

## Definição de Processos de Aquisição de Software com Uso de Abordagem baseada em Reutilização

Elaine Duarte Nunes<sup>1</sup>, Ana Regina Rocha<sup>1</sup>, Gleison Santos<sup>2</sup>

<sup>1</sup> COPPE/UFRJ – Universidade Federal do Rio de Janeiro  
Caixa Postal 68511 – CEP 21945-970 – Rio de Janeiro, Brasil

<sup>2</sup> Programa de Pós-Graduação em Informática – Universidade Federal do Estado do Rio de Janeiro (UNIRIO) - Av. Pasteur, 458, Urca, CEP 22290-240 – Rio de Janeiro, RJ

{elainenunes, darocha}@cos.ufrj.br, gleison.santos@uniriotec.br

**Resumo.** *As organizações precisam ser capazes de evoluir em qualidade e produtividade, reduzindo custos e ganhando competitividade. Muitas vezes as organizações são obrigadas a adotarem uma estratégia baseada na terceirização de parte de suas atividades de desenvolvimento. No entanto, a falta de processos adequados pode ser uma das causas para o insucesso de projetos que envolvam aquisição de software. Este artigo apresenta uma abordagem para definição de processos de aquisição de software baseado no uso de uma linha de processo de software (LPS) específica para aquisição. Com o uso desta abordagem, espera-se facilitar a definição de processos e a reutilização do conhecimento modelado na linha de processos, além de minimizar os problemas gerados quando se adquire software e de apoiar as organizações a executarem a aquisição de software de forma mais adequada.*

**Abstract.** *The organizations must develop both in terms of quality as well as productivity, reducing costs and becoming competitive to enable their survival in the market. One of the processes used by enterprises to gain competitiveness is by outsourcing their activities. The lack of adequate processes may cause the failure of software projects involving software acquisition. This paper aims at presenting an approach to the definition of acquisition processes, taking the different contexts into consideration, through the definition of a Software process Line (SPL). We expect that this approach will facilitate processes' definition, reuse of the knowledge model on the SPL, and minimize problems related to the software acquisition as well as support to execute software acquisition more efficiently.*

### 1. Introdução

Apesar da existência de diversos modelos de desenvolvimento de software, projetos ainda terminam com os prazos e custos além dos estimados, apresentam um percentual grande de falhas e esgotam prazo e custo sem produzir versões que possam ser utilizadas (HOFMANN *et al.*, 2007; JALIL e HANIF, 2009). Quando o desenvolvimento é terceirizado o problema se amplia, pois estão envolvidas múltiplas empresas, equipes e locais, dificultando a comunicação, levantamento e controle de mudança de requisitos e o gerenciamento do projeto. A responsabilidade do insucesso dos projetos é tanto do fornecedor quanto do cliente. Algumas ações podem ser executadas para evitar problemas, se o adquirente: (i) tiver um processo definido para a aquisição (KWAN e

LEUNG, 2004; TSUJI *et al.*, 2007; SHENG *et al.*, 2008a; CUI e XU, 2009), (ii) gerir o projeto contratado (JAMIESON *et al.*, 2005; JALIL e HANIF, 2009) e (iii) manter uma boa integração entre os envolvidos (KWAN e LEUNG, 2004; HUEN, 2007). Projetos de software adquiridos podem fracassar por motivos, como, por exemplo, falha na monitoração do fornecedor, não entendimento das responsabilidades das partes envolvidas, falta de integração entre os processos de aquisição e de desenvolvimento etc.

Nem sempre existe um processo de aquisição eficiente nas empresas que adquirem software ou não são abrangentes a todos os cenários possíveis. Quando a empresa terceiriza somente algumas atividades, o processo de desenvolvimento instanciado deveria apoiar esta aquisição. Infelizmente, nem sempre isto acontece, e a falta de processos adequados às necessidades da organização para a aquisição de software pode afetar o insucesso de alguns projetos adquiridos (KWAN e LEUNG, 2004; TSUJI *et al.*, 2007; SHENG *et al.*, 2008a; CUI e XU, 2009; SOFTEX, 2011a).

Para que um processo de aquisição possa ser utilizado com mais flexibilidade em uma organização é necessário que seja definido considerando as possibilidades de variação em função das diferentes características de projetos e das organizações adquirentes. A definição do processo baseado em reutilização de componentes de processo pode atender a este objetivo. Em função disto, uma abordagem para definição de processos baseada em reutilização de componentes de processo que atenda a diferentes cenários existentes nas organizações e facilite o uso das lições aprendidas pode apoiar empresas a gerir a aquisição de software e o projeto de software com aquisição de forma mais integrada, eficiente e adaptada à situação específica.

Nesse contexto, este artigo apresenta a definição de processos de Aquisição de Software utilizando técnicas de reutilização. Para atingir este objetivo foram definidos componentes, linhas e características de processos para esse domínio visando viabilizar a reutilização de componentes de processos na definição e geração de processos de Aquisição. O processo construído a partir das definições de componentes de processo está aderente a normas e modelos de referência para melhoria de processo de software, integrando a gestão do processo de desenvolvimento do software com aquisição e maximizando a reutilização de conhecimento relacionado à definição de processos.

Além desta introdução, este artigo está organizado em 5 seções. A Seção 2 apresenta uma revisão da literatura sobre aquisição. A Seção 3 descreve os objetivos de um estudo baseado em revisão sistemática elaborado e trabalhos similares identificados. A Seção 4 apresenta a abordagem para a definição da linha de processos de aquisição de software (LPAS). Por fim, as considerações finais são apresentadas na Seção 5.

## **2. Aquisição de Software**

As organizações precisam ser capazes de evoluir em qualidade e produtividade, reduzindo custos e ganhando competitividade para garantir a sobrevivência. Uma das abordagens adotadas pelas empresas para ganhar competitividade é a terceirização, liberando assim os gestores para a estratégia do negócio principal (JIANG *et al.*, 2010). A terceirização do desenvolvimento de software acontece quando uma empresa (cliente) contrata externamente todas ou parte das atividades de desenvolvimento do software em outra empresa (fornecedor) com níveis de acordo de remuneração (KHAN *et al.*, 2008).

Os acordos das partes envolvidas têm evoluído para operações complexas abrangendo vários sistemas e processos, representando uma grande transferência de ati-

vos, funções e pessoas. Os diversos cenários existentes neste contexto influenciam a definição e execução de processos de aquisição de software devido a características diferentes de cada organização e projeto.

A definição do processo de Aquisição minimiza riscos que podem comprometer os resultados esperados, como o não cumprimento de prazos, a falta de qualidade no produto adquirido, a falta de compatibilidade do produto adquirido com a arquitetura tecnológica definida, as dificuldades de integração e os problemas de suporte (SOFTEX, 2011a). Alguns benefícios com terceirização de software como redução de custo, ampliação da capacidade de desenvolvimento e flexibilidade no crescimento da equipe podem não se concretizar. Isto pode ocorrer em função da falta de clareza e mudança de requisitos pelo cliente, problemas de comunicação, restrições no contrato, falta de conhecimento no domínio do negócio e falta de disponibilidade em efetuar mudanças por parte do fornecedor (HUEN, 2007).

Devido à mudança nos ambientes empresariais em função da terceirização, a geração de valor já não está dentro dos limites de uma única empresa. Os acordos entre as partes envolvidas têm evoluído para operações complexas abrangendo vários sistemas e processos. Segundo GOTTSCHALK e KARLSEN (2005), a gestão do conhecimento deve transcender os limites organizacionais. Existem mudanças em diversas gerências de desenvolvimento de software e nos diversos papéis dos membros da equipe de projetos de Tecnologia da Informação que podem ser do adquirente ou do fornecedor. A Gerência de Conhecimento pode ser útil para coletar dados sobre as atividades e processos de aquisição tais como pedidos de propostas, avaliação e seleção de fornecedores, previsão dos recursos necessários, identificação dos riscos e medição de qualidade utilizada.

Em um processo de aquisição de software, todos os requisitos necessários precisam estar claramente definidos e as condições envolvidas na contratação acordadas. É preciso entender as necessidades de cada organização e formalizar em contrato os indicadores que representam a qualidade esperada, forma de aceitação, gestão de mudanças, artefatos esperados, prazo e custo. Desta forma, o processo de aquisição precisa ser formalizado desde a análise da necessidade da terceirização até a implantação e aceite do produto adquirido (HOFMANN *et al.*, 2007), evitando os diversos riscos entre as partes envolvidas e a ocorrência de conflitos na relação entre fornecedores e adquirentes de software (GOPAL *et al.*, 2003; MA *et al.*, 2007; TSUJI *et al.*, 2007; CUI e XU, 2009). Informações, tais como, forma de pagamento, prazo, medidas, custo e outras, precisam ser definidas, documentadas e acompanhadas pelo adquirente ao longo do projeto para mitigar os riscos, inclusive de custos, detectar variações em relação às *baselines* estabelecidas, notificar os interessados e ajudar o fornecedor na correção (PUTNAM e MYER, 2003).

Na aquisição de software é necessário medir e acompanhar os acordos estabelecidos. É importante acompanhar o custo de contratação dentro de uma organização com o objetivo de obter uma base de dados de métricas de projetos concluídos para tomada de decisão em projetos futuros de terceirização (HADDAD e RIBIERE, 2007). Neste contexto, se faz necessário medir desde os primeiros projetos adquiridos para ter uma base de dados consistente que possa apoiar as necessidades de informação do negócio, da organização e dos projetos e, assim, poder efetuar a melhoria no processo de aquisição. As medidas podem ser coletadas pelo adquirente ou pelo fornecedor. O contrato deve, portanto, estabelecer quais medidas devem ser coletadas e disponibilizadas pelo fornecedor (SOFTEX, 2011a).

No âmbito governamental as negociações e as aquisições de software são tratadas de forma distinta em relação a empresas privadas. No entanto, existem boas práticas de mercado nas empresas privadas que podem ser adotadas, tais como (REIFER, 2004): terceirizar apenas quando for bom para a empresa; não terceirizar as competências essenciais; estabelecer condições que gere um resultado positivo para ambas as partes (adquirente e fornecedor); nutrir o relacionamento com fornecedores; medir o desempenho o mais quantitativamente possível; fazer o desempenho ser financeiramente vantajoso e tratar terceirização como uma oportunidade de transferência de tecnologia. No Brasil, a Lei Nº 8.666 de 21 de junho de 1993 estabelece normas gerais sobre licitações e contratos administrativos pertinentes a obras, serviços, inclusive de publicidade, compras, alienações e locações no âmbito dos Poderes da União, dos Estados, do Distrito Federal e dos Municípios.

A aquisição de software e o desenvolvimento de um projeto com aquisição são objetos de estudos nos modelos e normas que definem boas práticas para o processo de aquisição de software, como os descritos no MR-MPS (Modelo de Referência para Melhoria de Processo do Software Brasileiro) (SOFTEX, 2011b), no CMMI-DEV (SEI, 2010) e nas normas ISO/IEC 12207 (ISO/IEC, 2008) e IEEE STD 1062:1998 (IEEE STD 1062, 1998). As normas ISO/IEC 12207 (ISO/IEC, 2008) e IEEE STD 1062:1998 (IEEE STD 1062, 1998) visam orientar relações técnicas e comerciais entre as organizações adquirentes e seus fornecedores. A primeira trata do processo de aquisição, com as etapas de preparação, comunicação, seleção de fornecedor, monitoração, aceitação do cliente e encerramento. A segunda apresenta 5 fases (Planejamento, Contratação, Implementação do Produto, Aceitação do Produto e Acompanhamento) com 9 passos. Cada passo possui entradas e saídas e define *checklists* para ajudar as organizações a estabelecerem seus próprios processos de aquisição de software.

O MR-MPS [SOFTEX, 2011b] define níveis de maturidade que permitem, dependendo do nível, a previsibilidade do desempenho futuro ao executar um ou mais processos. Como muitas organizações subcontratam etapas do processo de desenvolvimento ou componentes específicos do produto, essa atividade também deve ser controlada com o mesmo rigor que as questões internas, mas respeitando o modo com que outras organizações trabalham. Os requisitos úteis para que esse controle seja feito de forma adequada é definido no processo Aquisição (AQU) no nível F do modelo. Da mesma forma, o CMMI-DEV [SEI, 2010] define no nível 2 de maturidade a área de processo Gerência de Acordo com Fornecedores, com requisitos similares ao do MR-MPS.

### **3. Trabalhos Relacionados**

Existem muitos cenários nas organizações e estes precisam ser retratados no processo de aquisição. Dessa forma, é importante identificar os diversos cenários e questões envolvidas para apoiar a definição de processos que sejam adaptáveis a diferentes situações que existem quando se adquire software.

Para identificar as características dos diversos cenários da aquisição de software, foi realizado um estudo baseado em revisão sistemática da literatura visando identificar abordagens que apoiem organizações: (i) no processo de aquisição; (ii) no gerenciamento de um projeto de software com parte das atividades do processo adquiridas por meio de fornecedores; e (iii) na elaboração e no acompanhamento do contrato relativo às aquisições. O uso do estudo baseado em revisão sistemática da literatura visou reduzir os problemas de uma revisão informal e, também, permitir a constante atualização com

novas publicações disponibilizadas ao longo do tempo (KITCHENHAM, 2008). Mais detalhes sobre o protocolo utilizado e os resultados obtidos podem ser vistos em (NUNES, 2011) e em (NUNES *et al.*, 2010), artigo cujo foco é descrição do estudo baseado em revisão sistemática da literatura.

O objetivo do estudo foi *analisar* relatos de experiência e publicações científicas por meio de um estudo baseado em revisão sistemática, *com o propósito de* identificar as características de aquisição e gerenciamento do projeto de aquisição quando se adquire software e os riscos, métricas, estratégias, necessidades no gerenciamento de projetos de software com aquisição de software sob encomenda (excluindo aquisição de produtos de software do tipo *Commercial-off-the-shelf-software/COTS* e *Modified-off-the-shelf-software/MOTS*), *com relação às* abordagens de apoio empregadas na aquisição e gerenciamento de projetos com aquisição, *do ponto de vista do* pesquisador e *no contexto* acadêmico e de empresas (privadas e públicas) que adquirem software.

Foram pesquisados artigos publicados entre janeiro de 2000 e fevereiro de 2011. No total, 35 foram os selecionados. Destes, 3 eram referentes a ferramentas (sendo 1 de Arquitetura de processo, contemplando modelagem e simulação), 9 tratavam de riscos, 13 de aquisição em geral, 2 citavam métodos ágeis, 1 contextualizava o uso do MPS.BR, 1 tratava de portfólio de projetos, 2 de seleção de fornecedores, 1 do uso de Análise de Pontos por Função, 2 falavam sobre uso de modelos de qualidade e 1 relacionava aquisição e auditoria. Como conclusão do estudo baseado em revisão sistemática da literatura, verificou-se que a falta de um processo adequado às necessidades da organização para a aquisição de software (KWAN e LEUNG, 2004; TSUJI *et al.*, 2007; SHENG *et al.*, 2008a; CUI e XU, 2009) é uma das principais causas dos problemas existentes quando se adquire software. Além disto, evidenciou-se a existência de diferentes práticas para adquirir software para organizações e países. No entanto, não foram encontrados relatos de abordagens que permitam definir um processo com semelhanças e variações entre processos.

O trabalho que mais se aproxima deste contexto, propõe uma arquitetura de processos que permite simulação (CHOI e SCACCHI, 2001). Os autores consideram que fizeram um primeiro esforço em investigar e prover resultados na aplicação do domínio Aquisição de Software em conceitos, técnicas e ferramentas de modelagem e simulação de processos utilizando linhas de processo (arquitetura de processo), usando HLA (*high-level architecture*) e RTI (*run-time infrastructure*) com linguagem PML. No entanto, não fica claro como é tratada a variabilidade e nem explicam outros conceitos fundamentais das linhas de processo. Os autores consideram processo de aquisição de software um bom desafio por envolver várias empresas e patrocinadores, ser um projeto com custo alto, duradouro e frequentemente contaminado por problemas de processos. Porém, a abordagem tem como foco a modelagem e simulação de arquitetura de processo de software, usando o domínio aquisição.

#### **4. Definição do Processo de Aquisição de Software para Reutilização**

A definição de um processo de Aquisição minimiza riscos que podem comprometer os resultados esperados, como o não cumprimento de prazos, a falta de qualidade no produto adquirido, a falta de compatibilidade do produto adquirido com a arquitetura tecnológica definida, as dificuldades de integração e os problemas de suporte (SOFTEX, 2011a). Alguns benefícios com terceirização de software como redução de custo, ampliação da capacidade de desenvolvimento e flexibilidade no crescimento da equipe podem

não se concretizar. Isto pode ocorrer em função da falta de clareza e mudança de requisitos pelo cliente, problemas de comunicação, restrições no contrato, falta de conhecimento no domínio do negócio e falta de disponibilidade em efetuar mudanças por parte do fornecedor (HUEN, 2007). Considerando que estes fatores e riscos variam entre as empresas e até entre os projetos de uma mesma empresa, através da abordagem descrita neste artigo é possível definir essas variações. Esta abordagem pode ser usada em uma empresa que possua projetos que tenham necessidades diferentes ou pode ser utilizada por empresas de consultoria que precisem implementar processos de aquisição em diferentes empresas. Acredita-se que esta abordagem para definição de processo a partir de componentes de processos reutilizáveis pode ser aplicada a qualquer domínio.

Neste contexto, a possibilidade de definir os requisitos necessários e transformar os diversos cenários de um processo em componentes de processos reutilizáveis é bastante útil para o domínio da aquisição. Assim o processo de aquisição de software pode contemplar os diferentes cenários e permitir que as lições aprendidas possam ser incorporadas. Esta abordagem possibilita definir o processo através da escolha de componentes de processo reutilizáveis previamente definidos e disponíveis.

Para explicitar, preservar, disseminar e permitir a reutilização de conhecimento sobre processos, uma das formas possíveis é a definição e disponibilização de elementos reutilizáveis de processos, além de guias para orientar a escolha desses itens em cada situação. Diversas abordagens têm sido descritas para viabilizar a reutilização de processos de software (REIS, 2002; BARRETO *et al.*, 2008; BARRETO, 2011; ARMBRUST *et al.*, 2009). A abordagem escolhida para definição e disponibilização de componentes reutilizáveis de processo de Aquisição foi o trabalho de BARRETO (2011) que adapta técnicas de reutilização do desenvolvimento de produtos de software para o contexto da definição de processos de software. Esta abordagem visa tornar mais fácil a definição dos processos de software através do uso de Linha de Processo de Software (LPS), evitando o retrabalho na definição do processo e também diminuindo a demanda por engenheiros de processo mais experientes para executar esta tarefa, sem diminuir a qualidade dos processos definidos. Assim, através de um conjunto de características, uma LPS é instanciada para a derivação de diferentes processos de software.

Nesta abordagem, segundo BARRETO *et al.* (2010), a LPS possui as seguintes características principais: (i) é uma arquitetura de processo de software e, portanto, é composta por elementos de baixo nível e as interações entre eles, (ii) é capaz de representar pontos comuns e variabilidades entre os processos de software, (iii) existe se, e somente se, é capaz de derivar diferentes processos (ou seja, se ele permite que algum tipo de variabilidade), caso contrário, considera-se uma arquitetura de processo ordinária e (iv) está associada a características do processo, que condicionam e orientam a seleção dos elementos do processo que será escolhido em uma derivação.

Espera-se, assim, que organizações que adquirem software possam definir seu processo de aquisição de software e que possam reutilizar o conhecimento modelado na linha de processos e adaptar facilmente a seu contexto, independente da experiência da empresa nesta área. O processo construído a partir das definições apresentadas está aderente a normas e modelos de referência para melhoria de processo de software e integrar a gestão do processo de desenvolvimento do software com aquisição e com o objetivo de maximizar a reutilização de conhecimento relacionado à definição de processos.

Dessa forma, para apoiar as organizações que adquirem software a definirem um processo de aquisição, uma Linha de Processos de Aquisição de Software (LPAS) foi

definida, baseando-se nas normas, modelos e guias de qualidade de software.

Neste trabalho foi considerado o domínio de Aquisição de Software e não contempla aquisição de produtos de software do tipo *Commercial-off-the-shelf-software* (COTS) e *Modified-off-the-shelf-software* (MOTS). Dessa forma, a abordagem apoia projetos com aquisição de software sob encomenda ou sob medida. Esta aquisição pode ser necessária em várias fases do projeto, como, por exemplo, na especificação de requisitos, na modelagem e análise do projeto, na codificação, para as atividades de integração e teste, para efetuar a homologação, ou mesmo a implantação. Acredita-se que o processo de Aquisição resultante possa ser implementado em organizações públicas ou privadas que adquirem software e em diferentes tipos de negócio, como indústria, área financeira e consultoria de desenvolvimento de software.

O estudo baseado em revisão da literatura conduzido foi importante para identificação das possibilidades de variação nos processos de aquisição e guiou a criação das características e componentes de processo utilizados pela LPAS.

#### 4.1 Procedimento para definição de processos para reutilização

Para facilitar o entendimento, é importante apresentar brevemente os principais conceitos relacionados à reutilização de processos que foram utilizados (BARRETO, 2011):

- *Característica de Processo*: Funcionalidade ou classificação que o processo deve possuir, por exemplo: apoio ao nível G do MR-MPS, paradigma Orientado a Objetos, entre outros. Restringe a utilização de componentes, definindo se um conjunto de componentes pode ou não ser utilizado.
- *Componente de Processo*: Unidade básica de composição de processos. Definições de processo são sempre realizadas por meio da composição de diferentes componentes. Um componente de processo, segundo o SPEM – *Software Process Engineering Metamodel* (SPEM, 2006), é um agrupamento de descrições de processo (elementos de processo) que é internamente consistente e pode ser reutilizado com outros componentes de processo para compor um processo completo. Um componente é considerado algo relevante para ser reutilizado em outras definições de processo e encapsula uma série de informações de processo, tais como: atividades, produtos de trabalho requeridos e produzidos, responsáveis pela execução, entre outros. Podem ser concretos (componentes totalmente definidos que não admitem variabilidade) ou abstratos (componentes parcialmente definidos, que podem ser realizados de diversas maneiras, através de diferentes componentes concretos).
- *Conector de Componentes*: É uma forma de conectar os componentes de processo. Um conector possui elementos origem e destino, e alguma regra de conexão.
- *Linha de Processos de Software (LPS)*: Representa uma espécie de fluxo de trabalho, definindo um “esqueleto” que o processo deve possuir, determinando os principais elementos e como estes se relacionam, sem necessariamente definir como será o detalhamento desses elementos principais. É capaz de modelar variabilidades, através de pontos de variação e da determinação de quais componentes são opcionais. Um ponto de variação em uma LPS é uma indicação de que uma determinada ação pode ser realizada de diferentes maneiras. Assim, componentes abstratos em uma LPS são pontos de variação, pois indicam que a ação descrita pode ser realizada de diferentes maneiras. A cada ponto de variação estão associadas variantes (componentes concretos que “implementam” um dado componente abstrato). Através da seleção de características de processo, pode-se selecionar quais variantes são as mais adequadas para cada ponto de variação. Linha de processos de software é

uma linha de produtos cujos produtos são processos de software.

A utilização desses conceitos no contexto da aquisição de software tende a ser bastante útil, uma vez que é possível modelar: (i) as partes principais dos processos de aquisição (componentes de processo); (ii) as principais questões que poderiam levar à geração de diferentes processos (características de processo); e (iii) estruturas padrão para os processos de aquisição, no caso a Linha de Processos de Aquisição de Software (LPAS). A LPAS inclui os pontos que sempre devem estar nos processos (componentes concretos), os pontos que podem sofrer variação (pontos de variação, através de componentes abstratos), os componentes que são opcionais, além do conhecimento sobre qual componente selecionar para cada ponto de variação, dependendo do contexto (ou seja, dependendo das características selecionadas, selecionar os componentes a elas relacionados).

Para definir a linha de processos de aquisição, foi utilizada a abordagem definida como *top-down* por BARRETO *et al.* (2010), que consiste na execução de 4 passos. Estes quatro passos são descritos a seguir.

O **primeiro passo** é definir as características de processo. Sendo características de processo um conjunto de regras que atua sobre os componentes. As características são uma forma de definir o escopo da linha de processos gerada, uma vez que é necessária a criação de componentes de processo que atendam às características definidas. As características vão sendo definidas ao longo da definição de componentes de processo e têm grande influência para a escolha dos componentes que farão parte do processo. Nesta etapa, as características são escolhidas e impõem restrições aos componentes que podem ser selecionados de forma a atender às características. Uma característica pode ser conflitante em relação à outra, ou seja, não podem nunca estar presentes simultaneamente em um mesmo processo. As características servem como filtro e restringem bastante o número de componentes possíveis, tornando a definição mais simples.

O **segundo passo** é definir os componentes para o maior número de cenários de um processo. Um componente pode ser tão pequeno, como uma atividade ou abrangente como um processo inteiro e deverá sempre ser composto por, no mínimo, uma atividade, pois uma unidade de reutilização inferior a esta dificultaria a composição de processos. Deve ser definido como componente de processo o que é relevante para reutilização e ter granularidade adequada. Um componente pode ser (i) concreto, componente que não admite variabilidade e é executado como é definido ou (ii) abstrato, componente que admite variabilidade, ou seja, são configurados de várias formas e é necessária uma decisão sobre qual componente variante utilizar. Componentes podem também ser classificados como obrigatórios e opcionais. Logo um componente pode ser: opcional variante, opcional invariante, obrigatório variante ou obrigatório invariante. Para cada parte dos processos que possa ser realizada de forma diferente, devem ser definidos componentes abstratos. Para cada diferente maneira de realizar um componente abstrato, devem ser definidos componentes concretos. Componentes concretos também devem ser definidos para representar partes do processo que não sofrem variação. Devem ser definidos componentes suficientes para atender a todas as características definidas.

O **terceiro passo** é definir a(s) linha(s) de processo(s), as opcionalidades e integração entre os componentes de processo. Tendo os componentes definidos, é necessário selecioná-los para integrar a LPS, relacioná-los e ordená-los. Também devem ser definidos que elementos são opcionais. A arquitetura de processo com componentes abstratos ou elementos opcionais é chamada linha de processo de software.

O **quarto passo** é avaliar as características, os componentes e linhas de processos. O objetivo é avaliar se a LPS está definida de forma a satisfazer os requisitos estabelecidos. Esta avaliação pode ser feita através de revisão por pares. Esta revisão deve ser feita por especialistas que utilizarão sua experiência e conhecimento para verificar os componentes utilizados, o conjunto de características do processo, o sequenciamento dos componentes, as variantes disponíveis, entre outros. O objetivo é analisar e obter consenso sobre a melhor solução possível.

Estes passos não são necessariamente sequenciais, pois alguns passos podem ser conduzidos em paralelo. A execução destes passos possibilita a captura do conhecimento dos especialistas e a avaliação da qualidade do conhecimento.

## 4.2 Definição de Processos de Aquisição de Software para Reutilização

### Primeiro passo: definir as características de processo

Como descrito anteriormente, o primeiro passo a ser executado é definir as características de processo. Cada cenário que possua várias possibilidades de realização deve ser descrito como requisito e cada possibilidade será uma característica do processo deste requisito. As características são usadas para delimitar o escopo da linha de processos e delimitar os requisitos relacionados à aquisição a serem considerados. As características identificadas podem ser vistas na Tabela 1.

**Tabela 1** - Lista de características de processo de Aquisição de Software

Tipo de Característica	Descrição
Compatibilidade com Modelos de Maturidade	CMMI
	MR-MPS
Tipo de Desenvolvimento	Projeto com Aquisição de Software
	Projeto sem Aquisição de Software
Seleção de Fornecedor	Escolha de fornecedores a partir de lista de fornecedores previamente homologados
	Escolha de fornecedores a partir das especificações técnicas e comerciais
	Sem avaliação prévia de fornecedores (podendo participar qualquer um que tenha interesse).
Condições de pagamento	Contratação por valor de hora trabalhada
	Contratação por preço fixo
	Contratação por preço fixo com adicionais por hora trabalhada
Aplicação de Penalização	Aplicação de penalidade no contrato
Aplicação de Bonificação	Aplicação de bônus no contrato
Forma de pagamento de parcelas	Procedimento padrão da empresa para pagamento das faturas
	Procedimento diferente para pagamento das faturas
Pagamento de despesas	Pagamentos de despesas não incluídas no valor do contrato
Forma de estimar tamanho e prazo do projeto	Estimativa utilizando técnica de ponto de função
	Estimativa utilizando técnica de ponto de caso de uso
	Estimativa utilizando base histórica
Aquisição de Levantamento e Definição de Requisitos	Aquisição de Requisitos
Aquisição de Codificação	Aquisição de Codificação
Aquisição de Teste	Aquisição de Testes
Aquisição de Homologação	Aquisição de Homologação
Aquisição de Implantação	Aquisição de Implantação
Permissão para alterações em Contrato	Contrato permite alterações

Essas características foram definidas a partir dos requisitos considerados relevantes para a definição dos componentes. Dessa forma, a LPAS deve ser capaz de atender esses requisitos e as características definidas a partir dos requisitos restringem a utilização de componentes, definindo se um conjunto de componentes pode ou não ser utilizado. Foram identificados os seguintes requisitos: (i) a LPAS deve ser capaz de ser aderente a Modelos de Maturidade como CMMI-DEV e MR-MPS; (ii) a LPAS deve ser

capaz de atender a organizações que adquirem um software completo ou apenas algumas etapas do desenvolvimento do software; (iii) a LPAS deve possibilitar que fornecedores homologados possam participar da concorrência e também qualquer fornecedor de mercado que atenda as especificações estabelecidas no Pacote de Solicitação; (iv) a LPAS deve ser capaz de tratar diversas condições de pagamento; (v) a LPAS deve permitir aplicação de penalidades e bônus; (vi) a LPAS deve ser capaz de tratar diversas formas de pagamento; (vii) a LPAS pode tratar pagamento de despesas; (viii) a LPAS deve ser capaz de permitir estimativa de tamanho e prazo do produto feita pelo adquirente para análise das propostas; (ix) a LPAS pode permitir diferentes escopos de aquisição; (x) a LPAS pode possibilitar alterações nos contratos.

### **Segundo passo: definir os componentes dos processos**

Uma vez delimitado o escopo da LPAS, iniciou-se a definição dos componentes de processo a serem usados para compor processos de aquisição. Para facilitar a definição de linha de processos foi criado um formulário para preenchimento das informações necessárias. Componentes de processo que possuem componentes em sua arquitetura, também precisam ser definidos com o formulário para que se possa explicitar a conexão entre os componentes. A descrição do componente apresentado na Tabela 2 contém os dados presentes no formulário utilizado. Foram definidos ao todo 43 componentes de processo para atender as características de processo identificadas e as necessidades dos processos de aquisição de software. Devido a limitações de espaço, não é possível descrever neste artigo todos os componentes definidos, a listagem completa pode ser vista em (NUNES, 2011). No entanto, um exemplo é fornecido a seguir.

Para o planejamento da aquisição é necessário “Listar Requisitos do software”, “Planejar o Projeto Aquisição” e “Levantar Requisitos do Contrato”. Para cada uma dessas necessidades foram criados componentes de processo. Por exemplo, “Levantar Requisitos do Contrato” envolve definir as condições de pagamento, multa, bônus e despesas do contrato, além de estimar tamanho e prazo. Considerando especificamente as condições de pagamento do contrato, conforme se pode perceber nas características de processo foram consideradas três diferentes maneiras para a execução deste componente. Assim, isso levou a um componente abstrato e três componentes concretos que realizam o componente abstrato. A Tabela 2 ilustra o componente concreto “Definir condições de pagamento para contratação por hora trabalhada”.

Informações como ferramenta, artefatos, responsáveis e participantes são definições inerentes ao componente concreto. Há diferenças no tipo de conhecimento de apoio disponibilizado para cada componente: a organização terá *templates* de artefatos específicos, além de outras informações. Não fez parte deste trabalho provê-los.

### **Terceiro passo: definir as informações de linha(s) de processo(s)**

Uma vez definidos os componentes, pode-se definir a(s) linha(s) de processo(s). Para explicitar a arquitetura da LPAS e criar o fluxo de componentes, foram utilizados os conectores de componentes de processo: Fim-início, Início-Início, Fim-Fim e Início-Fim.

Com a linha de processos definida, é possível derivar diferentes processos de aquisição, dependendo das características escolhidas. Ou seja, a organização adquirente seleciona as características que correspondem ao cenário do projeto, por exemplo: a estimativa técnica deve ser feita por ponto de função, a contratação é por preço fixo, o não cumprimento de prazos resulta em multa e o processo de aquisição precisa ser aderente ao MR-MPS. A partir desta seleção, o conjunto de variantes que pode ser escolhido é

restrito e componentes associados a essas características podem ser selecionados de forma que a atividade de definição de processos é bastante simplificada.

**Tabela 2** – Definição de Componentes para definir condições de pagamento de contrato

<b>Identificador:</b>	<b>COP.AQU.PLA.CON.0010</b>
Nome:	Definir condições de pagamento para contratação por hora trabalhada
Descrição:	Definir condições de pagamento para contratação por hora trabalhada, avaliando os critérios que serão utilizados para efetivação do pagamento.
Tipo de Componente:	Concreto
Crítérios de Entrada:	Ter-se a necessidade da aquisição e o Plano de Aquisição (lista de requisitos de software e formalização da análise e estratégia da aquisição)
Crítérios de Saída:	Ter-se as condições de pagamento de contrato definidas.
Responsável:	Coordenador de Aquisição
Participantes:	Gerente de Projeto
Artefatos Requeridos:	Plano de Aquisição (lista de requisitos de software e formalização da análise e estratégia da aquisição) e <i>Template</i> de planilha de horas trabalhadas com tarefas executadas e critérios para aprovação.
Artefatos Produzidos:	Plano de Aquisição (lista de requisitos de software e de contrato: condições de pagamento), definição de valor/hora por perfil profissional; Planilha de horas trabalhadas com tarefas executadas e critérios para aprovação
Ferramentas:	MS Excel e Word
Definido por:	Instituição Implementadora COPPE/UFRJ
Características Atendidas:	CMMI, MR-MPS, Projeto com Aquisição, Contratação por hora trabalhada
Características Conflitantes:	Projeto sem Aquisição, Contratação por preço fixo e Contratação por preço fixo com adicionais por hora trabalhada
É Variante de:	COP.AQU.PLA.ABS.0009

A Tabela 3 apresenta o resumo da LPAS definida. Devido a limitações de espaço, a estrutura dos componentes não é detalhada. Os componentes abstratos são exibidos em itálico, os opcionais estão sublinhados.

**Tabela 3** – Linha de Processos de Aquisição de Software (LPAS)

<b>Identificador:</b>	<b>COP.AQU.ABS.0001</b>				
Nome:	Linha de Processo de Aquisição de Software				
Descrição:	Esta linha de processo abrange todos os componentes necessários para aquisição de software desde o início da identificação da necessidade do cliente até o encerramento com a aceitação do produto ou serviço.				
Tipo de Componente:	Abstrato				
Definido por:	Instituição Implementadora COPPE/UFRJ				
Características Atendidas:	CMMI, MR-MPS, Projeto com Aquisição				
Características Conflitantes:	Projeto sem Aquisição				
<b>Arquitetura e Conexão</b>					
Identificador	Nome	Conexões Originadas	Início	Fim	Opcional
COP.AQU.PLA.CON.0001	Listar Requisitos do software	COP.AQU.PLA.CON.0002 (Fim-Início)	X		
COP.AQU.PLA.CON.0002	Planejar o Projeto Aquisição	COP.AQU.PLA.ABS.0019 (Fim-Início)			
<i>COP.AQU.PLA.ABS.0019</i>	<i>Definir condições de pagamento ao fornecedor</i>	COP.AQU.PLA.ABS.0014 (Início-Início)			
<i>COP.AQU.PLA.ABS.0014</i>	<i>Estimar tamanho do produto e prazo para desenvolvimento</i>	COP.AQU.PLA.CON.0018 (Fim-Início)			
COP.AQU.PLA.CON.0018	Definir indicadores de contrato	COP.AQU.PRE.CON.0002 (Fim-Início)			
COP.AQU.PRE.CON.0002	Definir critérios de Avaliação de Fornecedor e Aceitação do Produto	COP.AQU.PRE.ABS.0003 (Fim-Início)			
<i>COP.AQU.PRE.ABS.0003</i>	<i>Avaliar Potenciais fornecedores</i>	COP.AQU.PRE.CON.0010 (Fim-Início)			

COP.AQU.PRE.CON.0010	Estabelecer, Distribuir e Manter o Pacote de Solicitação	COP.AQU.PRE.CON.0007 (Fim-Início)			
COP.AQU.PRE.CON.0007	Selecionar e Negociar com Fornecedor	COP.AQU.CONT.CON.0008 (Fim-Início)			
<i>COP.AQU.CONT.ABS.0008</i>	<i>Identificar Processos Críticos do Fornecedor</i>	COP.AQU.CONT.CON.0006 (Fim-Início)			
COP.AQU.CONT.CON.0006	Elaborar, avaliar e formalizar contrato	COP.AQU.MON.CON.0001 (Fim-Início)			
COP.AQU.MON.CON.0001	Planejar a monitoração da aquisição do produto	COP.AQU.MON.CON.0002 (Fim-Início)			
COP.AQU.MON.CON.0002	Monitorar a aquisição do produto	COP.AQU.MON.CON.0003 COP.AQU.MON.CON.0004 COP.AQU.ENC.CON.0001 Todos (Fim-Início)			
COP.AQU.MON.CON.0003	Obter acordo quanto às alterações necessárias	COP.AQU.ENC.CON.0001 (Fim-Início)			X
COP.AQU.MON.CON.0004	Aprovar e acompanhar pagamento ao fornecedor	COP.AQU.ENC.CON.0001 (Fim-Início)			
COP.AQU.ENC.CON.0001	Avaliar e aceitar o produto	COP.AQU.ENC.CON.0004 (Fim-Início)			
COP.AQU.ENC.CON.0004	Incorporar o Produto Adquirido ao Projeto	COP.AQU.ENC.CON.0005 (Fim-Início)			X
COP.AQU.ENC.CON.0005	Realizar Treinamento	COP.AQU.ENC.CON.0006 (Fim-Início)			
COP.AQU.ENC.CON.0006	Acompanhar utilização do software	COP.AQU.ENC.CON.0007 (Fim-Início)			
COP.AQU.ENC.CON.0007	Avaliar o fornecedor	COP.AQU.ENC.CON.0008 (Fim-Início)			
COP.AQU.ENC.CON.0008	Encerrar a aquisição			X	

Convém destacar que alguns dos componentes listados na Tabela 3 são decompostos em outros, conforme apresentam a Tabela 4.

**Tabela 4 – Decomposição dos componentes da LPAS**

<i>COP.AQU.PLA.ABS.0019</i>	<i>COP.AQU.PLA.ABS.0009 Definir condições de pagamento de contrato</i>
	<i>COP.AQU.PLA.ABS.0004 Definir forma de pagamento de parcelas de contrato</i>
	<i>COP.AQU.PLA.CON.0007 Aplicar penalidade ao contrato</i>
	<i>COP.AQU.PLA.CON.0008 Aplicar bônus ao contrato</i>
	<i>COP.AQU.PLA.CON.0013 Permitir pagamento de despesas não incluídas no valor do contrato</i>
<i>COP.AQU.PLA.ABS.0014</i>	COP.AQU.PLA.CON.0015 Estimar tamanho do produto a ser adquirido utilizando a técnica Ponto de Função e prazo para o desenvolvimento.
	COP.AQU.PLA.CON.0016 Estimar tamanho do produto a ser adquirido utilizando a técnica Caso de Uso e prazo para o desenvolvimento.
	COP.AQU.PLA.CON.0017 Estimar tamanho e prazo do produto a ser adquirido utilizando base histórica
COP.AQU.PRE.ABS.0003	COP.AQU.PRE.CON.0004 Avaliar os fornecedores homologados pela empresa
	COP.AQU.PRE.CON.0005 Avaliar os fornecedores de mercado
	COP.AQU.PRE.CON.0015 Liberar a participação de qualquer fornecedor interessado
<i>COP.AQU.CONT.ABS.0008</i>	COP.AQU.CONT.CON.0001 Identificar Processos Críticos do Fornecedor para qualquer aquisição
	COP.AQU.CONT.CON.0002 Identificar Processos Críticos do Fornecedor para aquisição de Requisitos
	COP.AQU.CONT.CON.0003 Identificar Processos Críticos do Fornecedor para aquisição de Codificação
	COP.AQU.CONT.CON.0007 Identificar Processos Críticos do Fornecedor para aquisição de Testes
	COP.AQU.CONT.CON.0004 Identificar Processos Críticos do Fornecedor p/ aquisição de Homologação
COP.AQU.CONT.CON.0005 Identificar Processos Críticos do Fornecedor para aquisição de Implantação	
<i>COP.AQU.PLA.ABS.0009</i>	COP.AQU.PLA.CON.0010 Definir condições de pagamento para contratação por hora trabalhada
	COP.AQU.PLA.CON.0011 Definir condições de pagamento para contratação por preço fixo
	COP.AQU.PLA.CON.0012 Definir condições de pagamento para contratação por preço fixo com possibilidade de pagamentos adicionais por hora trabalhada
<i>COP.AQU.PLA.ABS.0004</i>	COP.AQU.PLA.CON.0005 Estabelecer o padrão da organização para pagamento de parcelas de contrato
	COP.AQU.PLA.CON.0006 Definir um procedimento para pagamento de parcelas de contrato

Ao utilizar a LPAS descrita, há um limite teórico de definição de 55.296 proces-

sos diferentes, dependendo das necessidades e requisitos das empresas (o número real é menor pois não leva em consideração os componentes conflitantes entre si). Isto porque há 18 componentes obrigatórios, 10 opcionais, 3 obrigatórios com 3 variantes e 1 obrigatório com 2 variantes ( $2^{10} \times 3^3 \times 2^1 = 55.296$ ).

#### **Quarto passo: avaliar as características, os componentes e linhas de processos**

Este passo é importante para garantir a qualidade e adequação da linha de processo de software definida. A avaliação para este trabalho foi feita com dois enfoques: (i) quanto à forma das definições dos componentes; (ii) quanto ao conteúdo relativo a aquisição.

Para a revisão da forma é necessário especialistas com experiência em definição de processos e em reutilização de componentes. Esta revisão visou verificar se os componentes utilizados, o conjunto de características do processo, o sequenciamento lógico dos componentes, as variantes disponíveis e as características atendidas e conflitantes foram definidas de acordo com os conceitos de reutilização de processos. Para a revisão de conteúdo é necessário especialistas com conhecimento em definição de processos e em processo de aquisição de software de empresas que desenvolvem software e já foram avaliadas no MR-MPS. A avaliação foi realizada através da técnica de Revisão por Pares. Foram selecionados três especialistas, um para revisão quanto à forma (Revisor A) e dois para revisão de conteúdo (Revisores B e C). O pequeno número de revisores deve-se ao rigoroso critério definido para a identificação de especialistas no assunto.

O Revisor A (especialista em definição de processos para reutilização e com experiência em aquisição de software) sugeriu considerações de ajustes nos conectores definidos em algumas LPAS, sugeriu alteração no formulário quanto ao formato para definição de arquitetura facilitando a compreensão e tendo menos redundância de informações e, também, diversos ajustes nas definições quanto aos conceitos de reutilização. Este revisor também contribuiu com observações nas definições dos componentes de aquisição propriamente dita, como, por exemplo, sugeriu que os requisitos fossem explicados relacionando com a definição dos componentes de processo e as situações onde serão utilizados. Sugeriu também um levantamento mais amplo de variabilidades em relação a empresas públicas, além de sugestões de contribuições na melhoria de texto e definições. Os Revisores B e C fizeram observações quanto às definições dos componentes, sua aplicabilidade e abrangência do processo, assim como as definições relacionadas, melhoria no texto e também sugeriram alterações em alguns artefatos requeridos e produzidos. Como um dos requisitos é que a LPAS possa permitir diferentes escopos de aquisição, o Revisor B identificou a inexistência da possibilidade do adquirente realizar somente a aquisição de teste. As revisões foram acatadas e o conteúdo do trabalho já apresenta a versão final após as melhorias identificadas pelos revisores.

### **5. Considerações Finais**

Este trabalho fornece apoio às organizações que adquirem software a definirem seus processos de aquisição, maximizando a reutilização de conhecimento relacionado à definição de processos e permitindo que os processos atendam aos diversos cenários cada vez mais complexos. Os requisitos, as características e os componentes de processo e a Linha de Processos de Aquisição de Software (LPAS) foram definidos com base nas principais práticas e atividades de modelos e normas de qualidade de software e, também, como resultado do estudo baseado em revisão sistemática da literatura. Para definição da LPAS foi utilizado um conjunto de passos de acordo com a estratégia que adapta técnicas de reutilização do desenvolvimento de produtos de software para o con-

texto da definição de processos de software (BARRETO, 2007; BARRETO *et al.*, 2008; BARRETO *et al.*, 2010; NUNES *et al.*, 2010), usada no domínio de aquisição. Esta estratégia se mostrou adequada por facilitar a definição das diversas possibilidades de contexto e cenários que existem na aquisição de software.

Tendo processos para aquisição definidos previamente com lições aprendidas e melhores práticas, os riscos tendem a ser minimizados. A estratégia para definição da LPAS e seus componentes pode ser utilizada em diversos tipos de negócio como indústria, área financeira, consultoria de desenvolvimento de software e também em outros domínios de processo, além de possibilitar, sempre que necessário, a criação de novos componentes ou variantes, tornando, assim, a LPAS cada vez mais abrangente.

A estratégia utilizada permitiu que fosse possível retratar de forma clara as variações existentes para os diferentes cenários das organizações, possibilitando uma flexibilidade muito além do que a definição de um processo de aquisição de software formal seria capaz. Isso minimiza os vários problemas que existem no relacionamento entre adquirente e fornecedor e pode aumentar o número de projetos de sucesso em aquisição de software. Os componentes definidos também podem contribuir para a confecção de um contrato que possa minimizar problemas futuros entre as partes envolvidas.

Trabalhos futuros incluem: (i) ampliação da definição de componentes de processo com o intuito de tornar mais abrangente a LPAS, permitindo que outros cenários sejam contemplados, como, por exemplo, as diferentes formas de contratação na área governamental; (ii) uso de um processo de aquisição derivado da LPAS em uma empresa que tenha cenários diferentes, utilizando as definições apresentadas neste trabalho e a ferramenta de apoio desenvolvida por BARRETO (2011); (iii) ampliação da definição dos componentes de processo para reutilização na definição de processos de aquisição, com a contribuição de especialistas experientes e do uso de um processo derivado em um contexto real.

### **Agradecimentos**

Os autores agradecem à CAPES e à FAPERJ pelo auxílio financeiro (E-26/110.399/2011).

### **Referências Bibliográficas**

- ARMBRUST, O., KATAHIRA, M., MIYAMOTO, Y., *et al.*, 2009, "Scoping Software Process Lines", *Software Process: Improvement and Practice*, 14, 3 (2009), p. 181-197.
- BARRETO, A., 2011, "Uma Abordagem para Definição de Processos Baseada em Reutilização Visando a Alta Maturidade de Processos", *Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil*.
- BARRETO, A., MURTA, L., ROCHA, A., 2008, "Software Process Definition: a Reuse-based Approach", *XXXIV Conferencia Latinoamericana de Informática (CLEI'08), Santa Fe, Argentina, p.409-419*.
- BARRETO, A., NUNES, E., ROCHA, A.R., *et al.*, 2010, "Supporting the Definition of Software Processes at Consulting Organizations via Software Process Lines", *7th International Conf. on the Quality of Information and Communications Technology, Porto, Portugal*.
- CHOI, S.J., SCACCHI, W., 2001, "Modeling and simulating software acquisition process architectures", *Journal of Systems and Software*, v. 59, n. 3, p. 343-354.
- CUI, W., XU, M., 2009, "Software company process management and process audit research", *International Conference on Management and Service Science, MASS*
- GOPAL, A., SIVARAMAKRISHNAN, K., KRISHNAN, M.S., *et al.*, 2003, "Contracts in Off-shore Software Development: An Empirical Analysis", *Management Science*, v. 49, n. 12,

- p. 1671-1683.
- GOTTSCHALK, P., KARLSEN, J.T., 2005, "A comparison of leadership roles in internal IT projects versus outsourcing projects", *Industrial Management and Data Systems*, v 105, n 9, p. 1137-1149.
- HADDAD, M., RIBIERE, V., 2007, "The use of knowledge management in software acquisition", *VINE: The Journal of Inf. and Knowledge Management Systems* v37 n3, p.295-313.
- HOFMANN, H., D., Y., MISHLER, J., et al., 2007, "CMMI for Outsourcing, Guidelines for Software, Systems, and IT Acquisition", *SEI Series in Software Engineering*.
- HUEN, W.H., 2007, "Systems engineering of complex software systems", Milwaukee, WI, United states: Institute of Electrical and Electronics Engineers Inc, p. F1A16-F1A21.
- IEEE STD 1062, E., 1998, "IEEE Recommended Practice for Software Acquisition", *IEEE (Software Engineering Standards Committee of the IEEE Computer Society)*.
- ISO/IEC, 2008, "Systems and software engineering – Software life cycle processes", *The International Organization for Standardization and the International Electrotechnical Commission*, v. *ISO/IEC 12207:2008*.
- JALIL, Z., HANIF, A., 2009, "Improving management of outsourced software projects in Pakistan", *2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009*, p. 524-528.
- JAMIESON, D., VINSEN, K., CALLENDER, G., 2005, "Agile procurement: New acquisition approach to agile software development", v. 2005, p. 266-273, Porto, Portugal.
- JIANG, Y., CHEN, L., ZHOU, X., et al., 2010, "Process-oriented software outsourcing decision based on genetic algorithm", *IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2010*, p. 386-391.
- KHAN, S., NIAZI, M., AHMAD, R., 2008, "A readiness model for software development outsourcing vendors", p. 273-277, Bangalore, India.
- KITCHENHAM, B., 2008, "Procedures for Performing Systematic Reviews, Technical Report", *Department of Computer Science Keele University, Keele*.
- KWAN, T.W., LEUNG, H.K.N., 2004, "Project perspective of software acquisition practices", *Eighth IASTED International Conf. on Software Engineering and Applications*, p. 456-464.
- MA, J., LI, J., CHEN, W., et al., 2007, "An industrial survey of software outsourcing in China", *Product-Focused Software Process Improvement - 8th International Conference, PROFES*, v. *Lecture Notes in Computer Science*, v 4589 LNCS, p. 5-19, 200.
- NUNES, E., 2011, "Definição de Processos de Aquisição de Software para Reutilização", *Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil*.
- NUNES, E., BARRETO, A.S., ROCHA, A.R.C., et al., 2010, "Definição de Processos de Aquisição de Software para Reutilização", *Conferência Latino-americana de Informática (CLEI), 2010, Assunção - Paraguai*.
- PUTNAM, L.H., MYER, W., 2003, "Five Core metrics – The Intelligence Behind Successful Software Management", *Dorset House Publishing Co*.
- REIFER, D.J., 2004, "Seven Hot Outsourcing Practices", *IEEE Software*, v. 21, n. 1, p. 14-16.
- REIS, R.Q., 2002, "APSEE-Reuse: Um Meta-Modelo para Apoiar a Reutilização de Processos de Software", *Tese de D.Sc., PPGC, UFRGS, Porto Alegre, Brasil*.
- SEI, 2010, "CMMI® for Development (CMMI-DEV), V1.3", *Software Engineering Institute*.
- SHENG, Z., NAKANO, M., KUBO, S., et al., 2008, "Experimental risk estimation for offshore software outsourcing", *IEEJ Trans. Electrical and Electronic Eng.*, v. 3, n. 3, p. 338-344.
- SOFTEX, 2011a, "MPS.BR - Melhoria de Processo do Software Brasileiro, Guia de Implementação – Parte 8: 2011", In: [http://www.softex.br/mpsbr/\\_guias/default.asp](http://www.softex.br/mpsbr/_guias/default.asp).
- SOFTEX, 2011b, "MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral:2011", In: [http://www.softex.br/mpsbr/\\_guias/default.asp](http://www.softex.br/mpsbr/_guias/default.asp).
- SPEM, 2006, "SPEM - Software Process Engineering Metamodel, Object Management Group."
- TSUJI, H., SAKURAI, A., YOSHIDA, K., et al., 2007, "Questionnaire-based risk assessment scheme for Japanese offshore software outsourcing", *Software Engineering Approaches for Offshore and Outsourced Development (SEAFOD)*, LNCS v 4716, p. 114-127.