

LINHAS DE PRODUTO DE SOFTWARE: TORNANDO REALIDADE O REUSO SISTEMATIZADO DE SOFTWARE

SOFTWARE PRODUCT LINES: MAKING SYSTEMATIC REUSE OF SOFTWARE A REALITY

RESUMO

Reuso tem sido uma das formas pelas quais a engenharia de software tem buscado aumentar seus ganhos em produtividade e qualidade, bem como reduzir os custos de desenvolvimento e evolução. No entanto, o reuso granular e oportunista não tem se mostrado a forma ideal para se alcançar estes objetivos. Este artigo oferece uma reflexão sobre a abordagem de linhas de produto de software como forma planejada e estruturada de alcançar o esperado reuso sistematizado de software.

Palavras-chave: Linhas de Produtos de Software, Engenharia de Famílias de Produtos, Reuso de Software.

ABSTRACT

Reuse has been one of the ways that software engineering community has been applying to leverage productivity and quality, as well as to reduce the development and evolution costs. However, small grain and opportunistic reuse has not been shown as the perfect path to reach such goals. This paper offers a reflection about the software product line approach as a planned and structured means of reaching the desired systematic reuse.

Keywords: Software Product Lines, Product Family Engineering, Software Reuse.

Sheila dos Santos Reinehr

Professora da PUCPR
Engenheira Mecânica (UFPR)
Doutoranda em Engenharia de
Produção (USP)
sheila.reinehr@pucpr.br

*Marcelo Schneck de Paula
Pessoa*

Professor da Escola Politécnica
da USP
Engenheiro Eletrônico (USP)
Doutor em Engenharia
Eletrônica (USP)
mpessoa@usp.br

Roberto Cesar Durscki

Engenheiro de Computação
(PUCPR)
Mestrando em Engenharia de
Produção (USP)
roberto.durscki@gmail.com

Mauro de Mesquita Spinola

Professor da Escola Politécnica
da USP
Engenheiro Eletrônico (ITA)
Doutor em Computação (USP)
mspinola@usp.br

Marco Antonio Paludo

Professor da FESP
Bacharel em Ciência da
Computação (PUCPR)
Doutorando em Informática
Aplicada (PUCPR)

Robert Carlisle Burnett

Pró-Reitor de Graduação, Pós-
Graduação e Pesquisa da
PUCPR
Doutor (New Castle Upon Tyne
- Inglaterra)
robert.burnett@pucpr.br

Esse artigo é um subproduto do projeto MILPS de pesquisa científica do grupo Qualidade de Processo, Projeto e Produto de Software da PUC-PR, financiado parcialmente com recursos do CNPq (PI: 55.2217/02-6).

INTRODUÇÃO

Nas últimas décadas a Engenharia de Software vem buscando formas de conseguir aumentar a produtividade no desenvolvimento de software na mesma proporção que a indústria de hardware conseguiu aumentar a capacidade dos processadores. Para isto, tem buscado alternativas que auxiliem a remover as barreiras tradicionais: a intangibilidade do software, a dependência de atividades cognitivas durante todo o ciclo de desenvolvimento, a dificuldade no gerenciamento de requisitos, a imprecisão nas estimativas, e mais uma lista interminável de aspectos ainda sem uma solução definitiva na área de software.

No final da década de 80, com o início da divulgação dos métodos orientados a objetos, uma nova esperança surgiu para a indústria de software: a promessa de que isto facilitaria o reuso e, portanto, aumentariam as chances de se desenvolver produtos de maior qualidade, em menos tempo e com menor custo. O que se constatou em seguida foi que o reuso granular e oportunista de partes de código, ou de algumas classes, não traria os benefícios esperados e que seria necessário pensar de forma mais abrangente, buscando um reuso mais intensivo e planejado.

Antes disto, David Parnas (PARNAS, 1979) já apontava para uma abordagem que, posteriormente, viria a se tornar foco de atenção das indústrias, especialmente as que desenvolvem sistemas de software embarcado: a engenharia de famílias de produtos, como chamado na Europa ou, linhas de produto de software, como chamado nos Estados Unidos. Com este paradigma, as organizações que anteriormente abordavam o desenvolvimento de software projeto a projeto passariam a concentrar seus esforços na criação e manutenção de uma família de produtos. O foco passa então a ser a exploração das similaridades entre os produtos e o controle da variabilidade dos membros da família e não mais o desenvolvimento de um sistema único de cada vez.

Esse trabalho tem como objetivo apresentar o modelo de linhas de produto de software, realizando uma análise reflexiva a respeito dos benefícios que sua adoção pode trazer para as organizações, bem como os riscos envolvidos em iniciativas desta natureza. Através desta exploração, tem-se o objetivo de levar o

leitor a refletir sobre a seguinte questão: “Seriam as linhas de produto de software a forma pela qual o reuso será efetivamente sistematizado nas organizações?”.

Este trabalho está assim organizado: a seção 2 descreve a terminologia e os principais conceitos, a seção 3 descreve os benefícios de sua aplicação, a seção 4 descreve os riscos; a seção 5 mostra a análise de viabilidade e modelos de custo; apresenta a seção 6, os comentários finais e, em seguida, as referências bibliográficas.

1 A QUESTÃO DA TERMINOLOGIA E OS CONCEITOS ESTABELECIDOS

Em uma clara alusão às linhas de produção da indústria de manufatura, o termo linhas de produto de software (*software product lines - LPS*) foi adotado pela comunidade americana que trabalha com o tema. Sua inspiração remonta ao final do século XIX, onde as indústrias introduziram a revolução no processo produtivo, sugerindo o desenvolvimento seqüencial de produtos, baseado em tarefas repetitivas e especializadas. Por sua vez, a comunidade européia, percebendo a necessidade de focar a etapa de concepção dos produtos como forma de garantir o reuso efetivo, batizou a abordagem de engenharia de família de produtos (*product families engineering - PFE*). Apesar dos caminhos relativamente distintos e paralelos, atualmente, parece que vem se firmando o termo americano. Isto fica evidenciado com a recente união das duas principais conferências do setor (*Product Family Engineering*, da Europa e *Software Product Lines*, dos USA), que optaram pelo título: *International Conference on Software Product Lines (LPSC)*.

Segundo Cohen (COHEN, 2002), a definição de *software product line* com maior aceitação na indústria é a de Clements e Northrop:

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. (CLEMENTS; NORTHROP, 2002)

Dentro desta definição, alguns termos devem ser destacados, pois representam as características mais importantes de uma linha de produto e sustentam as principais vantagens propostas pelo modelo:

- conjunto comum de ativos: Os ativos são a essência da linha de produto e correspondem a um conjunto de elementos customizáveis, utilizados na

construção dos softwares produzidos (produtos). Entre os ativos estão, por exemplo: componentes de software, modelos de documentos utilizados no processo (artefatos), *design-patterns* utilizados pela equipe de desenvolvimento, documentação dos requisitos comuns à família de produtos, a arquitetura da linha de produtos (que será a base da arquitetura de cada produto gerado), cronogramas etc. Dentre estes elementos, a arquitetura é o elemento chave e normalmente é estudada a parte dos outros ativos;

- forma preestabelecida: O processo de produção de softwares através de uma linha de produto é realizado através de planos de produção. Esses planos são definidos para cada software (produto) que será produzido pela linha. Ao definir um plano de produção, que dará origem a um novo produto, devem-se relacionar quais ativos farão parte deste produto e assim estabelecer um vínculo aos processos anexos de cada ativo utilizado. Os processos anexos são pequenos processos de utilização contidos em cada ativo e que definem o que o ativo faz, qual a sua flexibilidade, qual a técnica de configuração do ativo (se ele for flexível). Essa natureza preestabelecida de funcionamento do processo produtivo da linha de produto é que garante o ganho de tempo e confiabilidade no desenvolvimento dos produtos;
- segmento particular ou missão: também chamado de domínio, refere-se ao corpo de conhecimento ou à área de especialização em que a linha de produto atua. O domínio está diretamente relacionado ao conjunto de funcionalidades correlacionadas que os produtos da linha visam atender. Como a flexibilidade dos ativos (especialmente da arquitetura) é limitada, é preciso delimitar um segmento de atuação. Sem essa delimitação, o escopo da linha de produto poderia ser muito abrangente, o que tornaria muito oneroso criar e manter o conjunto comum de ativos.

Como se pode observar pela definição, esses conceitos foram diretamente derivados da indústria de manufatura, ou seja, da filosofia de dividir para conquistar. Resumindo: o processo sugere dividir o produto em pequenas partes, mais compreensíveis e gerenciáveis (ativos); possuir especificações claras e sem ambigüidades para produção / montagem (forma preestabelecida) e, atender a um domínio específico (segmento particular ou missão) para que, como em uma linha de

montagem da indústria tradicional, cada novo produto seja uma reutilização de partes de produtos construídos anteriormente, pertencentes à mesma família, e, preferencialmente, com um mínimo de esforço adicional de construção.

É importante ainda, compreender o significado do termo domínio para o contexto de LPS. A literatura não apresenta um consenso em sua definição, no entanto, muitos autores concordam com o conceito de Clements e Northrop (2002): “Domínio corresponde a uma área de conhecimento ou atividade caracterizado por uma coleção de conceitos e terminologias compreendidos e aceitos pelos profissionais da área”.

Essa definição, no entanto, é muito vaga o que acaba gerando interpretações distintas. Em um sentido mais amplo, domínio pode ser considerado como uma área de atuação de mercado, a qual poderá ser o objetivo dos produtos de uma linha de produto de software.

Uma outra definição, mais restrita que a anterior, é a de que domínios são grupos de aplicativos de software voltados para um determinado nicho de mercado, como sistemas gestores de bancos de dados, sistemas de intranet, sistemas de CRM etc. Nestes casos a definição de domínio está mais voltada para os produtos do que para o mercado.

Finalmente, domínio pode ser considerado uma subárea dos conhecimentos envolvidos no desenvolvimento dos produtos de softwares em uma linha de produto. Assim, uma linha de produto que desenvolve sistemas bancários vai possuir ativos (componentes e artefatos) dos domínios de interface com usuário, comunicação, segurança, algoritmos financeiros etc.

2 BONS MOTIVOS PARA ADOTAR LINHAS DE PRODUTO DE SOFTWARE

Em seu trabalho com arquiteturas para reuso, especialmente focando a abordagem de famílias de produto, Jan Bosch destaca:

Os esforços para alcançar o reuso de software levaram à aceitação geral de duas importantes lições. A primeira é: o reuso oportunista (...) não é efetivo da prática; um programa de reuso bem sucedido em uma organização precisa ser um esforço planejado e pró-ativo. A segunda é que o reuso de baixo para cima (*bottom-up*), ou seja, a composição arbitrária de componentes para construir sistemas, não funciona na prática; programas de reuso de sucesso precisam adotar uma abordagem de cima para baixo (*top-down*). (BOSHC, 2000, p. 162)

O primeiro bom motivo para pensar na adoção de famílias de produtos é o constatado por Bosch: reuso granular e oportunista não trouxe para a indústria os benefícios esperados (redução no tempo do ciclo de desenvolvimento, menor custo para produzir novos produtos, mais qualidade nos produtos gerados etc.). Através do reuso ocasional de partes do código ou de partes da arquitetura não se conseguiu os ganhos de escala pretendidos. O Quadro 1 apresenta, de forma resumida, os benefícios tangíveis e intangíveis alcançados pelas organizações que adotam a abordagem, compilados por (COHEN, 2003).

Quadro 1 - Benefícios tangíveis de uma LPS

BENEFÍCIOS TANGÍVEIS	
Lucratividade	O repositório de ativos permite que a organização produza produtos voltados para um segmento específico de mercado. O benefício dessa focalização é observado no aumento da participação de mercado e aumento da lucratividade.
Qualidade	Uma redução no número de defeitos relatados é comum em sistemas desenvolvidos em linhas de produto. A qualidade também pode ser medida em termos de redução do tempo de correções e da redução do efeito dominó (geração de novos defeitos a partir de correções executadas).
Performance dos produtos de software	A utilização de ativos aumenta a performance em relação ao desenvolvimento tradicional, especialmente com o aumento da maturidade da linha, o que faz com que os ativos estejam cada vez mais otimizados.
Tempo de integração	O tempo de integração no desenvolvimento incremental é facilitado.
Produtividade	A equipe de desenvolvimento pode ser reduzida. O custo total de desenvolvimento é cortado consideravelmente. O cronograma é reduzido (maior velocidade de lançamento). O sistema possui uma flexibilidade documentada, o que facilita o atendimento das solicitações de modificações do cliente.
BENEFÍCIOS INTANGÍVEIS	
Desgaste de profissionais	Menor desgaste dos profissionais, o que resulta em uma redução do <i>turnover</i> de membros da equipe.
Aceitação dos desenvolvedores	Após um treinamento inicial, os desenvolvedores relatam satisfação em trabalhar com a abordagem baseada em ativos e arquitetura comuns.
Satisfação profissional	Os desenvolvedores relatam que o trabalho braçal já foi realizado (desenvolvimento dos ativos de software), assim eles podem se concentrar em atividades mais interessantes, como o aperfeiçoamento e / ou inovação de elementos específicos.
Satisfação do Cliente	Os ativos reduzem os riscos, aumentando a previsibilidade da entrega e diminuindo a taxa de defeitos. Esses fatores afetam positivamente o cliente (induzindo-o a preferir produtos derivados de linhas de produto).

FONTE: adaptado de Cohen (2003)

Em Brownsword (1996) o autor relata os números que quantificam os benefícios experimentados pela CelsiusTech, empresa nórdica que se dedica à confecção de produtos para a marinha sueca:

- Inversão da participação do software no custo total do produto final: de 65% para 20%;

- Redução da mão de obra de desenvolvimento de software (dos projetos de uma família) de 200 pessoas para menos de 50;
- O tempo de entrega (*time-to-market*) passou de alguns anos para meses;
- Constatou-se que 70% a 80% dos sistemas de software resultantes eram compostos de componentes do repositório de ativos;
- Houve um aumento na qualidade dos sistemas desenvolvidos e na satisfação dos clientes.

Em (ENGELSMA, 2001) os autores relatam experimentos realizados no âmbito do projeto europeu ESAPS, com as empresas: Philips, Bosch, Siemens, Nokia, Alcatel, Thales, Sainco, Market Maker e União Fenosa. Dentre as conclusões, os autores destacam a dificuldade que existe em quantificar os benefícios deste tipo de experimento, uma vez que dificilmente são estabelecidos objetivos quantitativos. Por exemplo, entre os objetivos iniciais de algumas destas empresas estavam termos genéricos como: processo de teste mais claro, melhor documentação dos testes, menor tempo de testes, melhor relacionamento entre teste e requisitos etc. Além disto, alguns benefícios são de natureza intrinsecamente qualitativa, impossibilitando uma comparação numérica. Dentro deste conjunto de experimentos, um dos que obteve informações quantitativas foi o executado pela Philips, relacionado a um sistema para a área de imagens médicas:

- O *throughput* do projeto foi reduzido para 70-80%, sem afetar a confiabilidade dos sistemas;
- O *throughput* de fabricação do produto foi reduzido em aproximadamente 40%.

Os resultados da Cummins, tradicional fabricante de motores, reportados em (CLEMENTS; NORTHROP, 2002) apontam para números ainda mais expressivos:

- Mais de 20 grupos de produtos foram lançados com sucesso, gerando mais de 1000 aplicações diferentes de motores.
- Na média, 75% de todos os seus produtos de software são atualmente provenientes de sua base de ativos;

- Antes da adoção de LPS, de cada 10 projetos: 3 estavam em dia, 4 falhavam e 4 estavam com problemas. Após a adoção, de 15 projetos, 15 estavam em dia.
- Desenvolvedores mais portáteis entre os projetos, com curva de aprendizagem bastante reduzida.
- Seriam necessários mais de 360 engenheiros de software usando a abordagem tradicional de desenvolvimento para produzir as mesmas funcionalidades que foram geradas com 100 engenheiros de software usando LPS.
- Estimam que a taxa de custo x benefício seja de 10:1.

3 CUIDADOS PARA UMA IMPLANTAÇÃO BEM SUCEDIDA

Conforme abordado anteriormente, uma implantação de linhas de produto de software não é simplesmente um reuso acidental de código. Em Bosch (2002, p. 257), temos: “Linhas de produto de software não aparecem acidentalmente, mas requerem um esforço explícito e consciente da organização interessada em empregar a abordagem de linha de produto”.

Diversos são os fatores que podem atrapalhar o sucesso na adoção da abordagem. Existem fatores de ordem técnica, que podem mais facilmente ser vislumbrados. No entanto, normalmente são os fatores relacionados com os aspectos não técnicos que mais dificultam qualquer iniciativa de introdução de melhorias organizacionais. Cohen levanta, em (COHEN, 2002), diversos fatores não técnicos, como:

- Falta de um líder comprometido: como qualquer iniciativa de mudança cultural e introdução de novas tecnologias, o processo exige a alocação de um líder que acredite nos princípios do modelo e que supervisionará o processo e motivará as pessoas. Seu apoio é fundamental para comunicar os resultados à gerência e apoiar os desenvolvedores em momentos de dificuldade, evitando que eles desviem do modelo. Caso esta pessoa não receba a autoridade necessária, não acredite no modelo, ou simplesmente não exista, é

muito difícil manter o foco e a confiança durante o processo e há um grande risco de insucesso.

- Falta de compromisso da gerência: a gerência deve estar convencida da viabilidade e das vantagens que o projeto vai trazer para a organização. É comum que projetos de implantação de linhas de produto comecem com força total, mas ao longo do tempo, por pressão do mercado e por falta de foco, tornem-se projetos coadjuvantes e logo caminhem para o esquecimento. Esse tipo de situação é muito ruim para a organização, pois desmoraliza os idealizadores além de desperdiçar recursos.
- Abordagem inadequada: a linha de produto deve ser um projeto estratégico para atingir objetivos corporativos. Apesar de seus objetivos variarem de empresa para empresa, eles estão sempre baseados no ganho de produtividade, custo e qualidade, baseados na exploração das similaridades de produtos de uma família. Caso os produtos planejados para a linha não possuam similaridades suficientes para garantir a viabilidade do modelo, a linha poderá nunca trazer o retorno planejado e tornar-se um fim em si mesma.
- Falta de compromisso da equipe: o líder da linha deve obter o compromisso da equipe desenvolvedora com os princípios e com o processo de desenvolvimento da linha. Uma equipe que desacredita o modelo impossibilita o processo. Uma maneira de evitar esse tipo de problema é envolver membros da equipe nas atividades de elaboração da linha fazendo-os sentir-se parte do projeto.
- Interação insuficiente entre as equipes: A iniciativa de uma linha de produto transcende os limites tradicionais da organização. O processo acaba, inevitavelmente, envolvendo diversas equipes como gerência, marketing e engenharia. Uma falta de interação e colaboração entre essas equipes pode impedir a obtenção dos objetivos planejados. Em (EBERT; SMOUTS, 2003) os autores detalham que o envolvimento de marketing na definição do escopo e dos requisitos de uma linha de produto é fundamental para o sucesso do projeto. Se isto não ocorrer, o pessoal de marketing pode vender produtos que a LPS não suporta,

ou, em contrapartida, o pessoal de engenharia pode permitir variabilidades na LPS que não são demandadas pelo mercado.

- Padronizações inadequadas: Há uma tendência nas organizações de assumir que a adoção de padrões irá guiar automaticamente a implantação de uma LPS. A institucionalização prematura e impensada de padrões, muitas vezes obsoletos, pode limitar as opções de tecnologia da linha, prejudicando seu escopo. Os padrões devem ser estudados caso a caso e, apesar de serem importantes para a linha, devem ser discutidos antes de sua implementação.
- Adaptação insuficiente: Assim como a arquitetura e os componentes devem permitir certo grau de adaptabilidade dentro da linha, as práticas organizacionais (padronizadas ou não) devem permitir certo grau de personalização voltado para as características da linha. A falta de adaptabilidade (ou uma adaptação mal estruturada) pode subotimizar a performance das equipes ou gerar desvios imprevistos das atividades envolvidas no processo da linha de produto.
- Evolução da abordagem: Caso o processo da linha de produto não seja revisado e atualizado periodicamente, suas práticas irão tornar-se obsoletas e inapropriadas às novas necessidades que surgirão. Neste caso, novas práticas não-previstas irão surgir informalmente devido às necessidades e poderão comprometer todo o modelo.
- Falha de disseminação: O líder da linha deve tomar a responsabilidade de desenvolver e distribuir a documentação em nível e tipo apropriados para a organização, treinar os envolvidos, e apoiar o processo no que for necessário.

Evidentemente, nenhuma organização irá enfrentar todas as dificuldades e desafios citados acima, no entanto, a pesquisa realizada em (COHEN, 2002) revelou que pelo menos três ou quatro destes elementos estiveram ou estão presentes nas organizações entrevistadas conforme o Quadro 2.

Quadro 2 - Problemas identificados pelas empresas

PROBLEMA	% EMPRESAS
Resistência Organizacional	52%
Resistência Gerencial	36%
Resistência dos Desenvolvedores	32%
Preocupações e desconfianças com o tamanho do investimento	45%
Falta de recursos devidamente treinados e capacitados	29%
Incapacidade de analisar e prever o impacto da implantação	19%
Preocupação sobre o longo tempo para retorno do investimento	18%

FONTE: adaptado de Cohen (2003)

Ahmed e Capretz estudaram os efeitos que os fatores relacionados aos objetivos estratégicos de negócio exercem sobre o desempenho das linhas de produto de software (AHMED; CAPRETZ, 2007). Foram pesquisadas 8 organizações, todas com mais de 5 anos de uso de LPS, mais de 2000 funcionários, localizadas na Europa e USA. Foram analisados aspectos como: planejamento estratégico, ordem de entrada no mercado, estratégia do nome da marca, orientação do mercado, gerenciamento de relacionamentos, visão de negócios, inovação e desempenho do negócio da LPS. As conclusões obtidas foram:

- A direção futura do negócio precisa considerar as LPSs como um ativo.
- O processo de LPS necessita de recursos que precisam ser delegados no plano estratégico.
- O planejamento estratégico deve delinear claramente o que deverá ser desenvolvido a partir da LPS de forma a obter vantagens competitivas e ganhar segmentos de mercado para alcançar objetivos de negócio.
- A gerência de relacionamentos desempenha um papel significativo no desenvolvimento com sucesso da LPS.
- A ordem de entrada no mercado possui uma correlação positiva com a posição ocupada pelo produto neste mercado.

Muitos são os riscos que estão envolvidos em uma adoção de linhas de produto de software, mas talvez o que mais mereça atenção seja a escolha da abordagem mais adequada para cada situação: evolucionária ou revolucionária. Na primeira, parte-se da evolução de componentes ou produtos existentes, de forma a atingir os objetivos de transformá-los em uma família. Na segunda, a linha é estabelecida e então começa o desenvolvimento de ativos e então a produção dos membros da família. De forma geral, a abordagem revolucionária pressupõe mais riscos, mas também ganhos mais elevados (BOSCH, 2000).

4 VIABILIDADE E JUSTIFICATIVA DA INICIATIVA

Os ganhos reportados pelas empresas que implantaram LPS oferecem uma visão positiva sobre a sua viabilidade. No entanto, cada empresa está inserida em um contexto diferente de negócios e possui condições igualmente diferenciadas de infra-estrutura, tecnologia e pessoas. Conforme exposto nos tópicos anteriores, o modelo apresenta muitas perspectivas de melhorias, assim como, de dificuldades e riscos. Essa é uma dicotomia tradicional nos projetos de mudança/ inovação tecnológica, e demanda muito cuidado da organização e, acima de tudo, um estudo criterioso para avaliar a viabilidade e o custo x benefício da aplicação da nova tecnologia na empresa.

4.1 Método SIMPLE

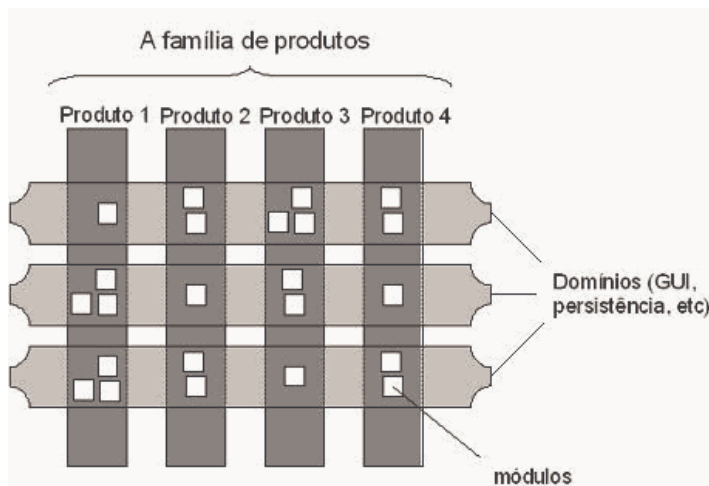
Diversos trabalhos vêm sendo desenvolvidos no sentido de apoiar esta análise, dentre eles o método SIMPLE (*The Structured Intuitive Model for Product Line Economics*) para avaliação baseada em indicadores financeiros de retorno, descrito em (CLEMENTS; MCGREGOR; COHEN, 2005). Os autores se baseiam em quatro funções básicas de custo, de forma a estimar medidas para a construção, sustentação e evolução das LPSs: custo para introduzir a LPS, custo para desenvolver a base de ativos, custo para desenvolver a parte única de um produto e custo para reusar da base de ativos.

4.2 Modelo de Avaliação Orientado a Objetivos para Domínios de LPS

Este modelo, desenvolvido pela Avaya Telecom, tem como objetivo priorizar os domínios que serão inseridos em uma linha de produto considerando que esta será implantada gradativamente, de forma incremental. O modelo apóia a identificação de quais módulos de uma família de sistemas são prioritários para o desenvolvimento no paradigma de LPS. Em (WEISS; GEPPERT, 2003), os autores consideram como domínio as subáreas de conhecimento envolvidas no desenvolvimento de uma família de sistemas.

Para avaliar a probabilidade de sucesso de um domínio no modelo de linha de produto, os autores decompõem os domínios em módulos para poder avaliá-los com mais clareza. Esses módulos são os componentes físicos das subáreas de conhecimento, conforme pode ser observado na Figura 1.

Figura 1 - Módulos e domínios



FONTE: adaptado de Weiss e Geppert (2003)

O modelo sugere entrevistas com os especialistas de cada domínio para obter uma pontuação de cada módulo em diversos critérios utilizados na análise dos domínios. Esses critérios são separados em quatro grupos:

- Atividade: quão ativos são os módulos de um domínio (frequência com que são modificados e a quantidade de manutenção que requerem).
- Retorno: importância do domínio para os objetivos de mercado da empresa, a importância do domínio para o cliente, seu potencial de gerar produtos diferenciados etc.
- Independência: grau de dependência de um módulo com seus módulos adjacentes (quanto mais dependente um domínio for de outros, mais difícil será para desenvolvê-lo independentemente em uma linha de produto);
- Viabilidade: questões como complexidade de código, maturidade dos módulos e restrições de recursos (capital e humano).

Cada um dos grupos é ainda composto por diversos critérios, e sua graduação é consequência da aplicação de fórmulas que relacionam de forma ponderada os critérios constituintes. Com o resultado da graduação de cada grupo, é possível obter uma graduação normalizada de dois parâmetros: impacto corporativo e probabilidade de sucesso. Mapeando os diversos módulos numa matriz que confronta essas variáveis, é possível selecionar os domínios mais

promissores para ingressarem no paradigma de LPS, ou seja, aqueles com alta probabilidade de sucesso e alto impacto corporativo (considerando que impacto é visto como o grau de influência positiva no negócio).

4.3 A Abordagem ABC para Linhas de Produto de Software

Essa abordagem utiliza o ABC de estudos de casos de negócio para a realidade das linhas de produto. Nessa abordagem (COHEN, 2003) utilizam-se três variáveis para avaliar a viabilidade da implantação do modelo:

- Aplicações: quantos sistemas a organização planeja entregar (produzir) usando os ativos da linha de produto durante o período do caso de negócio estudado; quanto esses produtos irão variar ao longo do tempo?
- Benefícios: as projeções de economia, ou outros retornos, provenientes da utilização dos ativos da linha de produto; Como o investimento contínuo no desenvolvimento dos ativos irá ajudar essas projeções?
- Custos: os custos de reuso, ou seja, gastos que a organização vai enfrentar ao desenvolver utilizando os ativos da linha de produto.

Em Cohen (2003) este modelo é utilizado para guiar a implantação de uma LPS, de forma incremental, no NRO (Escritório de Reconhecimento de Territórios) das forças armadas do EUA. Neste caso, são usados: o custo do reuso (COR), o grau de reuso (DOR) e a taxa de renovação.

Para o desenvolvimento do caso de negócio aplicando o modelo ABC, a empresa deve antes de tudo, planejar quantas iterações planeja realizar, ou seja, em quantas etapas será estabelecida a LPS. Para cada iteração, devem-se planejar quais aplicações serão desenvolvidas (mesmo que parcialmente), qual o custo do reuso (COR) e o grau do reuso (DOR) planejados e qual o retorno esperado ao fim de cada iteração.

Quanto aos benefícios, a organização deve estabelecer metas claras quanto aos objetivos de redução de custo, aumento de lucratividade, diminuição de *time-to-market* ou qualquer outro objetivo plausível e perceptível que espere ter com a iniciativa. Deve-se considerar que esses benefícios estão diretamente ligados à quantidade de produtos desenvolvidos e à maturidade dos processos da linha conforme a evolução das iterações.

Finalmente quanto aos custos, atividade que exige o maior esforço, a empresa deve utilizar dados, métricas e projeções para ilustrar os benefícios financeiros da adoção do modelo e determinar sua viabilidade ou não. Os principais elementos são: custos históricos (do paradigma anterior), custos de desenvolvimento dos ativos e o custo incremental e a previsão de economia de cada iteração.

Com estas projeções de custos e economia, o modelo se propõe a tornar mais clara a visibilidade para determinar se a implantação de uma linha de produto de software para um domínio é viável ou não. Além disso, serve como um instrumento para apoiar a negociação gerencial para aprovação e captação de fundos para o projeto.

5 COMENTÁRIOS FINAIS

O reuso de software é uma ambição da comunidade de engenharia de software desde o final dos anos 60 (BOSCH, 2002) quando se instalou o termo “crise de software”, na Conferência da NATO de 1968 (KRUEGER, 1992). Desde então, inúmeros estudos foram desenvolvidos para tentar mensurar os benefícios do reuso e o esforço necessário para efetivamente reutilizar componentes de uma base. Em 2002, Barry Boehm abre o prefácio de (CLEMENTS; NORTHROP, 2002) afirmando que o reuso é uma das maiores oportunidades para melhorar a produtividade, a qualidade e o tempo de ciclo do software, mas complementa enumerando as dificuldades para chegar lá. Isto nos remete à questão sobre a qual nos propusemos a refletir: “Seriam as linhas de produto de software a forma pela qual o reuso será efetivamente sistematizado nas organizações?”.

A abordagem de linhas de produto de software ou engenharia de famílias de sistemas pode ser considerada jovem, se analisarmos que sua adoção na indústria é ainda muito recente. No estudo realizado em (AHMED; CAPRETZ, 2007) os autores ressaltam a dificuldade que tiveram para selecionar empresas que atendessem a um dos critérios da pesquisa: utilizar LPS há mais de 5 anos. Por este motivo, ainda não existem dados quantitativos suficientes para afirmar com precisão que LPS seja a forma pela qual o reuso se estabelecerá de forma sistematizada nas organizações. No entanto, é possível afirmar que, ao adotar a abordagem de LPS, todas as empresas relatam melhorias sensíveis no tempo de ciclo dos produtos, atribuídas ao alto percentual de desenvolvimento baseado nos ativos da base. Este alto

percentual de reuso implica em uma institucionalização efetiva dos processos de reuso na instituição.

Este artigo apresentou o tema linhas de produto de software, alguns dos benefícios esperados com sua adoção, os fatores que devem ser levados em consideração para ter sucesso e a forma de se analisar a sua viabilidade econômica. Finalmente, para encerrar nossas reflexões, fica a afirmação de (BOSCH, 2002): “Linha de produto de software representa uma das mais promissoras abordagens para aumentar o reuso de software, aumentar a qualidade e reduzir os custos de manutenção”.

REFERÊNCIAS

AHMED, F.; CAPRETZ, L.F. Managing the business of software product line: an empirical investigation of key business factors. **Information & Software Technology**, v49, n.2, p. 194-208, 2007.

BOSCH, J. **Design & use of Software Architectures: adopting and evolving a product-line approach**. Harlow: Addison-Wesley & ACM Press, 2000. 354 p.

BOSCH, J. Maturity and Evolution in Software Product Lines: Approaches, Artifacts and Organization. In: SOFTWARE PRODUCT LINES: SECOND INTERNATIONAL CONFERENCE (SPLC), 2., 2002, San Diego, CA. **Anais...** Berlin: Springer-Verlag, 2002, p. 257-271 (Lecture Notes in Computer Science 2379).

BROWNSWORD, L.; CLEMENTS, P. **A Case Study in Successful Product Line Development (CMU/SEI-1996-TR-016)**. Pittsburg: Software Engineering Institute, 1996, 84 p.

CLEMENTS, P.; KAZMAN, R.; KLEIN, M. **Evaluating Software Architectures: methods and case studies**. Boston, MA: Addison-Wesley Publishing Company, 2001, 368 p.

CLEMENTS, P.; NORTHROP, L. **Software Product Lines: practices and patterns**. Boston: Addison-Wesley Publishing Company, 2002, 563 p.

CLEMENTS, P.; MCGREGOR, J.; COHEN, S. **The Structured Intuitive Model for Product Line Economics (SIMPLE) (CMU/SEI-05-TR-003)**. Pittsburg: Software Engineering Institute, 2005, 66 p.

COHEN, S. **Product Line State of the Practice Report (CMU/SEI-2002-TN-017)**. Pittsburg: Software Engineering Institute, 2002, 62 p.

_____. **Predicting when Software Product Lines Pays (CMU/SEI-2003-TN-017)**. Pittsburg: Software Engineering Institute, 2003, 20 p.

EBERT, C.; SMOUTS, M. Tricks and Traps of Initiating a Product Line Concept in Existing Products. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 25., 2003, Portland, OR. **Anais...** Los Alamitos: IEEE Computer Society Press, 2003, p. 520-525.

ENGELSMA, R. et al. **ESAPS Experiments**. (Relatório de Resultados de Validação do projeto ESAPS - ESI-WP4-0105-01), Junho 2001. Disponível em: <<http://www.esi.es/esaps/publicResults.html>>. Acesso em 20 abr. 2006.

KRUEGER, C. Software Reuse. **ACM Computing Surveys**. V.24, n. 2, p. 131-183, junho de 1992.

PARNAS, D.L. Designing Software for Ease of Extension and Contractions. **IEEE Transactions on Software Engineering**, v.SE-5, n.1, p.128-138, março 1979.

WEISS, D.; GEPPERT, B. Goal-Oriented Assessment of Product-Line Domains. IEEE Ninth International Software Metrics Symposium 2003, pp. 180-189. IEEE Computer Soc. Press. **Anais ...** Los Alamitos, EUA, 2003.

WIJNSTRA, J. G. Critical Factors for a Successful Platform-Based Product Family Approach. In: SOFTWARE PRODUCT LINES: SECOND INTERNATIONAL CONFERENCE (SPLC), 2., 2002, San Diego, CA. **Anais...** Berlin, Heidelberg: Springer-Verlag, 2002, p.68-89. (Lecture Notes in Computer Science 2379).