

Facilitando a Aprendizagem Organizacional em Melhoria de Processo de Software

Davi Viana¹, Tayana Conte², Cleidson de Souza³

¹Coordenação do Curso de Engenharia da Computação – Universidade Federal do Maranhão (UFMA)
São Luís – MA – Brasil

²Grupo de Usabilidade e Engenharia de Software (USES). Programa de Pós-graduação em Informática. Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

³Instituto Tecnológico Vale e Universidade Federal do Pará (UFPA)
Belém – PA – Brazil

davi.viana@ufma.br, tayana@icomp.ufam.edu.br, cdesouza@ufpa.br

Abstract. *Software Process Improvement involves (SPI) knowledge that can be new for the organization's members, which may need Organizational Learning (OL) to support the transfer of knowledge. However, applying OL in Software Organizations is not trivial. This work presents the KL-SPI (Knowledge and Learning to facilitate Software Process Improvement) Framework, that was defined from a systematic mapping and investigations of software development practice. The main results are the conceptual model to guide the OL practices identification and a set of practices that aims to facilitate OL in SPI initiatives.*

Resumo. *Melhorias em Processos de Software (MPS) envolvem diversos conhecimentos que podem ser novos para os colaboradores, e conseqüentemente, podem necessitar de Aprendizagem Organizacional (AO). Contudo, não é trivial aplicar a AO em Organizações de Software. Assim, este trabalho apresenta o Framework KL-SPI (Knowledge and Learning to facilitate Software Process Improvement). Este framework foi definido a partir de investigações em organizações de software e um mapeamento da literatura. Como principais resultados desta pesquisa, tem-se o desenvolvimento de um modelo conceitual para guiar a identificação de práticas de AO e um conjunto de práticas que visam facilitar a AO em iniciativas de MPS.*

1. Introdução

Melhorias de Processo de Software são iniciativas que buscam promover o aprimoramento dos processos de desenvolvimento de uma organização. Essas melhorias são ações executadas com o objetivo de evoluir os processos de software para que eles se tornem mais alinhados aos objetivos de negócio da organização. Essas ações podem ser executadas de acordo com as necessidades organizacionais ou podem seguir boas práticas de Engenharia de Software promovidas pelos modelos de maturidade em MPS

utilizados pela Indústria de Software, tais como o Capability Maturity Model Integration for Development - CMMI-DEV (Sei, 2010) e o Modelo de Referência de Melhoria de Processo de Software – MR-MPS-SW (Softex, 2016). As boas práticas podem contribuir para as organizações melhorarem a qualidade de seus produtos, aumentarem a produtividade e melhorarem sua vantagem competitiva (Santos, 2011).

Implantações de iniciativas de Melhoria de Processo de Software são atividades intensas em conhecimento. Essas iniciativas podem envolver conhecimentos como, por exemplo, Gerência de Projetos, Gerência de Requisitos, Garantia da Qualidade, Verificação de Software e entre outros. Além disso, os conhecimentos precisam ser transferidos e aprendidos pelos colaboradores das organizações de desenvolvimento de software (Santos *et al.*, 2015). Um fator comum entre a Gerência de Conhecimento (GC), Engenharia de Software e a Melhoria de Processo de Software é aprender com os fracassos e sucessos passados com o objetivo de melhorar futuras práticas do desenvolvimento de software (Bjørnson e Dingsøyr, 2008).

A aplicação de Aprendizagem Organizacional (AO) e a GC em relação ao que deve ser feito nas atividades dos processos de software e sua melhoria são fatores críticos em Engenharia de Software (Viana *et al.*, 2012; Montoni e Rocha, 2014). A GC é o método que simplifica o processo de captura, criação, distribuição, compartilhamento e entendimento do conhecimento de uma organização (Davenport e Prusak, 1998). A AO busca disseminar o conhecimento em organizações de desenvolvimento de software (Ruhe, 2001) através de uma série de interações entre a adaptação no nível individual (ou de subgrupo) e adaptação no nível organizacional (Bjørnson e Dingsøyr, 2008; Senge, 2014).

A falta de disseminação dos conhecimentos na organização pode ocasionar a perda de informações passadas. A transferência de conhecimento é um aspecto que pode influenciar os processos de software e sua melhoria em uma organização, pois é necessário que haja uma interação entre os colaboradores durante o MPS. Adicionalmente, é necessário que os conhecimentos sobre as práticas necessárias para executar essa melhoria sejam bem disseminados a fim de atingir o objetivo pretendido. Uma forma de disseminar o conhecimento e evitar sua perda durante os ciclos de melhoria de processos é aplicar estratégias de AO e GC (Nonaka e Takeuchi, 1995).

Uma organização que possui o conhecimento sobre qualidade de software distribuído entre os colaboradores, tem mais chances de manter as atividades do programa de melhoria (Almeida *et al.*, 2011). Desta forma, um dos grandes benefícios das iniciativas de Melhoria de Processo de Software é o suporte efetivo à gerência e aprendizagem dos conhecimentos (Dyba, 2005).

A melhoria dos processos de software é baseada em conhecimento humano, onde o nível de incerteza é muito elevado. Desta forma, é necessário analisar as diversas maneiras de abordagem da AO e GC voltadas para Engenharia de Software e Melhoria de Processo de Software visando diminuir os esforços adicionais empregados na execução das iniciativas de MPS. Desta maneira, a questão de pesquisa que norteia este trabalho é: *como facilitar a AO e a GC em Melhorias de Processo de Software?* Buscando responder esta questão, definiu-se o Framework KL-SPI (*Knowledge and Learning to facilitate Software Process Improvement*). Esse framework foi definido a partir da agregação de diversas evidências da literatura e da indústria de software. O

Framework contém três componentes principais: (a) Estratégia de diagnóstico do Estado da Prática de AO e GC nas organizações: essa estratégia busca verificar quais práticas de AO e GC estão sendo executadas na organização durante a melhoria dos processos de software. Essa estratégia contém um guia de como realizar o diagnóstico e uma ferramenta de apoio; (b) Catálogo de práticas de AO e GC: esse catálogo foi definido a partir da harmonização dos resultados identificados nas investigações da prática e no mapeamento sistemático através do modelo conceitual definido. O objetivo deste catálogo é apresentar um conjunto de práticas de AO e GC que podem ser aplicadas em organizações de software no contexto de MPS; e, (c) Ferramentas para auxiliar a AO em MPS: essas ferramentas foram definidas e/ou desenvolvidas no âmbito da pesquisa. Além disso, há ferramentas identificadas na literatura que podem ser utilizadas para apoio à execução das práticas de AO e GC nas organizações.

Além desta seção introdutória, o restante deste artigo está organizado da seguinte maneira: Seção 2 apresenta a teoria de AO e GC; Seção 3 descreve a metodologia seguida nesta pesquisa; Seção 4 apresenta os resultados obtidos com o Estudo do Estado da Arte e Estado da Prática. Seção 5 apresenta a definição do Framework KL-SPI e avaliação dos seus componentes. Por fim, a Seção 6 discute as considerações finais e possibilidades de trabalhos futuros desta pesquisa.

2. Aprendizagem Organizacional e Gerência do Conhecimento

Devido à grande competitividade das empresas, ao avanço das tecnologias e à necessidade de suprir o mercado em tempo hábil, o tratamento do conhecimento torna-se medida estratégica importante para se manter no mercado (Santos-Vijande *et al.*, 2012). Entre as atividades de GC, destaca-se (Schneider, 2009): coletar novos conhecimentos, tratar esses conhecimentos de forma que seja possível uma futura utilização, armazená-lo, disseminá-lo, e, por fim, disponibilizar estratégias de aplicação desses conhecimentos em novas situações.

Segundo Nonaka e Takeuchi (1995), existem dois tipos de conhecimento que precisam ser gerenciados: Tácito e Explícito. O Conhecimento Tácito é altamente pessoal, difícil de formalizá-lo e comunicá-lo aos outros através de palavras, números e sentenças escritas. Este tipo de conhecimento é encontrado na mentes dos colaboradores (Dingsoyr *et al.*, 2009). Por outro lado, o Conhecimento Explícito ou codificado é considerado transmissível em linguagem formal (ou semi-formal). O Conhecimento Explícito pode ser representado através de documentos, relatórios, base de dados e lições aprendidas. O conhecimento sempre se origina nas pessoas sendo criado através da interação entre o Conhecimento Tácito e o Explícito. Essa interação deu origem ao modelo SECI (*Socialization, Externalization, Combination e Internalization*), conforme apresentado na Figura 1. Cada processo deste modelo representa uma conversão de conhecimento. Inicialmente, é realizada uma socialização onde ocorre a troca de Conhecimento Tácito através de diálogo. Em seguida, esse conhecimento é externalizado por meio da escrita, além de ser combinado com outros conhecimentos através de vínculos entre os componentes do Conhecimento Explícito. O conhecimento externalizado é aplicado em alguma atividade de trabalho específica, neste momento ocorre o “aprender fazendo”. Por fim, verifica-se que há um campo de construção do conhecimento, onde a base de conhecimentos tácitos é enriquecida. Neste momento, o processo de aprendizagem se inicia novamente (Nonaka e Takeuchi, 1995).

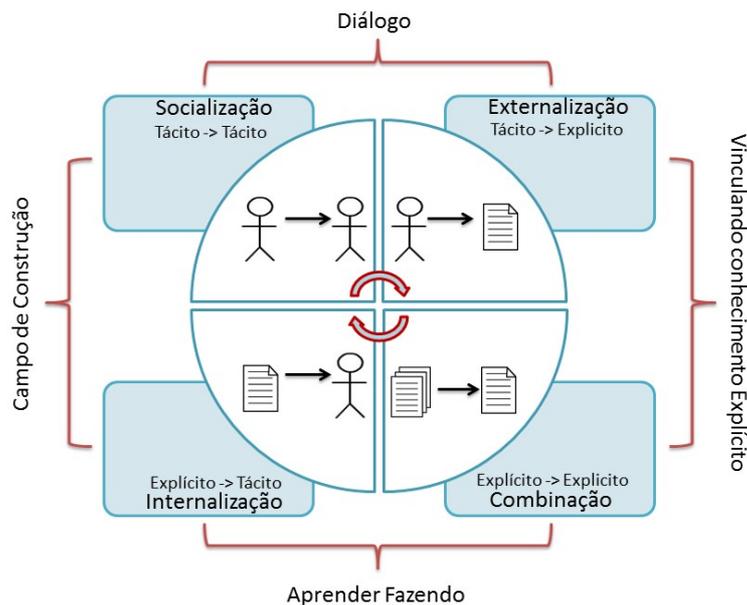


Figura 1. Modelo SECI e o Processo de Aprendizagem adaptado de Nonaka e Takeuchi (1995)

Em organizações, o conhecimento é encontrado na forma de rotinas, processos, práticas de trabalho, mentes de alguns colaboradores, normas organizacionais, assim como, documentos e repositórios. Esses conhecimentos precisam ser aprendidos pelos colaboradores das organizações de forma que as organizações obtenham sucesso na execução de suas atividades. Desta forma, verifica-se que somente gerenciar os tipos de conhecimento não é suficiente. É necessário que esse conhecimento seja aprendido a nível organizacional, de forma que agreguem sucesso às atividades de desenvolvimento de software realizadas.

A Aprendizagem permite que os membros das organizações possam construir uma interpretação e um entendimento organizacional do seu ambiente e comecem a avaliar estratégias viáveis para obter sucesso na execução das atividades (Menolli *et al.*, 2015). Um processo de aprendizagem pode ser gerado a partir da realização dos quatro processos do modelo SECI. Ao executar os quatro processos, tem-se a espiral do conhecimento (Schneider, 2009). Além disso, a Aprendizagem considera as experiências e conhecimentos passados, buscando não repetir os mesmos erros e evitando gastar recursos com o retrabalho.

A AO é uma abordagem que além de estimular a aprendizagem individual, deve criar uma coleção do conhecimento e incentivar a criação de uma infraestrutura para compartilhamento de conhecimento na organização como um todo (Schneider, 2009). A organização também deve prover um ambiente onde o trabalho em equipe e o interesse pelas pessoas deve ser privilegiado. Segundo Menolli *et al.* (2015), diversos pesquisadores apresentam diferentes definições de AO, contudo é importante ressaltar que todas estão voltadas a melhorar o conhecimento organizacional.

AO e a GC são visões complementares do processo de tratamento do conhecimento. A AO e GC se apoiam na tratativa do conhecimento da organização (Aurum *et al.*, 2008).

3. Desenvolvimento da Pesquisa

O desenvolvimento desta pesquisa foi realizado de acordo com as etapas que visavam agregar evidências da literatura com evidências da prática. Através das etapas apresentadas na Figura 2, pode-se acompanhar todo o desenvolvimento do Framework KL-SPI. Em seguida, os passos são detalhados.

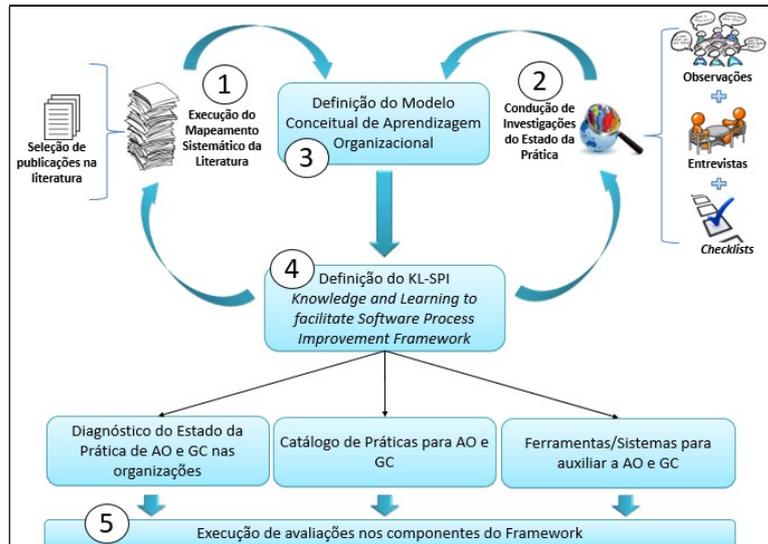


Figura 2. Passos seguidos durante a execução desta pesquisa.

1. Execução do Mapeamento Sistemático da Literatura – realizou-se uma caracterização da área de pesquisa. Analisaram-se os trabalhos disponíveis na literatura sobre AO e GC que auxiliaram na execução desta pesquisa e na definição do framework proposto;
2. Condução de Investigações do Estado da Prática – foram realizadas análises de aspectos da AO e GC em organizações de software que estavam inseridas no contexto de Melhoria de Processo de Software formal ou informal. Os resultados desses estudos apoiaram a definição dos componentes do framework;
3. Definição do Modelo Conceitual de Aprendizagem Organizacional – definiu-se um modelo conceitual a partir dos estudos de ontologias de organização e dos resultados das investigações do estado da arte e estado da prática. Em seguida, esse modelo conceitual guiou a harmonização dos resultados obtidos no mapeamento sistemático com os resultados das investigações da prática;
4. Definição do Framework KL-SPI – foram definidos e desenvolvidos os componentes do framework. Esse framework foi desenvolvido a partir do modelo conceitual definido para apoiar a aplicação das práticas de AO e GC. O Framework KL-SPI contém três componentes: (1) diagnóstico do estado da prática de AO e GC nas organizações – um processo foi definido para a realização desse diagnóstico; (2) catálogo de práticas de AO e GC – esse catálogo é um resultado obtido a partir das práticas identificadas na literatura e nas investigações; e, (3) ferramentas/sistemas para auxiliar AO e GC – esses apoios foram desenvolvidos durante a execução desta pesquisa e identificados no mapeamento sistemático da literatura;

5. Execução de avaliações nos componentes do Framework - foram conduzidas avaliações e análises para indicar a adequabilidade do Framework KL-SPI. Essas avaliações são importantes para que seja possível transferir o framework para a Indústria. Nesta etapa foram realizadas avaliações experimentais e avaliações de especialistas em determinados componentes do Framework.

4. Investigações do Estado da Arte e da Prática

Os dois primeiros passos desta pesquisa (Figura 2) foram realizados com o objetivo de obter evidências sobre a aplicação de AO e GC em Engenharia de Software e Melhoria de Processo de Software. O Estado da Arte foi identificado a partir do Mapeamento Sistemático da Literatura. O Estado da Prática foi identificado a partir da execução de Estudos de Caso. A seguir essas duas investigações são detalhadas.

4.1. Mapeamento Sistemático da Literatura

O mapeamento é uma revisão da literatura mais abrangente de estudos em uma área específica com o objetivo de identificar quais evidências estão disponíveis sobre um determinado tópico (Kitchenham *et al.*, 2011). Para guiar a execução do mapeamento sistemático desta pesquisa foram elaboradas duas questões de pesquisa: “Como é abordada a AO em empresas de desenvolvimento de software?” e “Como é abordada a AO em empresas de desenvolvimento de software inseridas no contexto de Melhoria de Processo de Software?”. Os resultados deste mapeamento sistemático são importantes para auxiliar na definição dos componentes do Framework KL-SPI.

De forma a caracterizar os principais aspectos dessas abordagens identificadas foram feitos modelos identificando possíveis fatores que podem facilitar a aplicação da abordagem e em que os fatores influenciam (como por exemplo, na aprendizagem e comunicação). Além disso, buscou-se identificar se a abordagem era baseada em outra abordagem previamente descrita na literatura e se ela possuía algum tipo de automação.

Como contribuição do mapeamento sistemático, tem-se a relação e análise de 35 abordagens voltadas para Engenharia de Software e Melhoria de Processo de Software. Algumas abordagens estão relacionadas a uma forma de desenvolvimento e podem necessitar de uma infraestrutura para poder ser utilizada. Por fim, verificou-se que parte dos trabalhos identificados se concentra em tratar o conhecimento ao invés de promover a aprendizagem pela organização.

Adicionalmente, esta análise de abordagens também permitiu a identificação de aspectos facilitadores da AO e GC. Entre esses fatores, pode-se destacar: ambiente de software com GC; criação de círculo de habilidades; colaboração de membros de uma equipe com outras; criação de seminários noturnos; documentação; equipes com diferentes funções; criação de grupos de interesse especial; incentivo a interação social; metodologias de desenvolvimento iterativas; programação em pares; criação de redes pessoais; *stand-up meetings*; comunicação.

Como mencionado na Seção 2, há dois tipos de conhecimento: conhecimento tácito e conhecimento explícito. As publicações foram analisadas no sentido de identificar qual o tipo de conhecimento estava sendo tratado. Buscou-se dividir as publicações de Engenharia de Software no geral e publicações que tratavam especificamente do contexto de MPS. Uma organização do resultado pode ser vista na

Figura 3. Em relação ao MPS, verifica-se que nenhum dos trabalhos identificados aborda de maneira específica o conhecimento tácito. De forma contrária, verifica-se que apenas seis trabalhos em MPS tratam de maneira específica o conhecimento explícito. Por fim, verificou-se que há 15 trabalhos em MPS que tratam de maneira conjunta o conhecimento tácito e explícito. Por outro lado, a maior dos trabalhos identificados na literatura, tratam do conhecimento em outros contextos da Engenharia de Software.

É importante ressaltar a quantidade de zeros na Figura 3. Isso ocorreu, pois analisou-se somente trabalhos que se enquadram nos conjuntos da Engenharia de Software e Melhoria de Processo de Software. Além disso, todos os trabalhos de Melhoria de Processo de Software são trabalhos de Engenharia de Software.

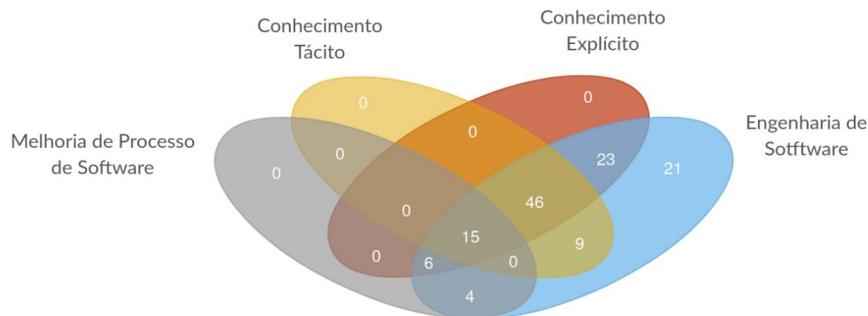


Figura 3. Diagrama de Venn sobre os resultados identificados nesta pesquisa.

Ao final da análise dos trabalhos obtidos pelo mapeamento sistemático verificou-se a falta de evidência experimental sobre a aplicação de AO e GC em ES/MPS. Desta forma, a realização de investigações da prática em organizações de software auxilia na criação dessas evidências e colabora na identificação de outras formas de aplicação da AO e GC no contexto real do desenvolvimento de software.

4.2. Investigações da Prática

As investigações da prática são estudos de caso executados devido à necessidade de analisar um fenômeno no seu contexto real (Runeson e Höst, 2009). Além disso, são importantes para esclarecer causas mais profundas por trás de um dado problema e suas consequências. Yin (2009) corrobora com a afirmação de que a execução de estudos de caso é relevante quando a questão de pesquisa exige uma descrição ampla e profunda de algum fenômeno social. Os estudos de caso permitem que os pesquisadores retenham as características holísticas e significativas dos eventos da vida real, como o comportamento de pequenos grupos, os processos organizacionais, as relações internacionais e a maturação das indústrias (Yin, 2009).

Os estudos de caso realizados foram exploratórios, isto é, são estudos de caso buscam explicações sobre um determinado evento ou problema. Para selecionar as três organizações que fizeram parte dos estudos de caso, buscou-se analisar a presença ou ausência de processo de software: (1) processo não documentado (há uma definição básica do processo de desenvolvimento nos projetos, contudo essa definição não é documentada); (2) processo documentado (o processo é documentado de acordo com as necessidades dos projetos e da organização, assim como seus ajustes e atividades de melhoria); e, (3) processo avaliado (o processo foi avaliado com base em modelos como: CMMI-DEV (Sei, 2010) e MR-MPS-SW (Softex, 2016)). O método de coleta de

dados adotado nesta pesquisa foi entrevista, onde utilizou-se questionários semiestruturados com questões abertas. Em relação ao método de análise de dados, utilizaram-se procedimentos de Grouded Theory (GT) (Strauss e Corbin, 1998).

Como contribuições dos estudos de caso, identificaram-se um conjunto de estímulos à aprendizagem que são ações com o objetivo de promover a aprendizagem do novo processo de software que continha as modificações necessárias ao programa de MPS. Entre as ações, tem-se: treinamentos, semana da qualidade/conformidade, execução de projeto piloto, *mentoring*, premiação de projetos conformes, integração entre os colaboradores, aprender na prática. Além disso, identificou-se também um conjunto de fontes de conhecimento da organização, como: wiki do processo, ferramenta de gerência de projetos (GP), colaboradores experientes, internet, intranet, código fonte dos projetos, blog/fórum organizacional e lições aprendidas.

Além desses resultados, também foi possível verificar aspectos de transferência do conhecimento, isto é, como o conhecimento é disseminado entre os colaboradores da organização através da análise de redes sociais. Outro aspecto observado foi sobre aprendizagem onde os membros das organizações relataram que o conhecimento prévio em melhoria de processo auxiliou na condução das atividades do programa de melhoria. Os resultados destas investigações da prática foram publicados em (Viana *et al.*, 2013; Viana *et al.*, 2014a; Viana *et al.*, 2014b; Viana *et al.*, 2015).

5. Definição do Framework KL-SPI

Os resultados do mapeamento sistemático da literatura apresentaram importantes aspectos de AO e GC para Engenharia de Software e MPS, como as práticas organizacionais e estratégias voltadas para AO e GC. Em seguida, as investigações do estado da prática de AO e GC em organizações de software identificaram como as organizações lidam com questões importantes do Conhecimento Organizacional para o sucesso da execução e melhoria dos processos/projetos de software. Contudo, são resultados que estão disponibilizados de maneira separada. Faz-se necessário harmonizar esses resultados de forma que eles possam ser úteis para outras organizações de software. Inicialmente, definiu-se um modelo conceitual que relaciona os resultados do mapeamento sistemático da literatura e das investigações da prática com aspectos relevantes da AO e GC para organizações de software. Em seguida, foram definidos os componentes do Framework KL-SPI. As subseções a seguir apresentam a definição da proposta desta pesquisa.

5.1. Modelo Conceitual

Os modelos conceituais buscam definir características e conceitos relevantes sobre determinado tema. Assim como modelos conceituais, as ontologias buscam fornecer uma descrição e entendimento sobre um universo que modelam (Villela, 2004). Para a definição do modelo conceitual de AO em Melhoria de Processo de Software, partiu-se inicialmente de uma ontologia de organização definida por (Villela, 2004). Esta ontologia de organização foi escolhida devido a presença de componentes voltados à organização de software.

A ontologia de organização possui subontologias voltadas a descrever aspectos do processo de software e da organização. Nesta presente pesquisa, definiu-se uma

subontologia de Conhecimento Organizacional que possui relacionamentos com os componentes das subontologias já definidas por (Vilela, 2004).

Para a construção da subontologia de Conhecimento Organizacional, foram utilizados conceitos da literatura voltados para AO e GC, além de terem sido considerados aspectos da Gerência de Conhecimento para o MPS.BR (Softex, 2016) e padrões de abordagens de AO e GC identificados no mapeamento sistemático da literatura. A Figura 4 apresenta a subontologia de Conhecimento Organizacional. Em seguida os conceitos relacionados são detalhados.

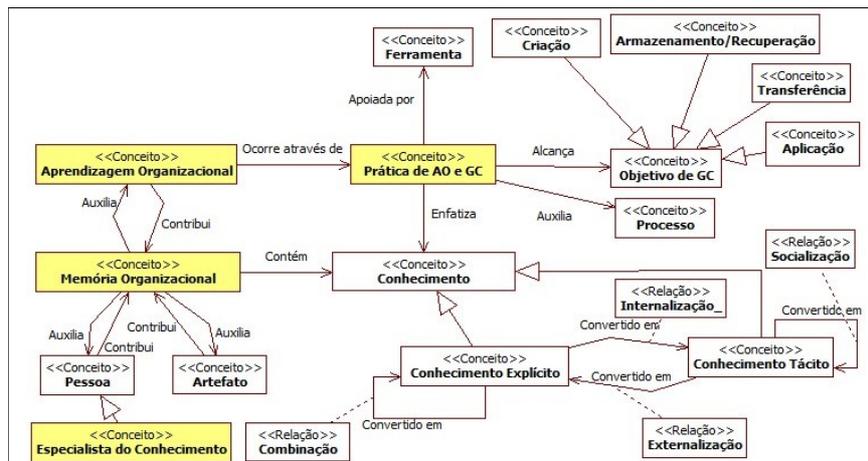


Figura 4. Subontologia de Conhecimento Organizacional.

Aprendizagem Organizacional: conceito que descreve a aprendizagem de indivíduos e grupos de pessoas em uma organização. A aprendizagem pode ocorrer através de mecanismos de AO e GC (Ruhe, 2001). Esses mecanismos são Práticas de AO e GC ou atividades da organização de software que buscam alcançar objetivos de GC. Os Objetivos de GC são aspectos que uma prática de GC visa alcançar, como: criação do conhecimento; armazenamento/recuperação do conhecimento, transferência e aplicação. Esses objetivos de GC foram baseados na definição de Alavi e Leidner (2001).

Práticas de AO e GC: enfatizam um dos tipos de conhecimento, o Conhecimento Tácito e/ou Conhecimento Explícito e o relacionamento entre esses conhecimentos. Para representar esse relacionamento, utilizou-se o ciclo de criação do conhecimento e processo de aprendizagem de Nonaka e Takeuchi (1995), o SECI. Além disso, as práticas de AO e GC podem ser apoiadas por Ferramentas, isto é, algum apoio ferramental que auxilie a execução da prática de AO e GC, como por exemplo, uma Wiki. Por fim, uma prática pode auxiliar a execução de um processo de software.

Memória Organizacional: pode auxiliar a aplicação sistemática de princípios de AO (Althoff *et al.*, 2000). A memória organizacional constitui uma base de conhecimento da organização, que pode conter tanto Conhecimento tácito quanto explícito. Novos conhecimentos para a Memória Organizacional são gerados a partir da Aprendizagem Organizacional. Essa Memória Organizacional é composta pelo conhecimento das Pessoas da organização e pelo conhecimento dos Artefatos da organização. Além disso, a Memória Organizacional auxilia às Pessoas durante a execução das atividades e auxilia com a criação dos Artefatos da organização. As Pessoas são os colaboradores

que podem ou não ser especialistas em um determinado conhecimento da organização. Por fim, a memória organizacional contém os artefatos necessários à execução dos processos de software.

Especialista do Conhecimento: conceito que descreve os colaboradores da organização que detêm o domínio de determinado tipo de conhecimento da organizacional (Softex, 2016). O conhecimento detido pelos especialistas também faz parte da memória organizacional. Os especialistas do conhecimento podem deter o conhecimento tanto de itens quanto de objetos de conhecimento.

Analisados os resultados específicos de GC no processo Gerência de Recursos Humanos do Nível E:MR-MPS-SW (Softex, 2016) com a subontologia definida, verifica-se que: (1) o primeiro resultado esperado é definido pelos conceitos de estratégia de GC e apoio ferramental, uma vez que essas estratégias podem conter fases que auxiliam a execução da GC na organização; (2) o segundo resultado esperado pode ser definido pelo conceito de especialista de conhecimento e de suas relações com as demais subontologias da ontologia de organização definida por Villela (2004). Além disso, esse resultado está diretamente relacionado à segunda questão de competência definida para a subontologia de Conhecimento Organizacional, além de auxiliar a evitar os problemas identificados nas planilhas de avaliações; e, (3) o terceiro resultado esperado pode ser definido através dos conceitos de memória organizacional e aprendizagem do conhecimento.

Para analisar a aplicabilidade desta subontologia como um modelo conceitual para a pesquisa também se realizou uma nova investigação da prática. O objetivo dessa nova investigação era identificar as práticas de AO e GC e representá-las de acordo com o modelo conceitual estabelecido. Foram identificadas onze práticas de AO e GC e essas práticas foram categorizadas de acordo com os componentes do modelo conceitual.

Após analisar a adequabilidade do modelo conceitual, buscou-se reanalisar as três investigações da prática realizadas (Seção 4.2) de forma que os resultados fossem adaptados para os componentes do modelo conceitual proposto nesta pesquisa. Após analisar os resultados, outro pesquisador verificou as relações estabelecidas de forma a avaliar os resultados gerados.

Por fim, os resultados do mapeamento sistemático também foram analisados a luz do modelo conceitual. Nem todas as publicações possuíam claramente práticas de AO e GC. Além disso, algumas publicações apenas descreviam abordagens. Para selecionar as práticas buscou-se verificar se a publicação apresentava atividades organizacionais que auxiliavam alcançar algum objetivo de gerência de conhecimento ou tratava o Conhecimento Tácito e/ou Explícito de alguma forma. Foram selecionadas 29 práticas da literatura. Um pesquisador convidado analisou as práticas vindas do mapeamento sistemático da literatura (diferente do pesquisador que avaliou as práticas vindas das investigações). Nesta análise foi verificada a adequação das práticas com os seus componentes, principalmente relacionados aos objetivos de GC e modelo SECI.

5.2. Framework KL-SPI

Apesar do modelo conceitual apresentado na Seção 5.1 buscar uma harmonização dos resultados das investigações da prática e da investigação do estado da arte é necessário desenvolver mecanismos que utilizem os resultados identificados nesta pesquisa como

forma de auxiliar as organizações em questões relacionadas à AO e GC. Desta forma, foi definido o Framework KL-SPI. Esse framework possui componentes que auxiliam a identificar gaps relacionados à AO e GC, sugerir práticas de AO e GC para serem executadas e sugerir ferramentas de apoio. Os componentes são descritos a seguir:

Estratégia de diagnóstico do Estado da Prática de AO e GC nas organizações: essa estratégia busca verificar quais práticas de AO e GC estão sendo executadas na organização durante a melhoria dos processos de software. Há um processo de como realizar o diagnóstico e uma ferramenta de apoio, KL-SPI *Diagnosis Tool*. As atividades do diagnóstico são: (1) identificar práticas relacionadas à AO e GC; (2) elaborar o diagnóstico das práticas de AO e GC para a organização; e, (3) sugerir práticas de AO e GC para a organização. Cada atividade foi decomposta em tarefas para que o objetivo principal fosse alcançado. Para cada tarefa foi criada uma especificação em relação aos critérios de entrada e saída, quem é o responsável, quem participa da atividade, quais os artefatos são requeridos e produzidos, ferramentas de apoio à execução da tarefa e quais são as pré-tarefas e pós-tarefas.

Catálogo de práticas de AO e GC: esse catálogo foi definido a partir da harmonização dos resultados identificados nas investigações da prática e no mapeamento sistemático através do modelo conceitual definido. Para seleção das práticas levou-se em consideração somente as práticas que possuem evidência experimental. O objetivo deste catálogo é apresentar um conjunto de práticas de AO e GC que podem ser aplicadas em organizações de software no contexto de Melhoria de Processo de Software. Ao final da pesquisa foram identificadas 33 práticas de AO e GC. A Tabela 1 apresenta uma parte das práticas identificadas.

Tabela 1. Exemplos de Práticas de AO e GC identificadas na pesquisa.

#	Práticas de AO e GC	Lit.	Inv.
1	Paralelismo de tecnologias novas e antigas em determinadas situações	X	
2	Atuação de especialistas da organização	X	X
3	Execução de treinamentos	X	X
4	Execução de Projeto Piloto	X	X
5	Utilização de ferramentas de comunicação escrita	X	X
6	Utilização de espaços de comunicação formal e informal entre as equipes	X	X
7	Utilização de <i>yellow-pages</i> de especialista	X	
8	Intercâmbio de colaboradores entre equipes	X	
9	Comunicação verbal entre colaboradores da organização	X	X
10	Utilização da Intranet	X	X

Ferramentas para auxiliar a Aprendizagem Organizacional em MPS: essas ferramentas foram definidas e/ou desenvolvidas no âmbito da pesquisa. Ao analisar a necessidade de apoios ferramentais através dos resultados do mapeamento sistemático da literatura, escolheram-se os seguintes requisitos: (1) apoios para externalização do conhecimento de software. Essa externalização é necessária para capturar os diversos conhecimentos dos colaboradores e torná-los disponíveis para outros colaboradores. Esse apoio deve ser independente de tecnologia; e, (2) ferramentas que apoiem a discussão de conhecimento de forma a realizar o compartilhamento e aplicação do conhecimento. Essas ferramentas são importantes para estimular o compartilhamento de conhecimento entre os colaboradores da organização. Para suprir os dois requisitos,

buscou-se definir abordagens que alcançassem esses objetivos. Desta forma, as abordagens identificadas foram:

- *PABC-Pattern* (Problema, Ação, Benefício, Contexto – Padrão): visa facilitar a codificação de lições aprendidas em organizações de software (Rabelo *et al.*, 2012; Rabelo *et al.*, 2014);
- *Knowledge Challenge*: Conjunto de regras para incentivar o compartilhamento de conhecimento dentro de organizações de software.

Essas abordagens desenvolvidas podem ser empregadas conforme a necessidade da organização. Além disso, também é possível verificar a aplicabilidade das abordagens já apresentadas pela literatura (ver Seção 4.1). Para isso, deve-se analisar o resultado do processo de diagnóstico. Com este resultado é possível identificar quais apoios podem ser empregados.

5.3. Avaliação do Framework KL-SPI

Foram conduzidas avaliações e análises para indicar a adequabilidade do Framework KL-SPI. Essas avaliações são importantes para que seja possível transferir o framework para ser utilizado pelas organizações de software. Foram realizadas as seguintes avaliações:

- Verificação do processo de diagnóstico por especialista em MPS: esta verificação auxiliou na evolução do processo de diagnóstico. Para escolha do especialista, buscou-se alguém com conhecimento em implementação e avaliação de processos de software. O feedback do especialista identificou pontos que necessitavam de mais esclarecimento de modo que qualquer profissional pudesse utilizar o processo de diagnóstico proposto. Entre os pontos identificados, destacam-se: criação de tarefas para obter o contexto da MPS da organização, uma vez que ela pode ser baseada em modelos de maturidade ou de acordo com as necessidades organizacionais; criação de tarefas para analisar os dados obtidos para serem utilizados no diagnóstico, pois alguns colaboradores podem deixar de indiciar especialistas, por exemplo;
- Estudo comparativo de práticas: Como forma de checar as práticas identificadas nesta pesquisa, realizou-se um estudo comparativo das práticas desta pesquisa com práticas já reportadas em Santos (2013). Neste trabalho, Santos (2013) descreve as práticas mais adotadas para compartilhamento de conhecimento. Escolheu-se este trabalho, pois as práticas identificadas foram avaliadas por um consultor da indústria que emprega aspectos de gerência de conhecimento em suas atividades. O resultado deste estudo mostrou que que 88,88% das práticas identificadas por Santos (2013) foram cobertas por práticas identificadas nesta pesquisa. Somente uma prática descrita por Santos (2013) não foi coberta por alguma prática desta pesquisa, o *Coding Dojo* que é uma prática aplicada por metodologias ágeis. Por fim, cabe ressaltar que foram identificadas, nesta pesquisa, 17 práticas além das reportadas por Santos (2013);
- Análise das tarefas do processo de diagnóstico relacionadas à identificação de redes sociais: foi conduzido um estudo para verificar como ocorre a disseminação do conhecimento sobre Melhoria de Processo de Software através da análise de redes

sociais. Neste estudo foi possível analisar algumas tarefas do processo de diagnóstico. Os resultados deste estudo foram publicados em Viana *et al.* (2014b). A execução dessas tarefas auxiliou na identificação de padrões de interação entre colaboradores. Esses padrões são importantes para auxiliar na transferência e aprendizagem dos conhecimentos sobre MPS;

- Análise de práticas de AO e GC por profissional da indústria que trabalha com GC: o objetivo desta análise foi verificar a concordância entre definição das práticas AO e GC do KL-SPI e a definição independente do profissional utilizando o SECI e objetivos de GC. Em relação ao modelo SECI, verificou-se que houve uma concordância completa e parcial de 71% das práticas. Enquanto em relação aos objetivos de GC, observou-se uma concordância completa e parcial de 88%. Por concordância completa considerou-se que tanto o KL-SPI quanto o profissional identificaram os mesmos componentes.

6. Considerações Finais e Trabalhos Futuros

Iniciativas de Melhoria de Processo de Software normalmente necessitam de aplicação de uma grande quantidade de conhecimentos novos na organização. Pelo fato da criação de conhecimento andar em paralelo à aprendizagem, é necessário garantir que o conhecimento seja gerenciado e aprendido por toda a organização para que se obtenha sucesso neste tipo de iniciativas. No entanto, não há uma forma padronizada para que esses conhecimentos sejam aprendidos pelas organizações. Portanto, deve-se considerar aspectos específicos da forma de trabalho em determinados contextos.

Esta pesquisa apresentou a concepção do Framework KL-SPI, cujo objetivo é facilitar a AO em MPS. Esse framework contém componentes que auxiliam a aplicação e o desenvolvimento de AO e GC nas organizações. São eles: (1) Definição do processo de diagnóstico do Estado da Prática em AO e GC nas organizações de software. Esse processo descreve os passos necessários para identificar as práticas correntes de AO e GC e atividades de sugestão de novas práticas que podem ser aplicadas; (2) Elaboração do catálogo de práticas de AO e GC. Este catálogo contém uma lista de práticas que pode ser utilizada como apoio ao diagnóstico do estado da prática, além de auxiliar na indicação de novas práticas para a organização; (3) Especificação, desenvolvimento e identificação de ferramentas de apoio à utilização do framework e apoio à execução de práticas de AO e GC. Durante a execução desta pesquisa foram desenvolvidos: KL-SPI Diagnosis Tool, PABC-Pattern e Knowledge Challenge. Além da análise de outras abordagens identificadas na Literatura.

Outra contribuição da pesquisa foi a definição de um Modelo Conceitual sobre AO e GC para MPS que foi definido a partir da extensão de uma ontologia de organização de software. Este modelo conceitual pode ser utilizado para harmonizar os resultados identificados na literatura e em investigações da prática.

Atualmente todo o processo de diagnóstico do KL-SPI está sendo empregado em uma nova investigação da prática para analisar sua aplicabilidade. Os resultados obtidos com esta pesquisa abrem novas perspectivas de pesquisa que podem ser exploradas em trabalhos futuros, como a investigação do relacionamento dos objetivos de GC com as etapas do modelo SECI e evolução do catálogo de práticas AO e GC do KL-SPI.

Agradecimentos

Agradecemos aos colaboradores das organizações participantes da pesquisa e a FAPEAM pelo apoio financeiro. Um agradecimento mais que especial à professora Emilia Mendes pelo apoio em toda a pesquisa e aos professores Ana Regina, Gleison Santos e Monalesa Barcellos pela avaliação da pesquisa.

Referências

- Alavi, M., Leidner, D.E., 2001, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues", *MIS quarterly*, pp. 107-136.
- Almeida, C.D.A., Albuquerque, A.B., Macedo, T.C., 2011, "Analysis of the continuity of software processes execution in software organizations assessed in MPS.BR using Grounded Theory". In: *Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*, pp. 792-797, Miami, USA.
- Althoff, K.-D., Bomarius, F., Tautz, C., 2000, "Knowledge Management for Building Learning Software Organizations", *Information Systems Frontiers*, v. 2, n. 3-4 (2000/10/01), pp. 349-367.
- Aurum, A., Daneshgar, F., Ward, J., 2008, "Investigating Knowledge Management practices in software development organisations – An Australian experience", *Information and Software Technology*, v. 50, n. 6 (5//), pp. 511-533.
- Bjørnson, F.O., Dingsøyr, T., 2008, "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used", *Information and Software Technology*, v. 50, n. 11 (10//), pp. 1055-1068.
- Davenport, T., Prusak, L., 1998, *Working knowledge: How organizations manage what they know*.
- Dingsøyr, T., Bjørnson, F.O., Shull, F., 2009, "What Do We Know about Knowledge Management? Practical Implications for Software Engineering", *Software, IEEE*, v. 26, n. 3, pp. 100-103.
- Dyba, T., 2005, "An empirical investigation of the key factors for success in software process improvement", *Soft. Eng., IEEE Transactions on*, v. 31, n. 5, pp. 410-424.
- Kitchenham, B.A., Budgen, D., Pearl Brereton, O., 2011, "Using mapping studies as the basis for further research – A participant-observer case study", *Information and Software Technology*, v. 53, n. 6 (6//), pp. 638-651.
- Menolli, A., Cunha, M.A., Reinehr, S., *et al.*, 2015, "'Old' theories, 'New' technologies: Understanding knowledge sharing and learning in Brazilian software development companies", *Information and Software Technology*, v. 58, pp. 289-303.
- Montoni, M.A., Rocha, A.R.C., 2014, "Applying grounded theory to understand software process improvement implementation: a study of Brazilian software organizations", *Innovations in Systems and Software Engineering*, v. 10, n. 1, pp. 33-40.
- Nonaka, I., Takeuchi, H., 1995, *The Knowledge-Creating Company*, 17th ed. Oxford Oxford University Press.
- Rabelo, J., Viana, D., Conte, T., *et al.*, 2012, "Comparing Knowledge Codification Approaches: An Empirical Study". In: *Proceedings of 2012 Brazilian Symposium on Collaborative Systems (SBSC)*, , pp. 136-145.
- Rabelo, J., Viana, D., Santos, G., *et al.*, 2014, "Usando o PABC-Pattern para Codificar o Conhecimento: Um Estudo Experimental". In: *Proceedings of XIII Simpósio Brasileiro de Qualidade de Software (SBQS 2014)*, v. 1, pp. 1 - 15, Blumenau - SC.

- Ruhe, G., 2001, "Learning Software Organisations", *Handbook of Software Engineering and Knowledge Engineering* (S.K. Chang, ed.), World Scientific Publishing 2001.
- Runeson, P., Höst, M., 2009, "Guidelines for conducting and reporting case study research in software engineering", *Emp. Soft. Engineering*, v. 14, n. 2, pp. 131-164.
- Santos-Vijande, M.L., López-Sánchez, J.Á., Trespalcios, J.A., 2012, "How organizational learning affects a firm's flexibility, competitive strategy, and performance", *Journal of Business Research*, v. 65, n. 8, pp. 1079-1089.
- Santos, G., 2011, "Influência e Impacto do Programa MPS.BR na Pesquisa Relacionada à Qualidade de Software no Brasil". In: *Proceedings of X Simpósio Brasileiro de Qualidade de Software (SBQS 2011)*, v. 1, pp. 73-87, Curitiba, PR.
- Santos, V., 2013, *Compartilhamento de conhecimento inter-equipes: O processo de socialização do conhecimento tácito em contextos ágeis de desenvolvimento de software*, Tese de Doutorado. Universidade de São Paulo. 197p.
- Santos, V., Goldman, A., De Souza, C.R., 2015, "Fostering effective inter-team knowledge sharing in agile software development", *Empirical Software Engineering*, v. 20, n. 4, pp. 1006-1051.
- Schneider, K., 2009, *Experience and Knowledge Management in Software Engineering* Heidelberg, Springer
- Sei, 2010, *CMMI® for Development, Version 1.3, Improving processes for developing better products and services*, Technical Report CMU/SEI-2010-TR-033. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2010. .
- Senge, P.M., 2014, *The fifth discipline fieldbook: Strategies and tools for building a learning organization*, Crown Business.
- Softex, 2016, *MPS.BR: Guia Geral MPS de Software*, Disponível em: <http://www.softex.br/mpsbr/guias/>. Acessado em 29 de Abril de 2016.
- Strauss, A., Corbin, J., 1998, *Basics of qualitative research: Techniques and procedures for developing grounded theory*, Sage.
- Viana, D., Conte, T., Marczak, S., et al., 2015, "Knowledge Creation and Loss within a Software Organization: An Exploratory Case Study". In: *48th Hawaii International Conference on System Sciences (HICSS 2015)*, pp. 3980 - 3989, Hawaii.
- Viana, D., Conte, T., Souza, C.D., 2014a, "Knowledge Transfer between Senior and Novice Software Engineers: A Qualitative Analysis". In: *Proceedings of the 26th SEKE*, Vancouver, Canada.
- Viana, D., Conte, T., Vilela, D., et al., 2012, "The influence of human aspects on software process improvement: Qualitative research findings and comparison to previous studies". In: *EASE 2012, on*, pp. 121-125, 14-15 May 2012.
- Viana, D., Rabelo, J., Conte, T., et al., 2013, "A qualitative study about the life cycle of lessons learned". In: *6th CHASE - ICSE 2013 Workshop*, pp. 73-76, San Francisco, United States.
- Viana, D., Souza, C.D., Cabral, R., et al., 2014b, "Usando Análise de Redes Sociais para Investigar a Disseminação do Conhecimento em Melhorias de Processos de Software". In: *XIII Simpósio Brasileiro de Qualidade de Software*, pp. 179 - 193, Blumenau - SC.
- Villela, K., 2004, *Definição e Construção de Ambientes de Desenvolvimento de Software Orientados à Organização*, Tese de D. Sc., COPPE, UFRJ, Rio de Janeiro, Brasil.
- Yin, R., 2009, *Case Study Research: Design and Methods (Applied Social Research Methods)* Beverly Hills, Sage Publications.