

Teorizando as Práticas dos Métodos Ágeis no Desenvolvimento de Software Visando ao Processo de Inovação das Empresas

Décio Bittencourt Dolci

Instituto de Ciências Econômicas, Administrativas e Contábeis – FURG – Brasil

Resumo

Executivos brasileiros de tecnologia da informação estão sendo desafiados a desempenhar um papel ativo e importante no processo de inovação em suas organizações. O objetivo do presente estudo foi explorar o uso de métodos ágeis de desenvolvimento de software diante deste desafio. Usou-se a teoria das práticas, em particular a visão de Orlikowski, para analisar práticas presentes em sete métodos ágeis – eXtreme Programming (XP), Dynamic Systems Development Method (DSDM), Scrum, Crystal, Agile Modeling (AM), Feature Driven Design (FDD) e Adaptive Software Development (ASD). Nos resultados, encontram-se as práticas agrupadas em cinco fatores subjacentes – aprendizagem, cliente, interativo, humano e técnico –, mostrando, na sequência, a mudança de foco do desenvolvimento ágil em relação aos métodos anteriores, centrando-se para o indivíduo e suas ações nas relações sociais em substituição a questões essencialmente técnicas. Os resultados sugerem que os métodos ágeis de desenvolvimento de software apresentam princípios e práticas mais adequados ao desafio da inovação do que ofereciam seus antecessores. Adicionalmente, suas práticas podem auxiliar a organização na constituição de propriedades institucionais úteis à gestão do processo de inovação de modelos de negócio.

Palavras-chave: desenvolvimento de software; métodos ágeis; inovação; modelos de negócio; teorias das práticas

Agile Methods in Software Development: The Search for Organizational Innovation

Décio Bittencourt Dolci
ICEAC - FURG - Brazil

Abstract

Brazilian IT executives are being challenged to play an active and important role in the innovation process within their organizations. The purpose of this study is to explore agile software development methods in facing this challenge. We deployed a practice theory framework, in particular that of Orlikowski, in order to analyze practices from seven agile methods - eXtreme Programming (XP), Dynamic Systems Development Method (DSDM), Scrum, Crystal, Agile Modeling (AM), Feature Driven Design (FDD) and Adaptive Software Development (ASD). Results show practices grouped into five underlying factors - learning, interactive, customer, human and technical; in addition, they explain the shift in focus of the agile development in reference to previous methods, centering on the individual and their actions in social relations in place of essentially technical issues. Findings suggest that agile software development methods have principles and practices better suitable to the innovation challenge than their predecessors do. Furthermore, their practices can aid the organization in the constitution of institutional properties useful to the process of business model innovation.

Keywords: software development; agile methods; innovation; business models; practice theory

1 Introdução

Inúmeras questões organizacionais, gerenciais e tecnológicas exigem a atenção dos executivos da área de tecnologia da informação (TI). Virtualização de servidores, computação nas nuvens, ferramentas colaborativas, dispositivos móveis e software como serviço são algumas das novas tecnologias em fase de difusão que se destacam apenas mais recentemente, em torno de dois anos, nas pesquisas sobre preocupações e intenções dos gestores de TI. Em contraste à volatilidade das questões tecnológicas, questões gerenciais – como planejamento estratégico de TI e alinhamento entre TI e negócio – apresentam certa estabilidade: aparecem há quase 10 anos entre as principais preocupações desses executivos (Luftman *et al.*, 2009; Luftman & Ben-Zvi, 2010). Entretanto, parece haver um fato novo: a área de TI estar sendo vastamente cobrada por liderar o processo de inovação das empresas.

O estudo *The State of the CIO 2009* no Brasil (CIO Magazine, 2009) mostra que 44% dos executivos de TI (entrevistados) afirmam que a inovação representa um assunto amplamente discutido em suas empresas e que a área de TI é constantemente cobrada por liderar esse processo, já havendo uma considerável dedicação dos gestores de TI a iniciativas inovadoras; e quase um terço (31%) dos profissionais que responderam ao estudo dizem que o tema consome de 5,1% a 10% das horas diárias de trabalho. Nesse sentido, amplia-se o papel do executivo de TI – também passa a ser o líder do processo de inovação das empresas (Fortino, 2008) e torna-se fundamental que esse executivo compreenda melhor questões relacionadas à estratégia, marketing, pesquisa e desenvolvimento.

Em grande parte, a nova atribuição deve-se à posição central que os artefatos de TI e de comunicação ocupam nas inovações contemporâneas, onde o sucesso normalmente depende de uma combinação adequada entre avanço tecnológico e inovação no modelo de negócio (Teece, 2010). Por exemplo, a associação da capacidade de produtos modernos monitorarem as atividades de seus usuários com informações administradas na empresa torna possível inovar como os produtos e serviços sejam criados, vendidos e usados (CIO Magazine, 2010). No entanto, há uma dificuldade que precisa ser superada. Apesar de já se ter cinco décadas de experiência no desenvolvimento de sistemas de informação (SI) computadorizados, a gestão do seu processo continua a ser desafiadora (Maruping *et al.*, 2009). Essa constatação não deveria surpreender caso se examine o produto do processo sob a perspectiva do que Baetjer chama de essência do software: “uma incorporação de conhecimentos coletados, destilados e organizados, à medida que o processo é conduzido” (Pressman, 2006, p. 16). É, de fato, uma atividade complexa, repleta de interdependências, com entradas de múltiplos indivíduos com conhecimentos diferentes. Seja qual for a área, seja qual for o sujeito, o conhecimento não permanecerá imóvel; assim, as funcionalidades requeridas de um software é um alvo em movimento, imprevisível (Nidumolu *et al.*, 2001).

Tal complexidade inerente aos softwares foi subestimada durante décadas. Com o ambiente moderno de negócios, mutável, acelerado e que exige a entrega rápida de sistemas computacionais bem-sucedidos, acentuou-se o problema. Em decorrência da falta de clareza sobre a essência do software, utilizaram-se processos de desenvolvimento de SI que se mostraram pouco eficazes. É inegável que diversas metodologias de desenvolvimento de sistemas não trataram adequadamente a imprevisibilidade, optando pela estruturação em vez da flexibilização. Além disso, também não deram atenção a processos de aprendizagem, a modos eficientes de gerar e compartilhar conhecimentos e a fatores humanos e sociais importantes em qualquer organização do trabalho. Adicionalmente, na academia, esses problemas não receberam a devida atenção, prevalecendo estudos interessados em aspectos de ordem técnica (Kotlarsky & Oshri, 2005; Bellini *et al.*, 2008).

Neste contexto, tendo como marco o *Manifesto para o Desenvolvimento Ágil de Software* (Pressman, 2006), as dificuldades são levantadas mais enfaticamente, abrindo espaço para os denominados “métodos ágeis”, onde são propostas formas de organizar o trabalho com base em novas premissas

(Beck & Fowler, 2001; Fowler & Highsmith, 2001; Pressman, 2006). Atualmente, segundo Conboy (2009), encontram-se, entre os métodos ágeis mais conhecidos, eXtreme Programming – XP (Beck, 1999), Dynamic Systems Development Method – DSDM (Stapleton, 1997), Scrum (Schwaber & Beedle, 2002), Crystal (Cockburn, 2001), Agile Modeling – AM (Ambler, 2002) e Feature Driven Design – FDD, (Coad *et al.*, 1999). Inicialmente criados e disseminados por desenvolvedores e consultores, tornaram-se foco de pesquisas científicas, revitalizando a área de pesquisa em desenvolvimento de SI, evidenciado pelo número de edições especiais de periódicos, conferências e *workshops* sobre o tema (Agerfalk *et al.*, 2009).

A relevância do tema de pesquisa associada a atividades de ensino e extensão exercidas pelo autor do presente artigo em uma universidade federal, trabalhando em conjunto com profissionais da área de TI e administradores, despertou o interesse na realização de um projeto de pesquisa sobre o tema em 2010. Percebe-se muita curiosidade sobre o assunto entre os profissionais da área de TI, em especial entre executivos, pois desejam compreender melhor as mudanças metodológicas que estão acontecendo, avaliar possíveis benefícios e problemas na sua adoção. Na academia, entre alunos da graduação em Administração, a discussão sobre métodos ágeis pouco se faz presente, pois fontes tradicionais como Laudon & Laudon (2007), O'Brien & Marakas (2007) e Turban *et al.* (2010), comumente usadas em disciplinas voltadas à Administração de Sistemas de Informação, não abordam esses métodos.

Diante do exposto, o presente artigo adota a linha investigativa da teoria das práticas – mais especificamente, o circunstanciado por Orlikowski (1992), Orlikowski *et al.* (1995), Orlikowski (2000) e Feldman & Orlikowski (2011) – para atingir os seguintes objetivos: compreender melhor os métodos ágeis por meio de suas práticas e articular possíveis contribuições do uso dos métodos ágeis na institucionalização de propriedades que auxiliem os processos de inovação nas empresas.

Realizada esta introdução ao tema de pesquisa, organiza-se o texto subsequente da seguinte forma: apresenta-se o referencial teórico, o procedimento metodológico da investigação empírica, a discussão de resultados – com foco especial em suas implicações práticas – e conclui-se o estudo com a indicação de limitações e sugestões para pesquisas futuras.

2 Referencial Teórico

Nesta seção, primeiramente, expõe-se o contexto teórico sobre desenvolvimento ágil, apresentando conceitos, breve histórico e princípios para se alcançar a agilidade; após, faz-se uma introdução à teoria de base usada na presente pesquisa.

2.1 Desenvolvimento Ágil: Conceitos, Histórico e Princípios

Conboy (2009) observa que os termos “método” e “agilidade” ainda são vagamente definidos nos estudos sobre desenvolvimento de sistemas de informação. Embora haja extensa pesquisa no sentido de distinguir o termo método em desenvolvimento de sistemas de informação de outros, como metodologia, método de gerenciamento de projetos, processo e práticas, o autor oferece a seguinte interpretação inclusiva (p. 329):

um método de desenvolvimento de sistemas de informação abrange toda a gama de práticas envolvidas no processo de concepção, construção, implementação e manutenção de um sistema de informação, orientando como essas atividades são realizadas e gerenciadas, a seqüência e a freqüência destas atividades, bem como os seus valores e objetivos.

Ainda observa que, recentemente, a noção de método como um pensamento em oposição a uma prescrição rigorosa, tornou-se amplamente aceita no desenvolvimento de sistemas de informação, sendo usada em seu estudo. De fato, ao incluir valores e objetivos ao final de sua definição, amplia nossa capacidade de examinar as práticas.

No que se refere à agilidade, Conboy buscou sua definição fazendo uma revisão da literatura sobre o uso da palavra em diversas áreas do conhecimento. Contextualizando ao utilizado em desenvolvimento de sistemas de informação, propõe a seguinte definição para agilidade (p. 340):

a disponibilidade contínua de um método para rapidamente ou inerentemente criar a mudança, proativamente ou reativamente adotar a mudança, e aprender com a mudança, enquanto contribui para o valor percebido pelo cliente (economia, qualidade e simplicidade), por meio de seus componentes coletivos e relação com seu ambiente.

A busca por agilidade e métodos ágeis está intimamente relacionada ao ambiente moderno de negócios em que prevalece a mudança (Wood, 1995; Mintzberg *et al.*, 2000; Cummings & Worley, 2009), cada vez mais veloz, exigindo a entrega rápida de sistemas computacionais bem-sucedidos. Diante desse desafio, engenheiros de software percebiam fraquezas na engenharia de software convencional e ideias de como vencê-las foram se cristalizando na década de 1990 (Pressman, 2006), originando o que foi denominado de “movimento ágil”. Um marco importante desse movimento é quando Kent Beck e 16 outros notáveis desenvolvedores, produtores e consultores de software formam a Aliança Ágil (<http://www.agilealliance.org>) e assinam o “Manifesto para o Desenvolvimento Ágil de Software”, em 2001. Declaram que estão descobrindo melhores modos de desenvolvimento de software, baseado em novos valores, a saber: indivíduos e interações em vez de processos e ferramentas; softwares funcionando em vez de documentação abrangente; colaboração do cliente em vez de negociação de contratos; e resposta a modificações em vez de seguir um plano.

A revisão da literatura sobre métodos ágeis remete em muitas passagens ao que foi escrito anteriormente sobre desenvolvimento iterativo e incremental (Larman & Basili, 2003), amplamente difundido em décadas passadas em reação ao processo sequencial, com pouquíssimo *feedback*, originalmente proposto pelo modelo cascata, ou ciclo de vida tradicional. Todavia, o postulado pelos métodos ágeis apresenta outras diretrizes até então não valorizadas pelos métodos apenas iterativos e incrementais. Questões humanas, participação dos clientes na gestão do processo e flexibilidade nunca antes foram tão enfatizadas. É válido destacar a definição de flexibilidade apresentada por Conboy (2009, p. 336): “capacidade de um método para criar a mudança ou adotar proativa, reativa ou inerentemente a mudança em tempo hábil, por meio de seus componentes internos e relacionamentos com seu meio ambiente”. Percebe-se que a definição de flexibilidade é bem próxima à de agilidade; no entanto, destaca-se principalmente que questões relacionadas à aprendizagem com a mudança não se fazem presentes.

Igualmente oportuno na consecução da presente pesquisa foi o levantamento dos princípios que regem o desenvolvimento ágil. Um panorama amplo para aqueles que querem alcançar a agilidade pode ser visto nos 12 princípios apresentados pela Aliança Ágil, resumidos como:

- p1. A maior prioridade da equipe de desenvolvimento é satisfazer ao cliente desde o início por meio de entrega contínua de software.
- p2. As modificações de requisitos são bem-vindas. Mesmo que tardias, as modificações devem ser aproveitadas como vantagens para a competitividade do cliente.
- p3. A entrega de software funcionando deve ser frequente, a cada duas semanas até dois meses, de preferência no menor espaço de tempo.
- p4. O pessoal de negócio e os desenvolvedores devem trabalhar juntos diariamente durante todo o projeto.

- p5. Os projetos devem ser construídos em torno de indivíduos motivados.
- p6. A conversa face a face é o modo mais eficiente de levar informação para dentro da equipe.
- p7. A principal medida de progresso é a entrega de software funcional.
- p8. Os processos ágeis promovem o desenvolvimento sustentável, buscando um ritmo constante, indefinidamente.
- p9. A atenção a excelência técnica e ao bom projeto deve ser contínua.
- p10. A busca pela simplicidade é essencial, maximizando a não-realização de trabalho que não seja útil.
- p11. O trabalho deve ser por meio de equipes auto-organizadas.
- p12. A equipe deve refletir regularmente sobre como se tornar mais efetiva, ajustando adequadamente seu comportamento.

Conforme Sidky & Arthur (2007), subjacente a esses doze princípios emergem cinco que capturam a essência deles, a saber: adoção da mudança para atender os valores do cliente; planejamento e entrega de software frequentemente; centralização no ser humano; excelência técnica; e colaboração do cliente. Os autores usam essas variáveis subjacentes na proposta de um *framework* com base em práticas ágeis. Dolci (2011) toma por base as cinco dimensões de Sidky & Arthur (2007) para definir cinco atributos a serem usados na construção de um modelo geométrico, associando a cada um deles os princípios propostos pela Aliança Ágil, conforme mostra a Figura 1. É válido destacar que, no contexto estudado, questões relacionadas à flexibilidade encontram-se no atributo aprendizagem.

Atributo	Descrição	Princípios
Aprendizagem	Predisposição à aprendizagem, adotando a mudança como geradora de vantagem competitiva para o cliente.	p2, p12
Iterativo	Atitude de planejar e entregar software frequentemente.	p3, p7, p8
Humano	Confiança nas pessoas.	p5, p11
Técnico	Comprometimento com a produção de código de alta qualidade.	p9, p10
Cliente	Interação entre cliente e equipe de desenvolvimento.	p1, p4, p6

Figura 1: Variáveis subjacentes aos 12 princípios ágeis

Nesta seção, apresentaram-se os principais conceitos envolvidos nos métodos ágeis. Após o breve histórico do Movimento Ágil, mostraram-se os princípios que orientam as práticas dos métodos ágeis. Os estudos citados servem para orientar a análise de dados e discussão de resultados.

2.2 Teoria das Práticas

No presente estudo, recorre-se à “teoria das práticas”, mais especificamente ao circunstanciado por Orlikowski (1992), Orlikowski *et al.* (1995), Orlikowski (2000) e Feldman & Orlikowski (2011), para a análise de dados e discussão de resultados. Considera-se a teoria das práticas um tipo de teoria social que foi esboçada por autores como Bourdieu, Giddens, Taylor, Foucault e outros (Recwitz, 2002; Feldman & Orlikowski, 2011). Central nessas investigações é a noção de que a vida social é uma produção em andamento e, assim, surge por meio de ações recorrentes das pessoas. Embora haja um amplo panorama intelectual, Feldman & Orlikowski (2011) destacam três princípios: (1) percebem que as ações cotidianas são consequentes na produção de contornos estruturais da vida social, (2) mostram rejeição a dualismos, reconhecendo o relacionamento inerente entre elementos que geralmente são tratados dicotomicamente, e (3) argumentam que relações são mutuamente

constitutivas. Ainda, no que se refere à utilização da teoria das práticas em pesquisas acadêmicas, observam: ao direcionar seu foco nas dinâmicas, relações e amarrações, oferece ferramentas poderosas para observar organizações contemporâneas – complexas, dinâmicas, distribuídas, móveis, transientes e sem precedentes.

Feldman & Orlikowski (2011) percebem a existência de um expressivo número de estudos realizados com base na teoria das práticas nos últimos anos. Normalmente, os estudos de Orlikowski fazem referência à teoria da estruturação de Giddens (1984). A teoria de Giddens oferece uma descrição da vida social em termos do desenvolvimento de práticas sociais ao longo do tempo e do espaço. Trata estrutura tanto como um produto da ação humana como condicionadora de tais ações, ou seja, embora reconheça que as ações humanas são habilitadas e restringidas por estruturas, estas também resultam de ações humanas prévias, havendo, deste modo, a recursão no processo. Nesse sentido, por um lado, o autor admite a noção determinista, objetiva e estática de estrutura, mas, por outro, aceita a visão voluntarista, subjetiva e dinâmica. Conforme Barley e Tolbert (1997), há coexistência de duas realidades: o ambiente institucional, constituído pelas características estruturais, e o ambiente da interação, formado pelas capacidades cognitivas dos agentes. A Figura 2 mostra como Giddens concebe a relação das duas realidades da organização social por meio do que ele denomina de “modalidades da estruturação”.

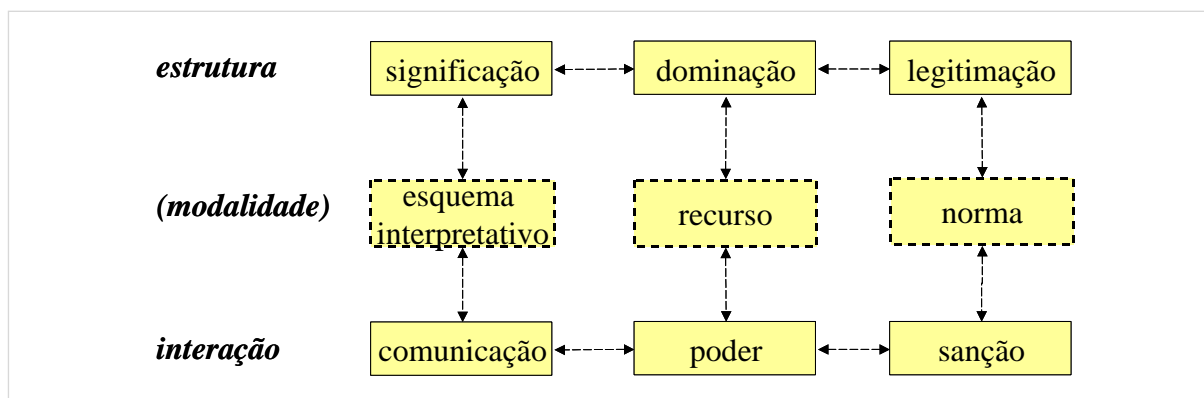


Figura 2: Modelo da estruturação
Fonte: Giddens (1984, p. 23)

São as modalidades as principais dimensões – esquema interpretativo, recurso e norma – da dualidade da estrutura em interação. Por meio delas, há o relacionamento entre as capacidades cognitivas dos agentes e as características estruturais. É válido salientar que se optou por empregar a palavra “recurso”, ainda que, em Giddens (1989), conste “facilidade”, pois se percebe que “recurso” expressa melhor a ideia do autor no nosso idioma.

Esquemas interpretativos são estoques de conhecimento compartilhados e padronizados que os agentes usam para interpretar comportamentos e eventos, assim alcançando sentido na interação. Recursos são os meios através dos quais suas intenções são realizadas, metas são concretizadas e poder é exercido. Normas são as regras que amoldam a conduta e definem a legitimidade da interação. Estas três modalidades determinam como as propriedades institucionais de um sistema social mediam a ação humana e como a ação humana constitui a estrutura social. Segundo Giddens, os atores apóiam-se nas modalidades da estruturação na reprodução de sistemas de interação, reconstituindo, justamente por isso, suas propriedades estruturais. Salienta, ainda, que apenas analiticamente as estruturas – de significação, de dominação e de legitimação – são separáveis; e, também, somente

neste sentido é válido apresentar os elementos da interação – de comunicação, de poder e de sanção – em separado.

Como conjunto de regras e recursos interativamente organizados como propriedades dos sistemas sociais, a estrutura está fora do tempo e do espaço (é uma abstração), exceto em suas exemplificações, e é marcada pela ausência do sujeito. Isto não impede conceber as propriedades estruturais como hierarquicamente organizadas, em termos de extensão espaço-temporal das práticas que os agentes recursivamente organizam. Às propriedades estruturais mais profundamente embutidas, implicadas na reprodução de totalidades sociais, Giddens denomina “princípios estruturais”. Aquelas práticas que possuem a maior extensão espaço-temporal, dentro de tais totalidades, podem ser designadas como “instituições”. Portanto, são as instituições os aspectos mais duradouros da vida social. Nesse sentido, os sistemas sociais compreendem as atividades localizadas de agentes humanos, reproduzidas através do tempo e do espaço, incluindo a estrutura recursivamente implicada.

Em outras palavras, a realidade institucional representa um quadro de referência de regras e tipificações derivadas do acúmulo de ações e interações ao longo da história. Conforme Giddens, a ordem institucional consiste de princípios gerais subjacentes a sistemas de significação, dominação e legitimação. Em contraste, a realidade da ação refere-se à organização de pessoas, objetos e eventos, revelando-se minuto a minuto no fluxo da vida social (Barley & Tolbert, 1997).

Na medida em que instituições são fixadas no estoque de conhecimento dos atores, influenciam como as pessoas comunicam-se, estabelecem poder e determinam que comportamentos aprovar e recompensar, e, segundo Giddens (1984, p. 20), analisar a estruturação dos sistemas sociais significa:

estudar os modos como tais sistemas, fundamentados nas atividades cognoscitivas de atores localizados que se apóiam em regras e recursos na diversidade de ação, são produzidos e reproduzidos em interação. (...) A constituição de agentes e estruturas não são dois conjuntos de fenômenos dados independentemente – um dualismo –, mas representam uma dualidade. De acordo com a noção de dualidade da estrutura, as propriedades estruturais de sistemas sociais são, ao mesmo tempo, meio e fim das práticas que elas mesmas organizam.

Na verdade, ao conciliar várias correntes teóricas, observa-se que Giddens apresenta mais uma forma de pensar sobre o mundo do que uma teoria sobre determinado fenômeno que pode ser explorada e testada empiricamente. Ao mesmo tempo, não deixa explícito como a tecnologia se insere neste processo de mudanças sociais. Um passo importante nesse sentido foi dado por Orlikowski (1992), ao desenvolver seu modelo estrutural da tecnologia.

Em Orlikowski & Robey (1991), os princípios da teoria da estruturação de Giddens são aplicados para auxiliar no entendimento da relação entre tecnologia da informação e estruturação das organizações. Adiantam, também, parte da formulação do modelo estrutural da tecnologia realizada por Orlikowski (1992) para levantar a questão da dualidade da tecnologia: produto social da ação humana em um específico contexto estrutural e cultural, exercendo simultaneamente um papel constitutivo em que um conjunto de regras e recursos contribui para mediar (facilitando ou restringindo) a ação humana e, assim, contribuindo para a criação, recriação e transformação desses contextos.

Ao fazer uma discussão sobre os resultados de sua pesquisa, Orlikowski (1992) afirma que inovações tecnológicas são resultantes do tempo e do contexto organizacional no qual elas estão inseridas e refletem conhecimentos, materiais, interesses e condições num dado período na história. Comparando a teoria da estruturação de Giddens e seu modelo estrutural da tecnologia, a autora salienta que, enquanto a teoria da estruturação de Giddens posiciona-se no entendimento da sociedade, possibilitando análises em seus múltiplos níveis, o modelo estrutural da tecnologia

permite examinar a interação entre tecnologias e organizações focalizando diferentes níveis de análises: interorganizacional, organizacional, grupos ou indivíduos. Entre outras sugestões de pesquisa, indica que o seu modelo pode ser empregado para examinar como diferentes formas organizacionais engendram certos tipos de tecnologia, e como essas tecnologias, em uso, podem reforçar e transformar as configurações estruturais da organização ao longo do tempo.

Gash & Orlikowski (1991) sugerem uma possível forma de avaliar as co-evoluções das inovações organizacionais e em TI, a partir do entendimento dos esquemas interpretativos norteadores dos gestores das organizações. Conforme as autoras, *frames* tecnológicos são subconjuntos de *frames* organizacionais, na medida em que são os pressupostos, significados e percepções que as pessoas usam para entender a natureza e o papel da tecnologia nas organizações, assinalando ser a tecnologia interpretativamente flexível, pois está aberta a diferentes interpretações, dependendo dos múltiplos grupos sociais. As autoras exemplificam que visões sobre como o trabalho deve ser feito, que divisão aplicar ao trabalho e quanto de autonomia empregados e unidades devem ter seguem pressupostos consciente ou implicitamente inseridos na tecnologia da informação.

Estudos posteriores de Orlikowski aprofundam duas questões. Primeiro, Orlikowski *et al.* (1995) mostram os processos de meta-estruturação que ocorre na fase de introdução da tecnologia e antecede a estruturação. Após, Orlikowski (2000) chama atenção para a necessidade de se distinguir as propriedades da tecnologia em uso da tecnologia *per se*, cunhando o termo “tecnologia na prática”. Salienta que, somente com o uso repetitivo de uma tecnologia (tecnologia na prática), é que as propriedades tecnológicas se tornam constituídas assim como outros recursos e regras que moldam a ação. É válido observar que, na presente investigação, estende-se a percepção sobre constituição das propriedades a partir das práticas em uso na mediação (fase de meta-estruturação). Mostra-se o modelo com os processos de estruturação e de meta-estruturação da tecnologia na seção 4.2.

3 Método

A coleta de dados sobre práticas dos métodos ágeis ocorreu no segundo semestre de 2010 e contou com a colaboração de 30 alunos da disciplina Administração de Sistemas de Informação do curso de Graduação em Administração da Universidade Federal do Rio Grande (FURG), sendo Desenvolvimento de Sistemas de Informação um dos tópicos estudados na referida disciplina. No planejamento da disciplina, concebeu-se uma sequência lógica de atividades e procedimentos de modo a aprofundar o conhecimento sobre métodos ágeis.

Primeiramente, realizaram-se três (encontros) aulas expositivas de 90 minutos sobre desenvolvimento de sistemas de informação. Nas duas primeiras, apresentaram-se as principais questões do processo de desenvolvimento e modelos tradicionais de processos de desenvolvimento, usando principalmente o que Laudon & Laudon (2007), O'Brien & Marakas (2007) e Turban *et al.* (2010) apresentam. É válido destacar que tais livros não abordam os métodos ágeis. Na terceira aula, apresentou-se o desenvolvimento ágil. Num primeiro momento, houve uma exposição sobre o que é desenvolvimento ágil, seu histórico, princípios, quem faz e por que é relevante o seu conhecimento. Utilizaram-se referências bibliográficas normalmente relacionadas à engenharia de software, como Pressman (2006). Após, assistiram-se alguns vídeos, um sobre o método Scrum e outro sobre Kanban no desenvolvimento ágil de software. Ao final da aula, propôs-se o desenvolvimento de um trabalho de pesquisa, destacando a emergência do tema, a proliferação dos métodos denominados ágeis e o desconhecimento de uma pesquisa com o intuito de investigá-los conjuntamente frente aos princípios do desenvolvimento ágil vistos em aula. Em uma quarta aula, apresentaram-se definições iniciais sobre o trabalho e distribuíram-se os alunos em sete grupos, sendo cada grupo responsável pela investigação de três métodos ágeis, conforme mostra a Figura 3.

Grupo	Métodos		
1	Scrum	XP	AM
2	DSM	FDD	Scrum
3	XP	Crystal	DSDM
4	FDD	ASD	XP
5	Crystal	AM	FDD
6	ASD	Scrum	Crystal
7	AM	DSDM	ASD

Figura 3: Distribuição dos métodos entre os grupos

Os métodos ágeis selecionados para análise são os apresentados em Pressman (2006). A distribuição propiciou que cada método fosse analisado por três grupos distintos. Para realizar a coleta de dados e análise, cada grupo deveria consultar no mínimo o conteúdo de quatro fontes: o *website* do método, um artigo não necessariamente científico sobre o método, a Wikipédia e o capítulo 4 de Pressman (2006). Deveriam destacar práticas associando-as aos princípios do desenvolvimento ágil. Ao final, cada grupo montou um quadro para cada um dos três métodos avaliados, apresentando uma lista de trechos destacados. Esse conjunto de instruções foi passada na quarta aula, colocando no *website* da disciplina o roteiro e um modelo de planilha a ser usado na coleta e tabulação dos dados. Foi estipulado um prazo de quatro semanas para realizar essa fase do trabalho. No entanto, foi necessário estender o prazo para seis semanas, quando foram postados no *website* da disciplina os últimos trabalhos. É válido salientar que, durante esse período, abordaram-se outros tópicos da disciplina, porém, frequentemente, os alunos organizavam dúvidas sobre o trabalho para serem discutidas ao final das aulas. Após a recepção e avaliação dos trabalhos, encerrou-se esta parte da coleta de dados e deu-se início a outra, tendo por objetivo responder questões mais quantitativas – mas que pouco contribuem aos propósitos deste artigo, estando disponíveis em Dolci (2011).

4 Resultados

Inicialmente, tendo em mente o primeiro objetivo desta pesquisa, apresenta-se parte das práticas percebidas nos métodos ágeis investigados relacionando-as aos cinco fatores teóricos subjacentes – iteratividade, humano, cliente, técnico e aprendizagem. Adicionalmente, buscando a compreensão teórica do fenômeno, faz-se uma discussão sobre o uso dos métodos ágeis à luz do modelo proposto por Orlikowski *et al.* (1995) – modelo estruturacional da tecnologia adaptado para tecnologia em uso e mediação da tecnologia em uso. Após, em função do segundo objetivo desta pesquisa, articula-se a possível contribuição da prática com métodos ágeis no processo de inovação das empresas.

4.1 Práticas Presentes nos Métodos Ágeis

Ao longo da exposição, apresentam-se práticas para cada fator, resgatando breves trechos destacados pelos grupos, para ilustrar a discussão; eles são identificados por siglas como G1, em que o número corresponde ao grupo apresentado na seção 3. Em alguns casos, há mais de um trabalho contendo a mesma ideia, porém com palavras equivalentes; para esses, coloca-se a sigla escolhida para a citação em negrito e, após, acrescentam-se as siglas dos similares. Com esses trechos, é possível perceber as conexões realizadas pelos grupos entre os princípios ágeis e suas práticas.

4.1.1 Práticas Relacionadas à Iteratividade

Entre as ações percebidas no DSDM que lhe dão alta iteratividade, estão a frequente entrega de software, o foco em parte da aplicação, a pronta resposta às mudanças e a aproximação ao software final de modo incremental. O DSDM mantém “*foco na entrega freqüente de produtos, com o pressuposto de que entregar algo ‘suficientemente bom’, já é sempre melhor do que entregar tudo ‘perfeitamente’ no final [...] nesse caso, 80% de uma aplicação pode ser entregue em 20% do tempo que levaria para entregar a aplicação completa 100% (G2)(G3)*”. Para isso, utiliza o “*desenvolvimento iterativo e aproximação incremental, sendo responsivo às exigências em mudança (G2)*” e toma como pressuposto em suas práticas que “*entregar algo bom logo é melhor que entregar algo perfeito somente no fim (G7)*”. Assim, “*iniciando a entrega do produto desde os primeiros estágios do projeto, o produto pode ser testado e revisado e a evidência do teste e revisão da documentação podem ser utilizados na próxima iteração ou fase (G7)*”. A origem do DSDM pode explicar boa parte da sua visão: “*uma metodologia originalmente baseada em ‘desenvolvimento rápido de aplicação’ (RAD) [...] e utiliza prototipagem incremental (G3)*”; ambas (RAD e prototipagem) as metodologias se preocupam em mostrar rapidamente programas em execução.

Quanto ao gerenciamento do processo, o DSDM “*valoriza muito mais entrega do produto que tarefa cumprida (G3)*”. Também se destaca que, nessa metodologia, o teste ganha fundamental importância, na medida em que, além de cumprir sua função original, também “*demonstra que o software foi usado (G3)*”. Por fim, favorece a iteração o fato de que “*esta metodologia apenas requer que cada etapa do desenvolvimento seja completada até que seja possível iniciar o passo seguinte. Isto faz com que cada fase do projeto possa começar sem ter que esperar que as fases que começaram anteriormente sejam totalmente terminadas (G7)*”.

Outro método que se destaca positivamente quanto à iteratividade é o AM. Observa a importância do *feedback* rápido, da interatividade e da fase de preparação (*setup*) para a próxima versão. “*O tempo entre uma ação e resposta é muito importante, pois, quanto mais rápida a reação, mais cedo pode ser corrigido um erro (G1)*”; nesse sentido, busca “*não só identificar os problemas rapidamente como também que eles possam ser resolvidos rapidamente [...] para que o software possa ser entregue funcionando o mais rápido possível (G1)*”. Este método também apregoa a importância de “*softwares funcionais serem entregues freqüentemente (semanas, ao invés de meses (G1))*”. Similarmente ao DSDM, também não procura a perfeição desde o princípio, destacando que “*os modelos geralmente não precisam ser 100% exato, apenas precisam ser precisos o suficiente (G1)*”. O método ainda considera em suas práticas o desenvolvimento sustentável em um ritmo constante indefinidamente, enquanto durar o projeto; para isso, “*todos os envolvidos no projeto, englobando patrocinadores, analistas e usuários, devem manter o projeto em um ritmo constante até a sua conclusão, com a finalidade de garantir o desenvolvido adequado (G5)(G1)*”. Na organização do trabalho deste método, busca-se “*um processo mais interativo e menos burocrático do que os atuais (G1)*”. É válido destacar que “*o objetivo principal não é produzir extensa documentação, artefatos e mesmo modelos, [...] qualquer atividade que não contribua diretamente à entrega do software funcionando deve ser questionada e evitada, caso não possa ser justificada (G7)*”.

4.1.2 Práticas Relacionadas à Técnica

Pode até surpreender, mas a análise dos dados revela que a excelência técnica não se estabelece pela capacidade de lidar com a complexidade; pelo contrário, o aprimoramento técnico para manter a qualidade e eficiência do software está fortemente associado à simplicidade. Nesse sentido, a solução mais simples é a melhor solução.

O método AM sugere que os desenvolvedores assumam a simplicidade. Propõe não colocar recursos adicionais nos modelos, mas apenas os que são necessários hoje. Considera que é possível modelar com os requisitos de que se dispõe no momento e o sistema será reconstruído e aperfeiçoado no futuro quando novos requerimentos surgirem. Ou seja, mantenha seu modelo tão simples quanto possível. É importante perceber o propósito da modelagem no AM: *“modele para entender, modelo para comunicar (G1)”*. *“Somente os modelos que são eficientes são levados em conta, os que apresentam algum problema são eliminados, e as modificações são implementadas sempre que necessário para alcançar o melhor resultado possível (G1)”*. Ainda sobre simplicidade, apresenta orientações como: *“crie conteúdo simples; descreva modelos simples; use a ferramenta mais simples (G1)”*. Nesse enfoque, a simplicidade torna-se possível pela própria característica evolutiva e iterativa das metodologias ágeis; sugere *“manter o modelo da forma mais simples e à medida que as necessidades da empresa aumentam, o modelo é atualizado de forma que possa suprir todas as novas necessidades da organização (G1)”*.

Percebe-se, também na associação entre excelência técnica e o método AM, a inteligibilidade. *“Em várias ocasiões, é mais simples e útil um protótipo que mostre o funcionamento de uma arquitetura do que um grande e complexo diagrama [...] é mais simples para o cliente analisar o funcionamento do sistema através de um protótipo, mesmo não sendo real, do que através de um conjunto de diagramas de casos de usos e projetos de interface (G5)”*. Outra questão que se aborda é o conteúdo, predominando a visão de que *“o conteúdo é mais importante que a representação. [...] A modelagem deve levar informações a sua pretensa audiência. Um modelo sintaticamente perfeito que leva pouco conteúdo útil não é tão valioso como um modelo com notação defeituosa, mas que fornece conteúdo valioso para a sua audiência (G7)”*. Assim, o AM sugere *“simplificar ao máximo o conteúdo dos modelos, acompanhando as necessidades e expectativas dos stakeholders [...] então, o modelo deve ser simples o bastante para representar de forma clara e concisa o seu objeto [...] isto também deixa o custo da atualização o mais baixo possível (G7)”*. No mesmo sentido, recomenda *“conservar apenas aqueles modelos que fornecerão valor em longo prazo, livre-se do resto (G7)”*. De outra forma, o produto de trabalho conservado precisa ser mantido quando ocorrem modificações, gerando desperdício, ou seja, modifica-se algo que não será usado.

Similarmente ao visto anteriormente no AM, simplicidade é um elemento primordial no XP quando se busca excelência técnica. *“O projeto XP segue rigorosamente o princípio KIS (keep it simple – mantenha a simplicidade). Um projeto simples é sempre preferível em relação a uma representação mais complexa (G3)”*. *“Projeto simples significa dizer que caso o cliente tenha pedido que na primeira versão apenas o usuário ‘teste’ possa entrar no sistema com a senha ‘123’ e, assim, ter acesso a todo o sistema, você vai fazer o código exato para que esta funcionalidade seja implementada, sem se preocupar com sistemas de autenticação e restrições de acesso. (G3)”*. A simplicidade favorece que *“a equipe se concentre em fazer, primeiro, apenas aquilo que é claramente necessário e evite fazer o que poderia vir a ser necessário, mas ainda não se provou essencial. (G3)”*.

Outro ponto destacado no XP que favorece a qualidade técnica do produto é trabalhar sob um processo que favoreça encontrar erros o mais cedo possível. *“Normalmente, quanto mais cedo descobrimos um problema, menos prejuízos ele pode causar e maiores são as chances de resolvê-lo de forma barata. [...] No XP, estabelecem-se formas de encurtar ao máximo a defasagem de tempo entre o momento em que uma ação é executada e o seu resultado é observado (G3)”*. Uma proposta do XP que visa ao aumento da qualidade técnica do projeto é o trabalho em duplas. *“Geralmente, a dupla é formada por um iniciante na linguagem e outra pessoa funcionando como um instrutor. O novato é que fica codificando em uma linguagem computacional, cabendo ao instrutor acompanhá-lo, ajudando a desenvolver suas habilidades. Desta forma, o programa sempre é visto por duas pessoas, minimizando a possibilidade de defeitos. Com isto, busca-se sempre a evolução da equipe, melhorando a qualidade do código-fonte criado (G1)”*. Outra prática é a escrita do código conforme um padrão. O objetivo é *“manter um padrão de código no projeto para que qualquer um da equipe*

possa rapidamente entender o código (G4). Considera-se que *”uma forma clara de se comunicar é através do código [...] Sendo um dos valores do XP a comunicação, encontram no código uma forma eficaz de a equipe se comunicar”*.

4.1.3 Práticas Relacionadas aos Clientes

Nesse atributo, observa-se não só a colaboração do cliente na construção do sistema, como também se as iterações são guiadas pelo cliente. Nesse sentido, no XP *“participam do processo as pessoas que mais entendem sobre o negócio e que possam estabelecer prioridades para as funcionalidades a serem entregues (G1)”*. *“No início da semana, desenvolvedores e cliente reúnem-se para priorizar as funcionalidades. Essa reunião recebe o nome de Jogo do Planejamento. Os planejamentos de um release e de iterações são feitos com base nas histórias (casos de uso simplificados) e conta com a colaboração de toda a equipe de desenvolvimento, incluindo o cliente (G1)”*. Ou seja, *“os clientes e a equipe XP trabalham juntos para decidir como agrupar histórias na versão seguinte (o próximo incremento de software) a ser desenvolvida pela equipe XP (G3)(G4)”*. Na implementação das funcionalidades descritas, *“os técnicos estimam qual o esforço e riscos envolvidos para implementar as funcionalidades e comunicam ao pessoal de negócios (G1)”*. Em outra etapa, *“os testes de aceitação XP, também chamados de testes do cliente, são especificados pelo cliente e focalizam as características e funcionalidades do sistema global que são visíveis e passíveis de revisão pelo cliente (G1)”*. O método opera com a premissa de que o cliente e o desenvolvedor devem trabalhar juntos para produzir um software que tenha real valor para o cliente. É válido destacar que *“o cliente dirige a equipe de desenvolvimento sobre a forma de entregar os negócios de valor durante todo o ciclo de vida de um projeto XP (G4)”*. Outro momento que possibilita a participação do cliente no XP são as reuniões chamadas de stand-up meeting. *“É uma breve reunião realizada diariamente, normalmente de manhã, pela equipe de desenvolvimento com o objetivo de compartilhar informações sobre o projeto e priorizar suas atividades. Trata-se de um diálogo entre todos os membros da equipe, se possível envolvendo também a presença do cliente. Em qualquer diálogo presencial, existem pelo menos duas coisas que são compartilhadas de maneira bastante rica: informações e emoções (G3)”*.

No que se refere à conexão do método AM com o atributo cliente, percebe-se que cabe ao cliente *“definir o que o sistema deve ou não fazer (G5)”*. Nesse sentido, *“o cliente e a equipe de desenvolvimento tratam dos problemas previsíveis, resolvendo-os com antecedência e formalizando as soluções em um acordo documentado (G5)”*. Outros pontos destacados no método AM é a importância em transformar a mudança *“numa vantagem competitiva para o cliente (G1)”* e que *“os melhores desenvolvedores têm a humildade de reconhecer que não sabem de tudo, [...] os clientes têm seus valores e esses valores adicionam qualidade ao projeto (G1)”*.

Similarmente ao encontrado na análise do método XP, no método AM *“os elementos do grupo de desenvolvimento e os clientes devem reunir-se frequentemente, com o objetivo de discutir o andamento e obter mais informações acerca do software (G7)”*. Também se observam certas questões mais pontuais: *“o rápido feedback buscado pelo método favorece tanto o entendimento dos requisitos como o desenvolvimento de interfaces do usuário, atendendo melhor suas necessidades (G7)”*. O método AM ainda sugere que *“cada modelo apresente uma visão diferente do sistema e apenas aqueles modelos que ofereçam valor à sua pretensa audiência sejam usados (G7)”*, ou seja, não adianta mostrar tudo o que é desenvolvido ao cliente, pois há de se aplicar um filtro. É preciso haver extrema atenção a essa observação, pois *“o modelo promove comunicação entre a equipe e os stakeholders do projeto, assim como entre os desenvolvedores da equipe (G5)”*, participantes que possuem conhecimentos diferentes. Sugere *“atenção para os instintos da equipe: ouça as sugestões/reclamações de sua equipe, pois talvez o problema que a equipe encontrou poderá dificultar o restante da implementação (G5)”*. Deve-se atentar para o fato de que *“poucas comunicações nos métodos*

ágeis são unilaterais, havendo uma preferência pela colaboração, o que facilita a disseminação do conhecimento (G5)”.

4.1.4 Práticas Relacionadas ao Fator Humano

Destacam-se práticas do método ASD e Scrum neste ponto. Valores como colaboração, comunicação e *feedback* guiam as práticas do método ASD, tendo como apoio filosófico “a colaboração humana e a auto-organização da equipe (G4)(G6)(G7)”. “A auto-organização aparece quando agentes individuais independentes cooperam para criar resultados emergentes. Um resultado emergente é uma propriedade além da capacidade individual (G6)”. Ao se procurar maiores detalhes sobre o método no que concerne à auto-organização, “o método ASD cita apenas como papéis o responsável executivo e os participantes de sessões (workshops). O responsável executivo é o responsável geral do processo e os participantes de sessões de desenvolvimento correspondem ao facilitador, ao escriba e ao cliente (G4), porém não apresenta uma definição detalhada das responsabilidades dos participantes. “Outra técnica usada no ASD é que a equipe monitore seu próprio desempenho [...] revisão de pessoas e processos (G4)”, também opera no sentido de que “a equipe obtenha seus objetivos totalmente definidos, podendo assim ter decisões com maior claridade (G4)”. No que se refere à colaboração, observa-se que “o pessoal motivado trabalha junto de um modo que multiplica seus talentos e resultados criativos além de seu número absoluto [...] a colaboração é acima de tudo, uma questão de confiança (G6)”.

No caso do Scrum, o atributo humano distingue-se pela dinâmica do processo que guia a *sprint* (uma iteração). *Sprint* é considerada a principal prática do Scrum, onde são implementados os itens de trabalho definidos no *product backlog* (lista do que há para fazer) pela equipe Scrum. “O padrão de processo scrum permite a uma equipe de desenvolvimento de software trabalhar de modo bem-sucedido em um mundo em que a eliminação da incerteza é impossível (G1)”. Nesse sentido, ter um processo para lidar com a incerteza parece ser fundamental para a satisfação no trabalho. Igualmente importante no processo são as reuniões, que levam à “socialização do conhecimento e promovem, assim, uma estrutura de equipe auto-organizada (G1)”. Pautam-se por “encorajar a comunicação verbal entre todos os membros da equipe e entre todas as disciplinas que estão envolvidas no projeto (G1)”. Ainda, observa-se que este método enfatiza bem a criação de equipes que se organizam e delegam tarefas a cada indivíduo. “O objetivo é que cada um saiba o que deve fazer (G1)”. Nesse sentido, faz parte da metodologia “a fixação de como o time deve trabalhar, sendo o primeiro passo para o desenvolvimento por meio do Scrum. Nesse método, “pequenas equipes de trabalho são organizadas de modo a maximizar a comunicação, minimizar a supervisão e maximizar o compartilhamento de conhecimento tácito operacional (G2)”.

4.1.5 Práticas Relacionadas à Aprendizagem

O processo de desenvolvimento de software precisa ser “adaptável tanto a modificações técnicas quanto de negócios (G1)”. Nesse sentido, o método Scrum “produz freqüentes incrementos de software que podem ser inspecionados, ajustados, testados, documentados e expandidos [...] itens podem ser adicionados às pendências a qualquer momento, é assim que as modificações são introduzidas (G1)”; no entanto, “pode-se alterar os requisitos a qualquer momento, desde que os itens alterados não estejam na *sprint* (G2)”. Ao final de cada iteração (*sprint* no Scrum), é feita uma reunião para revisão e demonstração do *sprint*. Nessa, todos os membros da equipe refletem sobre o *sprint* passado.

Percebem-se outras práticas associadas à aprendizagem no método ASD. “Como a aprendizagem é uma característica importante e contínua no método, assume que os planos e desenhos devem-se alterar na medida em que o desenvolvimento progride (G7)”. Diferente dos métodos tradicionais, no ASD, as

mudanças são bem-vindas. *“Operam segundo processos que se adaptam e prosperam na mudança, chegando ao ponto de mudarem a si próprios (G7)”*. Na construção do software, a proposta do método é que a equipe do projeto continuamente se pergunte: *“O que temos aprendido até agora, o que isso mudou nossa perspectiva sobre onde precisamos chegar? (G4)”*. Os membros costumam revisar os componentes de software que são desenvolvidos, aperfeiçoando a qualidade e aprendendo à medida que prosseguem. Observam que o ciclo dinâmico baseado em aprendizagem e adaptação contínua faz emergir o estado do projeto. Após *feedback* dos clientes e usuários, indicando se o produto está ou não satisfazendo às necessidades do negócio há a *“pós-conclusão: a equipe ASD torna-se introspectiva, cuidando do seu próprio desempenho e processo, com a intenção de aprender e depois aperfeiçoar a sua abordagem (G6)”*. De modo geral, aparecem quatro categorias gerais de coisas para se aprender ao final de cada interação: qualidade a partir da perspectiva do cliente, qualidade a partir de uma perspectiva técnica, o funcionamento da entrega e práticas que membros da equipe estão utilizando, e o status do projeto.

Similarmente ao visto no ASD, outros exemplos de práticas podem ser extraída da família Crystal de métodos ágeis. Busca o melhoramento da metodologia em cada incremento: *“o time pode aprender a usar o conhecimento ganho para otimizar o próximo incremento (G3)”*. O Crystal faz uso de *workshops* reflexivos: *“são reuniões que ocorrem antes e depois de cada iteração com o objetivo de analisar o progresso do projeto e formas de ajustar o seu andamento (G3)(G5)”*. Fora os *workshops*, o *“constante contato com o cliente e a entrega de incrementos em períodos regulares facilita visualizar uma possível mudança de requisitos (G6)”* e a realização dos ajustes no sistema. Segundo os criadores do método, *“suas fases – paralelismo e fluxo; monitoramento de progresso; construção, demonstração, revisão; utilização por parte dos usuários – auxiliam no processo de flexibilização e na introdução de novos conceitos ao projeto (G6)”*.

Buscou-se, nesta seção, compreender melhor os métodos ágeis por meio de suas práticas agrupadas em princípios. Na seção subsequente, especulam-se possíveis contribuições dessas práticas em questões que suplantam as relacionadas ao desenvolvimento de software, avançando na concepção de processos que potencializem a inovação.

4.2 Métodos Ágeis à Luz do Modelo de Meta-estruturação e Estruturação da Tecnologia

As estruturas, práticas e modalidades presentes nas organizações atualmente são bem diferentes das da época vivenciada pelos primeiros desenvolvedores de sistemas de informação computadorizados. Nessa dinâmica social, erguem-se novas formas de organizar o trabalho e não poderia ser diferente no desenvolvimento de software, despontando, no momento atual, o movimento em torno dos métodos ágeis.

No sentido de compreender melhor a inserção dos métodos ágeis nas organizações, desenvolve-se uma reflexão a partir do modelo proposto por Orlikowski *et al.* (1995). O modelo incluindo o processo de meta-estruturação da tecnologia em uso (Figura 4) é uma evolução do modelo estruturacional da tecnologia proposto por Orlikowski (1992), que teve origem principalmente na teoria da estruturação de Giddens (1984). Certamente que a pretensão desta reflexão, ao envolver elementos de tamanha profundidade, não é esgotar o assunto; pelo contrário, propõe-se iniciar a discussão.

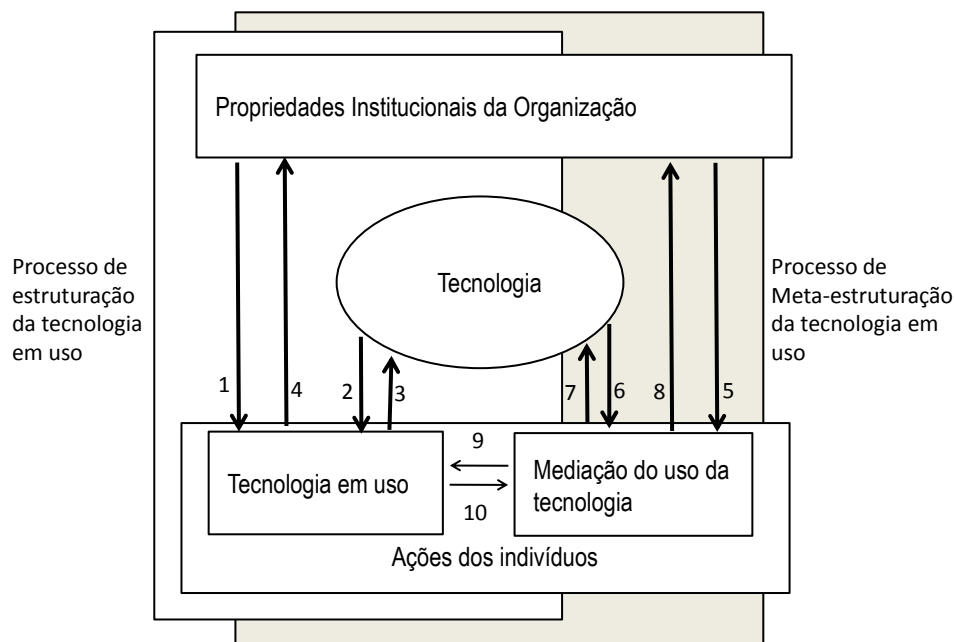


Figura 4: Processos de meta-estruturação e estruturação da tecnologia
Fonte: Orlikowski *et al.* (1995)

Por um lado, o modelo destaca o processo de meta-estruturação que ocorre durante a introdução de novidades tecnológicas; por outro, o processo de estruturação que ocorre em função do uso contínuo da tecnologia pelos usuários. À direita, tem-se que mediadores (equipe de desenvolvimento, no presente estudo) são influenciados pelo contexto institucional no qual eles trabalham, provendo-lhes certos recursos, autoridade e responsabilidades (seta 5). Nesse momento, já há uma concessão para usar práticas dos métodos ágeis no desenvolvimento de sistemas.

Na sequência, a equipe de desenvolvimento adota uma tecnologia mais apropriada para o desenvolvimento usando métodos ágeis, sendo habilitada e restringida por ela (seta 6). É válido destacar que, atualmente, já é possível adaptar reconhecidos softwares para controle de projetos, como o MS-Project, para certos métodos ágeis como o Scrum. À medida que as práticas dos métodos ágeis são adotadas, elas influenciam tanto a tecnologia a ser disponibilizada para os usuários como as que estão sendo adotadas no desenvolvimento. Na ação, mediadores criam aplicações, políticas, procedimentos, padrões, mecanismos de acesso e configurações físicas que alteram as características dos artefatos tecnológicos (seta 7). Nesse sentido, à medida que a equipe ganha mais conhecimento sobre a tecnologia, ocorre a melhoria no produto, como foi visto nas práticas relacionadas ao princípio da excelência técnica. Paralelamente, ainda em fase de meta-estruturação, as práticas influenciam as propriedades institucionais da organização (seta 8) e, adicionalmente, a mediação da tecnologia influencia a interpretação dos usuários, fornecendo-lhes entendimentos, imagens, conceitos, conhecimentos e heurísticas sobre a tecnologia (seta 9). Ao fazer isso, mediadores influenciam hábitos e percepções dos trabalhadores em torno da tecnologia e do método de desenvolvimento.

Percebe-se que inúmeros artefatos da TI amplamente adotados nas organizações tiveram seu início despertado pelo uso do pessoal de TI; para citar apenas alguns exemplos, diversos hábitos associados à Internet como uso de *e-mails*, fóruns de discussão, suporte *online* e *e-business* em suas diferentes formas foram rapidamente adotados pelo pessoal técnico, para, após, espriarem-se pela

organização. Em muitos casos, deu-se a meta-estruturação da tecnologia por meio da troca de conhecimento tácito sem qualquer processo formal.

Simetricamente ao processo de meta-estruturação, tem-se o processo de estruturação. A frequente entrega de software funcionando (seta 2) conjuntamente com outras práticas dos métodos ágeis favorece a aproximação entre a tecnologia disponibilizada e a tecnologia em uso, agilizando o *feedback* para a equipe de mediação (seta 10). Não obstante, políticas, padrões e configurações são aceitos, legitimados e reforçados na apropriação da tecnologia pelos usuários (setas 3 e 4), tanto em termos do que deriva da tecnologia em uso como do que provém do método de desenvolvimento em uso, de tal sorte que a recorrência da experiência da prática de uma nova forma de trabalho altere as propriedades institucionais da organização, influenciando (setas 1 e 5) as ações dos indivíduos, usuários e desenvolvedores.

Adotou-se, nesta seção, o modelo de Orlikowski *et al.* (1995) a fim de melhor perceber as relações e co-evolução de três elementos: tecnologia, organização do trabalho (desenvolvimento e uso da tecnologia) e propriedades institucionais da organização, destacando que essas propriedades são influenciadas tanto pela tecnologia como pelo método de trabalho.

Examinando as práticas dos métodos ágeis (seção 4.1) frente ao modelo de meta-estruturação e estruturação da tecnologia, percebe-se que, embora muitas práticas dos métodos ágeis estejam amparadas no princípio da iteratividade e questões técnicas, passando pelas relações dos atores com a tecnologia (setas 2, 3, 6 e 7), a grande mudança dos métodos ágeis em relação aos predecessores está no subsistema social. Ela reside na profunda atenção dedicada às relações entre os atores que desenvolvem a tecnologia e os que a colocam em uso, por meio de práticas associadas aos atributos cliente, humano e aprendizagem (setas 9 e 10) até então pouco valorizadas nas metodologias tradicionais de desenvolvimento de software, preferindo centrar o foco em questões essencialmente técnicas. Assim, nota-se que é possível aproximar, por meio dos métodos ágeis, a tecnologia disponibilizada à tecnologia em uso (tecnologia na prática). A abordagem sociotécnica auxilia tal análise. Nessa perspectiva, o subsistema técnico (tecnologia *per se*) possui uma eficiência potencial, porém é o subsistema social (as pessoas e as relações sociais) que transforma a eficiência potencial em eficiência real. No que diz respeito aos princípios institucionais da organização, os métodos anteriores nunca ousaram trabalhar fora de princípios rígidos geralmente oriundos da Engenharia e da Economia, subentendidos e reforçados na prática. Ao contrário, os métodos ágeis expõem novos princípios e suas práticas reforçam esses novos princípios, principalmente na fase de meta-estruturação (setas 5 e 8).

4.3 Métodos Ágeis e o Processo de Inovação das Empresas – Implicações para a Prática

Como observado na introdução deste artigo, a área de TI está sendo cobrada a liderar o processo de inovação das empresas. Esse movimento em breve encontrar-se-á com o do desenvolvimento ágil. Nesta seção, discutem-se alguns pontos desta intersecção.

A própria adoção e estruturação dos métodos ágeis em substituição a outros mais tradicionais no desenvolvimento de sistemas já caracteriza a introdução de uma inovação. Em princípio, poderia ser vista como uma inovação em processo. Conforme Albertin (2005, 16), “a inovação em processo é aquela que afeta o processo de produção em todas as suas ramificações, incluindo a transformação desde a matéria prima até o produto acabado e todas as atividades de suporte associadas a ela”. No entanto, com base no que foi exposto anteriormente sobre a estruturação do método de trabalho em conjunto com a da tecnologia, percebem-se muito mais características de uma inovação em atividades administrativas: “é aquela que envolve o componente administrativo e afeta o sistema social da organização, que inclui a estrutura social, regras, procedimentos, sistemas de informação e

comunicação e estruturas de autoridade que governam a relação entre seus membros” (Albertin, 2005, p. 17).

Em um estágio mais avançado, especula-se, ainda, a possibilidade de o desenvolvimento ágil fornecer elementos importantes para a criação de um novo programa de melhoria como gestão da qualidade total e *balanced scorecard*, tendo como objetivo principal o gerenciamento da inovação do modelo de negócio das empresas.

Conforme Albertin (2005), o desempenho empresarial é influenciado pelo uso de TI, porém a relação entre organização e TI é que dirigirá a TI como simples decorrência das diretrizes organizacionais até a TI como facilitadora e fornecedora de inovações de negócio. No que se refere a inovações de negócios, é válido observar que toda empresa estabelecida explícita ou implicitamente emprega um modelo de negócio. Assim, inovações de negócio são mudanças nesse modelo. A essência de um modelo de negócio está na definição da maneira pela qual a empresa entrega valor aos clientes, atrai os clientes para pagar o valor e converte os pagamentos em lucro. Assim, reflete a hipótese da administração da empresa sobre o que os clientes querem, como querem, como a empresa pode se organizar para melhor atender a essas necessidades, como receber o pagamento para atender e ter um lucro (Teece, 2010).

Para as empresas, inovações em modelos de negócio são tão importantes quanto em produtos. É válido observar que brilhantes invenções de produtos fracassaram por não ter um modelo de negócio apropriado e que muitas inovações em modelos de negócio proveram excelentes resultados para as empresas sem qualquer inovação nos produtos comercializados. No entanto, apenas mais recentemente, em função da TI e da redução dos custos de comunicação, principalmente com a Internet, é que as inovações nos modelos de negócio aparecem com maior frequência e diversidade, ganhando destaque nas empresas e na academia (Baden-Fuller, 2010). Não obstante, à medida que as empresas percebem o potencial das inovações nos modelos de negócio nesse novo contexto tecnológico, torna-se fundamental às empresas a capacidade de dispor de softwares diferenciados e adaptáveis, bem como melhorar o seu gerenciamento. Uma situação bem diferente da vivenciada nas décadas de 1980 e 1990, em que os pacotes de software ERPs estimularam gestores a pensar em software de gestão como *commodity*.

Projetar um novo modelo de negócios exige criatividade, percepção e uma boa dose de informação e inteligência sobre concorrentes, clientes e fornecedores, podendo haver um significativo componente tácito, observa Teece (2010). Um empreendedor pode ser capaz de intuir um novo modelo, mas não ser capaz de racionalizá-lo e articulá-lo plenamente; experimentação e aprendizagem fazem-se necessários. Além disso, conforme Teece, um bom modelo de negócio pode não ser configurado de imediato, sendo necessário aprendizado e ajustes. E, mesmo estabelecido, é provisório, no sentido em que, ao longo do tempo, é substituído por um modelo melhorado que tira proveito de outras inovações tecnológicas ou organizacionais. Assim, os empresários mais propensos ao sucesso são os que têm um bom (mas não perfeito) modelo de negócio e são capazes de aprender com ele, avaliá-lo contra o estado corrente do ecossistema de negócio e ajustá-lo. Alerta, ainda, que, embora o desenho de um bom modelo de negócio seja, em parte, uma “arte”, as chances de sucesso são maiores quando empreendedores e gerentes têm um profundo conhecimento das necessidades dos usuários, são bons ouvintes e aprendem rápido.

No momento, os métodos de gestão de inovação de modelos de negócios ainda estão em fase embrionária. Quando se pensa na necessidade das empresas com foco em inovação no modelo de negócio, em inúmeros fatores o desenvolvimento ágil se distingue positivamente em relação aos métodos de desenvolvimento que os precederam. Também se observa que muitas das questões apresentadas sobre inovação em modelos de negócio estão associadas ao papel que a TI ocupa na organização, em especial, o do software, e, por conseguinte, o seu método de desenvolvimento. Nesse

sentido, os objetivos dos métodos ágeis são mais bem entendidos por empresas que percebem a possibilidade de ganhar vantagem competitiva por meio do gerenciando da inovação do seu modelo de negócio. Nesse caso, a introdução das práticas dos métodos ágeis pode resultar em propriedades institucionais úteis às organizações desejosas em examinar e inovar mais sistematicamente seus modelos de negócio.

Nesse contexto, que integra inovação nos modelos de negócio e software, claramente, práticas relacionadas à aprendizagem auxiliarão o processo de inovação. É válido destacar o apresentado no referencial teórico (seção 2.1), em que um dos cinco fatores subjacentes aos princípios da agilidade é a “predisposição à aprendizagem adotando a mudança como geradora de vantagem competitiva para o cliente”. A interdependência dos modelos de negócio com a TI exige uma nova cultura de desenvolvimento de sistemas de informação que perceba não mais a mudança como um erro de definição ou uma adaptação necessária. Assim, associar mudança à vantagem competitiva da empresa parece ser um diferencial importante quando já se sabe que o processo de inovação do modelo de negócio exige aprendizagem, adaptações e ajustes.

Outro aspecto a considerar são as particularidades da iteratividade na gestão de projetos e programas. Pouco se domina em termos de gerenciamento de projetos (ou outro sistema produtivo), onde a relação entre as fases não seja do tipo sequencial ou sobreposta. Conforme o PMI (2008, p. 22), “numa relação iterativa, apenas uma fase está planejada a qualquer momento e o planejamento da próxima é feito à medida que o trabalho avança na fase atual e nas entregas”. Cita, ainda, que esta abordagem é muito útil em ambientes muito indefinidos, incertos ou em rápida transformação, como pesquisas. Embora o PMBOK seja referência em termos de conhecimento em gerenciamento de projetos, faz somente esse alerta sobre projetos com relação iterativa entre fases, tratando apenas questões relacionadas a uma fase.

Nesta seção, tomando como base a necessidade de as empresas administrarem e inovarem seus modelos de negócio e vislumbrando a adequação dos métodos ágeis a essa necessidade, especula-se a ideia de que, por meio dos métodos ágeis, seja possível obter propriedades institucionais na organização e resultados nos negócios que auxiliem os gestores de TI a desempenhar o tão solicitado papel de liderar o processo de inovação das empresas.

5 Conclusão

No presente artigo, primeiramente, apresentaram-se conceitos contextualizados às práticas do desenvolvimento ágil de software. Após, descreveram-se práticas dos métodos ágeis agrupadas por princípio, mostrando que, embora os métodos ágeis partam dos mesmos princípios, possuem práticas particulares. Preferiu-se realizar uma análise conjunta dos métodos, em vez de se concentrar e aprofundar no que dispõe um único método, citando os métodos apenas para efeito de ilustração. Assim, outros estudos podem detalhar mais incisivamente cada método e apresentar outras práticas.

A investigação do fenômeno dos métodos ágeis é propícia a inúmeras lentes teóricas, escolhendo-se, para esta investigação, a teoria das práticas e o modelo de meta-estruturação e estruturação da tecnologia de Orlikowski *et al.* (1995). Percebeu-se, por meio dele, a possibilidade de explorar a co-evolução da tecnologia, da organização do trabalho e das propriedades institucionais da organização, sendo essas propriedades influenciadas tanto pela tecnologia como pelo método de desenvolvimento; ao mesmo tempo, auxiliou a visualizar o direcionamento do foco dos métodos ágeis no indivíduo e suas ações nas relações sociais.

Certamente, fazem-se necessários outros estudos que aprofundem a investigação do fenômeno sob a mesma lente teórica, tanto no que se refere aos elementos do modelo como nas inter-relações destacadas. A abordagem adotada, meramente teórica, não havendo uma validação empírica, é uma

das limitações do estudo. No entanto, tal limitação já sugere futuras pesquisas que tenham por objetivo, entre outros, verificar a hipótese levantada de que a introdução dos métodos ágeis visando inicialmente a resolver problemas relacionados a software pode resultar em propriedades institucionais que auxiliem a tão demandada liderança dos gestores de TI na inovação das empresas.

Adicionalmente, percebe-se a possibilidade de investigar o impacto do uso de métodos ágeis em diversas questões-chave apontadas por executivos de TI, como, por exemplo, responder rapidamente às mudanças, prover flexibilidade, reduzir a complexidade, entre outras identificadas por estudos como os de Luftman (2004), Luftman & Ben-Zvi (2010) e Li *et al.* (2010). Até mesmo questões como reter profissionais da área de TI podem ser influenciadas positivamente por um processo de desenvolvimento de software adequado.

Desenvolvedores de sistemas de informação, a partir da prática, propuseram certos valores em substituição a outros que já estavam estabelecidos, formularam os princípios da agilidade e iniciaram a divulgação de seus métodos e respectivas práticas. No presente artigo, buscou-se compreender melhor tais métodos, teorizar a respeito e especular potencialidades de ordem prática.

Referências

- Agerfalk, P.; Fitzgerald, B.; Slaughter, S. Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research*, 20, 2009.
- Albertin, A. *Benefício do uso da tecnologia da informação no desempenho empresarial*. Relatório (GVPesquisa, 07/2005). São Paulo: Fundação Getúlio Vargas, 2005. <http://bibliotecadigital.fgv.br/dspace/handle/10438/3089>
- Ambler, S. *Agile modeling: Best practices for the unified process and extreme programming*. New York: John Wiley & Sons, 2002.
- Barley, S.; Tolbert, P. Institutionalization and structuration: Studying the links between action and institution. *Organization Studies*, 18, 1997.
- Beck, K. *Extreme programming explained: Embrace change*. Reading: Addison-Wesley, 1999.
- Beck, K.; Fowler, M. *Planning extreme programming*. Reading: Addison-Wesley, 2001.
- Bellini, C.; Pereira, R.; Becker, J. Measurement in software engineering: From the roadmap to the crossroads. *International Journal of Software Engineering & Knowledge Engineering*, 18, 2008.
- CIO Magazine. *Estudo inédito traça o perfil do líder de TI brasileiro*. 2009. <http://cio.uol.com.br/carreira/2009/10/02/um-retrato-do-lider-de-ti-brasileiro>
- CIO Magazine. *Gestores de TI terão de influenciar a inovação nas organizações*. 2010. <http://cio.uol.com.br/gestao/2010/09/28/gestores-de-ti-terao-de-influenciar-a-inovacao-nas-organizacaoes>
- Coad, P.; Luca, J.; Lefebvre, E. *Java modeling in color with UML*. Englewood Cliffs: Prentice Hall, 1999.
- Cockburn, A. *Crystal clear: A human-powered software development methodology for small teams*. Reading: Addison-Wesley, 2001.
- Conboy, K. Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20, 2009.
- Cummings, T.; Worley, C. *Organization development & change*. Mason: South-Western Cengage Learning, 2009.
- Dolci, D. Modelo geométrico de representação dos métodos ágeis de desenvolvimento de sistemas de informação em função dos princípios da agilidade. *III Encontro de Administração da Informação*. Porto Alegre: ANPAD, 2011.
- Feldman, M.; Orlikowski, W. Theorizing practice and practicing theory. *Organization Science*, 22, 2011.
- Fortino, A. Whither the CIO? Evolution from keeper of the infrastructure to firm innovator. *Portland International Conference on Management of Engineering & Technology 2008*. Cape Town: PICMET, 2008.
- Fowler, M.; Highsmith, J. The agile manifesto. *Software Development Magazine*, 9, 2001. <http://www.drdoobs.com/184414755>
- Gash, D.; Orlikowski, W. Changing frames: Towards an understanding of information technology and organizational change. *51st Annual Meeting of Academy of Management*. Miami Beach: Academy of Management, 1991.
- Giddens, A. *The constitution of society*. Berkeley: University of California, 1984.
- Giddens, A. *A constituição da sociedade*. São Paulo: Martins Fontes, 1989.
- Highsmith, J. *Agile software development ecosystems*. Reading: Addison-Wesley, 2002.
- Kotlarsky, J.; Oshri, I. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems*, 14, 2005.
- Larman, C.; Basili, V. Iterative and incremental development: A brief history. *Computer*, 36, 2003.
- Laudon, K.; Laudon, P. *Sistemas de informações gerenciais*. São Paulo: Pearson Prentice Hall, 2007.
- Li, D.; Huang, W.; Luftman, J.; Sha, W. Key issues in information systems management: An empirical investigation from a developing country's perspective. *Journal of Global Information Management*, 18, 2010.
- Luftman, J.; Ben-Zvi, T. Key issues for IT executives 2009: Difficult economy's impact on IT. *MIS Quarterly Executive*, 9, 2010.

- Luftman J.; Kempaiah R.; Rigoni E. Key issues for IT executives 2008, *MIS Quarterly Executive*, 8, 2009.
- Maruping, L.; Venkatesh, V.; Agarwal, R. A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20, 2009.
- Mintzberg, H.; Ahlstrand, B.; Lampel, J. *Safári de estratégia*. Porto Alegre: Bookman, 2000.
- Nidumolu, S.; Subramani, M.; Aldrich, A. Situated learning and the situated knowledge web: Exploring the ground beneath knowledge management. *Journal of Management Information Systems*, 18, 2001.
- O'Brien, J.; Marakas, G. *Administração de sistemas de informação: Uma introdução*. São Paulo: McGraw-Hill, 2007.
- Orlikowski, W. The duality of technology: Rethinking the concept of technology in organizations. *Organization Science*, 3, 1992.
- Orlikowski, W. Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science*, 11, 2000.
- Orlikowski, W.; Robey, D. Information technology and the structuring of organizations. *Information Systems Research*, 2, 1991.
- Orlikowski, W.; et al. Shaping electronic communication: The metastructuring of technology in the context of use. *Organization Science*, 6, 1995.
- PMI. *Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK)*. Pennsylvania: Project Management Institute, 2008.
- Pressman, R. *Engenharia de software*. São Paulo: McGraw-Hill, 2006.
- Reckwitz, A. Toward a theory of social practices: A development in culturalist theorizing. *European Journal of Social Theory*, 5, 2002.
- Sidky, A.; Athur, J. A disciplined approach to adopting agile practices. 2007. <http://arxiv.org/ftp/arxiv/papers/0704/0704.1294.pdf>
- Schwaber, K.; Beedle, M. *Agile software development with Scrum*. Upper Saddle River: Prentice Hall, 2002.
- Stapleton, J. *DSDM: Dynamic systems development method*. Harlow: Addison-Wesley, 1997.
- Teece, D. Business models, business strategy and innovation. *Long Range Planning*, 43, 2010.
- Turban, E.; Leidner, D., McLean; E., Wetherbe, J. *Tecnologia da informação para gestão*. Porto Alegre: Bookman, 2010.
- Wood, T. *Mudança organizacional*. São Paulo: Atlas, 1995.



Décio Bittencourt Dolci fez doutorado (2005) e mestrado (2000) em Administração pela Universidade Federal do Rio Grande do Sul (UFRGS), com estágio doutoral (2004) na Florida Atlantic University (FAU) no College of Business, Department of Information Technology. Possui especializações em Administração (1993) pela Universidade Federal do Rio Grande (FURG) e em Análise de Sistemas (1986) pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Graduou-se (1983) em Engenharia Mecânica pela FURG. Prestou consultoria em TI a diversas empresas da Região Sul do Brasil e foi empresário na área de software. Atualmente, é professor do Instituto de Ciências Econômicas, Administrativas e Contábeis da Universidade Federal do Rio Grande (FURG), atuando nos cursos de Administração, Sistemas de Informação e Tecnologia em Análise de Desenvolvimento de Sistemas. Seus trabalhos de pesquisa, extensão e inovação normalmente objetivam relacionar Administração e Tecnologia da Informação, sendo suas principais áreas de interesse: utilizações organizacionais da TI, inovações organizacionais, desenvolvimento de sistemas de informação e gestão de projetos.

Mais informações curriculares em: <http://lattes.cnpq.br/2325977323861550>

Contato: dbdolci@gmail.com