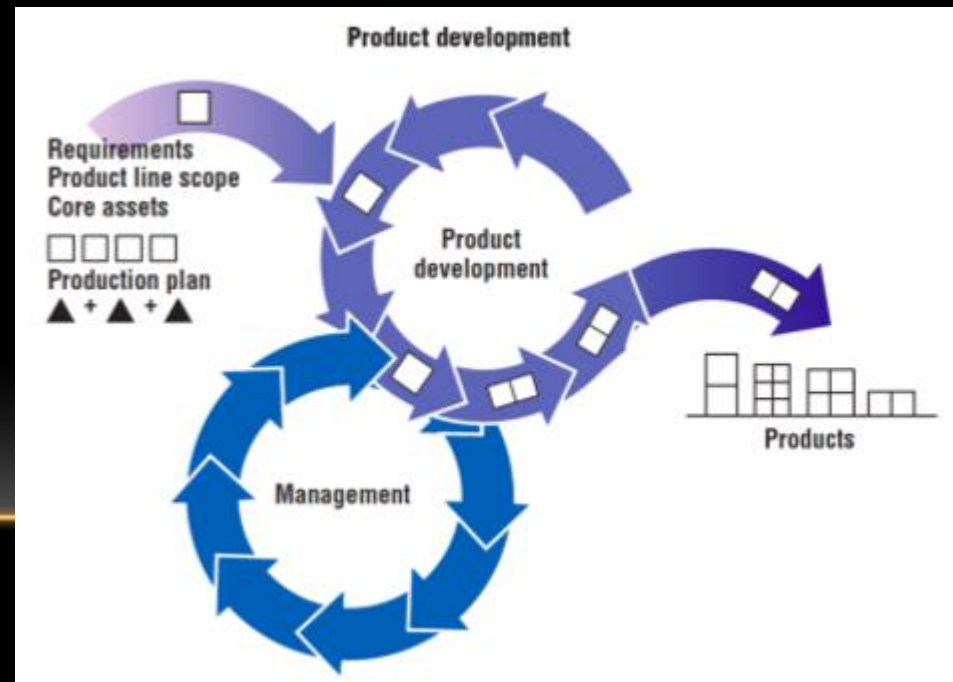
# ATIVOS BASE

- Core **Asset** Development é uma atividade na forma de ciclo de vida que resulta em ativos base que em conjunto compõem a plataforma da linha de produto.
- O objetivo desta atividade é definir os aspectos comuns e a variabilidade da linha de produtos, e, portanto, obter artefatos reutilizáveis para em seguida possuir uma capacidade de produção maior.
- Ativos base incluem, mas não estão limitados à arquitetura e sua documentação, especificações, componentes de software, ferramentas como geradores de componentes ou aplicação, modelos de desempenho, cronogramas, orçamentos, planos de teste, casos de teste, planos de trabalho e processo descrições.



# DESENVOLVIMENTO DE PRODUTOS

- Na atividade de **desenvolvimento de produto**, os produtos são desenvolvidos a partir dos ativos base, com base no plano de produção, para satisfazer as exigências da linha de produtos de software. Os insumos essenciais da atividade de desenvolvimento de produto são requisitos, escopo da linha de produtos, ativos base e o plano de produção.
- De posse do plano de produção, que detalha como os ativos base serão utilizados para construir um produto, o engenheiro de software pode montar as partes da linha de produtos.



# ATIVIDADE DE GESTÃO

- A atividade de **Management** desempenha um papel vital no sucesso da institucionalização da linha dentro de uma organização porque fornece e coordena a sua infra-estrutura necessária. Esta tarefa envolve atividades essenciais realizadas a nível técnico e organizacionais para apoiar o ciclo de vida do processo.
- O conjunto de ativos base e plano de como eles são usados para construir os produtos não nascem sem previamente estudar o ambiente, caracterizar o negócio, portanto deve existir investimento organizacional. A gestão deve dirigir, controlar e garantir a plena utilização dos ativos. Linhas de produtos de software está mais relacionado a práticas de negócios do que práticas técnicas.

# ABORDAGENS DE CONSTRUÇÃO DE LINHA DE PRODUTOS DE SOFTWARE

- **Proativa:** Com esta abordagem os ativos base são desenvolvidos primeiro para futuro produtos. Aqui são considerados todos os produtos a serem gerados previamente fazendo-se um planejamento inicial completo.
- **Reativa:** Nesta abordagem os ativos base já existem, bem como uma versão da linha de produtos, o que acontece é a evolução desta linha realizando-se incrementos na mesma à medida que novos requisitos aparecem.
- **Extrativa:** Com esta abordagem, inicialmente são analisados quais os produtos já existentes e como eles são estruturados de modo a extrair os requisitos comuns e variabilidades destes para então poder se derivar uma versão inicial da Linha de Produtos.

# LOCAL DE PESQUISA EM LPS

<http://www.sei.cmu.edu/productlines/start/>

**SIM! É**  
para a  
próxima  
aula!

## Atividades

Pesquisar e apresentar em aula um artigo que fale de pesquisa em LPS e faça algum link com Arquitetura de Software. A busca não precisa ficar centrada no site da SEI, pode-se buscar artigos na IEEE e ACM, bem como em periódicos do portal CAPES.

Como será somente uma apresentação por pessoa, cada aluno deve ser pontual e profundo na temática do artigo.

**Uma cópia do artigo pesquisado deve ser entregue em aula e o PPT (em formato PDF) enviado ao professor para que se possa identificar o grau de profundidade da apresentação frente ao artigo selecionado.**