

Um estudo de métodos ágeis sob a ótica do modelo do espiral do conhecimento

Cristiano Tolfo (UNIPAMPA/UFSC) cristianotolfo@unipampa.edu.br
Marcelo Gitirana Gomes Ferreira (UFSC) marcelo.gitirana@gmail.com
Fernando Antonio Forcellini (UFSC) forcellini@deps.ufsc.br
Cátia dos Reis Machado (UFSC) catia@egc.ufsc.br

Resumo:

Os métodos ágeis de desenvolvimento de software são caracterizados por princípios, valores e práticas que primam pela intensa interação entre pessoas e pela constante produção de conhecimento em pequenos ciclos de tempo. Tendo em vista que esta sistemática de trabalho normalmente requer uma conjunta gestão do conhecimento produzido pela equipe, este artigo apresenta uma análise de aspectos da gestão do conhecimento sob a ótica a adoção de práticas ágeis. Em uma revisão de literatura sobre métodos ágeis e gestão do conhecimento, elegeram-se o modelo do espiral do conhecimento para visualizar a criação de conhecimento no uso de práticas ágeis. Estudos preliminares realizados obtidos em um ensaio teórico permitem visualizar a relação entre algumas das práticas ágeis os ciclos que envolvem etapas de socialização, externalização, combinação e internalização do conhecimento por parte dos membros da equipe de software.

Palavras chave: Desenvolvimento de software, Métodos ágeis, Gestão do conhecimento, Modelo espiral de gestão do conhecimento.

A study of agile methods in light of the knowledge spiral model.

Abstract

Agile methods of software development are characterized by principles, values and practices that emphasize the intense interaction between people and the constant production of knowledge in small time cycles. Since this systematic work usually requires a joint management of the knowledge produced by the team, this article presents an analysis of aspects of knowledge management from the perspective the adoption of agile practices. In a review of the literature on agile methods and knowledge management, was elected as the knowledge spiral model to see the creation of knowledge in the use of agile practices. Preliminary studies obtained in a theoretical paper to visualize the relationship between some of the agile practices that involve cycles stages of socialization, externalization, combination and internalization of knowledge by members of the software team.

Key-words: Software development, Agile methods, knowledge management, knowledge spiral model.

1. Introdução

Diante da sociedade da informação, onde termos como a era da informação (Abreu, 2000; Castells, 2003), e a era do conhecimento (Toffler, 1990; Drucker, 1991; Quinn, 1992; Fialho, 2006) estão cada vez mais presentes, a criação e a gestão do conhecimento está cada vez mais presente nos sistemas produtivos. Na área de produção de software Cockburn (2002),

Demarco e Lister (1987), Poppendieck e Poppendieck (2003) e Sliger e Broderick (2008) são alguns dos autores que assemelham os profissionais do desenvolvimento de software a trabalhadores do conhecimento, termo este cunhado por Drucker (1967). Percebe-se consenso entre estes autores de que diferentemente dos trabalhadores manuais, que devem seguir os passos de um planejamento a eles determinados, os profissionais da engenharia de software estão envolvidos com tarefas intensivas em conhecimento que requerem considerável autonomia e capacidade de autogestão, abstração e criação.

Em decorrência do processo evolutivo na concepção de sistemas, do aprendizado oriundo da engenharia de software e inclusive de ensinamentos validados em outras áreas de produção, emergiu durante a década de 90 um modelo mental na indústria de software, trata-se de uma linha de pensamento que busca aumentar os índices de qualidade e produtividade enfatizando o lado humano deste setor produtivo.

Os profissionais da área de software adeptos desse modelo mental passaram a experimentar formas alternativas de criar e gerir conhecimento. No ano de 2001, os mesmos reuniram-se para discutir suas experiências vivenciadas, sendo que se definiu um conjunto de valores e princípios que delinearão o que ficou conhecido como Manifesto ágil (AGILE, 2001).

A propagação da filosofia ágil e o aumento de experiências na adoção de métodos ágeis reforçaram a premissa de que diferentemente de processos de desenvolvimento de software mais técnico prescritivos, as abordagens ágeis focam a explicitação de conhecimento implícito por meio da socialização do trabalho em equipe e interação entre pessoas. Além do indispensável conhecimento técnico, aspectos humanos, comportamentais e de liderança são decisivos para assegurar a qualidade e a produtividade no desenvolvimento ágil.

Tendo em vista que esta sistemática de trabalho proposta pelos métodos ágeis, normalmente, requer uma conjunta gestão do conhecimento produzido pela equipe, este artigo apresenta uma análise de aspectos da gestão do conhecimento sob a ótica a adoção de práticas ágeis. Em uma revisão de literatura sobre métodos ágeis e gestão do conhecimento, elegeu-se o modelo do espiral do conhecimento para visualizar a criação de conhecimento no uso de práticas ágeis. Estudos preliminares realizados obtidos em um ensaio teórico permitem visualizar a relação entre algumas das práticas ágeis os ciclos que envolvem etapas de socialização, externalização, combinação e internalização do conhecimento por parte dos membros da equipe de software.

2. O modelo do espiral do conhecimento

Nonaka & Takeuchi (1997) explicam o processo de criação do conhecimento nas empresas fazendo uso de duas dimensões nas quais o conhecimento pode ser analisado: a dimensão epistemológica e a dimensão ontológica.

Na dimensão epistemológica, pode-se distinguir o conteúdo tácito do conhecimento do seu conteúdo explícito (Polanyi apud Nonaka & Takeuchi, 1997). O conhecimento tácito é altamente pessoal e mais difícil de ser formalizado, o que dificulta a sua transmissão e o seu compartilhamento. O conhecimento tácito está profundamente enraizado nas ações e nas experiências dos indivíduos, bem como nas suas emoções, valores e ideais. *Insights* e palpites subjetivos podem ser incluídos nessa categoria de conhecimento. O conhecimento tácito pode ser dividido em duas outras dimensões: a técnica e a cognitiva. A dimensão técnica do conhecimento tácito se refere a um tipo de capacidade informal e difícil de definir, abrangendo as habilidades usualmente expressas pelo termo *know-how*. A dimensão cognitiva consiste em esquemas, modelos mentais, crenças e percepções, que apesar de não poderem ser facilmente articulados, moldam a forma como que as pessoas percebem o mundo à sua volta.

O conhecimento explícito, por sua vez, é aquele que pode ser transmitido através de uma

linguagem formal e sistemática. Tal conhecimento pode ser expresso por palavras e números e facilmente comunicado e compartilhado sob a forma de dados brutos, fórmulas científicas, procedimentos codificados ou princípios universais. A maior parte do conhecimento das pessoas é de natureza tácita, pois estas adquirem tal conhecimento através da ativa criação e organização das suas experiências.

Para Nonaka & Takeuchi (1997), o conhecimento tácito e o explícito não se apresentam como entidades isoladas, mas sim complementares. Eles interagem e se mesclam nas atividades criativas dos seres humanos. Tal modelo dinâmico da criação do conhecimento está ancorado na suposição de que o conhecimento humano é criado e expandido através da interação social entre o conhecimento tácito e o conhecimento explícito. Tal interação é denominada “conversão do conhecimento”. Já tal suposição permite postular os quatro modos de conversão de conhecimento ilustrados na Figura 1.



Figura 1 – Modos de conversão do conhecimento. Adaptado de Nonaka & Takeuchi (1997).

Socialização (de tácito para tácito). É o processo de aquisição de conhecimentos tácitos (modelos mentais ou know-how) a partir do compartilhamento de experiências. Dispensa o uso da linguagem. Um aprendiz trabalha com o seu mestre e aprende uma habilidade não através da linguagem, mas sim através da observação, imitação e prática. Nas empresas, o treinamento “on the job” se utiliza basicamente do mesmo princípio. A chave para a aquisição do conhecimento tácito é a experiência. Sem alguma forma de experiência compartilhada é extremamente difícil para uma pessoa se projetar para dentro do processo de pensamento de outra pessoa. A mera transferência de informações frequentemente fará pouco sentido se abstraída das emoções associadas e o do contexto no qual estão inseridas. A socialização também ocorre pela da interação entre os membros de uma equipe de desenvolvimento de um produto e os usuários do mesmo.

Externalização (de tácito para explícito). É o processo de articulação do conhecimento tácito em conceitos explícitos. Dentre os quatro modos de conversão do conhecimento, a externalização é a chave para a criação do conhecimento, pois cria novos conceitos, explícitos, a partir do conhecimento tácito. A externalização é comumente ativada pela por diálogos e pela reflexão coletiva. A dedução e a indução são métodos utilizados no processo de criação de conceitos. Porém, quando não se pode uma expressão adequada para uma imagem através da dedução ou da indução, deve-se utilizar um método não analítico. A externalização pode então ser guiada por metáforas e analogias. A utilização de metáforas e analogias é altamente efetiva na promoção do comprometimento com o processo criativo.

Combinação (de explícito para explícito). É o processo de organização de conceitos em um

sistema de conhecimentos. Envolve a combinação de diferentes corpos de conhecimentos explícitos. Trocas de conhecimentos por meio de documentos, reuniões, conversas telefônicas ou pela Internet são exemplos de combinação. A reconfiguração de informações existentes através do ordenamento, da adição, da combinação e da categorização de conhecimentos explícitos – tal como feito em bancos de dados eletrônicos – também leva a novos conhecimentos.

Internalização (de explícito para implícito). É o processo de incorporação de conhecimentos explícitos em conhecimentos tácitos. Quando experiências são internalizadas sob a forma modelos mentais compartilhados ou know-how técnico, eles tornam-se recursos valiosos.

A dimensão ontológica do conhecimento se refere aos níveis das entidades criadoras de conhecimento: individual, grupal, organizacional, interorganizacional. O conhecimento só pode ser gerado por indivíduos. Uma organização não pode gerar conhecimento sem os seus indivíduos. A organização suporta os indivíduos criativos ou provê um contexto onde estes geram conhecimento.

A socialização objetiva o compartilhamento de conhecimentos tácitos. A menos que os conhecimentos tácitos tornem-se explícitos, estes não podem ser facilmente utilizados pela organização como um todo. Também a mera combinação de informações explícitas não amplia a base de conhecimento da organização. Entretanto, quando os conhecimentos tácitos e explícitos integrarem, surge a inovação. A criação do conhecimento organizacional se constitui desse modo, de uma interação contínua e dinâmica entre o conhecimento tácito e o dinâmico. Esta interação é conformada pela alternância de diferentes modos de conversão de conhecimento, como mostra a Figura 2.

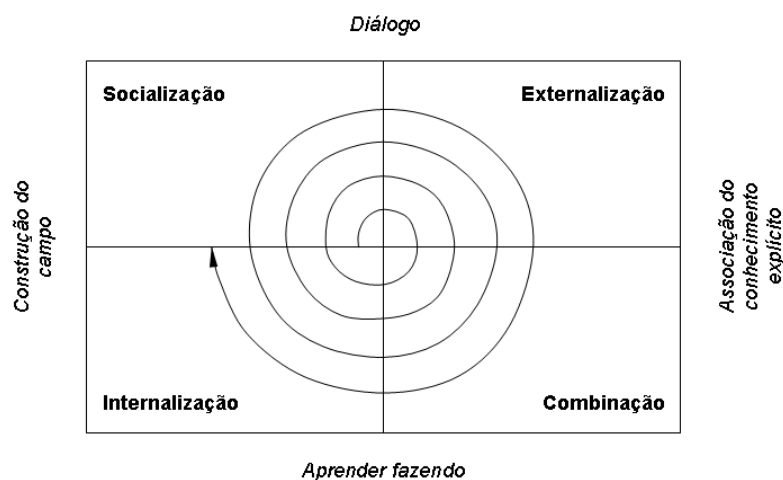


Figura 2 – Espiral do conhecimento. Adaptado de (Nonaka & Takeuchi, 1997).

Primeiramente a socialização inicia estabelecendo um campo para as interações. Este campo facilita o compartilhamento de experiências e de modelos mentais entre os membros da organização. A seguir, a externalização é deflagrada através de diálogos significativos e reflexões coletivas, onde o uso apropriado de metáforas e analogias ajuda os membros do time a articular seus conhecimentos tácitos que por uma outra forma seriam dificilmente comunicados. Em seguida, o modo combinação é ativado pela interligação dos novos conhecimentos criados com os conhecimentos previamente existentes de outras seções da organização, cristalizando-os assim em novos produtos, serviços ou sistemas de gerenciamento. Finalmente o “aprender fazendo” provoca a internalização do conhecimento

produzido.

A organização não pode por si só criar conhecimentos. O conhecimento tácito dos indivíduos serve de base para a criação do conhecimento organizacional. A organização deve mobilizar o conhecimento tácito gerado e acumulado pelos indivíduos. O conhecimento tácito mobilizado é amplificado, dentro da organização, através dos quatro modos de conversão do conhecimento e cristalizado em níveis mais altos de ontologia. É denominada “espiral do conhecimento” na qual a interação entre conhecimento tácito e explícito se amplia em escala e progride em níveis ontológicos. Desse modo, a criação do conhecimento organizacional é um processo em espiral, começando no nível dos indivíduos e se movendo através de comunidades cada vez maiores de interação, cruzando as fronteiras das seções, dos departamentos e da organização

Segundo Nonaka & Takeuchi (1997), a função da organização no processo de criação do conhecimento organizacional é fornecer o contexto apropriado para a facilitação das atividades em grupo e para a criação e acúmulo de conhecimento em nível organizacional

3. Métodos ágeis de desenvolvimento de software

As abordagens de desenvolvimento de software que seguem os valores e princípios definidos neste manifesto passaram a denominar-se de métodos ágeis. Entre tais métodos, cita-se: Scrum (SCHWABER, 1995), Dynamic Systems Development Method – DSDM (STAPLETON, 1997), Lean Agile (POPPENDIECK e POPPENDIECK, 2006), Feature Driven Development - FDD (PALMER, 2002) e Extreme Programming - XP (BECK, 2000). Trabalhos como os realizados por Rising e Janoff (2000), Cohn e Ford (2003), Williams e Cunningham (2004), Svensson (2004), Dyba e Dingsoyr (2008), Sharp e Robinson (2008), Cabral et al (2009), Moe et al (2009), Pikkarainen et al (2009), Xiaofeng et al (2009) são alguns dos relatos empíricos de experiências com a adoção de métodos ágeis. Já autores como Palmieri (2002), Holz e Maurer (2002) e Chau e Maurer (2004) são alguns dos que abordam a gestão do conhecimento no contexto dos métodos ágeis.

Referindo-se ao significado dos valores e princípios ágeis, autores como Highsmith (2002) e Tate (2006) lembram que os ambientes empresariais estão cada vez mais dinâmicos e instáveis, assim para o primeiro autor, no contexto do que foi declarado no manifesto ágil, o termo agilidade refere-se à habilidade para criar e responder a mudanças nestes ambientes. Já o segundo percebe que as premissas ágeis, por aceitar melhor as mudanças que venham ocorrer em projetos de software, estão são mais ajustados aos ambientes empresariais do que os métodos excessivamente prescritivos.

Diante de percepções como as apresentadas pelos autores acima citados, faz-se necessário uma melhor compreensão das diferenças das entre abordagens clássicas e ágeis, para a partir disso, verificar a hipótese de melhor adequação dos métodos ágeis a ambiente dinâmicos. A Tabela 1 mostra algumas comparações realizadas entre as referidas abordagens.

Desenvolvimento de software	Métodos tradicionais	Métodos ágeis
Concepções fundamentais	Os sistemas são totalmente especificados e construídos sobre metódico e extensivo planejamento.	Os sistemas devem primar pela qualidade, construídos de forma adaptativa por pequenas equipes que se utiliza de princípios de melhoria contínua e testes, baseados em mudanças e rápido <i>feedback</i> .
Controle	Centrado no processo	Centrado nas pessoas
Estilo de gerenciamento	Comando e controle	Liderança e colaboração

Gestão do conhecimento	Explícito	Tácito
Funções e responsabilidades	Individual – favorece a especialização	Equipes auto-organizadas – encoraja a intercambialidade de funções e responsabilidades.
Comunicação	Formal	Informal
Papel do cliente	Importante	Crítico
Ciclo do projeto	Guiado por atividades e tarefas	Guiado por entregas do produto
Modelo de desenvolvimento	Modelo de ciclo de vida, abordagens clássicas e variações, ex. modelo cascata.	Modelo de entregas evolutivas.
Estrutura organizacional	Mecanicista (burocrática com alta formalização)	Orgânica (flexível e participativa encoraja a interação e a cooperação).

Fonte: adaptado de Nerur et al (2005))

Tabela 1 - Comparação entre métodos tradicionais e métodos ágeis.

Conforme exposto na Tabela 1, em abordagens tradicionais as tarefas, normalmente, são mais prescritivas, documentadas e distribuídas para os desenvolvedores, ou seja, centradas no processo. Nas abordagens ágeis a equipe possui autonomia e flexibilidade na escolha e realização das tarefas, priorizando sempre a qualidade e entrega de módulos de software funcional, sendo assim, centrado em pessoas. Já a respeito da gestão do conhecimento o autor chama a atenção para o fato de que os métodos ágeis primam são mais voltados para o conhecimento tácito.

3.1 Método SCRUM

O termo Scrum, no âmbito empresarial, foi proposto por Takeuchi e Nonaka (1986), publicado como resultado de uma pesquisa que aborda as boas práticas comuns em dez empresas inovadoras japonesas. Schwaber e Beedle (2002) e Schwaber (2004) apresentam o Scrum como uma alternativa permite a gestão de projetos de software, e afirma que o mesmo é válido inclusive para projetos de outra natureza. Os autores ressaltam que não se trata de uma metodologia prescritiva, mas de um framework de gestão de projetos que auxilia no planejamento das atividades e controle do tempo de realização das mesmas. O referido framework possui um conjunto de práticas organizadas em um processo iterativo e incremental. Conforme ilustrado na Figura 4.

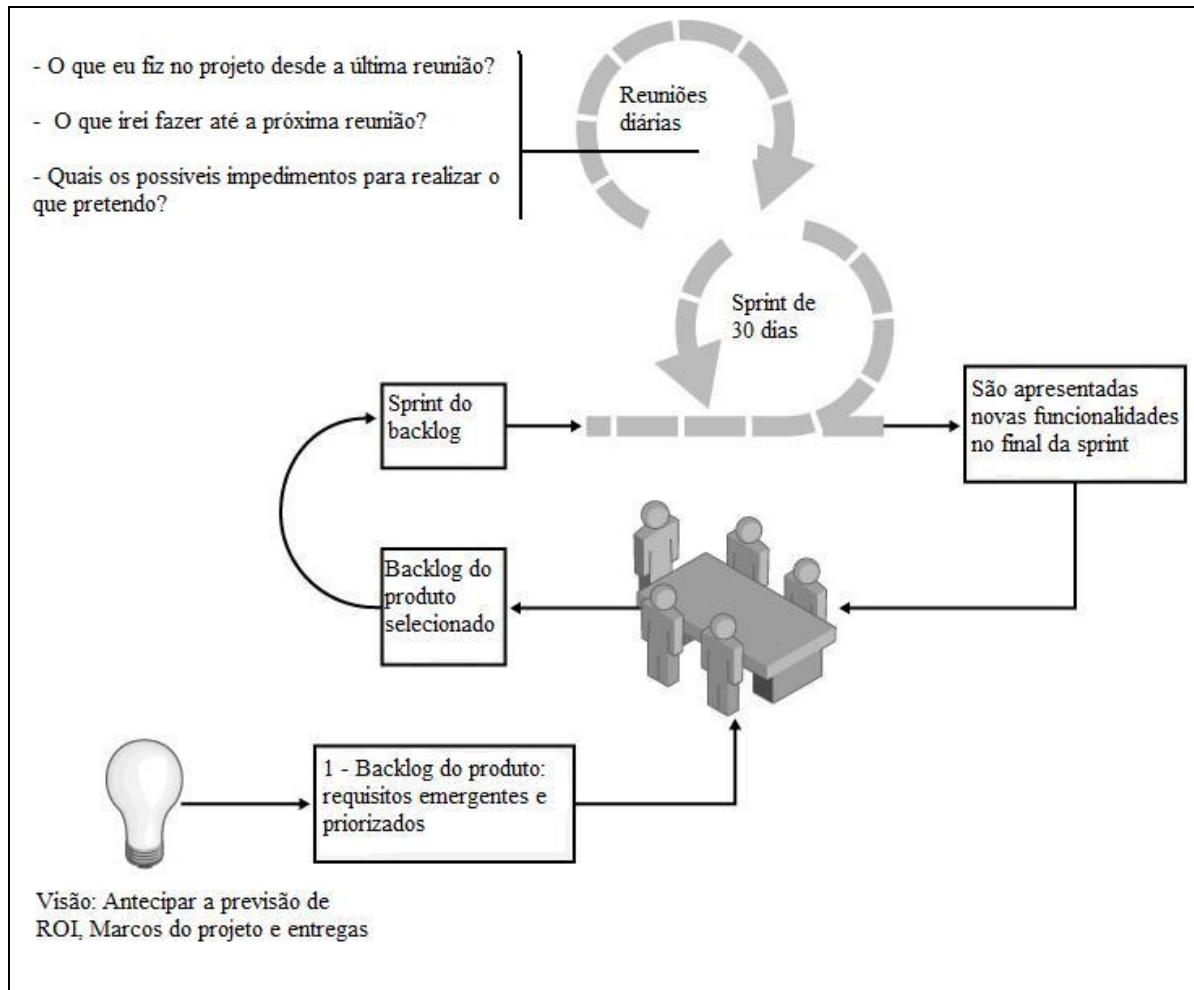


Figura 4 – Estrutura da Metodologia Scrum. Fonte: Adaptado de Schwaber (2004).

Conforme esquematizado na Figura 10, Schwaber e Beedle (2002) e Schwaber (2004) sugerem que um projeto de desenvolvimento de software baseado no Scrum tenha seu início a partir de uma visão do sistema de software a ser desenvolvido. Nesta fase inicial geralmente não há uma visão completa e definitiva do sistema, mas define-se uma previsão do retorno do investimento para o cliente e quais são os principais marcos e entregas do projeto.

Por se tratar de uma abordagem que prima por equipes auto-geridas e de alto desempenho, uma das práticas Scrum consideradas vitais durante a execução de uma Sprint é a prática de pequenas reuniões diárias. As reuniões diárias prescritas por Schwaber e Beedle (2002) são compostas de três perguntas básicas que cada membro da equipe deverá responder: O que eu fiz no projeto desde a última reunião? O que irei fazer até a próxima reunião? Quais são os impedimentos? Como este encontro diário tem aproximadamente quinze minutos de duração, as pessoas não devem tentar resolver os problemas durante a reunião, mas apresentar a dificuldade encontrada para que algum membro da equipe possa ajudá-lo em outro momento.

3.2 Extreme programming – XP

De acordo com Beck (2000) o método XP é composto por um conjunto de práticas que depende de alguns valores compartilhados pela equipe do projeto para que elas efetivamente sejam executadas. Desta forma, inicialmente se faz necessário entender o significado destes valores no contexto das práticas. Os valores são Comunicação, simplicidade, feedback e coragem. Os referidos valores devem estar presentes nas práticas prescritas pelo método, as práticas são: Cliente presente, Jogo de planejamento, Metáfora, Projeto simples, Liberações

pequenas, programação em pares, desenvolvimento orientado por testes, Refatoração, Integração contínua, Ritmo sustentável, Código padronizado, Código coletivo.

O método XP consiste de um conjunto de práticas complementares entre si e que são suportadas por valores necessários para que as mesmas possam ser executadas de forma satisfatória. Autores como Abrahamsson (2003) e Huang e Holcombe (2009) afirmam parte destas práticas já eram conhecidas na área de engenharia de software, antes mesmo do manifesto ágil. Contudo, Abrahamsson (2003) chama a atenção para o fato de que no âmbito do método XP, estas práticas se diferenciam pela complementaridade entre elas. Beck (2000) ressalta que com exceção da prática de testes, as demais práticas XP podem não fazer muito sentido ou não funcionarem bem sozinhas, requerendo outras práticas para manterem-se equilibradas, ou seja, em um dado contexto, as deficiências encontradas em uma prática podem ser compensadas pelos pontos fortes de outra. A complementaridade entre as práticas XP está representada na Figura 5.

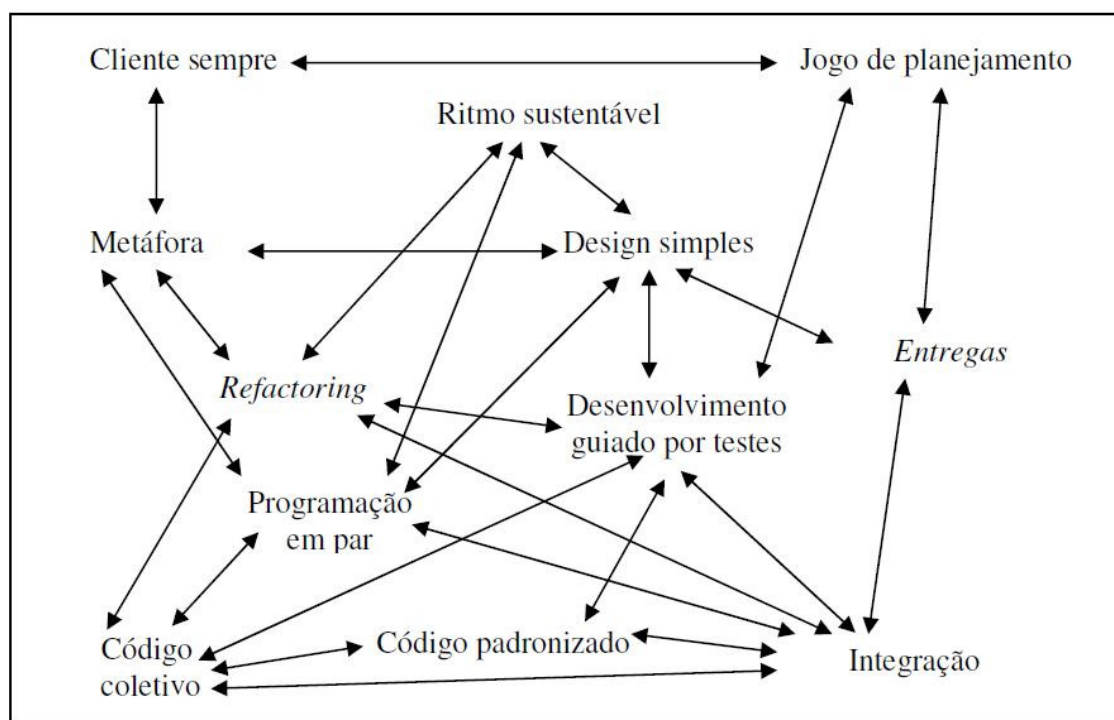


Figura 5 – A interligação das práticas XP. Adaptado de Beck (2000).

De acordo com Beck (2000) a relação de complementaridade das práticas no método XP pode ser observada na forma em que elas estão organizadas. Por exemplo, no XP, estabelecer o desenvolvimento de software baseados em pequenas entregas em curtas iterações de tempo, requer, inicialmente, a obtenção de uma visão geral do sistema de software a ser desenvolvido, para, a partir disso, determinar a prioridade do que será produzido nestas iterações. Em outras palavras, a prática Pequenas entregas (small releases) está diretamente relacionada a prática Jogo de Planejamento. Uma interligação semelhante pode ser observada na prática integração contínua que minimiza o custo de ter que realizar integrar pequenas e frequentes versões do sistema, também se observa que o desenvolvimento orientado por testes deve diminuir a taxa de erros neste processo de intensa produção módulos de software funcional.

4. Métodos ágeis sob a ótica do modelo do espiral do conhecimento

Analisando o espiral do conhecimento sob a ótica das práticas adotadas pelos métodos ágeis,

conforme pode ser observado na Figura 6, práticas como programação em par, código coletivo e cliente presente podem ser entendidas como formas de socialização do conhecimento na equipe de desenvolvimento de software. As práticas de criação de metáforas e primar por um design simples são formas de representação do conhecimento, ou seja, trata-se da externalização do conhecimento gerado na socialização. A combinação do conhecimento gerado até o momento acontece no momento em que a equipe utiliza as práticas de integração contínua e de entregas frequentes. Já a internalização do conhecimento gera subsídios para que a equipe possa realizar refactoring e adotar um ritmo sustentável de desenvolvimento de código.

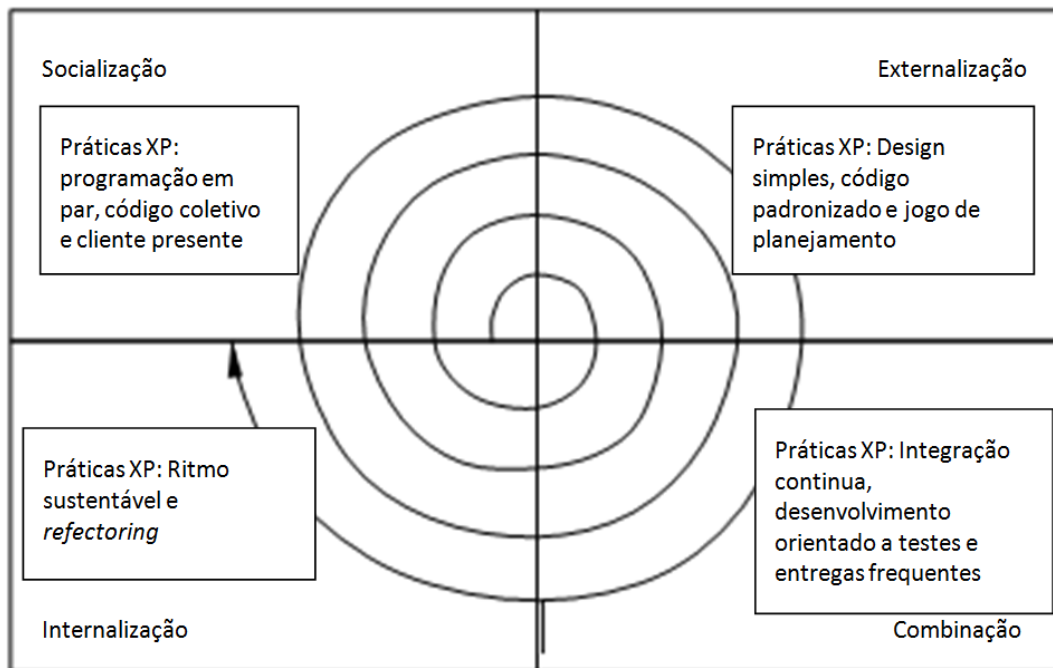


Figura 6 – Práticas XP sob a ótica do espiral do conhecimento.

Tendo em vista que o desenvolvimento de software resulta na criação de novos produtos, sendo este processo dependente da pesquisa e da atividade de criação das pessoas, Schwaber e Beedle (2002) entendem que esta pesquisa e criatividade envolvidas representam a criação de conhecimento. De acordo com os autores, o fato dos desenvolvedores de software precisar eliciar os requisitos que deverão compor o sistema, significa que o conhecimento encontra-se de forma tácita. Segundo os mesmos, durante o desenvolvimento de software, os desenvolvedores tomam suas decisões a partir do conhecimento adquirido por meio de modelos gerados ou trechos de código executável, sendo que nesta perspectiva, ambos representam conhecimento explícito. Para Schwaber e Beedle (2002), o conhecimento em sua forma explícita contrasta com o conhecimento de natureza tácita. Enquanto o primeiro é formalizado, o segundo é baseado em experiências e se reflete em intuições e reações não externalizadas, o autor cita como exemplo, o caso de programadores experientes que são dotados de habilidades e conhecimentos não externalizados.

Utilizando-se do modelo de criação do conhecimento proposto por Nonaka e Takeuchi (1997), ilustrado na Figura 7, Schwaber e Beedle (2002) afirma que o método Scrum se aplicado adequadamente pode exercer a função de promotor do conhecimento por meio de ciclos de socialização, externalização, combinação e internalização do conhecimento.

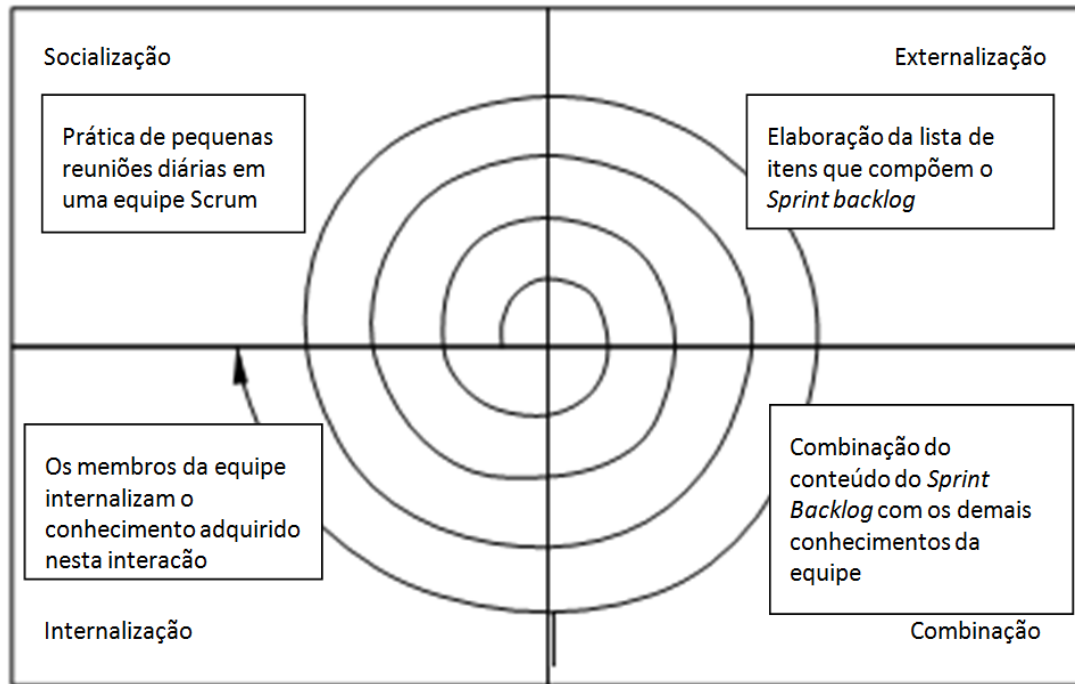


Figura 7 – Dinâmica do Scrum sob a ótica do modelo do espiral do conhecimento.

As ilustrações apresentadas na Figura 6 e na Figura 7 foram elaboradas tendo como base o fato de que a representação do conhecimento pode ocorrer em forma código de software funcional é o caso dos métodos extreme programming e scrum, os quais são formados por um conjunto de práticas e valores, no qual a socialização do conhecimento por meio da interação entre as pessoas gera a possibilidade de obter-se maior qualidade e produtividade em projetos de software.

5. Considerações finais

A análise do modelo de espiral do conhecimento sob a ótica das práticas e valores ágeis representa uma etapa inicial de uma pesquisa que em etapas posteriores pretende verificar em pequenas empresas produtoras de software a coerência dos relacionamentos realizados e também as técnicas utilizadas para a gestão do conhecimento.

O relacionamento observado entre modelo do espiral do conhecimento e algumas das práticas ágeis permite um entendimento da dinâmica que envolve a adoção de métodos ágeis e principalmente a forma em que ocorre a gestão do conhecimento. Neste estudo verificou-se que os ciclos de socialização, externalização, combinação e internalização do conhecimento sob a ótica das práticas ágeis está mais relacionado a um estilo de gestão de conhecimento voltado para o conhecimento tácito, contudo, por meio das práticas a explicitação é possível.

Neste estudo verificou-se também que métodos ágeis são baseados no trabalho em equipe, interação entre pessoas e representação do conhecimento em forma de código funcional, assim, apresentam-se como um contraponto às metodologias de desenvolvimento de software mais prescritivas e baseadas na representação do conhecimento por meio de artefatos e documentos.

Referências

ABRAHAMSSON, P.; SALO, O.; RONKAINEN, J.; WARSTA, J. *New directions on agile methods: A comparative analysis*. IEEE Proceedings of the 25th International Conference on Software Engineering, 2003. p.114-120.

ABREU, A. F., REZENDE, D. A. *Tecnologia da informação aplicada a sistemas de informação empresariais*. São Paulo: Atlas, 2000.

AGILE. *Agile manifesto*. Disponível em: <www.agilemanifesto.org>. Acesso em: 22 agosto 2009.

BECK, K. *Extreme programming explained: embrace change*. Addison-Wesley, 2000.

CABRAL, A. Y. et al. *A Case Study of Knowledge Management Usage in Agile Software Projects*. Enterprise Information Systems: 11th International Conference Enterprise Information Systems - ICEIS 2009. Milan, Italy, May. Pp. 6-10. Springer Berlin Heidelberg, 2009. DOI: 10.1007/978-3-642-01347-8_52.

CASTELLS, M. *A Galáxia da Internet*. São Paulo: Zahar, 2003.

CHAU, T.; MAURER, F. *Knowledge Sharing in Agile Software Teams*. In W. Lenski (Ed.), *Logic versus Approximation* (Vol. 3075, pp. pp. 173-183). Berlin: Springer-Verlag Heidelberg, 2004.

COCKBURN, A. *Agile Software Development: The cooperative game* (2nd Edition). Addison Wesley, 2006.

COHN, M.; D. FORD. *Introducing an agile process to an organization*. IEEE Software, 36(6):74–78, June 2003.

DEMARCO, T.; LISTER, T. *Peopleware: Productive Projects and Teams*. New York: Dorset House, 1987.

DEMING, W. E. *Out of the crisis*. MIT Press: Cambridge, MA, 1986.

DRUCKER, P. *Effective Executive*. Harper Colophon Books, New York, 1967.

DRUCKER, P. F. *The new productivity challenge*. Boston: Harvard Business Review, 1991.

DYBA, T.; DINGSOYR, T. *Empirical studies of agile software development: a systematic review*. Information and Software Technology 50 (9–10) (2008), pp. 833–859.

FIALHO F. A. P. et al. *Empreendedorismo na era do conhecimento*. Florianópolis: Visual Books, 2006.

GARVIN, D.A. *Managing quality: The strategic and competitive edge*. Free Press: New York, 1988.

HIGHSMITH, J. *Agile Software Development Ecosystems*. Addison Wesley: Boston, 2002.

HOLZ, H.; MAURER, F. *Knowledge management support for distributed agile software processes*. In *Advances in Learning Software Organizations*, 4th International Workshop, LSO 2002, Chicago, IL, USA, August 6, 2002, Revised Papers., volume 2640. Springer, 2002.

HUANG, L.; HOLCOMBE, M. *Empirical investigation towards the effectiveness of Test First programming*. Information & Software Technology 51(1): 182-194, 2009.

JURAN, J. M. *Juran on leadership for quality*. Free Press: New York, 1989.

LARMAN, C. *Agile and Iterative Development: A Manager's Guide*. Addison Wesley, 2003.

MOE, N. B.; DINGSOYR T.; DYBA T. *A teamwork model for understanding an agile team: A case study of a Scrum project*. Information and Software Technology, In Press, Corrected Proof, Available online 20 November 2009.

NERUR, S.; MAHAPATRA, R.; MANGALARAJ, G. *Challenges of migrating to agile methodologies*. Communications of the ACM, v.48 n.5, p.72-78, May 2005 [doi>10.1145/1060710.1060712]

NONAKA, I; TAKEUCHI, H. *Criação do conhecimento na empresa: como as empresas japonesas geram o conhecimento na empresa*. 7. ed. Tradução de Ana Beatriz Rodrigues e Priscilla Martins Celeste. Rio de Janeiro: Campus, 1997.

NONAKA, I; TAKEUCHI, H. *Criação do conhecimento na empresa: como as empresas japonesas geram o conhecimento na empresa*. 7. ed. Tradução de Ana Beatriz Rodrigues e Priscilla Martins Celeste. Rio de Janeiro: Campus, 1997.

PALMER, S. R.; FELSING, J. M. *A Practical Guide to Feature-Driven Development*. Upper Saddle River. NJ: Prentice-Hall, 2002.

PALMIERI, D. *Knowledge Management through Pair Programming*, Master's Thesis, Department of Computer Science, North Carolina State University, Raleigh, NC, 2002.

PIKKARAINEN, M. et al. *What Skills Do We Really Need in Agile Software Development? – Discussion of Industrial Impacts and Challenges*. Agile Processes in Software Engineering and Extreme Programming,

Lecture Notes in Business Information Processing, Volume 31. ISBN 978-3-642-01852-7. Springer Berlin Heidelberg, 2009, p. 267-280. DOI: 10.1007/978-3-642-01853-4_61.

POPPENDIECK, M.; POPPENDIECK T. *Lean Software Development – An Agile Toolkit for Software Development Managers.* Addison-Wesley, Boston, 2003.

QUINN, J. B. *The intelligent enterprise: A new paradigm.* New York: The Free Press, 1992.

RISING, L.; JANOFF, N. S. *The Scrum software development process for small teams.* IEEE Software, 17(4):26–32, July/August 2000.

SCHWABER, K. *Agile Project Management with Scrum.* Microsoft Press, 2004.

SCHWABER, K. *Scrum Development Process.* OOPSLA'95 workshop on Business Object Planejament and Implementation. Springer-Verlag. 1995.

SCHWABER, K.; BEEDLE, M. *Agile Software Development with SCRUM.* Prentice Hall, 2002.

SCHWABER, K.; BEEDLE, M. *Agile Software Development with SCRUM.* Prentice Hall, 2002.

SHARP, H.; ROBINSON H. *Collaboration and co-ordination in mature eXtreme programming teams.* International Journal of Human-Computer Studies, v.66 n.7, p.506-518, July, 2008

SLIGER, M.; BRODERICK, S. *The Software Project Manager's Bridge to Agility.* Addison Wesley Professional, 2008.

STAPLETON, J. *Dynamic Systems Development Method – The Method in Practice.* Addison Wesley, 1997.

SVENSSON, H. *A Study on Introducing XP to a Software Development Company.* Xp-2003. Lectures notes in computer science. Vol. 2675, 433-434. Springer, 2003.

TATE, K. *Sustainable Software Development: An agile perspective.* Addison Wesley: Boston, 2005.

TOFFLER, A. *Powershift: As mudanças do poder.* Rio de Janeiro: Record, 1990.

XIAOFENG, W. et al. *Where Agile Research Meets Industry Needs: Starting from a 10-Year Retrospective.* Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Business Information Processing, Volume 31. ISBN 978-3-642-01852-7. Springer Berlin Heidelberg, 2009, p. 221. DOI: 10.1007/978-3-642-01853-4_40.