Aspectos Sociotécnicos do Desenvolvimento de Software Utilizando Scrum em um Caso Prático

Antonio F. dos Santos Júnior[1], Rodrigo Pereira dos Santos[2]

[1] CETELI, Universidade Federal do Amazonas, Brasil
CEP 69.077-000 – Manaus, AM

[2] COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Brasil
Caixa Postal 68511 – CEP 21.945-970 – Rio de Janeiro, RJ

ajunior.analista@ig.com.br, rps@cos.ufrj.br

Abstract. *In business atmosphere, quality and agility are important principles to keep a company competitive. Considering that software development companies live in business atmospheres subject to frequent changes, Software Engineering needs to consider technical and non-technical factors together, in software processes, analyzing their different styles. So, this paper presents the practical case of a company assessed at CMMI which implemented Scrum (compliant to ISO 9001:2000), in order to capture sociotechnical aspects in sceneries where Scrum is used during software development.*

Resumo. *No ambiente empresarial, qualidade e agilidade compreendem princípios fundamentais para manter uma empresa competitiva. Considerando que empresas de desenvolvimento de software convivem em ambientes de negócios sujeitos a mudanças freqüentes, cada vez mais a Engenharia de Software precisa considerar, em seus processos, os fatores técnicos e não-técnicos envolvidos em conjunto, analisando suas diferentes vertentes. Este artigo apresenta o caso prático de uma empresa avaliada CMMI, que implementou Scrum (aderente à ISO 9001:2000), a fim de capturar aspectos sociotécnicos em cenários que utilizam o Scrum durante o desenvolvimento de software.*

1. Introdução

No ambiente empresarial, *qualidade* e *agilidade* compreendem princípios fundamentais para manter uma empresa competitiva [Boehm & Turner, 2004]. Nesse sentido, deve-se almejar a redução do distanciamento que ocorre entre as etapas de concepção e de entrega do produto de software [Boehm, 2003] e encarar a mudança como uma característica de qualquer projeto, que requer espírito de equipe e que propicia a aprendizagem, e não como um problema ou falha de projeto [Ferreira & Lima, 2006]. Considerando esse contexto, percebe-se que cada vez mais a Engenharia de Software (ES) precisa considerar os fatores não-técnicos envolvidos, a fim de verificar que o sucesso de um projeto não se deve apenas ao correto cumprimento de atividades e boas práticas e que as falhas existentes não decorrem apenas desses fatores [Teixeira & Cukierman, 2007]. Assim, destaca-se a importância de se relacionar fatores técnicos e não-técnicos e entender a influência dessa combinação no cenário industrial e a sua "co-modificação" a partir da experiência social, somente percebida por uma aproximação concomitantemente social e técnica. Conforme Cukierman *et al.* (2007), isso representa o *olhar sociotécnico*, o qual almeja apreender a ES sem fragmentá-la em fatores (ou aspectos) técnicos de um lado e

fatores (ou aspectos) não-técnicos de outro; sem fatorá-la em quaisquer outras dualidades ("fatores técnicos" *versus* "fatores humanos, organizacionais, éticos, políticos, sociais etc.") que terminem por desfigurar o "pano sem costura" que imbrica, na ES, o técnico e o social em um mesmo e indivisível "tecido".

Conforme Santos (2008a), torna-se interessante para a ES investigar elementos para uma abordagem sociotécnica de processos de software, uma vez que existem diferentes vertentes de desenvolvimento, tais como a tradicional (e.g., CMMI, ISO etc.) e a ágil (e.g., *Extreme Programming*, *Scrum* etc.), e tais vertentes podem apresentar possíveis combinações e similaridades. Visando capturar aspectos sociotécnicos em cenários que se baseiam no desenvolvimento de software utilizando *Scrum*, este artigo apresenta o caso prático de uma empresa, cujos processos, aderentes ao CMMI num primeiro momento, sofreram alterações decorrentes de algumas mudanças ambientais, de forma que a empresa passou a utilizar o *Scrum* (posteriomente aderente à ISO). Com essa finalidade, será apresentada uma visão geral do desenvolvimento de software com *Scrum* (Seção 2). A partir disso, será analisado o caso prático de uma empresa de software e sua fase de transição durante a adoção de vertentes de processos distintas, buscando-se por aspectos sociotécnicos e impactos desta transição (Seção 3). Por fim, traçam-se as considerações finais (seção 4).

2. Uma Visão Geral do Desenvolvimento de Software com *Scrum*

De acordo com Schwaber (2004), o *Scrum* é uma metodologia[1] ágil para gestão e planejamento de projetos de desenvolvimento de software. O *Scrum* foi desenvolvido por Ken Schwaber e Mike Beedle na década de 1990, baseando-se em experiências no desenvolvimento de sistemas e processos, a partir do reconhecimento de que o desenvolvimento de software é muito complexo para ser planejado corretamente desde o início. O *Scrum* contempla uma visão empírica baseada em aspectos teóricos de controle de processos, ou seja, parte do pressuposto de que nem todas as características do produto são conhecidas na análise e que os requisitos mudarão com o passar do tempo. Contempla, ainda, duas atividades principais: *inspeção* e *adaptação*. Como o processo não é definido, o gerente de projeto tem que inspecionar a execução diariamente, o que requer transparência, e fazer as adaptações necessárias com o passar do tempo.

O desenvolvimento é dividido em iterações (*sprints*), de até trinta dias, e equipes pequenas, de até dez pessoas, reunindo projetistas, programadores, engenheiros e gerentes de qualidade – a lista de tarefas que o time (*team*) se compromete a fazer em um *sprint* se chama *sprint backlog*. As equipes trabalham em cima de funcionalidades (i.e., requisitos) definidas no início de cada *sprint* e cada equipe fica responsável pelo desenvolvimento destas funcionalidades, isto é, dos *product backlog items* (conjunto de requisitos priorizados, na forma de uma lista de itens, que devem ser desenvolvidos para o produto). Visando congregar a equipe, existem reuniões de acompanhamento diárias (*daily meetings*), preferencialmente de curta duração (aproximadamente quinze minu-

[1] Existe uma confusão entre os termos *método*, *metodologia* e *processo* de desenvolvimento de software devido à inexistência de um consenso universal, o que se percebe pela divergência entre as definições apresentada por diferentes dicionários [Improve It, 2009]. Conforme o Dicionário Priberam da Língua Portuguesa [DPLP, 2009], *método* é o processo racional que se segue para chegar a um fim, *metodologia* é a subdivisão da lógica que estuda os métodos técnicos e científicos e *processo* é a maneira de operar, de agir. Neste artigo, considera-se o *Scrum* como uma metodologia ágil para facilitar a gestão de projetos de desenvolvimento de software.

tos), nas quais são discutidos três pontos: o que foi feito desde a última reunião (dia anterior), quais as dificuldades e os fatores de impedimento (*bottlenecks*) e o que precisa ser priorizado no dia que se inicia. Existem também as chamadas *retrospective meetings*, que consistem em reuniões realizadas ao final de cada *sprint* com o objetivo de relatar os pontos fortes e fracos deste *sprint*, bem como definir ações para manter os pontos fortes e eliminar ou reduzir os pontos fracos relatados [Schwaber & Beedle, 2002]. Nessa lista de reuniões, pode-se incluir também a *review meeting*, na qual o time mostra o que foi alcançado durante o *sprint*, tipicamente na forma de *demos* das novas funcionalidades. Por fim, a *sprint planning meeting* é uma reunião na qual todos os *stakeholders* envolvidos estão presentes para planejar a próxima *sprint*. Nesse contexto, o ciclo de vida do *Scrum* é baseado em três fases principais, divididas em sub-fases: pré-planejamento (*pre-game phase*), desenvolvimento (*game phase*) e pós-planejamento (*post-game phase*) [Schwaber & Beedle, 2002].

Conforme Siqueira (2007), processos empíricos aceitam as falhas como conseqüência natural da produção e tentam torná-las visíveis e passíveis de correção, o mais cedo possível, para que a abordagem empírica se torne ideal para o ambiente de desenvolvimento de software. Nesses ambientes, as funcionalidades consideradas "terminadas" muitas vezes não estão suficientemente testadas e aceitas pelo cliente, fazendo-se necessárias atividades de inspeção e de adaptação. Além disso, deve-se considerar a existência de diferentes papéis (agentes independentes) no *Scrum*, com o intuito de atacar a complexidade do desenvolvimento e gerenciamento de projetos de software a partir da implantação de um controle descentralizado, capaz de lidar com contextos pouco previsíveis e de forma mais eficiente [Schwaber, 2004]: (i) *product owner* (cliente): responsável pelo gerenciamento do retorno sobre o investimento do projeto, verificando se o produto entregue atende aos requisitos do projeto e priorizando as funcionalidades que devem ser entregues e aquelas de maior valor ao projeto; (ii) *Scrum team* (time): *stakeholders* envolvidos no desenvolvimento do projeto (e.g., analistas, desenvolvedores etc.), que almejam objetivos estabelecidos; (iii) *Scrum master*: responsável por guiar o time e agir como *broker*, intermediando negociações entre o *product owner* e a equipe do projeto, além de ensinar e acompanhar a utilização do *Scrum*.

O processo *Scrum* realiza uma distinção entre os *stakeholders* responsáveis (*pigs*) e aqueles que estão apenas envolvidos, tais como clientes e executivos (*chickens*) [Schwaber, 2004]. Isso é importante para a visibilidade das responsabilidades sobre o projeto (e.g., da mesma forma que são atribuídas as responsabilidades para os *pigs*, o *Scrum* estabelece que todos os *chickens* não "perturbem" o time). Assim, todo e qualquer interesse de *stakeholders* não vinculados à equipe responsável é filtrado e levado em consideração apenas pelo *product owner* e pelo *Scrum master*, deixando o time livre para prosseguir com o projeto [Siqueira, 2007].

3. Analisando um Caso Prático de uma Empresa: "do CMMI ao *Scrum*"

Visando capturar aspectos sociotécnicos em cenários que se baseiam no desenvolvimento de software com *Scrum*, esta seção apresenta um caso prático: *uma empresa (**inicialmente de grande porte**, **multinacional**) cujos processos, aderentes ao CMMI num primeiro momento, sofreram alterações decorrentes de algumas mudanças ambientais, de forma que a empresa passou a utilizar o Scrum (posteriormente aderente à ISO).* Essa empresa iniciou suas atividades na área de qualidade de software em 2004, quando

foi aberto um projeto para a definição de processos aderentes ao modelo CMMI, desenvolvido pela *Siemens Corporate Tecnology* [Siemens CT, 2008]. Segundo [Mehner *et al.*, 1998], esses processos correspondem a uma mistura dos modelos CMMI e *Bootstrap* (refinamentos e extensões específicos da *Siemens*), especialmente para as disciplinas de engenharia. Em 2005, a empresa foi avaliada CMMI Nível 2.5 pela *Siemens Corporate Research* [Siemens USA, 2008], cujo processo, denominado ICO (*Integrated Component Order*), está relatado em [Santos Júnior & Yellayi, 2007]. Em 2006, a empresa obteve a certificação ISO 9001:2000 [ABNT, 2001], utilizando os processos aderentes ao CMMI. Na época, aproximadamente 400 profissionais de sete institutos (subcontratados) trabalhavam em projetos de desenvolvimento de software para dispositivos móveis. Em 2006, devido a mudanças estratégicas, a empresa passou a utilizar metodologias ágeis (*Scrum*), com o objetivo de ganhar agilidade e manter a qualidade dos produtos de software, mas ainda manteve o uso do processo ICO por alguns meses.

A partir do cenário delineado, a seguir, apresenta-se um relato do início da implantação do processo *Scrum* no **departamento de pesquisa e desenvolvimento da empresa**. A descrição realizada se baseia nos comentários referentes às observações do analista de qualidade durante o acompanhamento dos projetos, ao passo que as opiniões e relatos dos participantes dos projetos provêm das *retrospective meetings*. O departamento era formado por pessoas com papéis definidos pelo processo avaliado CMMI, tais como analistas e gerentes de qualidade, de configuração e de testes, gerentes de projeto, desenvolvedores, arquitetos de software, coordenadores técnicos, gerentes de linha e diretor do departamento, bem como um EPG (*Engineering Process Group*) com cinco membros. No entanto, durante o uso do *Scrum*, os papéis foram reduzidos para *Scrum master*, *product owner* e time, sendo este formado por um conjunto de pessoas com habilidades necessárias para o desenvolvimento de software.

3.1. O Primeiro Projeto com *Scrum*: "novas percepções de valor"

No **primeiro projeto** a utilizar o *Scrum* (iniciado em janeiro de 2006), cujo objetivo era o desenvolvimento de software para comunicação entre um PC (*personal computer*) e dispositivos móveis, o time trabalhou usando o conhecimento adquirido a partir da leitura sobre o processo descrito em [Schwaber & Beedle, 2002] e [Larman, 2004], sem ter recebido um treinamento prévio. Em decorrência desse cenário, aspectos técnicos e sociais se "misturavam" de forma que o *técnico* e o *não-técnico* geravam uma "co-modificação". Nesse momento, uma análise "falha" preliminar pode ser apontar os *stakeholders* como a base dos problemas do processo, ou ainda apontar o *Scrum* como "emergente" ou "flexível demais", considerando o cenário anterior (CMMI), fundamentado em processos tradicionais e "universais" [Teixeira & Cukierman, 2007]:

- as *daily meetings* que, de acordo com o processo *Scrum*, deveriam ser realizadas todos os dias, com duração de 15 minutos, eram feitas dia sim e dia não e com duração de 30 a 50 minutos, apesar ter sido definido o tempo de 30 minutos para as reuniões. Em muitas *daily meetings*, alguns participantes faziam comentários além das três perguntas que deveriam ser respondidas, aumentando o tempo das reuniões;
- apenas 70% dos *product backlog items* planejados foram realizados, por não ter ocorrido a fase *pre-game* (estimativa de custo e tempo para a realização dos *product backlog items*, e a divisão destes em partes que "caibam" dentro de um *sprint*);

- durante a *review meeting*, o *product owner* nitidamente demonstrou ter dúvidas quanto à qualidade do software apresentado, por não serem exibidos os resultados dos testes, apesar de não terem ocorrido falhas no software durante a apresentação. O processo aponta a necessidade de testes de aceitação para cada *product backlog item*;
- o *sprint backlog* não era sempre atualizado pelo time diariamente, como era pedido pelo processo, que exigia que as horas remanescentes de cada atividade planejada fossem atualizadas, bem como novas atividades fossem incluídas (caso necessário) e atividades não necessárias fossem canceladas.

Apesar dos aspectos negativos apontados anteriormente, alguns aspectos positivos se destacaram, visando à congruência sociotécnica, uma vez que os aspectos negativos são *técnicos* e *não-técnicos*, em uma "co-modificação", assim como os aspectos positivos. Ou seja, o "técnico" modificou o "não-técnico" e vice-versa, de maneira que ocorreu uma mudança cultural em razão do *Scrum* e ocorreu uma mudança no processo em razão de uma nova cultura a ser desenvolvida e implementada, relacionada à agilidade. Esses aspectos apontam "sensações" percebidas nos *stakeholders*, que indicam a influência dos aspectos não-técnicos em ambientes com agilidade [Santos, 2008b]:

- o time se sentia bastante motivado, por ter liberdade de definir as atividades que seriam necessárias e pela agilidade com que as elas aconteciam;
- foram desenvolvidas planilhas que implementavam o *product backlog* e o *sprint backlog*, permitindo maior controle sobre os requisitos e tarefas do projeto;
- o time trabalhava no mesmo ambiente, de forma integrada, o que facilitou muito a comunicação entre seus membros, bem como a solução de problemas de maior urgência – mesmo os membros de institutos (fornecedores subcontratados), que antes trabalhavam em suas próprias empresas, passaram a trabalhar neste mesmo ambiente.

O EPG, que no processo ICO tinha a missão de definir e institucionalizar processos aderentes ao Nível 3 do CMMI e avaliar a qualidade dos produtos de trabalho de cada projeto, passou a estudar o *Scrum* (devido à mudança) e acompanhar o projeto, agindo como "facilitador" para a implementação do novo processo. Isso não era uma tarefa fácil, dado que os membros do EPG estavam aprendendo a teoria, ao passo que os membros do projeto haviam estudado o *Scrum* antes mesmo da unidade brasileira da empresa ter decidido adotá-lo. Além disso, apesar de indicar que, ao final de cada *sprint*, deve ser entregue um software com qualidade necessária para inseri-lo no mercado, o próprio *Scrum* não define o papel do EPG e nem do analista de qualidade, indicando o *Scrum master* como responsável por garantir que a metodologia seja seguida. Este fato fazia com que os membros do projeto não dessem tanto crédito às opiniões do EPG, quando este sugeria tarefas e processos aderentes ao CMMI para solucionar problemas de projeto. Essa falta de crédito gerava dúvidas nos membros do EPG quanto à sua utilidade para a empresa. Mas o EPG utilizou a experiência no acompanhamento do primeiro projeto para ministrar treinamentos sobre *Scrum* para o restante do departamento.

Para facilitar a implantação do *Scrum* no departamento, em março de 2006, um membro do EPG foi enviado à matriz da empresa, no exterior, para participar de um curso denominado CSM (*Certified Scrum Master*) [Ruby, 2007], que o certificou como *Scrum master*. O conhecimento adquirido foi repassado a todo o departamento por meio de treinamentos práticos sobre o uso do *Scrum*. Os membros do EPG passaram a acompanhar os projetos mais de perto, fazendo com que o *Scrum* fosse implantado sem maio-

res problemas. Quanto aos pontos sensíveis citados anteriormente, relacionados aos aspectos técnicos, o EPG e os membros do projeto definiram algumas soluções durante uma *retrospective meeting*, poucos dias após a *review meeting*. Torna-se importante considerar que, nesse momento, as características da agilidade favoreceram o equilíbrio entre as percepções de valor dos diferentes *stakeholders*, visando conciliar e/ou reconciliar as divergências geradas pela tecnicidade dos processos [Boehm, 2003]:

- as *daily meetings* seriam realizadas todos os dias com duração de 15 minutos, e os participantes deveriam se ater às três perguntas. Se um tema precisasse de uma discussão maior, seria tratado após a reunião, com participação das pessoas envolvidas;
- um testador participaria como *pig* para especificar e executar os testes de aceitação, ajudando o time a resolver os *bugs* encontrados e apresentando os resultados dos últimos testes durante a *review meeting*;
- caso o *sprint backlog* não fosse atualizado durante algum dia, o *scrum master* poderia apresentar esse fato durante a *daily meeting* como um impedimento, alertando todo o time de que isso não pode ocorrer e solicitando que todos façam a atualização logo após a reunião, mesmo que somente um membro não tenha feito a atualização;
- antes do próximo *sprint planning*, o time deverá detalhar e estimar os *product backlog items* considerados mais prioritários pelo *product owner*, tarefa esta sempre definida no *sprint backlog* do *sprint* atual;
- logo após a *review meeting*, o time deverá realizar a *retrospective meeting*.

As soluções foram implantadas com sucesso, exemplificando que a flexibilidade e o dinamismo de uma metodologia ágil trazem benefícios ao processo, desde que de forma disciplinada [Santos, 2008a]. O time do primeiro projeto passou a trabalhar de acordo com [Schwaber & Beedle, 2002] e, ao final de quatro meses, entregou não apenas os requisitos solicitados pelo cliente, mas outros requisitos acrescentados ao software. O bom uso do *Scrum* foi confirmado pelas *retrospective meetings* seguintes, pois tanto os pontos fortes como os pontos fracos apontados se referiam mais a detalhes técnicos (inerentes ao projeto) do que ao uso do *Scrum* em si. Assim, verifica-se a importância de considerar os fatores não-técnicos em conjunto com os fatores técnicos, visando promover a melhoria da qualidade a partir da calibragem do processo, com base na experiência e nas percepções de valor dos *stakeholders*. Essa experiência foi repassada pelo EPG aos outros projetos do departamento, onde o uso do *Scrum* foi bem recebido pelos desenvolvedores, em função da "liberdade" para a definição das atividades e da "grande interação" com os usuários nas primeiras *retrospective meetings* de todos os treze projetos que usaram *Scrum*. Por outro lado, de acordo com o resultado da avaliação do CMMI, realizada em outubro de 2005, observou-se que o processo anterior era visto pelos integrantes do departamento como "**muito bom**", mas "**muito exigente**", que de certa forma é "natural", considerando a característica hierárquica e centralizada dos processos tradicionais [Teixeira & Cukierman, 2007].

Dentre os aspectos negativos do uso do *Scrum*, os mais comentados foram a falta de artefatos e processos para realizar os estudos de viabilidade, da arquitetura de software, da análise e projeto do sistema e dos testes, bem como do planejamento do projeto como um todo. Apesar do *Scrum* definir o *sprint backlog* como ferramenta para gerenciar um *sprint*, de acordo com os *Scrum masters*, não havia uma ferramenta eficiente para controlar o projeto como um todo. Ou seja, se por um lado as pessoas gostavam da

"liberdade" para definir suas atividades, por outro lado "queriam" artefatos e definições de processos. Conforme Teixeira (2006) coloca, pode-se entender essa postura como fruto da racionalidade científica ocidental, que valoriza o método como diretriz para a verdade, o que torna sempre necessário a construção/existência de modelos "universalizantes". Nesse momento, deve-se recordar que a ES necessita reconhecer a relevância da co-produção entre fatores técnicos e não-técnicos para o alcance dos objetivos dos projetos de software ao longo do processo [Cukierman *et al.*, 2007]. Um bom exemplo do que a falta de definição de processos gerou foi a falta de participação da equipe de testes no início do primeiro projeto que utilizou *Scrum*. No primeiro *sprint*, após o término da codificação de um incremento do produto de software, um membro da equipe de testes foi chamado para realizar os testes, mas ele não pôde fazer o trabalho, em razão da falta de especificação de testes, o que causou um impacto no primeiro *sprint*. Justificativa: durante a apresentação ao *product owner*, na reunião *sprint review*, não foi apresentado nenhum relatório atestando qualidade do software, apesar deste estar funcionando.

3.2. Influências das Características do CMMI durante o Uso do *Scrum*

Considerando a transição entre CMMI e *Scrum*, que apresentam tendências divergentes em vários pontos [Glazer *et al.*, 2008], não é de se estranhar que os *stakeholders* queiram "*um pouco mais de processo e artefatos*", uma vez que o *Scrum* permite definir as atividades que deverão ser executadas para atingir aos objetivos de um *sprint*. Para a empresa, uma explicação advém da própria formação do departamento de pesquisa e desenvolvimento, cujas pessoas foram contratadas, inicialmente, para assumir funções descritas no processo avaliado CMMI Nível 2. Esse processo, definido na matriz da empresa, trata do desenvolvimento distribuído de um ou mais componentes de software, e é apoiado por outros processos, visando a integração de componentes, testes de sistema e entrega do produto. Esse processo permitiu o desenvolvimento e testes de componentes em unidades isoladas da empresa localizadas na América, Europa e Ásia, com posterior integração, testes de sistema e aprovação na Europa – o que deixava clara a atribuição dos papéis durante o desenvolvimento de um componente. Sendo assim, para a unidade brasileira, foram contratadas pessoas para assumir os papéis já definidos nesse processo, tais como gerentes de linha, gerentes de projetos, arquitetos de software, testadores, gerentes de configuração, desenvolvedores, coordenadores técnicos, analistas e gerentes de qualidade, analistas de processo (EPG), dentre outros.

Outro critério importante utilizado para selecionar essas pessoas na *fase CMMI* da empresa: experiência de trabalho em empresas avaliadas pelo menos Nível 2 do SW-CMM. Ou seja, essas pessoas vinham de uma cultura organizacional de "processos claros" (ambientes controlados, conforme apresenta Santos (2008a)), o que facilitou muito a institucionalização do processo de desenvolvimento de componentes. Isto se evidenciou 18 meses após o início das atividades da unidade brasileira da empresa, quando esta teve seus processos avaliados (CMMI) pelos auditores da *Siemens Corporate Research* e todas as pessoas demonstraram saber exatamente seu papel (os processos, documentos e ferramentas que utilizavam, com que outros papéis interagiam e de que forma deviam interagir). O processo cobria 14 das 25 áreas de processo do CMMI [Chrissis *et al.*, 2003], com um número estimado de 140 práticas específicas atendidas (**ou parcialmente atendidas**), de um total de mais de 300 práticas específicas. De acordo com Marçal *et al.* (2007), o *Scrum* atende totalmente ou parcialmente 29 práticas em três

áreas de processo da área de *Gerência de Projetos*. Esses números (baixos) explicam o quanto as pessoas que utilizariam o *Scrum* poderiam sentir falta de processos aderentes ao CMMI, pois áreas de processo, como *Solução Técnica*, *Verificação e Validação*, *Garantia da Qualidade de Processos e Produtos*, *Gerência de Configuração*, entre outras, não seriam cobertas pelo processo de desenvolvimento de componentes.

No início da implantação do *Scrum*, gerentes de linhas se tornaram *product owners*, gerentes de projetos se tornaram *Scrum masters* e desenvolvedores, gerentes de configuração e arquitetos se tornaram o time. Os testadores e os analistas de qualidades, no momento, não iriam fazer parte do time. Os coordenadores técnicos fizeram parte do time, mas utilizando suas habilidades como desenvolvedores. No curso CSM, realizado na matriz da empresa, participaram quatro analistas de qualidade, que tinham dúvidas quanto à sua participação no processo. Mas o instrutor do curso informou aos analistas que eles poderiam participar avaliando a qualidade dos produtos gerados pelo time.

Na unidade brasileira, os analistas de qualidade não faziam parte dos times. Entretanto, eles eram freqüentemente solicitados, juntamente ao EPG, para indicar artefatos que documentassem determinadas atividades, tais como estudos de viabilidade, arquitetura de software, análise e projeto, especificação de testes etc. No geral, estes artefatos eram versões mais "enxutas" de artefatos usados no processo ICO, de forma não padronizada, mas por demanda, ou seja, os artefatos eram adaptados segundo a necessidade dos projetos. Os membros do EPG, assim como os analistas de qualidade, tinham dúvidas quanto à sua importância em face dessas mudanças, mas a solicitação por artefatos reduziu esse receio. Outra razão para o receio foi a própria mudança da missão do EPG, que até então era definir e institucionalizar processos aderentes às práticas do CMMI Nível 3, e em março de 2006 passou a implantar agilidade, em que não tinham experiência. Durante essa mudança, havia três projetos que utilizavam o processo ICO, assim como um projeto utilizando o *Scrum* e mais três que iriam usar este processo.

3.3. A Evolução do *Scrum* e a ISO: "a falha está nos fatores não-técnicos?"

Decorridos poucos meses após o início do uso do *Scrum*, em abril de 2006, surgiu a necessidade de que o departamento certificasse seus processos conforme a ISO 9001:2000 [ABNT, 2001], devido a uma exigência legal. Esses processos deveriam deixar clara a ligação entre o departamento de pesquisa e desenvolvimento e os demais departamentos da unidade brasileira, permanecendo no escopo apenas o desenvolvimento de software embarcado para dispositivos móveis. O EPG, juntamente com a diretoria do departamento, decidiu complementar os processos aderentes ao CMMI com a documentação oficial da empresa utilizada em seu *Sistema Integrado de Gestão*, aderente à ISO 9001:2000, documentando também o ciclo de vida definido no *Scrum*. Não houve necessidade de criação de novos artefatos nem de mudança dos processos ora utilizados. No entanto, deveria ser criado um "macro-processo" para descrever o ciclo de vida do processo ICO e informar cada procedimento deste processo quando do atendimento de cada etapa desse ciclo de vida. O processo *Scrum* também fez parte deste "macro-processo", mas sem conexão com o ICO.

O trabalho durou dois meses e envolveu documentação e treinamento. A empresa foi certificada na ISO 9001:2000 e na ISO 14001, em julho de 2006. Devido à pressão causada pelo pouco tempo para documentar os processos, o EPG quase foi des-

feito, pois seus membros não se sentiram satisfeitos, apesar do reconhecimento do trabalho. Uma das razões para isso foi a dificuldade para conseguir informações dos membros dos projetos, que estavam pressionados pelos seus próprios prazos e queriam pouca interação durante a documentação dos novos processos. Assim, os membros do EPG tiveram que alocar "horas extras" e "madrugadas de trabalho" em razão do pouco apoio dos *stakeholders* envolvidos – isso reflete o desequilíbrio na percepção de valor dos *stakeholders* [Boehm, 2003], além de "ansiedades" advindas de processos tradicionais [Santos, 2008b]. A redução desse problema se deu após o departamento ter obtido 69% de aderência à ISO 9001:2000, em uma auditoria interna, e os membros dos projetos terem sido alertados de que se a empresa não obtivesse a certificação, sua licença para produção de dispositivos móveis seria cancelada – nesse ponto, percebe-se a mistura entre o "técnico" e o "não-técnico" [Teixeira & Cukierman, 2007], de forma que o *universal*, o *generalizante*, o *atemporal* e o *escrito* conclama o *particular*, o *local*, o *temporal* e o *oral* [Boehm, 2006] [Cukierman *et al.*, 2007]. Com isso, o EPG teve apoio total para definição dos documentos e para os treinamentos, resultando em 100% de aderência na avaliação oficial para re-certificação da ISO 9001:2000 – este sucesso reforça a mistura social, cultural, histórica, política e técnica, que pode gerar efeitos sobre os recursos humanos (desgaste, desânimo a curto prazo, pressão etc.) [Teixeira, 2006]. Em agosto de 2006, foi realizado um treinamento em CSM na unidade brasileira da empresa, certificando 30 profissionais, sendo 17 da empresa, incluindo os membros do EPG, gerentes de projeto, desenvolvedores e fornecedores de serviço.

Em dezembro de 2006, houve uma mudança drástica que reduziu o tamanho do departamento à metade e tornou a unidade brasileira da empresa uma unidade autônoma, **de médio porte** – é interessante notar que, diante da estrutura montada, congregando ISO 9001:2000 e *Scrum*, existem "outros fatores" que podem interferir na organização [Teixeira & Cukierman, 2007]. O EPG foi desfeito e apenas uma pessoa, o membro mais experiente, se tornou responsável pela área de qualidade e processos, agora chamada de TPMQ (*Training, Process, Metrics and Quality)*. Essa área teve a responsabilidade de definir processos adequados à nova realidade, que consistia em desenvolvimento de software e produtos para atender às necessidades dos clientes brasileiros ("local") e das outras unidades da empresa. Os projetos de desenvolvimento continuaram a utilizar o *Scrum*, sem padronização, enquanto os novos processos eram definidos. Como a cultura de processos fazia parte do departamento e as pessoas deixavam clara essa necessidade, não houve dificuldade para que o TPMQ fizesse a definição dos processos, disponibilizados na Intranet do departamento.

Em cinco meses de trabalho, o TPQM definiu processos de Gerência de Projetos, Gerência de Configuração, Gerência de Testes e um ciclo de vida de desenvolvimento, buscando atender às práticas específicas do CMMI, e desenvolveu uma ferramenta para a solicitação de serviços de pesquisa e desenvolvimento (P&D), utilizada pelos outros departamentos para solicitar serviços. Os participantes e os envolvidos na definição dos processos estavam "ansiosos" para institucionalizar os processos mencionados ("geral"). Isso não ocorreu por não se ter definido um projeto piloto adequado para utilizá-los, de forma que os projetos em andamento continuavam utilizando *Scrum* sem padronização.

Até abril de 2007, em torno de 71% dos 13 projetos em andamento obtiveram sucesso na entrega de seus produtos de software usando *Scrum*, mas era necessária uma nova certificação da ISO 9001:2000. No entanto, os processos aderentes à ISO

9001:2000, definidos no ano anterior, não eram mais adequados à nova realidade do departamento (e da empresa). Além disso, como os processos descritos pelo TPMQ não foram divulgados e institucionalizados, decidiu-se por não inclui-los nos processos da ISO. Assim, reuniram-se o Coordenador de Gestão Integrada (CGI) e os gerentes dos projetos em andamento, para que a definição dos processos da ISO estivesse de acordo com as atividades realizadas nestes projetos. Apesar dessas atividades não estarem padronizadas, não foi difícil descrever os processos, uma vez que todos os projetos utilizavam *Scrum*, faltando apenas definir alguns artefatos a serem eleitos como padrão. A descrição dos novos processos foi realizada de forma a unir as cláusulas 7.2.x e 7.3.x do padrão ISO 9001:2000 ao *Scrum* (mas não aderente ao CMMI), considerando a experiência anterior nos 13 projetos, que obteve um aumento de produtividade e 100% de aderência ao padrão [Santos Júnior *et al.*, 2008]. Isso envolveu um grupo de nove *Scrum masters* (incluindo o CGI) e partiu da premissa de que, no *Scrum*, o time define as atividades necessárias para o cumprimento dos objetivos a ele endereçados. Após a institucionalização do novo processo, a equipe ganhou novos artefatos úteis para documentar seus trabalhos, utilizando-os em todos os novos projetos. 25% dos *Scrum masters* enxergavam a finalidade do processo única e exclusivamente para atingir a ISO 9001, enquanto os 75% restantes achavam-no útil para a qualidade do processo – mas todos eram unânimes em afirmar que nem todos os processos eram cobertos.

Diante desse contexto, pode-se depreender que o sucesso na combinação de *Scrum* e ISO (não aderente ao CMMI) mostra a viabilidade de integração entre diferentes vertentes de processos de desenvolvimento [Boehm & Turner, 2004]. No entanto, o histórico apresentado chama a atenção para as dificuldades enfrentadas, sobretudo no movimento de "co-modificação" entre fatores técnicos e não-técnicos e nas posturas apreendidas (ansiedade, pressão, desânimo, pessimismo), além das diferenças culturais apresentadas por modelos tradicionais e "universais" e modelos alternativos e "emergentes" [Teixeira & Cukierman, 2007] [Santos, 2008a]. Além disso, percebe-se que uma porcentagem dos *stakeholders* vê a institucionalização de processos como instrumento para alcançar certificações, ainda mais diante da redução do porte da empresa e dos projetos alocados ao departamento em questão. Esse ponto é crucial, pois mostra que, diante de tantos aspectos positivos e negativos identificados, um processo de co-modificação entre "técnico" e "não-técnico" é fundamental para se entender e gerenciar o sucesso de projetos de software [Boehm, 2003].

4. Considerações Finais

Existem diversas vertentes de processos de desenvolvimento de software e uma das questões discutidas na literatura (e enfrentada pela indústria) consiste na escolha de uma vertente que melhor satisfaça os objetivos da organização [Jiang & Eberlein, 2008]. Essa questão ressalta a importância de se explorar uma discussão baseada em observação e em evidência no campo da ES, o que reflete um dos seus desafios: *tornar-se experimental e basear-se nos resultados provenientes deste processo* [Kitchenham *et al.*, 2002]. Estudos de campo (na indústria) se tornam importantes, considerando uma abordagem sociotécnica em ES, podendo abrir boas oportunidades para pesquisa acerca de processos de software ao considerar combinações e similaridades sob uma perspectiva baseada em valor e focada nos *stakeholders* [Boehm, 2003]. Visando capturar aspectos sociotécnicos em cenários que se baseiam no desenvolvimento de

software com *Scrum*, este artigo apresentou o caso prático de uma empresa, cujos processos, aderentes ao CMMI num primeiro momento, sofreram alterações decorrentes de algumas mudanças ambientais, de forma que a empresa passou a utilizar o *Scrum* (posteriormente, aderente à ISO). O caso prático apresentado foi desenvolvido com base em entrevistas realizadas com profissionais da empresa, considerando comentários referentes às observações do analista de qualidade durante o acompanhamento dos projetos e opiniões e relatos dos participantes de projetos conduzidos.

Alguns aspectos interessantes puderam ser observados, considerando pontos positivos e negativos, dificuldades e casos de sucesso, além de sensações como ansiedade, pressão, alívio, medo, liberdade etc. Ao longo da transição entre o uso do CMMI e o uso do *Scrum*, aspectos inerentes ao pensamento ocidental puderam ser detectados, como a busca por modelos tradicionais, "universais", "seguros" e "estruturados", mesmo diante da ênfase da agilidade sobre os *stakeholders*. Isso pode estar relacionado ao fato de que o perfil dos recursos humanos que compunham a empresa era baseado em profissionais experientes, oriundos de empresas que utilizavam CMMI. Ademais, situações passíveis de falha no *Scrum* requereram a estruturação de processos e inconscientemente despertavam comparações com o momento anterior (uso do CMMI). Em alguns momentos, isso contribuiu para a corrente de que métodos ágeis são tão flexíveis que são mais propensos a riscos e que funcionam apenas em empresas de pequeno e médio porte [Boehm, 2006]. Por fim, um momento importante do caso prático apresentado se deu durante a aderência dos processos da empresa à ISO, mantendo-se a utilização do *Scrum* – quando, pouco tempo depois de estar experimentando uma combinação entre *Scrum* e ISO, condições ambientais levaram a empresa a reduzir sua participação no mercado. Esse caso reforça a necessidade de se *tratar a ES sob um olhar sociotécnico*, uma vez que a ES é constituída de forma indissociavelmente técnica, social, histórica, econômica, ética e política, aspectos estes inextricavelmente articulados e passíveis de co-produção [Cukierman *et al.*, 2007].

Dado que o presente trabalho se concentrou em estruturar o cenário para a extração de elementos para uma abordagem sociotécnica do *Scrum*, novas oportunidades de pesquisa podem ser identificadas, tais como mapear características do CMMI e do *Scrum*, visando identificar impactos, e mapear elementos do *Scrum* e da ISO, visando verificar possibilidades de combinação. Como trabalhos futuros, pretende-se explorar as entrevistas realizadas no caso prático, bem como explorar diferentes perspectivas de valor [Boehm, 2003] dos *stakeholders*, visando a sua conciliação e/ou reconciliação e o controle e monitoramento baseado em valor, considerado a transição e composição das vertentes de desenvolvimento de software distintas (CMMI, *Scrum* e ISO).

Agradecimentos

Os autores agradecem ao CNPq e à FAPEAM pelo apoio financeiro neste trabalho.

Referências

ABNT (2001) "*NBR ISO 9001 Sistemas de Gestão da Qualidade – Requisitos*". ABNT CE-25:002.18 – Comissão de Estudos de Sistemas da Qualidade, Brasil.

Boehm, B. (2003) "Value-Based Software Engineering". *Software Engineering Notes* 28, 2 (March),1-12.

Boehm, B. (2006) "A View of 20th and 21st Century Software Engineering", In: *Proc. of the 28th ICSE*, Shanghai, China, 12-29.

Boehm, B.; Turner, R. (2004) "Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-driven Methods", In: *Proc. of the 26th ICSE*, Edinburgh, UK, 718-719.

Chrissis, M.; Konrad, M.; Shrum, S. (2003) "*CMMI: Guidelines for Process Integration and Product Improvement*". Addison-Wesley, 688p.

Cukierman, H. L.; Teixeira, C.; Prikladnicki, R. (2007) "Um Olhar Sociotécnico sobre a Engenharia de Software". *Revista de Informática Teórica e Aplicada* 14, 2, 207-227.

DPLP (2009) "Dicionário Piberam da Língua Portuguesa". Dicionário On-line. Disponível em: http://www.priberam.pt/dlpo/dlpo.aspx. Acessado em: 11/05/2009.

Ferreira, R. B.; Lima, F. P. A. (2006) "Metodologias Ágeis: Um Novo Paradigma de Desenvolvimento de Software", In: *Anais do II WOSES*, V SBQS, Vila Velha, Brasil, 107-114.

Glazer, H.; Dalton, J.; Anderson, D.; Konrad, M.; Shrum, S. (2008) "*CMMI or Agile: Why Not Embrace Both!*". Technical Note CMU/SEI-2008-TN-003, Software Engineering Institute, 41p.

Improve It (2009) "Metodologia, Método ou Processo?". *Artigos On-line*. Disponível em: http://improveit.com.br/xp/metodologia. Acessado em: 11/05/2009.

Jiang, L.; Eberlein, A. (2008) "Towards a Framework for Understanding the Relationships between Classical Software Engineering and Agile Methodologies", In: *Proc. of the 1st Intern. Workshop on Scrutinizing Agile Practices or Shoot-Out at the Agile Corral*, 30th ICSE, Leipzig, Germany, 9-14.

Kitchenham, B. A.; Pfleeger, S. L.; Pickard, L. M.; Jones, P. W.; Hoaglin, D. C.; El Emam, K.; Rosenberg, J. (2002) "Preliminary Guidelines for Empirical Research in Software Engineering". *Transactions on Software Engineering* 28, 8 (August), 721-734.

Larman, C. (2004) "*Agile and Iterative Development: A Manager's Guide*". Addison-Wesley, 368p.

Marçal, A. S. C.; Freitas, B. C. C.; Soares, F. S. F.; Maciel, T. M. M.; Belchior, A. D. (2007) "Estendendo o SCRUM segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI", In: *Anais da XXXIII Conferencia Latinoamericana de Informática* (CLEI'07), San Jose, Costa Rica, 1-12.

Mehner, T.; Messer, T.; Paul, P.; Paulisch, F.; Schless, P.; Volker, A. (1998) "Siemens Process Assessment and Improvement Approaches: Experiences and Benefits", In: *Proc. of the 22nd Annual International Computer Software and Applications Conference*, Vienna, Austria, 186-195.

Ruby, K. (2007) "*Scrum Alliance Certifications*".

Santos, R. P. (2008a) "Elementos para uma Abordagem Sociotécnica do Desenvolvimento de Software com *Extreme Programming*". *Scientia* 19, 2 (Julho-Dezembro), 102-116.

Santos, R. P. (2008b) "Uma Visão Sociotécnica sobre o Desenvolvimento de Software com *Extreme Programming*", In: *Anais do IV WOSES*, VII SBQS, Florianópolis, Brasil, 25-36.

Santos Júnior, A. F.; Silva, V. J.; Lucena Júnior, V. F. (2008) "Desenvolvimento de um Processo de Software Aderente à ISO 9001:2000 Baseado no Processo Ágil Scrum", In: *Anais do VII SBQS*, Florianópolis, Brasil, 359-367.

Santos Júnior, A. F.; Yellayi, S. (2007) "Software Process Improvement in an Outsourced EPG Environment", In: *Proc. of the 19th Annual Premier Conference SEPG*, Austin, USA.

Schwaber, K. (2004) "*Agile Project Management with Scrum*". Microsoft Press, 192p.

Schwaber, K.; Beedle, M. (2002) "*Agile Software Development with Scrum*". Prentice Hall, 158p.

Siemens CT (2008) "*Siemens AG – Corporate Technology*".

Siemens USA (2008) "*Siemens USA – Siemens Corporate Research*".

Siqueira, H. B. A. (2007) "*Mapeamento das Práticas de Scrum nas Áreas de Processo do CMMI e Uma Proposta para sua Aderência*". Monografia. CIn/UFPE, Recife, Brasil, 57p.

Teixeira, C. A. N. (2006) "Algumas Observações sobre os Vínculos entre a Engenharia de Software e o Pensamento Moderno", In: *Anais do II WOSES*, V SBQS, Vila Velha, Brasil, 39-50.

Teixeira, C. A. N.; Cukierman, H. L. (2007) "Por que Falham os Projetos de Implantação de Processos de Software?", In: *Anais do III WOSES*, VI SBQS, Porto de Galinhas, Brasil, 1-12.