

# Usando PABC-Pattern para Codificar o Conhecimento: Um Estudo Experimental

Jacilane Rabelo<sup>1</sup>, Davi Viana<sup>1</sup>, Gleison Santos<sup>2</sup>, Tayana Conte<sup>1</sup>

<sup>1</sup>Instituto de Computação – Universidade Federal do Amazonas (UFAM)  
Manaus - AM - Brasil

<sup>2</sup>Universidade Federal do Estado do Rio de Janeiro (UNIRIO)  
Rio de Janeiro - RJ - Brasil

{jaci.rabelo, davi.viana, tayana}@icompu.ufam.edu.br,  
gleison.santos@uniriotec.br

**Resumo.** *Gerência do conhecimento tem um papel fundamental no desenvolvimento de software. A coleta de lições aprendidas é uma prática comum em organizações de software para gerenciar e transferir o conhecimento. A codificação do conhecimento pode facilitar o registro das lições aprendidas para sua consulta posterior. Este trabalho apresenta um estudo qualitativo da PABC-Pattern (Problema, Ação, Benefício e Contexto – Padrão), uma nova abordagem para codificar e compartilhar o conhecimento na forma de lições aprendidas. Os resultados mostraram que há uma facilidade de uso nos campos da abordagem. Adicionalmente, foi feita uma comparação do esquema proposto pela PABC-Pattern com os esquemas de outras abordagens baseadas em padrão.*

**Abstract.** *Knowledge management plays a fundamental role in software development. Lessons learned capture is a common practice on software organizations in order to manage and transfer knowledge. Knowledge codification can facilitate lessons learned record for a future usage. This paper presents a qualitative study of the PABC-Pattern, a new approach for knowledge codification and sharing on the form of lessons learned. Our results showed that that it was easy to use the fields of the proposed approach. Additionally, we compared PABC Pattern with other approaches based on patterns.*

## 1. Introdução

Produtos e tecnologias mudam rapidamente, portanto, o investimento em conhecimento tornou-se uma estratégia competitiva e sustentável para as organizações [Nonaka e Teece 2001]. Porém, a realização das atividades que necessitam da aplicação de muito conhecimento não é fácil [Gomes *et al.* 2001]. Em atividades de engenharia de software, tratar o conhecimento pode ser problemático, pois essas atividades dependem intensamente do conhecimento [Bjørnson e Dingsøyr 2008] e exigem tanto esforço humano quanto níveis de conhecimento especializado elevados [Gomes *et al.* 2001].

A Gerência de Conhecimento (GC) contribui para o processo de tratamento do conhecimento organizacional [Prost *et al.* 1999]. O enfoque em GC pode auxiliar as organizações de software a compartilhar conhecimento relevante entre os seus colaboradores [Bjørnson e Dingsøyr 2008]. Logo, essas organizações evitam cometer falhas semelhantes, e aumentam a produtividade no trabalho [Alavi e Leidner, 1999]. O conhecimento a ser gerenciado em organizações de software pode ser encontrado em diferentes formas: rotinas, processos, tecnologias, documentos e repositórios.

De acordo com Davenport e Prusak (1999), o conhecimento pode ser classificado em dois tipos: tácito e explícito. O conhecimento tácito é baseado na experiência da pessoa e é subjetivo, o que o torna difícil de expressar com palavras, números e sentenças. Por outro lado, o conhecimento explícito, também chamado de conhecimento codificado, é considerado transmissível em linguagem formal e sistemática. Nonaka e Teece (2001) informam que esse conhecimento é objetivo, e pode ser representado por meio de documentos, relatórios e base de dados. Além disso, o conhecimento explícito deve estar estruturado adequadamente para que outros colaboradores possam utilizá-lo. Uma maneira de fazer isso é através de lições aprendidas, que são relatos de experiências contendo sucessos ou fracassos detectados durante o desenvolvimento de software [Falbo *et al.* 2004].

A utilização de GC é esperada nas organizações que implantam o MR-MPS-SW [SOFTEX, 2012] nível E, devido à existência do processo de Gerência de Recursos Humanos. O objetivo deste processo é manter os recursos humanos e competências necessárias para atender suas demandas, de forma consistente com as necessidades do negócio. Adicionalmente, o processo requer uma manutenção do conhecimento organizacional através de estratégias de GC. Assim, pode-se considerar que institucionalizar atividades de GC em uma organização influencia diretamente na qualidade das atividades desenvolvidas pelos colaboradores, auxiliando-os no compartilhamento das lições aprendidas e na colaboração durante o processo de desenvolvimento de software [Carneiro *et al.* 2009]. Lições aprendidas são comumente registradas em formato de texto e armazenadas em repositórios de conhecimento. No entanto, observações da prática mostram que é comum encontrar registros de lições aprendidas sem informações que facilitem a consulta e a futura utilização, quando necessário. O fato de não haver informações que expliquem a situação em que as lições aprendidas podem ser aplicadas, pode fazer com que seja difícil para o engenheiro de software consultá-las e utilizá-las.

A fim de melhorar o uso de lições aprendidas, algumas organizações criam mecanismos para codificar o conhecimento. No entanto, apesar de permitir o tratamento de uma grande quantidade de informações e reutilização do conhecimento, essas organizações não utilizam essas informações de forma efetiva [Andrade e Lima Reis 2011; Hughes 2006]. Dessa maneira, é necessário registrar as circunstâncias em que as lições aprendidas foram adquiridas e relacioná-las com práticas, situações e atividades do processo de software. Segundo McDermott (1999), se as pessoas que trabalham em um grupo não estão aptas a compartilhar conhecimento, nem conseguem entender quais *insights* e informações serão úteis, o desenvolvimento de mecanismos não auxiliará a codificação e compartilhamento dos conhecimentos. Consequentemente, devido a essas dificuldades, é necessário analisar o tratamento do conhecimento pelos colaboradores buscando responder questões relacionadas à: *Como os profissionais explicitam seu conhecimento? Quais os problemas e dificuldades enfrentados ao fazer este compartilhamento? Uma estratégia de codificação do conhecimento estruturada pode facilitar o registro e compartilhamento do conhecimento?*

Com o propósito de responder essas perguntas foi realizado um estudo experimental utilizando uma nova abordagem de codificação de conhecimento, a PABC-Pattern (Problema, Ação, Benefício e Contexto – Padrão), que tem por objetivo transformar informação ou compreensão adquirida pela experiência em conhecimento explícito. Em trabalhos anteriores, mostrou-se resultados de estudo comparativo de duas

abordagens de codificação: PABC-Pattern e Mapa Mental (Rabelo *et al.*, 2012). Neste presente trabalho apresenta-se o uso de PABC-Pattern por profissionais de tecnologia da informação. Os resultados desse estudo foram analisados através de procedimentos qualitativos que possibilitaram apontar que PABC-Pattern teve uma boa aceitação como abordagem para compartilhar e codificar o conhecimento. A principal razão para essa aceitação é devido à assistência prestada pela sua forma estruturada. No entanto, essa aceitação não foi unânime. Foram identificadas algumas dificuldades em relação à aceitação dos resultados alcançados, o que também permitiu identificar as facilidades e dificuldades para codificação e compartilhamento do conhecimento.

O restante deste artigo está estruturado da seguinte forma: a Seção 2 discute estratégias de codificação do conhecimento. A Seção 3 detalha a proposta da PABC-Pattern. A Seção 4 mostra os detalhes de configuração do estudo experimental, incluindo objetivo e procedimentos. Na Seção 5 são apresentados e discutidos os resultados do estudo. Seção 6 apresenta uma análise de características (*feature analysis*) comparando abordagens baseadas em padrões para codificar o conhecimento. Finalmente, a Seção 7 mostra as considerações finais.

## **2. Gerência de Conhecimento e Estratégias de Codificação**

Na Engenharia de Software (ES) tem se discutido como tratar as informações por meio da gerência do conhecimento [Bjørnson e Dingsøyr 2008]. Bjørnson e Dingsøyr (2008) apresentam uma revisão sistemática da literatura sobre os conceitos, resultados e métodos de investigação que têm sido utilizados para GC em ES. Os autores concluíram que o tipo de atividades de GC a ser aplicado em uma organização depende de como o desenvolvimento de software é feito. Isso é explicado pelas diferentes implicações de GC para o desenvolvimento ágil e o tradicional.

Organizações têm adotado diferentes estratégias, ferramentas e técnicas para GC [Andrade e Lima Reis 2011; Tang e Chiang 2009]. Aurum *et al.* (2008) apresentam as práticas atuais e questões facilitadoras de GC em processos de engenharia de software. Os autores realizaram um estudo qualitativo em duas organizações australianas e concluíram que os colaboradores acreditavam na utilidade de compartilhar o conhecimento. Além disso, os autores observaram que os colaboradores formavam redes informais, e que era necessário formalizar as práticas de conhecimento, a fim de apoiar essas redes com GC. Por fim, foi verificado que as ferramentas, técnicas e metodologias utilizadas nas organizações estudadas eram inadequadas para um gerenciamento eficiente do conhecimento [Aurum *et al.* 2008].

A maior parte do conhecimento dentro de uma organização é tácito. No entanto, para que uma organização utilize o conhecimento de forma eficiente, é necessário gerenciar tanto o conhecimento tácito quanto o explícito. A utilização desses dois conhecimentos é importante para acumular o conhecimento corporativo e explorar novos potenciais [Choi e Lee 2003]. Neste artigo, o foco é no gerenciamento do conhecimento explícito. Uma das estratégias relacionadas a esse tipo de conhecimento é a codificação, a qual é feita de “pessoa-a-documentos” [Hansen *et al.* 1999]. Na codificação, o conhecimento é extraído da pessoa, podendo ser reutilizado para diversos fins. Além disso, muitas pessoas podem buscar e reutilizar o conhecimento codificado sem ter contato com quem o criou [Hansen *et al.* 1999]. Na subseção seguinte serão apresentadas algumas estratégias para codificar o conhecimento.

### 2.1. Estratégias de Codificação

Tautz *et al.* (2000) apresentam os resultados de um estudo que comparou a eficácia e eficiência do compartilhamento de conhecimento tradicional (“pergunte aos seus colegas”) versus uma “abordagem baseada em repositório” (ou seja, sem a necessidade de saber quem criou a experiência). Muitos participantes afirmaram que em um projeto real primeiro usariam a “abordagem baseada em repositório” para identificar quem poderia ser entrevistado utilizando a abordagem “pergunte aos seus colegas”.

Andrade e Lima Reis (2011) apresentam os resultados do uso de uma Wiki como repositório para armazenar e compartilhar o conhecimento intelectual em uma organização de desenvolvimento de software. No entanto, apesar da existência da Wiki, o conhecimento era compartilhado por e-mails e conversas informais. Uma resistência também foi observada em relação à criação e consulta de itens de conhecimento (durante dois anos foram cadastrados apenas quatorze itens de conhecimento).

O Mapa Mental é um diagrama que pode ser usado para criar, visualizar, organizar e classificar ideias, assim como, auxiliar a estudar e organizar a informação, resolver problemas e tomar decisões [Tang e Chiang 2009]. Essa abordagem oferece informações úteis para o compartilhamento do conhecimento entre as organizações. Tang e Chiang (2009) apresentam um estudo no qual avaliaram o Mapa Mental como ferramenta para o compartilhamento do conhecimento organizacional. O principal benefício citado pelos funcionários foi a clareza do conteúdo e do gerenciamento da informação. Por outro lado, um dos aspectos negativos dos Mapas Mentais é a ausência de um contexto para a lição aprendida.

Rech e Ras (2011) propõem ABCDE (Ação, Benefício, Contexto, Descrição e Evidência). ABCDE é uma técnica para a agregação do conhecimento que organiza as lições aprendidas em cinco elementos: (a) ação, esclarece uma atividade que foi aplicada ao sistema; (b) benefício, um relatório que contém os efeitos (positivos ou negativos) que foram causados pela ação; (c) contexto, caracteriza o ambiente em que a ação foi realizada; (d) descrição, mostra uma explicação detalhada do problema, solução, e outros, a partir do padrão de software no seu nível de abstração; e (e) evidência, mostra um relatório e uma lista de evidências. ABCDE foi desenvolvida para atender a realidade da Fábrica de Experiência (*Experience Factory* ou EF) [Rech e Ras 2011] e sua adoção é limitada pelas organizações que implementam esse modelo. EF divide os esforços de desenvolvimento pelas organizações com responsabilidades distintas para desenvolver o software e capturar experiências. Essa abordagem é responsável pelo desenvolvimento, atualização e por proporcionar experiências reutilizáveis que podem ser utilizadas pela equipe de desenvolvimento do produto de software [Basili *et al.* 2001]. No entanto, EF requer um investimento significativo de tempo e esforço. Assim, é necessário desenvolver abordagens alternativas para distribuir conhecimento rapidamente. Por isso, foi desenvolvida uma nova abordagem de codificação de lições aprendidas, conforme apresentada na próxima seção.

### 3. Abordagem PABC-Pattern

PABC-Pattern é uma abordagem para codificar lições aprendidas de uma organização de software que não utilize o modelo de Fábrica de Experiência. Essa abordagem não está associada a ferramentas ou processos de software específicos, o que possibilita que diferentes organizações possam utilizá-la e colocar em práticas as atividades de GC. PABC-Pattern é estruturada e seus elementos visam armazenar o conhecimento. Além

disso, a PABC-Pattern possibilita consultar as lições de forma mais eficiente por meio de: identificação; uso de palavras-chaves; relação com outras lições; detalhamento maior do contexto em que essa lição pode ser aplicada. O contexto da lição aprendida é uma das formas mais eficazes de capturar e reutilizar o conhecimento em uma determinada situação, pois, o contexto de um artefato representa o "quando", "onde", "porquê", "o que" e "por quem" [Ras *et al.* 2009].

A estrutura da PABC-Pattern é baseada em evoluções no modelo ABCDE proposto em [Rech e Ras 2011]. Essas modificações visaram diminuir a quantidade de interações e pessoas envolvidas, bem como diminuir o esforço necessário para a codificação do conhecimento. Por isso, a abordagem PABC-Pattern utiliza padrão inspirado pela ideia dos padrões de processo<sup>1</sup>, a fim de permitir a codificação das lições aprendidas por uma única pessoa. A descrição de um padrão de processo deve conter os elementos estruturais necessários para a compreensão da ideia do padrão, assim, os elementos do PABC-Pattern são:

- **Identificação:** uma identificação única para cada lição;
- **Título:** descrição do nome da lição de forma resumida;
- **Problema:** detalhamento do problema ou uma pergunta que a lição aprendida deve solucionar;
- **Causa do Problema:** detalha a causa do problema, nessa descrição deve conter o que fez o problema ocorrer;
- **Consequência do Problema:** descrição da consequência do problema, ou seja, o que aconteceu após o problema ocorrer;
- **Ação:** detalha a solução para o problema, ou seja, esclarece uma atividade que foi aplicada para resolver o problema;
- **Benefício:** descrição contendo os efeitos (positivos e/ou negativos) que foram causados pela Ação;
- **Palavra-chave:** expõe as palavras-chaves que identificam a lição;
- **Relação com outras lições:** lista a identificação de outras lições;
- **Contexto:** caracterização do ambiente no qual a ação foi executada:
  - **Tipo de projeto:** seleção do tipo de projeto (projeto de desenvolvimento, projeto de manutenção, ou ambos);
  - **Tamanho do projeto:** seleção do tamanho do projeto (projeto pequeno, projeto médio, ou grande);
  - **Fase do projeto:** marcação da fase do projeto (levantamento de requisitos, análise de requisitos, projeto, implementação, teste, implantação, atividades de gerência, atividades de apoio, outros);
  - **Função/cargo do criador da lição:** descrição da função/cargo de quem criou a lição;
  - **Domínio relacionado:** descrição do domínio em que a lição pode ser aplicada;
  - **Outra informação relevante:** descrição de outra informação que o criador da lição possa julgar necessário.

A abordagem PABC-Pattern possui um roteiro que auxilia na utilização para codificar as lições aprendidas e um checklist (opcional) de validação. A Figura 1 e 2

<sup>1</sup> Um padrão de processo descreve uma solução geral, abordagens bem sucedidas e/ou séries de ações para um problema específico do processo de desenvolvimento de software [Ambler 1998].

mostram, respectivamente, parte do roteiro e do checklist. Além disso, uma parte do template da PABC-Pattern é apresentado na Figura 3 (Seção 6).

Em Rabelo *et al.* (2012), é apresentado um estudo qualitativo que comparou duas abordagens de apoio a codificação do conhecimento: PABC-Pattern e Mapa Mental. Os resultados do estudo indicaram que os participantes tinham uma preferência em utilizar a abordagem PABC-Pattern, devido à sua facilidade de uso e utilidade. Foram identificadas algumas razões para essa preferência: a) mais detalhes de codificação; b) uma descrição mais detalhada do cenário do conhecimento, que é mais esclarecedora; c) a possibilidade de adquirir mais informações com menos esforço; e d) a possibilidade de descrever tanto o problema e a solução.

Após o estudo qualitativo de comparação, houve uma avaliação na estrutura da PABC-Pattern para verificar se os elementos propostos estavam sendo utilizados corretamente pelos usuários. Com base nos resultados do estudo anterior e na avaliação da estrutura da abordagem, a PABC-Pattern foi evoluída e avaliada em um novo estudo experimental com profissionais da área de tecnologia. A seguir serão mostrados os procedimentos da realização desse estudo.

Procedimento Iniciais	Exemplos
<ul style="list-style-type: none"> <li>• Capture a <b>IDEIA CENTRAL (TÍTULO)</b> da lição aprendida – <i>Qual foi o ponto principal da lição aprendida que poderá ajudar outras pessoas?</i></li> </ul>	<p><b>Título:</b> Utilização dos protótipos de tela para validação de requisitos.</p>
<ul style="list-style-type: none"> <li>• Explique o <b>PROBLEMA</b> relacionada ou apresente uma <b>PERGUNTA</b> que a lição aprendida deve solucionar</li> </ul>	<p><b>Problema(s):</b> Dificuldade em avaliar requisitos com usuários com baixa experiência técnica em engenharia de software.</p>
<ul style="list-style-type: none"> <li>• Descreva a <b>CAUSA DO SITUAÇÃO</b> – <i>O que fez a situação ocorrer? Por que aconteceu a situação?</i></li> </ul>	<p><b>Causa(s) do(s) Problema(ões):</b> Falta de mecanismos adequados para avaliar requisitos com usuários com baixa experiência técnica em engenharia de software</p>
<ul style="list-style-type: none"> <li>• Explique a <b>CONSEQUÊNCIA DA SITUAÇÃO</b> – <i>O que a situação causou? O que aconteceu após ocorrer a situação? Qual foi o resultado da situação?</i></li> </ul>	<p><b>Consequência(s) do(s) Problema(ões):</b> Requisitos avaliados de forma inadequada e cliente insatisfeito devido à falta de atendimento de suas reais necessidades.</p>
<ul style="list-style-type: none"> <li>• Explique a <b>AÇÃO</b> que soluciona a <i>situação</i> – <i>Explique o que foi feito para solucionar a lição aprendida</i> – <i>A ação descreve o que fazer – Inicie a AÇÃO com verbos no infinitivo: Ex.: Criar, Fazer</i></li> </ul>	<p><b>Ação(ões):</b> Deve-se utilizar protótipo de tela para validação dos requisitos em fases iniciais do projeto.</p>
<ul style="list-style-type: none"> <li>• Descreva os <b>BENEFÍCIOS</b> da <i>ação</i> – <i>Descreva os efeitos positivos e/ou negativos que foram causados pela ação</i></li> </ul>	<p><b>Benefício(s):</b> Requisitos aprovados de forma mais adequada, sendo representativos das necessidades reais dos usuários do produto.</p>

Figura 1: Parte do Roteiro de utilização da PABC-Pattern

Contexto		
<b>Tipo de Projeto:</b> <i>Foi selecionado o tipo de projeto em que ocorreu a lição aprendida?</i>		
<input type="checkbox"/> Projeto de Desenvolvimento <input type="checkbox"/> Projeto de Manutenção <input type="checkbox"/> Ambos		
<b>Tamanho do Projeto:</b> <i>Foi selecionado o tamanho do projeto em que a lição aprendida pode ser aplicada?</i>		
<input type="checkbox"/> Projeto Pequeno <input type="checkbox"/> Projeto Médio <input type="checkbox"/> Projeto Grande <input type="checkbox"/> Projeto de qualquer tamanho		
<b>Fase do Projeto</b> (pode marcar mais de uma): <i>Foi marcado a(s) fases(s) do projeto em que o problema ocorreu?</i>		
<input type="checkbox"/> Levantamento de Requisitos	<input type="checkbox"/> Análise de Requisitos	<input type="checkbox"/> Projeto
<input type="checkbox"/> Implementação	<input type="checkbox"/> Teste	<input type="checkbox"/> Implantação
<input type="checkbox"/> Atividades de Gerência	<input type="checkbox"/> Atividades de Apoio	<input type="checkbox"/> Outros

Figura 2: Parte do Checklist da abordagem PABC-Pattern

#### 4. O Experimento

O objetivo do experimento foi verificar a aceitação da PABC-Pattern de acordo com a percepção dos profissionais da área de tecnologia da informação. Esses profissionais apresentaram sua percepção em relação à codificação e compreensão (decodificação) de lições aprendidas. Uma vez que já havia sido realizado uma comparação com Mapa

Mental descrito em [Rabelo *et al.* 2012], optou-se por realizar um estudo observacional. Esse estudo é, então, descrito abaixo.

A metodologia experimental seguida para apoiar o desenvolvimento e melhoria da PABC-Pattern consistiu em dois estudos experimentais. No primeiro estudo de viabilidade, resultados apresentados em Rabelo *et al.* 2012, o objetivo foi determinar se PABC-Pattern teria uma aceitação em comparação com uma abordagem visual, o Mapa Mental. No estudo experimental mostrado neste presente trabalho, o objetivo foi ‘olhar para dentro’ a fim de entender os fatores que facilitam ou não a utilização da PABC-Pattern por profissionais. Além disso, foi observado se esses profissionais iriam utilizar novamente e recomendariam a PABC-Pattern como abordagem para codificar lições aprendidas na organização em que trabalham.

#### 4.1 Objetivo

O objetivo desse estudo, baseado no paradigma GQM (Goal, Questions, Metrics) [Basili e Rombach 1998], é mostrado na Tabela 1.

**Tabela 1: Objetivo do estudo baseado no paradigma GQM**

<b>Analisar</b>	abordagem PABC-Pattern
<b>Com o propósito de</b>	compreender
<b>Em relação a</b>	percepção sobre: (a) compartilhamento do conhecimento, (b) codificação do conhecimento, (c) compreensão do conhecimento, (d) utilização da abordagem para codificar o conhecimento.
<b>Do ponto de vista de</b>	pesquisadores em Engenharia de Software
<b>No contexto de</b>	codificação do Conhecimento de Situações Reais vivenciadas em Projetos de Software

#### 4.2 Participantes

Participaram do estudo 19 profissionais da área de Tecnologia da Informação (TI) de 7 diferentes organizações do Estado do Amazonas. A maioria dos profissionais trabalham em organizações de TI por mais de dois anos e possuem experiência em mais de dois projetos de desenvolvimento. Os papéis/funções dos profissionais são: desenvolvedor, analista, testador, técnico de hardware e gerente de configuração. Esses profissionais trabalham para alguma dessas empresas: (a) uma organização que presta serviços de tecnologia para os órgãos públicos do Estado do Amazonas – Brasil; (b) uma empresa que desenvolve soluções tecnológicas para empresas da Zona Franca de Manaus; (c) dois institutos de pesquisa e desenvolvimento para dispositivos móveis; (d) uma empresa de consultoria especializada no desenvolvimento e melhoria de soluções de software; e (e) duas agências governamentais diferentes, responsáveis pelo desenvolvimento de sistemas computacionais e outras soluções tecnológicas.

Todos os participantes concordaram em participar do estudo de forma voluntária (não houve nenhuma bonificação). Antes da execução do estudo, os profissionais responderam ao questionário de caracterização do perfil a fim de verificarmos: a) experiência em participação de projeto; b) tempo de trabalho em empresas de desenvolvimento de software; e, c) nível de conhecimento em GC. A distribuição dos dados por participantes são mostrados na Tabela 2.

#### 4.3 Procedimentos

O estudo foi realizado individualmente e teve duração total de duas semanas para a sua execução. Para garantir que todos os participantes realizassem as mesmas atividades foi feito um planejamento com as seguintes etapas da execução do experimento:

- O participante respondia o questionário de caracterização do perfil;

**Tabela 2: Caracterização dos profissionais que participaram do estudo**

<b>Experiência sobre a participação em desenvolvimento de software</b>	Experiência em 1 projeto	0
	Experiência em 2 a 4 projetos	6
	Experiência em mais de 5 projetos	13
<b>Tempo de trabalho em empresas de desenvolvimento de software</b>	Entre 1 a 2 anos	1
	Entre 2 a 5 anos	11
	Acima de 5 anos	7
<b>Conhecimento sobre gerência de conhecimento</b>	Nenhum conhecimento prévio de GC	9
	Algumas noções de GC adquiridas através de leituras/palestras ou tópicos de disciplinas	7
	Estudou GC em disciplina específica para este assunto ou livro específico	3

- O pesquisador informava que estava realizando uma pesquisa sobre lições aprendidas e pedia ao participante que relatasse uma lição aprendida vivenciada e que considerava importante para compartilhar com outros colaboradores da empresa em que trabalha;
- Após relatar a lição aprendida, o pesquisador apresentava ao participante o template da PABC-Pattern para que fosse codificada a lição. O participante também recebia o Roteiro e o Checklist da PABC-Pattern (ver Figura 1 e 2) a fim de ajudar na atividade de codificação da lição aprendida. A Figura 3 mostra um exemplo de lição aprendida codificada;
- Depois de codificar a lição aprendida, o participante respondia um questionário contendo perguntas sobre a facilidade de uso e utilidade da PABC-Pattern;
- Uma lição aprendida codificada era entregue ao participante, a fim de verificar a sua compreensão. Essa lição aprendida foi extraída de um projeto real de desenvolvimento de software;
- Após o participante informar que já havia compreendido a lição, o pesquisador fazia perguntas para validar essa compreensão utilizando o Checklist da PABC-Pattern;
- O pesquisador recolhia a lição aprendida e fazia novos questionamentos ao participante. O objetivo era verificar se o participante realmente havia entendido a lição;
- O participante respondia o questionário de avaliação do entendimento da lição aprendida que foi codificada usando a PABC-Pattern.

Os dados dos questionários de avaliação foram analisados utilizando procedimentos de Grounded Theory (GT) [Strauss e Corbin 1998].

<b>PABC-Pattern Template</b>
<b>Título:</b> <i>Experiência no desenvolvimento com Postgree SQL.</i>
<b>Problema(s):</b> <i>Lentidão na consulta de informações no Banco de Dados.</i>
<b>Causa(s) do Problema(s):</b> <i>Implementação de maneira não eficiente das consultas SQL Cliente/ Servidor.</i>
<b>Consequência(s) do Problema(s):</b> <i>Aumento de produtos na linha de produção; Não cumprimento de metas; Desempenho muito ruim do software para sua demanda.</i>
<b>Ação(ões):</b> <i>As consultas que eram feitas do lado do cliente foram transferidas para o lado servidor, através de <i>functions</i>. Agora o software contém apenas <i>functions</i> e não toda instrução SQL, fazendo agora apenas referência as <i>functions</i>.</i>
<b>Benefício(s):</b> <i>O desempenho da consulta foi significativa, no caso, as consultas que costumavam levar 20-30 segundos, agora leva de 2 a 3 segundos.</i>
<b>Palavra(s)-chave:</b> <i>Postgree SQL; Functions; Consulta banco de dados.</i>

**Figura 3: Exemplo de lição aprendida codificada na PABC-Pattern**



#### 4.4. Análise Qualitativa

GT é um método qualitativo para a análise de dados, que descreve um conjunto de procedimentos sistemáticos de coleta e análise dos dados para gerar e validar teorias substantivas sobre fenômenos essencialmente sociais, ou processos sociais abrangentes. O método GT é baseado em codificação, um processo analítico pelo qual os dados são divididos, conceituados e integrados para formar uma teoria [Strauss e Corbin 1998].

O processo de codificação é dividido em três fases: (a) codificação aberta que envolve a quebra, a análise, a comparação, a conceituação e a categorização dos dados; (b) codificação axial onde se explicitam as relações entre os códigos, mostrando relações de causas e efeitos, condições intervenientes e estratégias de ação; e, (c) codificação seletiva que integra todas as outras categorias e expressar a essência do processo social que ocorre entre os envolvidos [Strauss e Corbin 1998]. Neste presente artigo, foram realizados somente as fases a e b. Apesar do propósito ser a construção de teorias sociais, o uso do método pode ser feito de acordo com os objetivos de pesquisa [Strauss e Corbin 1998]. Após a análise, três outros pesquisadores verificaram esses códigos e categorias a fim de auditar o processo de codificação.

#### 5. Discussão

Seaman (1999) descreve que os dados qualitativos podem ser usados para ir além das estatísticas e auxiliam na explicação das razões por trás das hipóteses e relacionamentos. A utilização de métodos qualitativos auxilia em uma compreensão mais profunda de todo o fenômeno. A análise qualitativa dos resultados do estudo está descritas a seguir.

**Entendimento do Conhecimento:** os participantes do estudo tiveram que entender uma lição aprendida codificada na PABC-Pattern. O pesquisador que realizou o estudo fez algumas perguntas relacionadas à lição aprendida para verificar se os participantes realmente entenderam. Todos os participantes tiveram **facilidade de entendimento**. Por isso, reforça-se que todos os participantes realmente haviam entendido a lição. Algumas causas citadas para essa facilidade de entendimento são: “Formulário da PABC-Pattern estava bastante claro”, “Conhecimento registrado na PABC-Pattern permite a resolução de um problema com mais rapidez e eficiência” e “Roteiro da PABC-Pattern facilita o entendimento da lição aprendida”. A citação do participante 15 mostra uma dessas facilidades:

*“A lição observada com o uso do PABC-Pattern estava bastante clara, e muito amigável sua compreensão”.* Participante 15

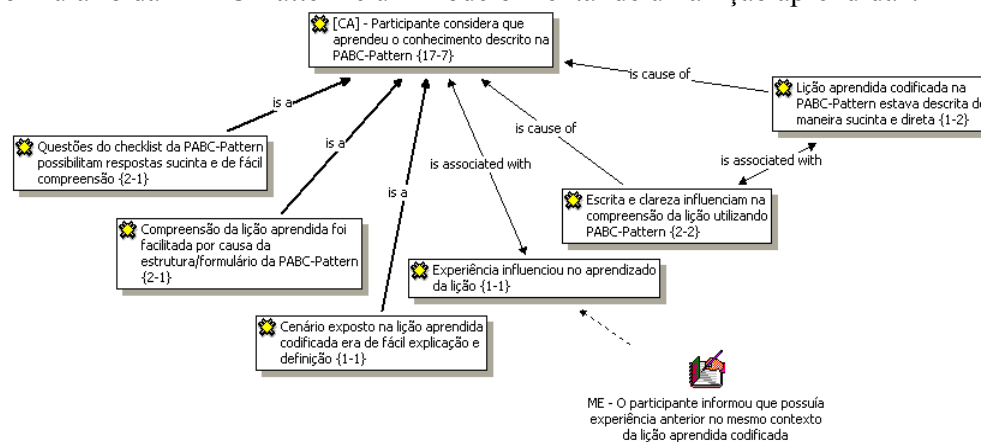
**Codificação do conhecimento:** em relação à atividade de codificação do conhecimento, onze participantes afirmaram que a PABC-Pattern **é fácil de ser usada para codificar o conhecimento**. Algumas das razões para essas afirmações são: “Roteiro da PABC-Pattern facilita na codificação da lição aprendida” e “PABC-Pattern mantém uma linha de raciocínio que torna o processo bem intuitivo”. A afirmação do participante 5 ilustra uma das causas identificadas:

*“(…) o uso de exemplos simples e o fato de ‘manter’ uma linha de raciocínio torna o processo bem intuitivo”.* Participante 5

**Compartilhamento do Conhecimento:** Todos, exceto dois participantes, consideraram que PABC-Pattern possui **facilidade para compartilhar o conhecimento**. Algumas causas relacionadas que poderiam contribuir para essa facilidade são: “PABC-Pattern permite reunir diversas experiências” e “PABC-Pattern mostra como descrever os pontos essenciais da lição (problema, ação e outros)”.

Os participantes foram questionados, após a atividade de entendimento da lição, se consideravam que houve um aprendizado do conhecimento codificado na PABC-Pattern. Além disso, foram questionados se os campos da PABC-Pattern haviam auxiliado nesse aprendizado. No que diz respeito ao **aprendizado do conhecimento** descrito na PABC-Pattern, 17 participantes afirmaram que o processo de aprendizado ocorreu. Algumas das razões para essa facilidade são: “Questões do checklist da PABC-Pattern possibilitam respostas sucinta e de fácil entendimento”, e “Entendimento da lição aprendida foi facilitada por causa do formulário da lição aprendida”. A Figura 4 mostra a relação entre esses códigos. Apenas um participante considerou que **não aprendeu o conhecimento** descrito na PABC-Pattern, pois julgou que para um real aprendizado é necessário ter um estudo mais aprofundado da lição. Finalmente, os participantes também afirmaram que os **campos da PABC-Pattern** auxiliam no aprendizado da lição.

Alguns aspectos relacionados à **facilidade de entendimento** sobre os **campos** são: “campos da PABC-Pattern são diretos e exigem poucas palavras”, “campos da PABC-Pattern ajudam a organizar e tornar o conhecimento disponível”, e “campos da PABC-Pattern são autoexplicativos”. Alguns aspectos que tornaram **mais fácil de usar** a PABC-Pattern estão relacionados com o **formulário**, tais como: “Seções do formulário da PABC-Pattern contribuem para explicitar mais lição(ões) aprendida(s)”, “Formulário da PABC-Pattern extrai a lição aprendida de quem preenche” e “Formulário da PABC-Pattern é um modelo mental de uma lição aprendida”.



**Figura 4: Códigos relacionados ao aprendizado do conhecimento na PABC-Pattern**

Após a aplicação da abordagem, os participantes foram questionados sobre a recomendação da PABC-Pattern na empresa em que trabalham. Adicionalmente, buscou-se identificar se a lição aprendida codificada na PABC-Pattern poderia ajudar outros colaboradores. Os resultados mostraram que 14 participantes **recomendariam a utilização da PABC-Pattern na empresa em que trabalham**. Algumas afirmações que mostram essa recomendação são: “Formulário da PABC-Pattern facilita a entrada de novos funcionários” e “PABC-Pattern facilita na percepção de erros conhecidos”. Ao analisar os dados, observou-se, ainda, algumas **Restrições para recomendar a utilização da PABC-Pattern**. São elas: “Para adotar PABC-Pattern deve-se assegurar que os colaboradores realmente compartilhem o conhecimento”, “Uma forma simplificada de coleta de informações facilitaria a recomendação de utilização da PABC-Pattern”, “Registro do conhecimento deve ser acessível a qualquer colaborador”

e “Deve-se assegurar que PABC-Pattern se enquadra na política e processo da empresa”.

A fim de verificar se os **participantes já tinham usado algum mecanismo (abordagem/ferramenta/documento) para codificar a lição aprendida e externalizar o conhecimento** (para responder a questão “*Como os profissionais explicitam seu conhecimento?*”), uma nova questão foi adicionada ao questionário de avaliação. Alguns dos mecanismos citados pelos participantes foram: “Listagem de lições no final do *sprint*”, “Listagem em um framework para armazenar ideias e lições”, “Sistema similar a Wikipédia”, “Listagem de lições nas reuniões de *postmortem*”. Oito **participantes não tinham utilizado algum mecanismo** e três **não compreenderam como relatar outras abordagens** para codificação de lição aprendida. Os participantes também foram questionados sobre que **tipo de lição aprendida eles** gostariam de codificar e em qual **contexto**. Cinco participantes citaram “Importante codificar todos os tipos de lições”. Outras sugestões foram mais específicas, tais como lições relacionadas a: “Elaboração de pré-projetos”, “Manutenção de base de dados”, “Levantamento de requisitos”, “Testes em aplicação”, “Atividades com muitas dificuldades para ser resolvidas”.

Os participantes destacaram algumas razões para a **dificuldade em utilizar a PABC-Pattern** para codificação do conhecimento: “Descrever uma lição aprendida pode requerer muitos detalhes e acaba sendo difícil repassar isso para os colaboradores”, “Preenchimento dos campos na PABC-Pattern é complexo”, “Nem sempre pode ser fácil utilizar PABC-Pattern para codificar lições aprendidas” e “Facilidade em utilizar a PABC-Pattern pode ser influenciada pela complexidade do projeto a ser utilizado na codificação de lições”. Um participante que teve **dificuldade em compartilhar o conhecimento** usando a PABC-Pattern afirmou: “Fatores externos podem afetar os resultados da PABC-Pattern” e outro afirmou “Campos poderiam ser mais claros”. No entanto, os participantes não listaram os fatores externos. Foram identificadas duas **dificuldades** relacionadas ao **formulário**: “Diagramação do formulário da PABC-Pattern poderia ter uma interface mais amigável” e “Espaço disponível para os campos no template induz a quem utilizar a resumir ou informar menos dados”.

Apesar dos resultados apresentados não serem generalizáveis, os resultados mostrados nesse estudo qualitativo permitem contribuir para a compreensão do processo de codificação e compreensão do conhecimento através do uso da PABC-Pattern. Também foi possível observar sugestões que podem auxiliar evoluções futuras da abordagem proposta. Além dessas sugestões, realizou-se uma análise de características, com o objetivo de melhorar a estrutura da PABC-Pattern.

## 6. Análise de Características de Abordagens baseadas em Padrões

Um Padrão é uma solução genérica para um problema ou questão a partir do qual uma solução específica pode ser obtida [Coplien 1995]. Ambler (1998) afirma que os padrões são úteis para capturar e refinar o conhecimento de uma forma reutilizável. A abordagem PABC-Pattern é baseada em padrões inspirada pela ideia dos padrões de processo. PABC-Pattern teve seus campos definidos a fim de facilitar a codificação e compreensão das lições aprendidas. Assim, uma das questões que os participantes tiveram que responder no questionário de avaliação relatado nesse artigo foi: “*Na sua opinião, existe algum campo da PABC-Pattern que não é necessário para codificar/entender a lição aprendida? Por favor, comente*”. No entanto não houve retorno satisfatório em relação a essa questão. Visando melhorar PABC-Pattern, foi

executada uma análise de características com outras três abordagens baseada em padrões encontradas na literatura. Essa análise é descrita a seguir.

Hugles (2006) descreve sua abordagem no contexto de um sistema de gerenciamento de conhecimento da usabilidade. O sistema é baseado nos princípios da linguagem padrão e de casos de usos da escrita, transformando as lições aprendidas a partir de teste de usabilidade em conhecimento organizacional. Essa abordagem não foi validada experimentalmente, no entanto, de acordo com o autor, serve com um ponto de partida para outras propostas. Os elementos da abordagem de Hugles (2006) são mostrados na Tabela 3.

**Tabela 3: Elementos propostos em Hugles (2006)**

Componente Padrão	Descrição
Usuários queriam...	Representa o contexto do padrão. É a declaração do cenário ou um objetivo específico do usuário dentro de um cenário
Ação do Usuário	Detalhes das ações realizadas dentro da interface do usuário na tentativa de cumprir o seu objetivo. Está relacionada com as forças dentro de um contexto
Ação do Sistema	Relatório de como o sistema respondeu ou acomodou a ação do usuário
Data	Usado para apoiar as ações do usuário, exibe a data/hora dos relatórios originais ou dos dados
Solução	Detalhes da intervenção tomadas ou alteração do <i>design</i> que resultaram do achado
Comentários	Outras informações ou dados que podem ser úteis

Feldhusen e Bungert (2007) discutem a aplicação de linguagem padrão para *Product Lifecycle Management* (PLM) e engenharia de *design*. O objetivo dessa aplicação é desenvolver uma teoria coerente que associa o conhecimento corrente e faz com que seja utilizado durante todo o ciclo de vida do produto. PLM acrescenta diferentes objetos de design com vários níveis de detalhes, diferente do ciclo de vida e várias disciplinas da engenharia. A estrutura usada no trabalho de Feldhusen e Bungert (2007) é mostrada na Tabela 4.

**Tabela 4: Elementos propostos por Feldhusen e Bungert (2007)**

Componente Padrão	Descrição
Nome	Nome descritivo e único do padrão
Posição	Classificação relacionada à sua qualidade (opcional)
Imagem	Ilustra um exemplo do padrão
Contexto	Contexto detalhado do problema
Problema	Descrição detalhada do problema. Adição de comentário sobre as forças que causam o problema
Exemplo	Descrição das situações em que problema ocorreu em conjunto com uma solução
Solução	Solução arquitetural detalhada do problema ilustrado por um diagrama
Suporte padrão	Lista de referências ao suporte padrão

Com o objetivo de incentivar a GC em pequenas-médias empresas (PME), Wildner (2008) propõe uma abordagem que tem como objetivo fornecer descrições estruturadas para problemas recorrentes nas PME. A descrição da abordagem foi desenvolvida com base em padrões conhecidos e é mostrada na Tabela 5.

**Tabela 5: Elementos propostos por Wildner (2008)**

Componente Padrão	Descrição
Nome	Identificação única do padrão
Contexto	Breve descrição da situação em que o problema ocorreu
Problema	Pequena descrição do problema
Forças	Descrição detalhada dos fatores de influência sobre o problema
Solução	Descreve detalhes sobre a solução para o problema
Justificativa	Justifica a solução proposta
Resultado do contexto	Descrição das consequências da implementação da solução
Padrões relacionados	Indica padrões a uma parte do contexto do problema e fornece uma solução
Usos conhecidos	Descrição ou referências para exemplos de implementação conhecidas da solução

### 6.1. Comparação das Abordagens

A Tabela 6 mostra a comparação entre os elementos (campos) das abordagens descritas nessa Seção 6 com os elementos da PABC-Pattern (apresentada na Seção 3).

**Tabela 6: Comparação das abordagens**

PABC-Pattern		Hughes (2006)	Feldhusen and Bungert (2007)	Wildner (2008)
Título		-	Nome	Nome
Problema		-	Problema	Problema
Causa do Problema		-	-	-
Consequência do Problema		-	-	-
Ação		Ação do usuário, Ação do sistema e Solução	Solução	Solução
Benefício		-	-	Justificativa e Padrões Relacionados
Palavra-chave		-	-	-
Relação com outras lições		-	Suporte padrão	-
Contexto	Tipo de projeto	Usuários querem... e Comentário	Contexto	Contexto
	Tamanho do projeto			
	Fase do projeto			
	Responsável/papel de quem criou a lição			
	Domínio relacionado			
Outras informações relevantes				
-		Data	-	-
-		-	Posição	-
-		-	Imagem	-
-		-	-	Usos conhecidos
-		-	-	Forças
-		-	Exemplo	-

Na Tabela 6, pode-se observar que a PABC-Pattern possui elementos semelhantes a outras abordagens, mas com nomenclaturas diferentes: *Título*, *Problema*, *Ação*, *Contexto* e *Benefícios*. PABC-Pattern tem alguns elementos adicionais: *Causa do Problema*, *Consequência do Problema* e *Palavra-chave*. Os campos (*Consequência* e *Causa*) do problema são úteis para mostrar o que causou o problema e qual o resultado quando esse ocorre. No estudo experimental relatado nesse presente trabalho foi possível verificar que esses campos foram julgados úteis pelos participantes. Assim, esses campos serão mantidos na nova versão da PABC-Pattern.

O campo *palavra-chave* é benéfico para ser usado em mecanismos de recuperação da informação. O campo *Contexto* da PABC-Pattern é o mais detalhado em comparação com os elementos correspondentes das outras abordagens. Esse elemento é usado para detalhar os pontos mais importantes relacionados ao contexto em que a lição apreendida pode ser aplicada no futuro. O campo *Data* proposto por Hughes (2006) é importante porque mostra dados da criação da lição. Esse campo será adicionado aos detalhes do campo contexto da PABC-Pattern. Será recomendada a inclusão de *exemplos*, *usos conhecidos* e *imagens* no campo *Contexto – Outras informações relevantes*. Adicionar o campo *Imagens* também foi uma sugestão de um participante do estudo (ver Seção 6).

### 7. Conclusões e Comentários sobre Trabalhos Futuros

Este trabalho apresentou um estudo qualitativo com o objetivo de verificar o uso da abordagem PABC-Pattern em relação à codificação e compreensão das lições aprendidas. Para realizar esse estudo utilizou-se procedimentos do método *Grounded Theory* para analisar os dados. O estudo foi realizado com profissionais da área de Tecnologia da Informação de diferentes empresas do Estado do Amazonas.

Os resultados apresentados permitiram observar como os profissionais codificam e compreendem o conhecimento através da abordagem PABC-Pattern. Em relação à facilidade de codificação do conhecimento, identificou-se que PABC-Pattern auxilia a resolver problemas semelhantes. Além disso, observou-se que o formulário da PABC-Pattern ajuda no entendimento da lição aprendida e algumas dificuldades foram apontadas. Por exemplo, os profissionais mencionaram que o preenchimento de alguns campos foi uma tarefa complexa. Também foi indicado que a cultura da organização pode inviabilizar a execução da PABC-Pattern.

O estudo possui algumas limitações. A generalização dos resultados dos trabalhos qualitativos é limitada pelo fato dos resultados encontrados estarem relacionados diretamente ao contexto onde o estudo foi aplicado. No entanto, os resultados são relevantes para um entendimento mais detalhada da codificação e compreensão do conhecimento. Além disso, apesar da maioria dos colaboradores nunca ter utilizado formas de codificar o conhecimento, suas experiências com tecnologia da informação são importantes *inputs* para observar a aplicabilidade da PABC-Pattern

Como trabalhos futuros, espera-se a execução de novos estudos a fim de aplicar PABC-Pattern em diferentes organizações de software e verificar sua viabilidade em vários contextos. Por exemplo, as lições aprendidas serão definidas por um participante e usadas por outros em um contexto real, com a finalidade de avaliar o apoio da PABC-Pattern.

### **Agradecimentos**

Os autores agradecem o apoio concedido pela: CAPES (Doutorado); FAPEAM através do Edital N.009/2012 - RHTI - (Doutorado), por meio do Projeto Universal Amazonas sob o processo n. 062.00146/2012 e por meio do Edital 015/2013 - PROTI-Mobilidade. Os autores agradecem aos profissionais que participaram do estudo e aos comentários dos revisores deste artigo.

### **Referências**

- Alavi, M. e Leidner, D. (1999) “Knowledge Management Systems: Emerging Views and Practices from the Field”, *Communications of the AIS*, p. 1-5.
- Ambler, S. (1998) “Process Patterns: Building Large-Scale Systems Using Object Technology”, Cambridge University Press, New York.
- Andrade, G. e Lima Reis, C. (2011) “Considerações sobre Implantação de Gerência de Conhecimento em Uma Organização de Desenvolvimento de Software”, *ESELAW*, Rio de Janeiro, p. 17-26.
- Aurum, A., Daneshgara, F., e Warda, J. (2008) “Investigating Knowledge Management practices in software development organisations – An Australian experience”. *Information and Software Technology*, 50(6), p. 511-533.
- Basili, V. e Rombach, H. (1998) “The tame project: towards improvement-oriented software environments”, *IEEE Transactions on Software Engineering*, v. 14, n. 6, pp. 758-773.
- Basili, V., Costa, P., Lindvall, M., Mendonça, M., e Seaman, C. (2001) “An Experience Management System for a Software Engineering Research Organisation”, *Proceedings of 26th Annual NASA Goddard Software Engineering Workshop*, p. 29-35.
- Bjørnson, O. e Dingsøyr, T. (2008) “Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts, Findings and Research Methods Used”, *Information and Software Technology*, Elsevier, p. 1055 – 1068.
- Carneiro, P., Costa, R., e Meira, S. (2009) “MaRKSoNe: A Tool for Supporting Knowledge Management in Software Projects”. *Simpósio Brasileiro de Sistemas Colaborativos*, p.70-79.

- Choi, B. e Lee, H. (2003) “An Empirical investigation of KM Styles and their Effect on Corporate Performance”, *Information & Management*, v. 40, p. 403-417.
- Coplien, J. (1995). *A Development Process Generative Pattern Language*. In *Pattern Languages of Program Design Addison Wesley*, Reading, Mass.
- Davenport, T. H. e Prusak, L. (1997) “Working Knowledge: How Organizations Manage What They Know”, 1. ed.: Harvard Business School Press, Boston, 224pp.
- Feldhusen, J. e Bungert, F. (2007) “Pattern languages: an approach to manage archetypal engineering knowledge”, 16th International Conference on Engineering Design, p. 581-592.
- Falbo, R., Arantes, D. e Natali, A. (2004) “Integrating Knowledge Management and Groupware in a Software Development Environment”, 5th International Conference on Practical Aspects of Knowledge Management - PAKM'2004, v. 3336, Berlin, p. 94-105.
- Gomes, M., Carolyn, S., Basili, V. e Kim, Y. (2001). “A Prototype Experience Management System for a software Consulting Organization”, 13th Conference on Software Engineering and Knowledge Engineering, SEKE'01, Buenos Aires, p. 29-36.
- Hansen, M., Nohria, N. e Tiernev, T. (1999) “What is Your Strategy for Managing Knowledge?”, *Harvard Business Review* 77(2), p. 106-116.
- Hughes, M. (2006) “A pattern language approach to usability knowledge management”, *Journal of Usability Studies (JUS)*, 2(1), p. 76-90.
- McDermott, R. (1999) “Why information technology inspired but cannot deliver knowledge management”, *California Management Review* 41, p. 103–117.
- Nonaka, I. e Teece, J. (2001) “Managing Industrial Knowledge: Creation, Transfer and Utilizations”, Chapter 3, Sage Publications, London, p. 330-335.
- Sharma, N. Kawaljeet, S. e Goyal, D. (2012) “Adoption of Knowledge Management Practices in Software Engineering Organizations: A Survey of Software Engineers' Perceptions”. Second International Conference on Advanced Computing & Communication Technologies, p. 24-29.
- Probst, B. Raub, S. e Romhardt, K. (1999) “Managing Knowledge: Building Blocks for Success”, 1. ed., New York: Wiley, 368p.
- Rabelo, J. Conte, T. Viana, D. e Santos, G. (2012) “Comparing Knowledge Codification Approaches: An Empirical Study”, IX Simpósio Brasileiro de Sistemas Colaborativos (SBSC), p. 136-145.
- Ras, E. Rech, J. e Weber, S. (2009) “Knowledge Services for Experience Factories”, WM2009: 5th Conference on Professional Knowledge Management.
- Rech, J. and Ras, E. (2011) “Aggregation of Experiences in Experience Factories into Software Patterns”, ACM SIGSOFT Software Engineering Notes, p. 1-4.
- Seaman, C. B. (1999) “Qualitative Methods in Empirical Studies of Software Engineering”, *IEEE Transactions on Software Engineering*, v. 25, n. 4, p. 557-572.
- SOFTEX (2012) Associação para Promoção da Excelência do Software Brasileiro - SOFTEX. MPS.BR - Guia Geral MPS de Software, dezembro 2012. Disponível em [www.softex.br](http://www.softex.br).
- Strauss, A. e J. Corbin (1998) “Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory”, 2ed., SAGE Publications, London.
- Tang, J. e C. Chiang, (2009) “Organizational Knowledge Sharing through Mind Mapping”, *Fuzzy Systems and Knowledge Discovery – FSKD'09*. Sixth International Conference On, v.2, p. 305-309.
- Tautz, C., Althoff, K.-D e Nick, M. (2000) “A Case-Based Reasoning Approach for Managing Qualitative Experience”, *Workshop on Intelligent Lessons Learned Systems*.
- Wildner, S. (2008) “Problem-oriented knowledge management - towards a pattern-based implementation approach for KM in SME”, 9th European Conference on Knowledge Management, p. 967-976.