



Conceitos de Linhas de Produtos de Software (LPS)

Prof.^a Maiara Cancian

Descrição

A ideia da Linha de Produto de Software (LPS) com base no reuso, conceitos envolvidos, engenharia utilizada e *framework* de apoio para o seu desenvolvimento.

Propósito

Linha de Produtos de Software (LPS) disponibiliza ao engenheiro de software uma abordagem de desenvolvimento baseada em reuso, em que o objetivo é o desenvolvimento de um software para um domínio específico, no qual se compartilha um conjunto comum de funcionalidades, permitindo que ele seja adotado em escala por empresas da mesma área.

Objetivos

Módulo 1

Princípios da LPS

Identificar os princípios da Linha de Produto de Software e os conceitos envolvidos.

Módulo 2

Engenharia da LPS

Reconhecer a engenharia envolvida na Linha de Produto de Software.

Módulo 3

Framework para LPS

Reconhecer o *framework* da Linha de Produto de Software.

Módulo 4

Benefícios e dificuldades da adoção de LPS

Descrever os benefícios da adoção da Linha de Produto de Software.



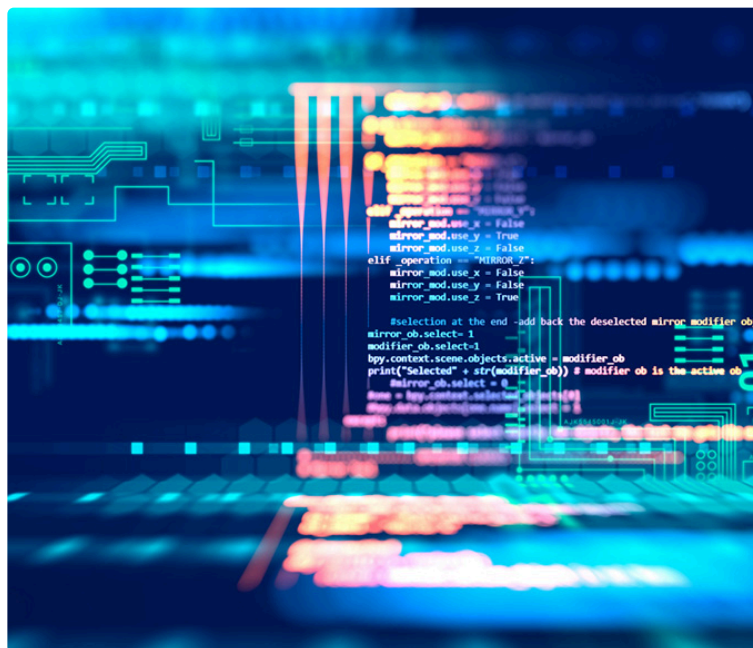
Introdução

O software já faz parte do negócio de todas as empresas, já que ele é algo indispensável para a condução e acompanhamento dos processos e da sua gestão. Mas, para obter vantagens competitivas, é necessário que o software esteja disponível rapidamente para atender às demandas crescentes por novos produtos e serviços.

É muito comum que já existam softwares para as empresas em seus segmentos, e uma pergunta a ser feita é: por que não se reutiliza softwares já criados para alguma empresa, visto que muitos dos processos são similares?

Isso já ocorre e é a chamada **Linha de Produtos de Software**, conjunto de sistemas de software que compartilha um mesmo domínio e recursos que satisfazem as necessidades específicas de um determinado segmento de mercado.

Se executado adequadamente, o desenvolvimento utilizando a abordagem de Linha de Produtos traz inúmeras vantagens competitivas. A reutilização estratégica de software por meio de uma abordagem de Linha de Produtos de Software bem gerenciada atinge as metas de negócios para: aumentar a eficiência, produtividade e qualidade; e, dessa forma, consegue um lançamento no mercado em tempo surpreendente.



1 - Princípios da LPS

Ao final deste módulo, você será capaz de identificar os princípios da Linha de Produto de Software e os conceitos envolvidos.

Reuso de software

Estratégias de software

A grande maioria dos produtos, após terem sido feitos, são entregues aos seus clientes, pois são produtos físicos, como uma cadeira, um carro ou uma decoração. Com o software é diferente, pois ele é virtual, e não físico. Quando entregamos um software, estamos entregando uma cópia virtual do produto de software.

E quais vantagens podemos ter no caso de software? Uma delas é o reuso.

A Engenharia de Software apresenta três estratégias principais para a boa condução e sucesso na área de TI, que são:



Melhoria de processo



Inovação tecnológica



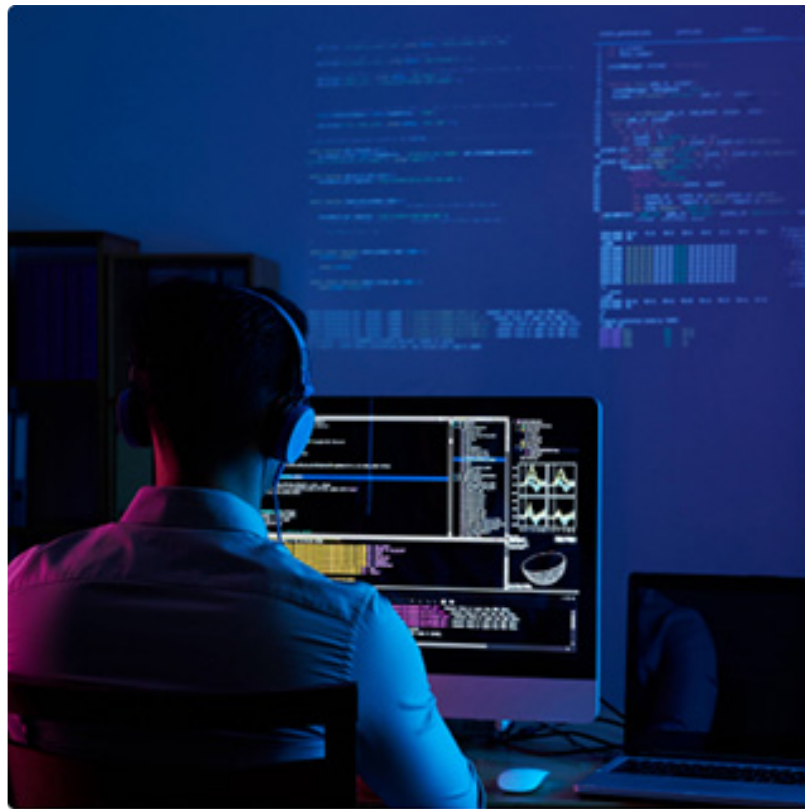
Reuso

Vamos ver um exemplo bem usual: os smartphones. Grandes empresas como a Apple ou Samsung não desenvolvem um software para cada celular e para cada pessoa. O software é o mesmo, mas com variações, dependendo do modelo; e ainda assim pode ser customizado (pelo menos algumas funcionalidades) pelo usuário, fazendo com que cada celular tenha características de funcionamentos escolhidas por eles.

Comentário

Você já notou que muitas funcionalidades existem em quase todos os softwares? Como, por exemplo, o cadastro, o login, pagamentos etc. Essas funcionalidades passam a ser muito mais similares quando a área de atuação das empresas é a mesma.

Imagine uma loja de roupas: o software dela deve ser muito parecido (ou idêntico) ao software de outras lojas de roupas. Pense no controle de estoque de um armazém ou no sistema de gerenciamento de um banco. Se eles têm a mesma área de negócio, é muito provável que tenham rotinas de trabalhos similares.



Veja que os artefatos de um projeto de software são, em sua maioria, reutilizáveis, visto que depois de prontos podem ser utilizados em novos projetos. Dessa forma, é comum que um projeto se baseie no reuso de sistemas já existentes.

Engenheiros normalmente não especificam do zero um projeto em que cada componente tenha que ser fabricado especialmente. Eles baseiam seus projetos em componentes já testados e experimentados em outros sistemas.

A engenharia de software baseada em reuso é uma abordagem de desenvolvimento que tenta maximizar o reuso do software existente. Reusar software tem grandes vantagens; entre as principais, podemos citar: aumento na produtividade no processo de desenvolvimento, aumento da qualidade do software e diminuição do prazo de entrega (PRESSMAN, 2016).

As unidades de software a serem reusadas podem ser de tamanhos muito diferentes, como, por exemplo: objetos e funções (unidades pequenas); componentes (podem ser pequenas e médias); e sistema de aplicação (unidades grandes, sendo família de aplicação com arquitetura comum).

Reuso do sistema de aplicações

O reuso do **sistema de aplicações** envolve o reuso de sistemas de aplicações inteiros pela configuração de um sistema para um ambiente específico. O sistema central é especificamente projetado para ser configurado e adaptado às necessidades específicas de diferentes

clientes do sistema. Nesse caso, temos a criação de Linhas de Produtos de Software (LPS), ou Família de Aplicações.

Atenção!

Uma LPS consiste então em uma estratégia de realizar o reuso de software de forma sistemática para a construção de sistemas, e com isso investir menos esforço. Mas para que seja considerada LPS, os novos produtos devem pertencer a uma mesma família, ou seja, devem pertencer a um mesmo domínio de mercado.

Clements e Northrop (2002) entendem que uma linha de produto de software é um conjunto de sistemas que usam software intensivamente, compartilhando um conjunto de características comuns e gerenciadas que satisfazem as necessidades de um segmento particular de mercado ou missão, e que são desenvolvidos a partir de um conjunto comum de ativos principais e de uma forma preestabelecida.

A ideia principal é a possibilidade de customização para cada cliente/*stakeholder*, e a LPS deve fornecer meios para isso. Nesse caso, veja que é criado o conceito de “Variabilidade” que se refere à customização, ou seja, o que pode variar nesse sistema de um *stakeholder* para outro. A variabilidade pode gerar consequências como diferentes artefatos para diferentes aplicações, e nesse caso as modelagens de apoio auxiliam na sua compreensão.

As adaptações a serem feitas na LPS, para se adequar à necessidade de cada cliente, também são conhecidas como especialização, e podem ser:

Especialização de plataforma



Quando versões da aplicação são desenvolvidas em plataformas diferentes, como, por exemplo em Windows e Linux. Neste caso, a funcionalidade da aplicação permanece inalterada, somente os componentes que fazem interface com o sistema operacional são modificados.

Especialização de ambiente



Quando versões da aplicação são criadas para lidar com dispositivos periféricos específicos.

Especialização funcional



Quando versões da aplicação são criadas para clientes específicos com requisitos diferentes.

Especialização de processo



Quando o sistema é adaptado para lidar com processos de negócio específicos.

Feature Model

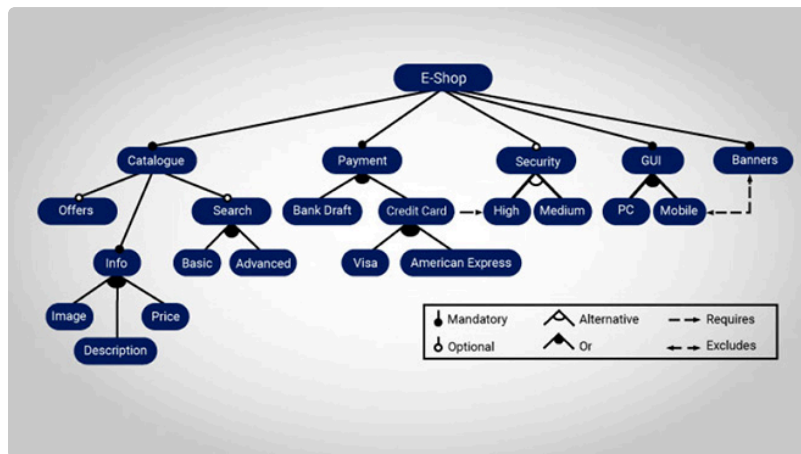
Resgatando as informações vistas até agora, vemos que a implantação da LPS em diferentes empresas é possível pela variabilidade, visto que a arquitetura do sistema é a mesma, e a variabilidade define as partes que seriam específicas para cada negócio.

Nesse sentido, temos a *Feature Model*, principal mecanismo para modelagem de variabilidade.

O conceito de *feature* é derivado do método *Feature Oriented Domain Analysis* (**Análise de domínio orientada a recursos**). De acordo com o método, a *feature* corresponde a uma característica do sistema visível ao usuário final, ou seja, trata-se de um contexto do sistema em que o usuário final tem contato direto (CLEMENTS; NORTHROP, 2002).

Uma *feature* é correspondente a uma unidade comportamental (relevante para um ou mais *stakeholders* desse produto), que pode ser utilizada para representar um conjunto de requisitos (um agrupamento de requisitos relacionados). O objetivo de uma *feature* é aproximar o programador do *stakeholder* do produto, criando características principais (em alto nível) no produto de software, facilitando a inclusão dessa *feature* no ciclo de produção.

A figura a seguir ilustra um exemplo de representação para o modelo de *features*, neste caso de um Sistema e-Shop.



Exemplo de representação de *feature model*.

Neste exemplo, as *features* obrigatórias são: Catalogue, Payment, GUI e Info; e as opcionais são: Security, Banners, Offers, Search. Veja alguns detalhes:

Security

Esta *feature* possui um grupo de *subfeatures* em uma alternativa mutuamente exclusiva (XOR) que indica que apenas uma delas é escolhida para a aplicação.

Payment

Esta é composta por duas *subfeatures* em uma alternativa OR, que indica que uma ou as duas podem ser escolhidas.

Uma *feature* filha pode aparecer em um produto apenas caso sua *feature* superior apareça. A *feature* raiz é parte que compõe todos os produtos dentro da LPS.

Uma característica importante da *feature model* é que ela é utilizada somente para modelagem dos produtos na etapa de domínio, pois não consegue representar “como” estes produtos serão gerados através dos artefatos, que estão presentes nos ativos-base.

Uma solução para esse problema é o uso de *Configuration Knowledge* (Conhecimento de configuração), técnica utilizada para mapear a *feature model* para os artefatos de implementação. Esses artefatos podem ser modelados no *Configuration Knowledge*, garantindo a associação entre os artefatos de um modelo com os itens que fazem parte da *feature model*.

Sistemas ERP: *Enterprise Resource Planning*

O sistema ERP é o exemplo mais difundido de reuso de software na abordagem LPS, traduzindo do inglês, a sigla significa “Planejamento dos Recursos da Empresa”. Logo, é um software de gestão empresarial que serve para automatizar processos manuais, armazenar dados e unificar a visualização de resultados. Como esses processos são comuns em muitas áreas, este é um tipo de produto fortemente candidato para ser escalável (SOMMERVILLE, 2011).



Ilustração de um ERP.

É possível encontrar ERPs para diferentes setores do mercado, chamados de módulos verticais de setores específicos. Dentre os mais comuns, podemos destacar:

- Automação comercial;
- Comércio exterior;
- Gestão de acervos;
- Plano de saúde;
- Gestão educacional;

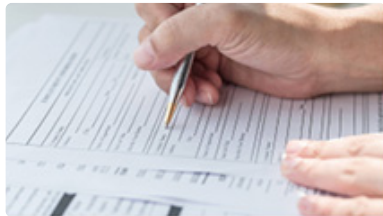
- Gestão de transportadora.

Dentre as funcionalidades, chamadas de módulos principais, podemos citar:

Recursos Humanos



Fiscal



Estoque



Compras



Produção





Financeiro



Faturamento



Gerenciamento de projetos



O uso do ERP permite unificar informações e facilitar o fluxo de trabalho entre as áreas. Considerando que cada departamento precisa de uma ferramenta própria para poder gerar seus relatórios e executar tarefas rotineiras, o ERP passa a ter um papel fundamental. Nesse caso, funciona como um centralizador do fluxo de trabalho, alinhando a comunicação de todas as áreas e facilitando a tomada de decisão.

Saiba mais

O desenvolvimento do sistema ERP começou há mais de 100 anos. O engenheiro Ford Whitman Harris criou um modelo EOQ (*Economic Order Quantity*: traduzindo, Quantidade de pedido econômico) para programar

a produção. Ele foi utilizado por décadas até que a fabricante de ferramentas Black & Decker adotou outra solução para planejar as necessidades de materiais.

O novo método ficou conhecido como MRP (*Material Requirements Planning*: traduzindo, Planejamento de Requisitos de Materiais). Na prática, alguns conceitos do EOQ foram utilizados e integrados a um computador. Essa ferramenta foi usada por muitos anos até ser desenvolvida uma nova metodologia em 1983, a MRP II (*Manufacturing Resource Planning*: traduzindo, Planejamento de Recursos de Manufatura).

Nessa evolução, foram adotados módulos diferenciados e componentes da arquitetura de software para integrar as atividades da companhia relacionadas às compras e ao gerenciamento de contratos, bem como a integração de dados e o compartilhamento de informações entre os diversos departamentos da companhia, a fim de melhorar a produção e reduzir os desperdícios. Em virtude disso, a nova ferramenta passou a incorporar os setores de finanças, RH e vendas. Em 1990, esse sistema foi denominado ERP.

A SAP é a empresa líder de mercado no segmento de software para aplicativos empresariais e ajuda empresas de todos os portes e setores a alcançarem a excelência operacional: 77% das receitas de transações no mundo são geradas com a ajuda dos sistemas da SAP.



ERP SAP.

A empresa SAP desenvolveu bastante o conceito do software e hoje já integra em seus produtos tecnologias de *Machine Learning*, Internet das Coisas (IoT) e funções analíticas avançadas que ajudam as empresas a transformar seus negócios em Empresas Inteligentes.

Veja algumas das vantagens da adoção de um ERP:

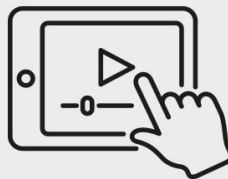
- Processos automatizados em todos os departamentos.
- Redução no tempo investido em controle de entregas e fechamento de envios.
- Melhor qualidade de dados informados.
- Aumento de produtividade, já que todas as informações são compartilhadas entre áreas.
- Redução de riscos e erros.
- Redução de custos e tempo na gestão de *compliance*, no fechamento contábil e fiscal, na apuração de impostos e contribuições e no controle de bens patrimoniais.
- Acompanhamento de todo o processo de compra, recebimento, contrato e estoque, dando fim aos arcaicos processos manuais e uso de papéis e documentos por escrito.
- Gestão financeira integrada e independente.



Conceitos de *Feature Model*

Neste vídeo, apresentaremos os principais conceitos sobre o *Feature Model*.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

A engenharia de software baseada em reuso é uma estratégia da engenharia em que o processo de desenvolvimento é orientado para o reuso de softwares existentes. Nesse contexto, analise as afirmativas a seguir:

- I. Preencher uma biblioteca de componentes reusáveis e garantir que desenvolvedores de software possam utilizar essa biblioteca são ações simples e rápidas, pois processos de desenvolvimento não precisam ser adaptados para utilizar essa biblioteca.
- II. Os custos dos componentes reutilizáveis já são conhecidos (pois já estão prontos) e, nesse caso, o risco de processo é reduzido, pois algumas estimativas já são conhecidas.
- III. Em questões financeiras, o reuso é uma ótima opção, já que desenvolver tudo novamente elevaria o custo do projeto.
- IV. O reuso de software ajuda as empresas a atenderem melhor as demandas de mercados, já que conseguem prazos de entregas mais atrativos, devido ao tempo economizado.

Assinale a opção correta:

- A Apenas as opções I e II estão corretas.
- B Apenas as opções II e III estão corretas.
- C Apenas as opções II e IV estão corretas.
- D Apenas as opções III e IV estão corretas.
- E Apenas as opções I e IV estão corretas.

Parabéns! A alternativa C está correta.

O reuso, de acordo com as premissas da Engenharia de Software, é um processo que envolve várias etapas, como a análise da unidade de reuso, teste, catalogação e armazenamento. Para o uso efetivo da unidade de reuso é preciso analisar e documentar sua utilização.

O reuso é fortemente indicado nas empresas de desenvolvimento, já que os artefatos gerados para produzir software são altamente

reutilizáveis; dessa forma, as empresas estão mais preparadas para atender às demandas de mercado.

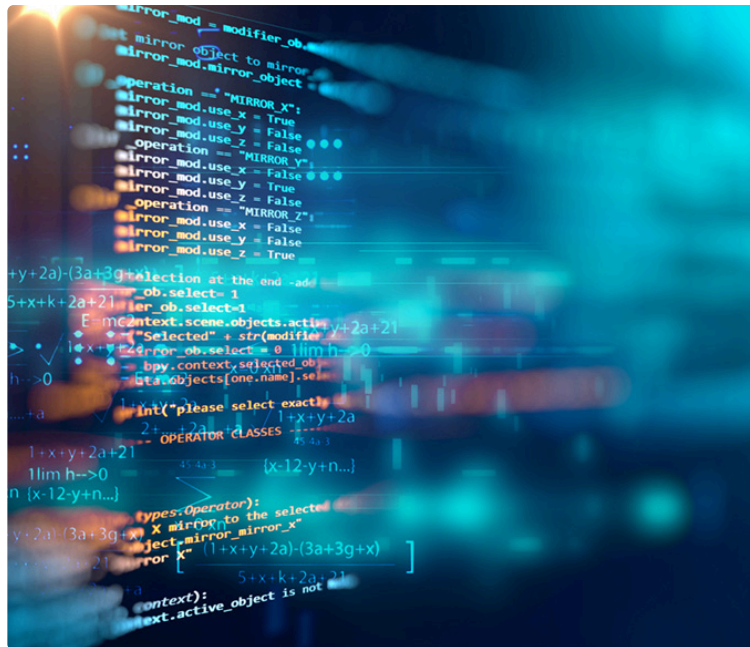
Questão 2

Uma empresa de distribuição e venda de produtos deseja instalar um sistema constituído por módulos integrados permitindo a administração de diversas operações, tais como financeira, contábil, logística e recursos humanos, o que possibilita um maior controle dessas operações e dos custos devido à forte integração das áreas citadas. Baseado nas características citadas, a escolha deve ser direcionada a um sistema de informação do tipo:

- A Sistema de Informação para Executivos (EIS).
- B Sistema de Gestão Empresarial (ERP).
- C Gerenciamento do Relacionamento com o Cliente (CRM).
- D Sistema Especialista (SE).
- E Sistema Operacional (SO).

Parabéns! A alternativa B está correta.

A adoção do ERP traz vários benefícios, desde sua aquisição e implantação mais rápida, visto que o software está pronto e basta fazer as customizações. Além disso, o seu uso integra os processos das organizações, pois são todos os módulos na mesma aplicação, e isso agiliza muito os processos.



2 - Engenharia da LPS

Ao final deste módulo, você será capaz de reconhecer a engenharia envolvida na Linha de Produto de Software.

Processo da LPS

Atividades essenciais para LPS

Denominamos **Engenharia da LPS** as etapas, processos, metodologias e demais recursos que dão suporte ao seu desenvolvimento. A adoção da engenharia traz benefícios no desenvolvimento da LPS, como a redução de custos e geração de produtos de qualidade, já que, ao estabelecer os processos, o ciclo de vida fica mais confiável.

Existem três atividades essenciais que são interativas e fazem parte do processo da Linha de Produto de Software:



Core Asset Development

(Desenvolvimento de Ativos Principais)



Product Development

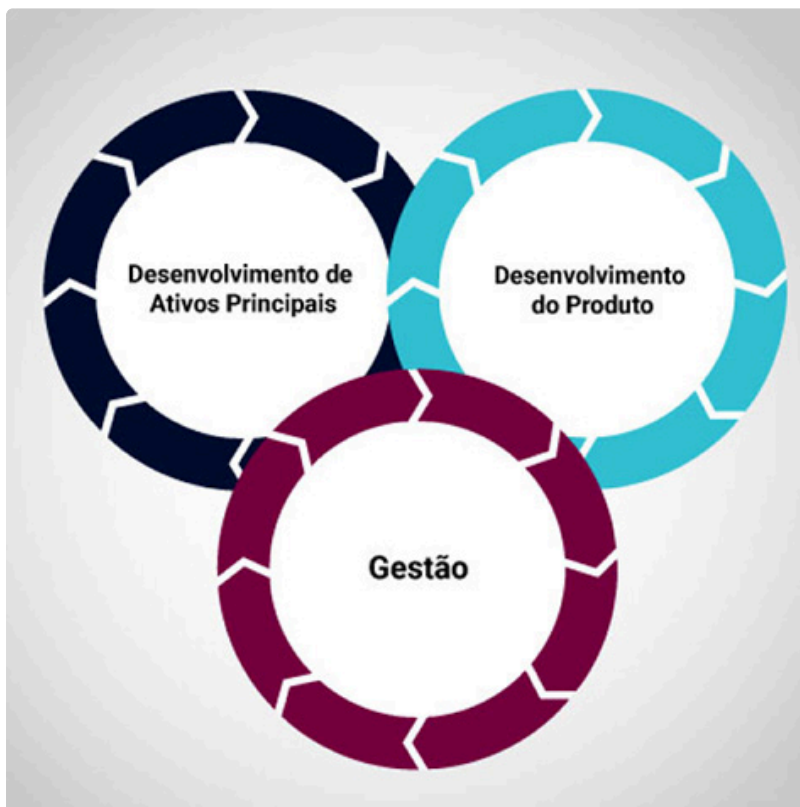


Management

(Gestão)

A interatividade se deve ao fato destas atividades poderem ocorrer a qualquer momento durante o desenvolvimento da LPS (DA SILVA *et al*, 2011).

A imagem a seguir mostra as três atividades essenciais, cada uma representada em um círculo giratório. Note que os três círculos são ligados entre si e em movimento contínuo, mostrando que todos os três são essenciais e estão intimamente ligados, podendo ocorrer em qualquer ordem, bem como são altamente iterativos.



Atividades essenciais LPS.

No que tange à implantação da LPS, basicamente temos duas formas: instantânea ou gradual. Na instantânea, são feitas poucas customizações, e as que são feitas já são liberadas nas primeiras versões, otimizando o processo. Isso faz com que o custo seja menor e a implantação mais rápida, o que torna a operação economicamente atrativa.



Já na implantação gradual é feito um estudo mais profundo de todas as customizações necessárias, e normalmente ocorrem muitas etapas para isso. Essa opção é mais lenta, mas também traz ganhos e diminui os riscos, já que o produto vai ficar mais aderente aos processos da empresa. O custo desta implantação também é maior, mas costuma ser a opção mais adotada.

Quanto às abordagens para desenvolvimento, destacam-se a proativa, reativa e extrativa. Na abordagem proativa, os ativos-base são desenvolvidos para depois construir produtos. Na abordagem reativa, os ativos-base já existem e vão sendo evoluídos com incrementos na medida em que novos requisitos aparecem. Na abordagem extrativa, é feita uma análise dos produtos existentes e suas estruturas para poder extrair as características comuns e variáveis para derivar a implantação da LPS.

A seguir, descreveremos as atividades essenciais para a sua melhor compreensão.

Core Asset Development (Desenvolvimento de Ativos Principais)

O objetivo dessa atividade é definir os aspectos comuns e a variabilidade da linha de produtos, identificando os artefatos reutilizáveis. A imagem a seguir mostra o núcleo dessa atividade juntamente com suas saídas e insumos necessários.



As setas na figura sugerem que não há um momento certo de adicionar uma restrição ou novos padrões no desenvolvimento, e essas entradas afetam diretamente as saídas no processo. Em alguns contextos, os produtos existentes são a base para os ativos-base; em outros, estes podem ser desenvolvidos do zero para futuro reuso.

Mas o que são os ativos-base?

Os ativos-base podem ser: a 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 descrições do processo.

Comentário

Um ativo-base pode ser construído para ser utilizado integralmente em todos os produtos. Mas, veja: se ele for passível de adaptações, ele poderia ser mais largamente utilizado, pois essas adaptações o tornariam apto a satisfazer as necessidades de diferentes clientes. Dessa forma, algumas adaptações podem ser necessárias para tornar o ativo-base mais utilizável em uma linha de produtos.

Nesses casos, mecanismos de variação de ativos-base podem ser utilizados para ajudar a controlar as adaptações necessárias e suportar as diferenças entre os produtos de software. Essas adaptações devem (obrigatoriamente) ser planejadas antes do desenvolvimento, já que alteram sua arquitetura. Isso facilita o trabalho para a equipe de desenvolvimento do produto sem colocar em risco as propriedades existentes dos ativos-base.

Product Development (Desenvolvimento do Produto)

Dentro da engenharia da LPS, a segunda atividade essencial é a *Product Development* (Desenvolvimento do Produto), que parte dos ativos-base já desenvolvidos anteriormente (na primeira atividade essencial), em que 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. A imagem a seguir ilustra o produto atividade de desenvolvimento, juntamente com suas saídas e fatores contextuais.



Product Development (Desenvolvimento do Produto).

Veja que nessa figura as setas de rotação indicam iterações entre as partes envolvidas. Por exemplo, a existência e a disponibilidade de um determinado produto podem afetar os requisitos de produtos subsequentes (já que irão pertencer ao mesmo ciclo de produção). Conhecer aspectos do produto previamente fornece uma base para explorar essas diferenças em futuros produtos. Além disso, esta atividade tem a obrigação de dar feedback sobre quaisquer problemas ou deficiências encontradas com os ativos-base.

Management (Gestão)

A terceira atividade essencial da LPS é a de *Management* (gestão) e desempenha um papel vital no sucesso da institucionalização da LPS dentro de uma organização. A gestão fornece e coordena a infraestrutura necessária. Esta atividade envolve tarefas realizadas em nível técnico e organizacional para apoiar o ciclo de vida do processo.



A gestão traz vários benefícios, já que analisa o cenário geral da organização. No desenvolvimento da LPS, ela supervisiona a construção dos ativos-base e tarefas de desenvolvimento do produto, garantindo que os grupos que constroem os ativos-base e os produtos estão plenamente envolvidos nas tarefas individuais, acompanhando o processo definido para a linha de produtos e seu progresso. Isto representa não apenas os aspectos técnicos, mas também aspectos gerenciais e organizacionais.

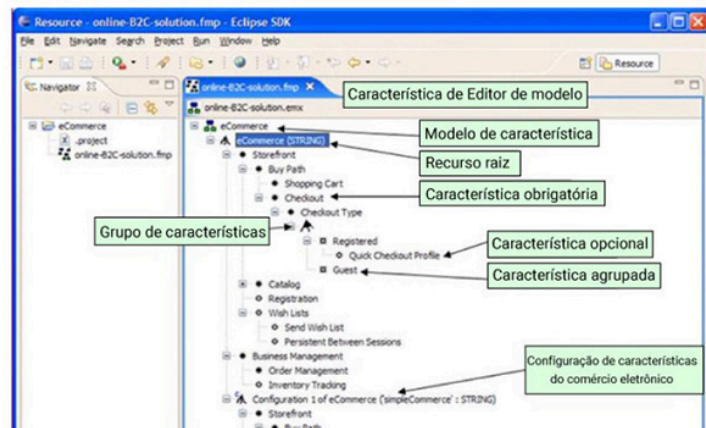
O estudo do ambiente, do negócio e das questões organizacionais de cada empresa é importante para construir o plano base dos ativos. A gestão deve dirigir, controlar e garantir a plena utilização desses ativos. Linhas de produtos de software estão mais relacionadas a práticas de negócios do que práticas técnicas.

Ferramentas

Existem algumas ferramentas disponíveis no mercado que auxiliam na modelagem do *Feature Model*. Algumas delas serão descritas a seguir (DA SILVA et al, 2011).

FMP

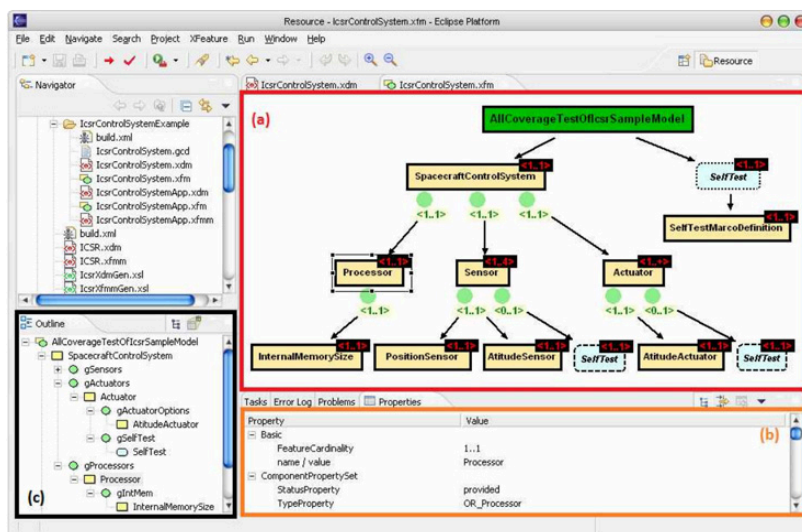
A ferramenta FMP é gratuita, pois é um plugin utilizado na IDE Eclipse. Ela oferece suporte à modelagem de domínio, apoiando a LPS. Os modelos são gravados no formato XML, padrão mundialmente conhecido. A modelagem de *features* é baseada em cardinalidade, que serve para restringir a quantidade de filhos a que uma *feature* pode estar relacionada. A seguir, temos um exemplo de interface da ferramenta FMP.



Interface da ferramenta FMP.

XFeature

Mais um plugin do Eclipse, a ferramenta XFeature permite acompanhar o processo de modelagem e configurar artefatos reusáveis, com suporte à modelagem de domínio, sendo também possível customizar meta-modelos, otimizando o desenvolvimento. Os meta-modelos são estruturas que descrevem como as *features* estão sendo modeladas, facilitando sua implementação.



Interface da ferramenta XFeature.

Originalmente, o XFeature foi desenvolvido para aplicações espaciais (sistemas embarcados de controle para naves espaciais, por exemplo);

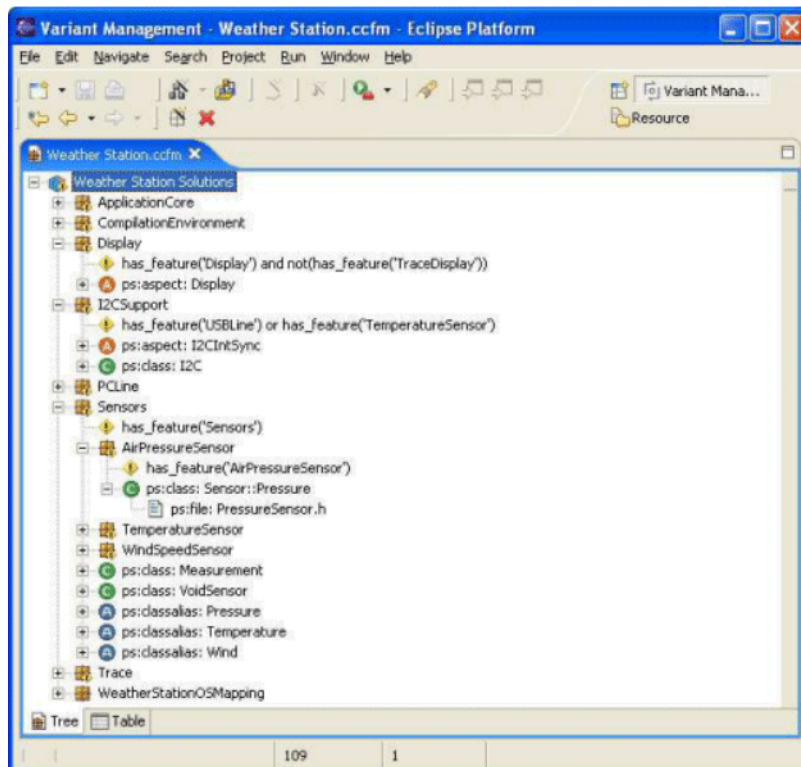
porém, não há nenhuma restrição na maneira em que foi desenvolvido para uso em outros contextos.

O XFeature foi implementado baseado na versão 3.3 do Eclipse; ao contrário do FMP, ele não utilizou o EMF para definição do meta-modelo de *feature model*. Seu editor é baseado no GEF (*Graphical Editing Framework*: traduzindo, *Framework* de Edição Gráfica) do Eclipse.

O XFeature depende fortemente de XML e de transformações XSL. Scripts são utilizados para a verificação e geração de arquivos de configuração da aplicação, tais como os que customizam o editor dos modelos.

Pure::variants

A ferramenta *pure::variants* também é um plugin do Eclipse. A ferramenta tem suporte ao desenvolvimento e à implantação de LPS e famílias de software, dando suporte às atividades de análise, modelagem, implementação e implantação. Ela permite a modelagem de domínio de *Configuration Knowledge* (Conhecimento de Configuração), permitindo o mapeamento entre *features* e artefatos de LPS.

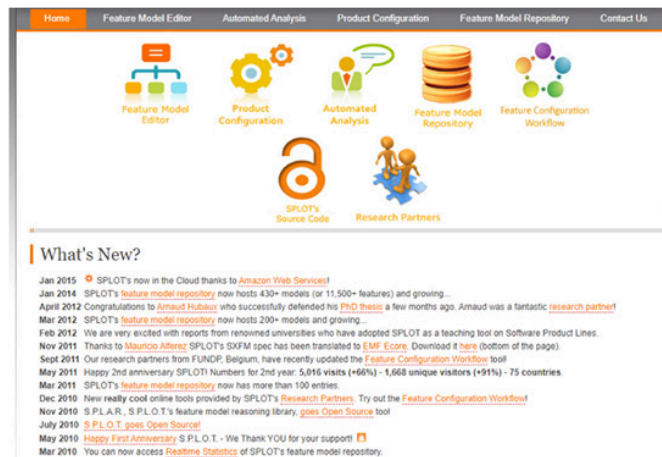


Interface de Edição do *Family Model* do *Pure::variants*.

SPLIT (*Software Product Line Tools*)

A ferramenta SPLIT (*Software Product Line Tools*: traduzindo, ferramentas da linha de produtos de software) é composta por um conjunto de funcionalidades que apoiam o desenvolvimento da LPS.

Essas funcionalidades são: editor de modelo de *feature*, análise automatizada, configuração de produto e repositório de modelo de *feature*. A seguir, é possível verificar a interface da ferramenta SPLOT.



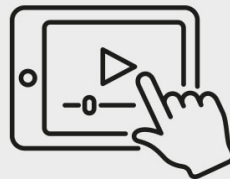
Interface da ferramenta SPLOT.



Atividades essenciais para LPS

Neste vídeo, apresentaremos as principais atividades essenciais para LPS e seus relacionamentos.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

A engenharia de software baseada em reuso é uma abordagem de desenvolvimento que maximiza o reuso do software existente. A Linha de Produto de Software faz uso dessa abordagem, com sucesso, reduzindo custos e aumentando a qualidade e confiabilidade. Para apoiar no desenvolvimento da Linha de Produto de Software, a literatura apresenta a Engenharia da LPS, que consiste em:

- A Diagramas modelados pela engenharia de software de um determinado produto.
- B Etapas, processos, metodologias e demais recursos que dão suporte ao seu desenvolvimento.
- C Uma norma ISO para LPS.
- D Processo de aquisição do produto de software.
- E Ciclo da manutenção do produto de software.

Parabéns! A alternativa B está correta.

A Engenharia da Linha de Produto de Software é muito importante, pois ela apresenta as etapas, processos, metodologias e demais recursos que irão dar suporte ao seu desenvolvimento. Com a sua adoção, se diminui consideravelmente possíveis problemas, pois todas as suas práticas são exclusivas para LPS.

Questão 2

A engenharia da LPS apresenta três atividades essenciais a serem seguidas no desenvolvimento do produto. Elas são interativas, pois podem ocorrer a qualquer momento durante o desenvolvimento da LPS. Essas atividades essenciais são:

- A Análise, Desenvolvimento do Produto e Gestão.
- B Desenvolvimento de Ativos Principais, Análise e Desenvolvimento do Produto.

- C Desenvolvimento de Ativos Principais,
Desenvolvimento do Produto e Análise.
- D Desenvolvimento de Ativos Principais,
Desenvolvimento do Produto e Gestão.
- E Desenvolvimento do Produto, Gestão e Entrega.

Parabéns! A alternativa D está correta.

São três as atividades essenciais do processo da Linha de Produto de Software: *Core Asset Development* (Desenvolvimento de Ativos Principais), *Product Development* (Desenvolvimento do Produto) e *Management* (Gestão). Estas três atividades são interativas e interligadas, sendo capazes de serem desenvolvidas em qualquer momento e em qualquer ordem durante a produção da LPS.



3 - *Framework* para LPS

Ao final deste módulo, você será capaz de reconhecer o *Framework* da Linha de Produto de Software.

Paradigma de desenvolvimento para LPS

Um *Framework* para LPS é uma estrutura para engenharia de produto que incorpora os conceitos centrais da engenharia de LPS por meio do uso de plataformas que oferecem a capacidade de fornecer personalização em massa.

No contexto de software, uma plataforma é uma coleção de artefatos reutilizáveis. Esses artefatos precisam ser reutilizados de maneira consistente e sistemática para construir aplicativos.

Para facilitar a customização em massa, a plataforma deve fornecer os meios para satisfazer os diferentes requisitos das partes interessadas. Para isso, o conceito de variabilidade é introduzido na plataforma. Como consequência da aplicação desse conceito, os artefatos que podem diferir nas aplicações da linha de produtos são modelados usando variabilidade (POHL, BOCKLE, LINDEN; 2005).

O paradigma de desenvolvimento para LPS é separado em dois processos:

Engenharia de domínio

Processo responsável por estabelecer a plataforma de reutilização definindo o que é comum e o que é variável 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

Processo 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.

Na imagem a seguir, observe uma representação do Framework LPS:

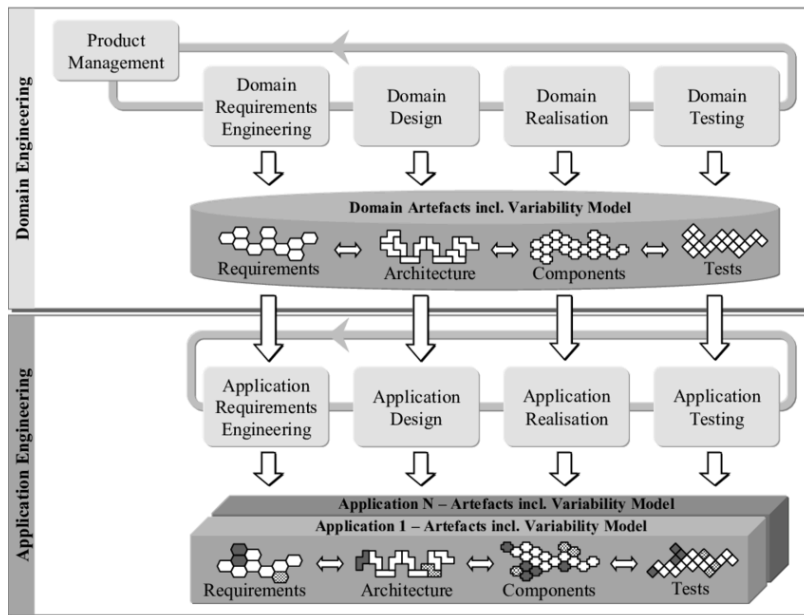


Ilustração do Framework LPS.

A imagem anterior ilustra o processo de desenvolvimento seguindo a ideia da linha de produto de software. Inicialmente, é preciso compreender o plano de negócio da empresa, informações sobre os produtos contidos nos ativos-base e dos novos produtos que serão construídos, bem como os novos requisitos, para entender qual demanda está sendo requerida.

Na fase de engenharia de domínio, é feita uma análise dos potenciais artefatos para reuso sistematizado nos ativos-base de acordo com as etapas de processo de desenvolvimento.

Por exemplo: análise de domínio nos artefatos de requisitos, arquitetura de domínio nos projetos arquiteturais, implementação de domínio nos ativos de implementação e testes de domínio nos documentos de testes reutilizáveis.

A vantagem dessa divisão é a separação dos dois pontos de atenção: construir uma plataforma robusta e criar softwares específicos aos clientes em pouco tempo. Para serem eficazes, os dois processos devem interagir de uma maneira que seja benéfica para ambos.

A seguir, detalharemos um pouco mais o *Framework LPS*.

Engenharia de domínio

Objetivos

Os principais objetivos do processo de engenharia de domínio são:

- Definir a similaridade e a variabilidade da LPS.
- Definir o conjunto de aplicativos para os quais a LPS está planejada (definir o escopo da linha de produtos de software).
- Definir e construir artefatos reutilizáveis que atinjam a variedade desejada.

A imagem a seguir mostra a parte da engenharia de domínio, dentro do Framework LPS.

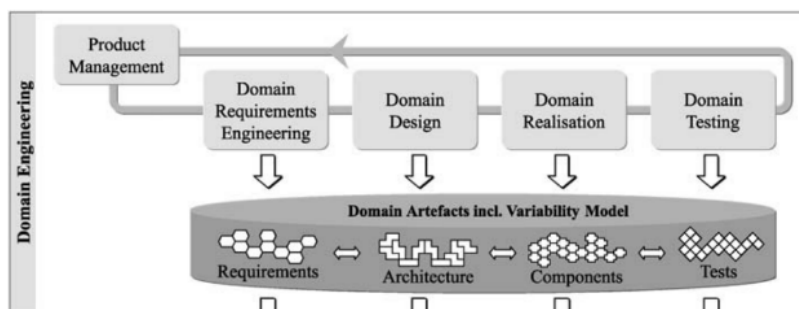


Ilustração do Framework LPS, parte de Domain Engineering.

Etapas

A seguir, veja o que cada uma dessas etapas (subprocessos) representa.

Gerenciamento de produto (*Product Management*)

Trata dos aspectos econômicos da LPS e, em particular, da estratégia de mercado. Sua principal preocupação é a gestão do portfólio de produtos da empresa ou unidade de negócios. Aqui são empregadas técnicas de escopo para definir o que está dentro do escopo da linha de produtos e o que está fora.

A entrada para a gestão de produtos consiste nos objetivos da empresa definidos pela alta administração. A saída do gerenciamento de produtos é um roteiro de produtos que determina os principais recursos comuns e variáveis de produtos futuros, bem como um cronograma com suas datas de lançamento planejadas. Além disso, o gerenciamento do produto fornece uma lista de produtos existentes ou artefatos de desenvolvimento que podem ser reutilizados para estabelecer a plataforma.

O gerenciamento de produtos antecipa alterações em recursos, restrições legais e padrões para as futuras aplicações da linha de produtos de software e formula (modela) os recursos. Isso significa que a evolução das necessidades do mercado, da tecnologia e das restrições para aplicações futuras é levada em consideração.

Engenharia de requisitos de domínio (*Domain Requirements Engineering*)



Engloba todas as atividades para elicitar e documentar os requisitos comuns e variáveis da linha de produtos. A entrada para este subprocesso consiste no *roadmap* do produto. A saída compreende requisitos reutilizáveis, textuais e baseados em modelos e, em particular, o modelo de variabilidade da linha de produtos. Assim, a saída não inclui a especificação de requisitos de uma determinada aplicação, mas os requisitos comuns e variáveis para todas as aplicações previsíveis da linha de produtos.

Projeto de domínio (*Domain Design*)



Reúne todas as atividades para definir a arquitetura de referência da linha de produtos. A arquitetura de referência fornece uma estrutura comum de alto nível para todas as aplicações da linha de produtos. A entrada para este subprocesso consiste nos requisitos de domínio e no modelo de variabilidade da engenharia de requisitos de domínio. A saída abrange a arquitetura de referência e um modelo de variabilidade refinado que inclui a chamada variabilidade interna (por exemplo, variabilidade necessária por motivos técnicos).

Realização de domínio (*Domain Realisation*)



Envolve o projeto detalhado e a implementação de componentes de software reutilizáveis. A entrada para este subprocesso consiste na arquitetura de referência, incluindo uma lista de artefatos de software reutilizáveis a serem desenvolvidos na realização do domínio. A saída da realização de domínio abrange

o design detalhado e os ativos de implementação de componentes de software reutilizáveis.

Teste de domínio (*Domain Testing*)



Responsável pela validação e verificação de componentes reutilizáveis. O teste de domínio testa os componentes em relação à sua especificação, ou seja, requisitos, arquitetura e artefatos de design. Além disso, o teste de domínio desenvolve artefatos de teste reutilizáveis para reduzir o esforço de teste de aplicativos.

A entrada para o teste de domínio compreende os requisitos de domínio, a arquitetura de referência, os designs de componentes, de interfaces e os componentes reutilizáveis implementados. A saída abrange os resultados dos testes realizados no teste de domínio, bem como artefatos de teste reutilizáveis.

Engenharia de aplicação

Objetivos

Os principais objetivos do processo de engenharia de aplicação são:

- Alcançar uma alta reutilização ao definir e desenvolver um aplicativo de linha de produtos.
- Explorar a semelhança e a variabilidade da LPS durante o desenvolvimento de um aplicativo de linha de produtos.
- Documentar os artefatos do aplicativo, ou seja, requisitos do aplicativo, arquitetura, componentes e testes, e relacioná-los aos artefatos do domínio.
- Vincular a variabilidade de acordo com as necessidades do aplicativo, desde os requisitos sobre a arquitetura, até os componentes e os casos de teste.
- Estimar os impactos das diferenças entre os requisitos do aplicativo e do domínio na arquitetura, nos componentes e nos testes.

A imagem a seguir mostra a parte da engenharia de aplicação, dentro do *Framework LPS*.

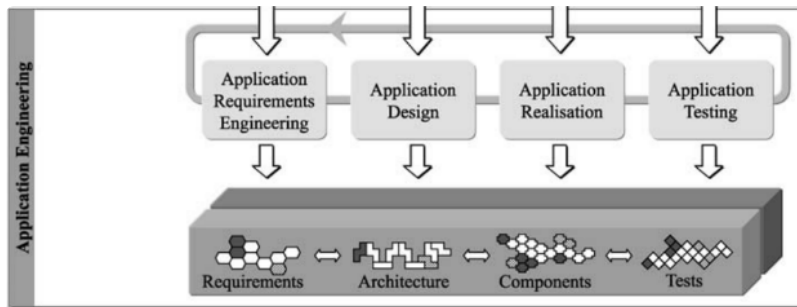


Ilustração do *Framework LPS*, parte de *Application Engineering*.

Etapas

A seguir, veja o que cada uma dessas etapas (subprocessos) representa.

Engenharia de requisitos de aplicação (*Application Requirements Engineering*)

Subprocesso que engloba todas as atividades para o desenvolvimento da especificação de requisitos de aplicação. A quantidade alcançável de reutilização de artefatos de domínio depende muito dos requisitos do aplicativo. Portanto, uma grande preocupação da engenharia de requisitos de aplicativos é a detecção das diferenças entre os requisitos de aplicativos e os recursos disponíveis da plataforma.

A entrada para este subprocesso compreende os requisitos de domínio e o *roadmap* do produto, com as principais características da aplicação correspondente.

Além disso, pode haver requisitos específicos (por exemplo, de um cliente) para o aplicativo específico que não foram capturados durante a engenharia de requisitos de domínio. A saída é a especificação de requisitos para o aplicativo específico.

Projeto de aplicação (*Application Design*)

Envolve as atividades para produzir a arquitetura do aplicativo. O projeto do aplicativo usa a arquitetura de referência para instanciar a arquitetura do aplicativo. Ele seleciona e configura as partes necessárias da arquitetura de referência e incorpora adaptações específicas da aplicação. A variabilidade limitada no

design do aplicativo está relacionada à estrutura geral do sistema considerado (por exemplo, os dispositivos de hardware específicos disponíveis no sistema).

A entrada para o design do aplicativo consiste na arquitetura de referência e na especificação dos requisitos do aplicativo. A saída compreende a arquitetura de referência do aplicativo para a arquitetura final do aplicativo em questão.

Realização da aplicação (*Application Realisation*)



Cria a aplicação em questão. As principais preocupações são a seleção e configuração de componentes de software reutilizáveis, bem como a realização de ativos específicos de aplicativos. Os ativos reutilizáveis e específicos do aplicativo são reunidos para formá-lo.

A entrada consiste na arquitetura do aplicativo e nos artefatos de realização reutilizáveis da plataforma. A saída consiste em um aplicativo em execução junto com os artefatos de design detalhados.

Teste da aplicação (*Application Testing*)



Compreende as atividades necessárias para validar e verificar um aplicativo em relação à sua especificação.

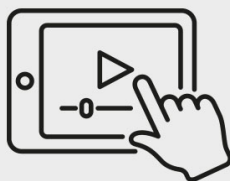
A entrada para teste de aplicativos compreende todos os tipos de artefatos de aplicativos a serem usados como referência de teste, o aplicativo implementado e os artefatos de teste reutilizáveis fornecidos pelo teste de domínio. A saída compreende um relatório de teste com os resultados de todos os testes realizados. Além disso, os defeitos detectados são documentados com mais detalhes em relatórios de problemas.



Paradigma de desenvolvimento para LPS

Neste vídeo, apresentaremos os principais conceitos sobre o paradigma de desenvolvimento para LPS.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

O *Framework* para Linha de Produto de Software (LPS) é uma estrutura para engenharia do produto que se utiliza de plataformas que garantem personalização de um determinado produto. Dois processos compõem esse *Framework*, sendo um deles a Engenharia de Domínio, na qual outros subprocessos são realizados. A seguir, eles são apresentados. Relacione a primeira coluna com a segunda, indicando a descrição correta de cada um:

Coluna1	Coluna 2
A - Gerenciamento de Produto	() Envolve todas as atividades para definir a arquitetura de referência da linha de produtos
B - Engenharia de Requisitos de Domínio	() É responsável pela validação e verificação de componentes reutilizáveis
C - Projeto de Domínio	() Trata dos aspectos econômicos da LPS, em particular, com a estratégia de mercado
D - Realização de Domínio	() Envolve o projeto detalhado e a implementação de componentes de software reutilizáveis
E - Teste de domínio	() Engloba todas as atividades para elicitar e

Coluna1	Coluna 2
	documentar os requisitos comuns e variáveis da linha de produtos

Assinale a alternativa correta, que indica a sequência de letras da Coluna 2, de acordo com a correta descrição dos subprocessos da coluna 1:

A A – B – C – D – E

B A – C – E – B – D

C C – A – E – D – B

D E – C – A – D – B

E C – E – A – D – B

Parabéns! A alternativa E está correta.

O processo de engenharia de domínio é responsável por estabelecer a plataforma de reutilização, definindo o que é comum e o que é variável da linha de produtos. O processo tem cinco subprocessos que fazem parte da sua execução: gerenciamento de produto, engenharia de requisitos de domínio, projeto de domínio, realização de domínio e teste de domínio.

Questão 2

A Engenharia de Aplicação é um dos processos que compõem o *Framework* para Linha de Produto de Software (LPS), oferecendo apoio para que seja possível personalizar os produtos. Dentro da Engenharia de Aplicação, outros subprocessos são apresentados, entre eles, a Engenharia de Requisitos de Aplicação. Sobre esse subprocesso, analise as afirmativas a seguir:

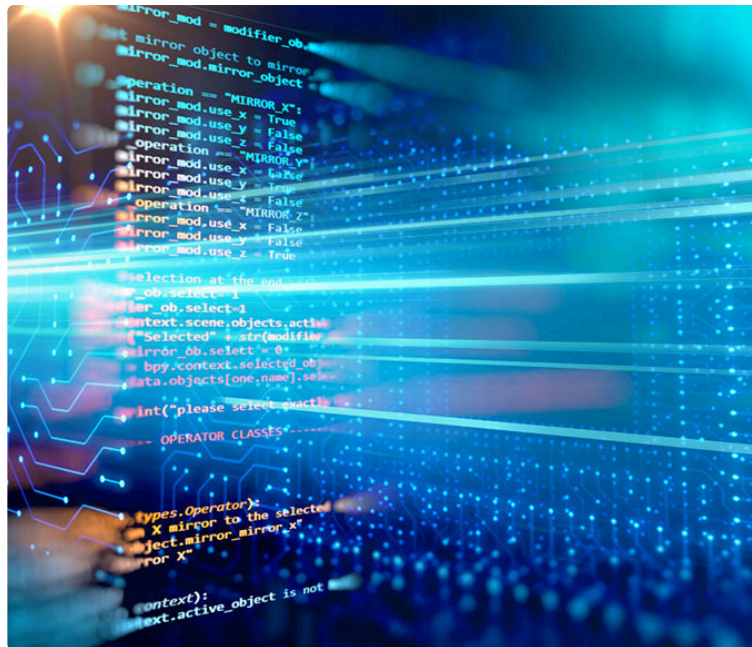
- I. É referente a todas as atividades para o desenvolvimento da especificação de requisitos de aplicação.
- II. A entrada para este subprocesso compreende os requisitos de domínio e o *roadmap* do produto.
- III. A saída deste subprocesso é a precificação da aplicação após suas alterações.
- IV. Apoia na detecção das diferenças entre os requisitos de aplicativos e os recursos disponíveis da plataforma.

Assinale a alternativa correta:

- A Apenas as afirmações I e II são corretas.
- B Apenas as afirmações I e III são corretas.
- C Apenas as afirmações I, II e III são corretas.
- D Apenas as afirmações I, II e IV são corretas.
- E Apenas as afirmações III e IV são corretas.

Parabéns! A alternativa D está correta.

O subprocesso de engenharia de requisitos de aplicação tem como saída a especificação de requisitos para o aplicativo específico e não a precificação da aplicação após suas alterações. As demais afirmativas estão corretas, pois a engenharia de requisitos de aplicação possui requisitos específicos como os de domínio e o roadmap do produto, sendo responsáveis pela diferenciação entre as condições de um aplicativo e os requisitos encontrados em uma plataforma.



4 - Benefícios e dificuldades da adoção de LPS

Ao final deste módulo, você será capaz de descrever os benefícios da adoção da linha de produto de software.

Tipos de benefícios

A adoção de LPS traz diversas vantagens que impactam positivamente nas empresas de várias formas. Essas vantagens podem ser classificadas em três tipos:

Benefícios organizacionais

Agrupam vantagens como uma melhor compreensão do domínio, maior facilidade de treinar pessoas, redução de desgaste profissional e de turnover (rotatividade de profissionais), um produto de maior qualidade e, consequentemente, confiança do cliente (já que há redução de riscos e defeitos, e aumento de previsibilidade de entrega).

Benefícios de 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, e estabelecimento de padrões

de programação. Esses aspectos melhoram muito a qualidade do produto, já que existe uma redução no número de defeitos, bem como redução no tempo de correções.

Benefícios de negócio



Dizem respeito à lucratividade, já que há uma 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, uma vez que há maior controle dos componentes do produto final.

Benefícios individuais para *stakeholders*

A Software Engineering Institute - SEI (2016) apresenta em seu *Framework* alguns benefícios individuais para cada *stakeholder* envolvido no processo da LPS. A seguir apresentaremos mais detalhes sobre eles:

Chief executive officer (CEO)

Opções para desenvolver rapidamente novos produtos; grandes ganhos de produtividade; tempo de colocação no mercado muito melhorado; crescimento sustentado e presença no mercado; opções e capacidade de capturar economicamente um nicho de mercado.

Chefe de operação

Uso eficiente da força de trabalho; capacidade de explorar novos mercados, novas tecnologias e/ou novos produtos.

Gerente técnico

Maior previsibilidade; funções e responsabilidades bem estabelecidas; produção eficiente.

Desenvolvedor do produto de software

Moral mais alta; maior satisfação no trabalho; pode se concentrar em aspectos verdadeiramente únicos dos produtos; maior integração de produtos; menos atrasos no cronograma; maior mobilidade dentro da organização; mais comercializável; tempo para aprender novas tecnologias; parte de uma equipe de construção de produtos com qualidade estabelecida.

Arquiteto ou desenvolvedor de ativos

Maior desafio; o trabalho tem mais impacto; prestígio dentro da organização; torna-se tão comercializável como a linha de produtos.

Vendedores

Produtos previsíveis de alta qualidade; entrega previsível; pode vender produtos com garantia de qualidade.

Cliente

Produtos de alta qualidade; data de entrega previsível; custo previsível; custos conhecidos para requisitos exclusivos; materiais e documentação de treinamento bem testados; custos de manutenção compartilhados; potencial para participar de um grupo de usuários.

Usuário final

Menos defeitos; melhores materiais de treinamento e documentação; uma rede de outros usuários.

Vários estudos apresentam informações que ilustram as melhorias de custos, tempo de mercado e produtividade da adoção da LPS. Veja alguns exemplos:

- A Nokia é capaz de produzir dezenas de modelos diferentes de celular por ano com a adoção da Linha de Produto de Software.
- A empresa Cummins, Inc. reduziu o tempo que leva para produzir o software de um motor diesel de cerca de um ano para uma semana.
- A Motorola observou uma melhoria de produtividade em um determinado modelo de celular.
- Hewlett-Packard (HP) reportou uma redução no *time to market* por um fator de sete e um aumento na produtividade por um fator de seis, em um modelo de sistemas de impressoras.

Dificuldades na adoção da LPS

Como pôde ser visto na Engenharia da LPS, sua adoção não é simples e requer muito esforço da organização. Esse esforço pode envolver custos, reorganização de processos e mudança da cultura da equipe. Esse tipo de fator pode ser uma barreira inicialmente, bem como comprometer sua adoção e a mudança da mentalidade na organização (PRESSMAN, 2016).

Vamos listar aqui alguns fatores que podem constituir dificuldades no uso de LPS:

Liderança comprometida com a adoção



É necessário que a adoção de uma nova abordagem como a LPS seja feita por líderes especializados, que conheçam muito bem a LPS e que possam apoiar, supervisionar, gerenciar e motivar a equipe.

Comprometimento da gerência



É necessário um acompanhamento muito próximo da gerência para que as pressões externas da organização não façam com que o projeto de LPS fique em segundo plano ou não seja seguido.

Adoção da abordagem equivocada



Não é sugerido “forçar” a adequação de um produto dentro de um produto LPS. É necessária uma análise cuidadosa e, caso o produto pretendido não tenha uma similaridade suficiente com a família, não é possível garantir a viabilidade do modelo.

Comprometimento da equipe



A falta de compromisso pode ocorrer devido à mudança da cultura organizacional da empresa e seu processo de desenvolvimento. Naturalmente, os humanos têm resistência a mudanças, e a falta de motivação pode impactar o comprometimento da equipe.

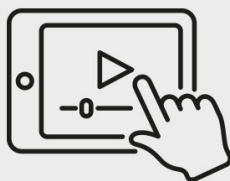
Infelizmente, a adoção da LPS ainda não é largamente utilizada em empresas de desenvolvimento de software. Porém, a engenharia de software vem introduzindo esse assunto, bem como apresentando melhores práticas. Outro incentivo tem sido as pesquisas na área da LPS, mostrando casos reais de sucesso, um ótimo indicativo de que a LPS ainda tem muito a evoluir em sua adoção nas organizações.



Vamos conhecer os benefícios para *stakeholders*?

Neste vídeo, apresentaremos os principais conceitos sobre o paradigma de desenvolvimento para LPS.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

São várias as vantagens que a LPS traz para as empresas, principalmente pela sua abordagem de reuso, em que um mesmo software (com adaptações/customizações) pode ser adquirido por várias empresas do mesmo domínio de negócio. São vários os *stakeholders* envolvidos positivamente nesta adoção, conforme a SEI apresenta em seu *Framework*. Assinale a opção a seguir que indique o benefício trazido ao usuário final, na adoção da LPS:

- A Moral mais alta, maior satisfação no trabalho, poder se concentrar em aspectos verdadeiramente únicos dos produtos.
- B Integração de produtos, menos atrasos no cronograma, maior mobilidade dentro da organização, mais comercializável, tempo para aprender novas tecnologias.
- C Produtos previsíveis de alta qualidade, entrega previsível, poder vender produtos com garantia de qualidade.
- D Produtos de alta qualidade, data de entrega previsível, custo previsível, custos conhecidos para requisitos exclusivos mentos, materiais e documentação de treinamento bem testados.

- E Menos defeitos, melhores materiais de treinamento e documentação, uma rede de outros usuários.

Parabéns! A alternativa E está correta.

A adoção de LPS traz diversas vantagens, que impactam positivamente nas empresas e em seus stakeholders envolvidos, incluindo o usuário final, que é beneficiado de várias formas, trabalhando com um sistema que apresenta menos defeito, com materiais e documentação, visto que o sistema já é utilizado por uma ampla comunidade.

Questão 2

A adoção da LPS tem uma engenharia envolvida, em que temos o Framework de apoio para auxiliar nos processos. Mas não é simples se adequar a todos os processos e isso requer muito esforço da organização. Questões culturais podem ser problemáticas também: mesmo tendo seus artefatos altamente reutilizáveis, a maioria das empresas de software não adota o reuso. Nessa linha, analise as afirmativas a seguir:

- I. A adoção de uma nova abordagem como a LPS deve ser realizada por líderes que a entendam e que garantam um acompanhamento cuidadoso junto à sua equipe.
- II. O gerenciamento cuidadoso é importante para que o projeto LPS siga em primeiro plano na organização.
- III. A adequação deve sempre ocorrer, independentemente da aderência da área de negócio do cliente, pois, se o software for reusado, vai sempre trazer lucro para as empresas.
- IV. A mudança da cultura organizacional da empresa e seu processo de desenvolvimento pode gerar falta de compromisso por parte dos colaboradores.

Assinale a alternativa correta:

- A Apenas as afirmativas I e II estão corretas.
- B Apenas as afirmativas I, II e III estão corretas.

- C Apenas as afirmativas I, II e IV estão corretas.
- D Apenas as afirmativas III e IV estão corretas.
- E Apenas as afirmativas II e IV estão corretas.

Parabéns! A alternativa C está correta.

A adoção da abordagem equivocada é um problema para LPS, pois não é sugerido “forçar” a adequação de um produto dentro de um produto LPS. É necessária uma análise cuidadosa e, caso o produto pretendido não tenha uma similaridade suficiente com a família, não é possível garantir a viabilidade do modelo, o que vai contra a afirmativa de que a adequação deve sempre ocorrer, independentemente da aderência da área de negócio do cliente, pois, se o software for reusado, vai sempre trazer lucro para as empresas.

Considerações finais

A engenharia de software apresenta abordagens e técnicas com o objetivo de otimizar o reuso dos artefatos de software e obter todas as vantagens que ele oferece, como agilizar o processo de desenvolvimento e atender às demandas do mercado.

Entre as opções disponíveis, uma estratégia é a Linha de Produto de Software (LPS), que consiste em realizar o reuso de produtos que pertençam a uma mesma família, ou seja, a um mesmo domínio de mercado. Para sua compreensão, existe uma Engenharia bem definida que descreve os processos e subprocessos, bem como os artefatos de cada etapa.

Mas, para reusar componentes de um software ou até mesmo um software inteiro, é preciso planejamento. O reuso é uma estratégia que já vem sendo utilizada pelas grandes empresas para agilizar e reduzir os custos de desenvolvimento.

Para apoiar sua adoção, existe um *Framework* de melhores práticas, que auxilia no desenvolvimento das etapas com base na Engenharia da LPS.

Para obter sucesso em sua adoção, trabalhar com LPS requer uma mudança cultural, com foco no reuso, que nem todas as organizações possuem no seu dia a dia. Apoio da liderança e foco na qualidade também são questões importantes para o sucesso de sua adoção.

Podcast

Para encerrar, ouça a entrevista com os principais pontos abordados neste estudo.

Para ouvir o *áudio*, acesse a versão online deste conteúdo.



Explore +

Ouçã o podcast *Reuso de Software*, com Eduardo Almeida, do canal Fronteiras da Engenharia de Software.

Procure no YouTube por vídeos sobre linhas de montagens de automóveis e celulares para entender como a arquitetura do produto pode ser similar, com pequenas variações, e adotar o reuso.

Referências

CLEMENTS, Paul; NORTHROP, Linda. **Software product lines: practices and patterns** [book review]. Software, IEEE, 131 –132, 2002. Consultado na internet em: 1 set. 2022.

DA SILVA, F. A. P. *et al.* **Linhas de produtos de software**: uma tendência da indústria. Porto Alegre: Sociedade Brasileira de Computação, 2011.

MENDONÇA, Marcílio; BRANCO, Moises; COWAN, Donald. **S.P.L.O.T. - Software product lines online tools**. Companion to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA, Orlando, USA. 2009. Consultado na internet em: 1 set. 2022.

POHL, Klaus; BOCKLE, Günter; LINDEN, Frank van der. **Software product line engineering**: foundations, principles, and techniques. Essen: Springer, 2005.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de software**. Porto Alegre: AGMH, 2016.

SAP, Portal ERP. **iX Tecnologia e Educação Ltda**, 2022. Consultado na internet em: 15 jul. 2022.

SEI, Software Engineering Institute. **A framework for software product line practice**, version 5.0. 2016.

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson Prentice Hall, 2011.

Material para download

Clique no botão abaixo para fazer o download do conteúdo completo em formato PDF.

Download material

O que você achou do conteúdo?



 Relatar problema

