

A Utilização de Práticas Scrum no Desenvolvimento de Software com Equipes Grandes e Distribuídas: um Relato de Experiência

Alex N. Borges Júnior¹, Fernando K. Kamei¹, Pablo R. C. Alves¹, Rodrigo G. C. Rocha^{1,2}, Ryan R. Azevedo^{1,3}, Silvio Meira¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7.851 – 50.732-970 – Recife – PE – Brasil

²Universidade Federal Rural de Pernambuco (UFPE - UAG)
55292-270 – Garanhuns – PE – Brasil

³Universidade Federal do Piauí (UFPI)
64600-000 – Picos – PI – Brasil

{anbj, fkk, prca, rgcr, rra2, srlm}@cin.ufpe.br

Abstract. *With the business globalization and the growing of the software market, the distributed software development has emerged. This approach has brought competitive advantages, as well as inherited problems of traditional approaches and added new challenges. In this context, a movement can be observed around the use of agile methodologies. However, there are few practical experiences that can provide solutions for the adaptation of agile processes in this environment. This paper is about an experience of adapting Scrum agile practices in a distributed software project that involves a big team. Here will be present the results, the difficulties and lessons learned.*

Resumo. *Com a globalização dos negócios e o crescimento do mercado de software, o Desenvolvimento Distribuído de Software tem se destacado. Esta abordagem trouxe vantagens competitivas, como também herdou problemas da abordagem tradicional e acrescentou novos desafios. Nesse contexto, é possível observar uma movimentação em torno da utilização de metodologias ágeis, porém, ainda são poucas as experiências práticas capazes de apresentar soluções para a adaptação de processos ágeis neste ambiente. Dessa forma, este artigo relata a experiência de adaptar práticas ágeis do Scrum no contexto de um projeto distribuído de software que envolve uma equipe grande. São apresentados os resultados, as dificuldades e as lições aprendidas.*

1. Introdução

Como reflexo da globalização dos negócios e do crescimento da economia, ocorreu uma migração do mercado local para o mercado global, criando novas formas de competição e colaboração. Esses fatores também atingiram o mercado de software. Neste contexto, surge o Desenvolvimento Distribuído de Software (DDS), onde as equipes de desenvolvimento estão distribuídas em diferentes localizações [Carmel 1999]. Segundo Audy e Prikladnicki (2008), a distribuição do processo de desenvolvimento de software faz ampliar os problemas inerentes ao desenvolvimento tradicional e gera novos desafios ao adicionar distância física, dispersão temporal e diferenças culturais.

Com o mercado de software mais dinâmico e global, as organizações têm buscado a implantação de processos produtivos visando melhorar a qualidade e reduzir custos [Ferreira et al. 2006]. Além disso, o mercado busca por maior agilidade e flexibilidade no desenvolvimento e entrega de software. Com isso, as abordagens ágeis ganham mais importância, apresentando vantagens como redução de prazos de entrega, menor burocracia no desenvolvimento, maior produtividade e mais eficiência.

Neste cenário, destaca-se o crescimento da utilização de abordagens ágeis no DDS, em especial o *Scrum*. No entanto, pesquisas na área ainda não são consolidadas, sendo escassos os trabalhos que abordam processos específicos para este contexto. Assim, é fundamental para o crescimento da área relatar experiências práticas capazes de apresentar soluções para a adaptação de processos ágeis nos mais diversos contextos.

O objetivo deste trabalho é apresentar a experiência de utilizar práticas *Scrum* no processo de desenvolvimento e gerenciamento de um projeto distribuído de software com equipes grandes. O projeto, intitulado *The SeC (The Supply e-Commerce)*, surgiu em 2010, da disciplina de Engenharia de Software do Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de Pernambuco (UFPE) e envolveu estudantes de mestrado e doutorado, todos trabalhando de forma distribuída. O seu objetivo foi o desenvolvimento de um sistema de compras e vendas coletivas na Web.

O restante do trabalho está estruturado da seguinte forma: a seção 2 aborda os fundamentos teóricos relacionados à pesquisa. A seção 3 trata do contexto do projeto, suas características e de seu processo de desenvolvimento, além de como se deu a gestão do projeto. A seção 4 aborda os resultados obtidos ao fim do projeto e as lições aprendidas. Por fim, na seção 5, são feitas as considerações finais acerca do trabalho.

2. Fundamentação Teórica

Esta seção apresenta uma breve revisão bibliográfica com o objetivo de fornecer um embasamento teórico dos principais conceitos utilizados neste trabalho.

2.1. Desenvolvimento Distribuído de Software

O desenvolvimento de software de forma co-localizada tem se tornado cada vez mais custoso e menos competitivo para as organizações. Visando a redução de custos, a melhoria da qualidade de seus produtos, o aumento de produtividade e a competitividade global, várias empresas optam por distribuir seus processos de desenvolvimento em lugares diferentes [Audy e Prikladnicki 2008].

O Desenvolvimento Distribuído de Software (DDS) é um modelo onde os envolvidos em um determinado projeto estão dispersos. De acordo com Prikladnicki (2003), as vantagens que o DDS oferece são:

- Possibilidade de desenvolvimento *follow-the-sun*¹, que permite o aumento de produtividade e a redução dos prazos de entrega dos produtos;
- Disponibilidade de recursos globais com baixos custos e a qualquer hora;
- Disponibilidade de recursos qualificados em áreas especializadas;
- Oportunidade de realizar o desenvolvimento do produto perto do cliente;
- Possibilidade de formação de equipes virtuais.

¹ Desenvolvimento *follow-the-sun* significa 24 horas contínuas de trabalho.

2.2 Scrum

O *Scrum* é baseado nas metodologias ágeis voltado especificamente para o gerenciamento de software com base na colaboração com o cliente, trabalho em equipe, desenvolvimento iterativo e incremental, e com respostas às mudanças [Rico et al. 2009]. Tais características objetivam eliminar a dependência em realizar um extenso planejamento inicial e documentação de todos os requisitos do sistema.

Segundo Schwaber, o *Scrum* é desenvolvido por meio de três papéis principais: i) o *Product Owner* (PO) que representa os interesses do dono do produto; ii) o Time que desenvolve as funcionalidades do produto e; iii) o *Scrum Master* que garante que as regras e práticas do *Scrum* estejam sendo utilizadas no projeto, além de ser responsável por remover qualquer tipo de impedimento e ajudar ao Time [Schwaber 2004].

O trabalho do *Scrum* é realizado em iterações com duração de 02 à 04 semanas cada, chamadas de *Sprints*. As *Sprints* são iniciadas com uma reunião e planejamento, conhecida como *Sprint Planning Meeting 1*, onde o Time decide quais estórias do *Product Backlog*² serão trabalhadas na *Sprint*. Após, ocorre a *Sprint Planning Meeting 2* com a presença dos membros do Time e o *Scrum Master*, para definirem e planejarem as tarefas de cada estória que deverão compor a *Sprint* (*Sprint Backlog*).

Durante as *Sprints* devem ocorrer reuniões diárias, conhecidas como *Daily Scrum Meeting*, visando o acompanhamento diário do projeto. No final de cada *Sprint*, ocorrem as reuniões de revisão, chamadas de *Sprint Review*, para apresentar ao *Product Owner* o que foi desenvolvido, e obter o *feedback* [Schwaber 2004]. Também deve ser realizada ao final da *Sprint* a reunião de retrospectiva (*Sprint Retrospective*), para avaliar os pontos positivos que ocorreram na *Sprint* e o que deve ser melhorado.

3. Projeto The SeC

O estudo de caso apresentado consiste no projeto The SeC (*The Supply e-Commerce*), idealizado por um aluno de mestrado durante uma disciplina de Engenharia de Software de Pós-Graduação da Universidade Federal de Pernambuco (UFPE). Apesar de envolver estudantes de vários estados do nordeste, a disciplina tinha encontros presenciais, que ocorriam uma vez por semana, nas instalações da UFPE. A disciplina tinha por objetivo permitir a experiência prática de desenvolvimento distribuído de software, juntamente com as práticas de desenvolvimento ágil de software em um ambiente utilizando plataformas virtuais focadas em computação nas nuvens.

3.1. Características do Projeto

O projeto The SeC consistiu no desenvolvimento de um sistema de compras e vendas coletivas na Web, que tem como princípio a união de compradores ou vendedores interessados em um mesmo produto, com o intuito de através da lei da oferta e da procura, permitir a compra ou venda com preços mais competitivos. O The Sec é uma aplicação nas nuvens que apresenta duas características principais: ser uma rede social e ter ênfase em comércio eletrônico (Leilão e Pregão).

O projeto teve duração de 04 meses, e foi desenvolvido sobre a plataforma Salesforce³, que é uma plataforma de computação nas nuvens que permite executar e desenvolver aplicativos de negócios. Vale ressaltar que foi verificada a possibilidade de utilizar outros ambientes que possuíam plataformas sob o conceito de computação nas

² Lista das funcionalidades desejadas para o produto, definidas pelo *Product Owner*.

³ www.salesforce.com

nuvens, como: Google AppEngine, AmazonWS e Windows Azure. No entanto, a escolha do Salesforce foi decidida pela sua facilidade, esforço reduzido para compreensão do desenvolvimento nas nuvens e por prover soluções de negócio pertinentes ao projeto.

As atividades do projeto ocorreram, na maioria das vezes, de forma distribuída, com exceção de algumas reuniões que ocorriam de forma presencial, nas instalações da universidade. O projeto envolveu cinquenta estudantes, entre mestrandos e doutorandos, residentes nas cidades de Maceió, João Pessoa, Campina Grande, Recife e Caruaru. Os estudantes foram divididos em 05 equipes, com 08 ou 09 componentes, e um time de apoio formado por 05 estudantes. Também participaram do projeto o professor titular da disciplina e dois professores auxiliares, que auxiliavam durante todo o processo.

3.2. Processo de Desenvolvimento do Projeto

O processo de desenvolvimento do The SeC foi baseado nas práticas do *Scrum*, que mesmo sendo indicado para projetos com equipes pequenas e co-localizadas, foi levado em consideração o curto espaço de tempo relacionado à complexidade do produto, além da oportunidade de utilizar as vantagens dos processos ágeis dentro do projeto. Também influenciou na decisão a oportunidade de ter uma experiência prática com processos ágeis de software, sendo um dos objetivos da disciplina em questão.

O *Scrum* enfatiza a comunicação face a face entre os membros do time. Porém, diante do contexto do projeto The SeC, não foi possível que os membros das equipes trabalhassem juntos em um mesmo ambiente constantemente. Além da distância física e da infra-estrutura pouco adequada, ocorreram dificuldades em estabelecer horários compatíveis para todos os envolvidos. Sendo assim, o desenvolvimento do projeto seguiu a proposta de modelo elaborada por Sutherland et al. (2007) chamada de *Scrum de Scrums*, voltada para projetos que trabalham com o DDS, que divide o trabalho em partes (times *Scrum*) isoladas, eliminando a maioria das dependências.

Desse modo, com base no modelo de *Scrum de Scrums*, o desenvolvimento do projeto The SeC foi dividido em equipes, como apresentado na Figura 1.

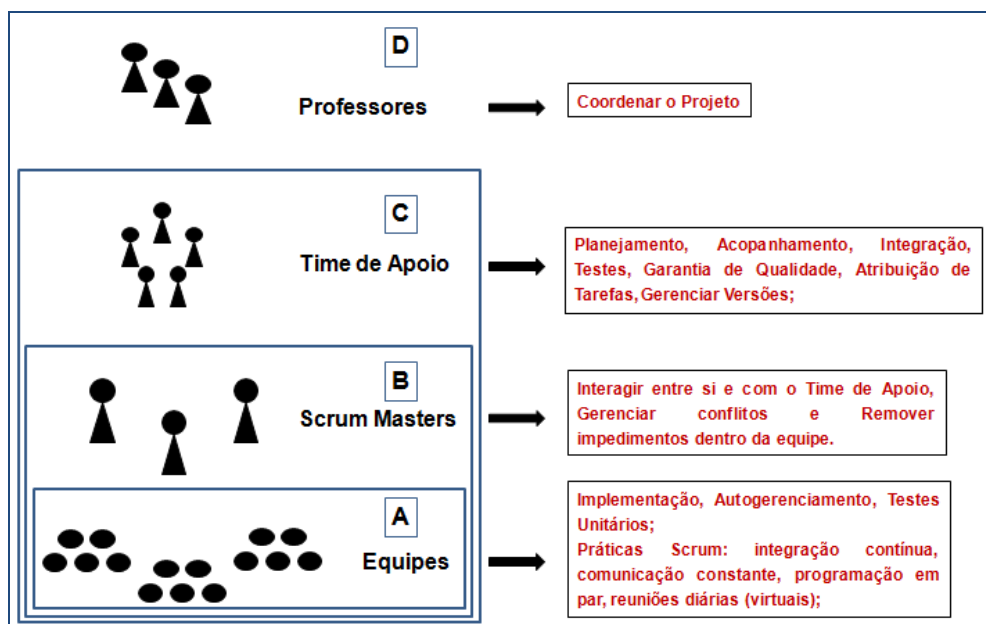


Figura 1. Participantes do Projeto

A Figura 1 apresenta a distribuição dos participantes do projeto, onde: (A) são as equipes de desenvolvimento; (B) são os *Scrum Masters* de cada equipe; (C) os membros do time de apoio, que ofereciam todo o suporte para o desenvolvimento do projeto; e (D) os professores, que coordenaram os ciclos de iterações e apoio entre os envolvidos.

No início do desenvolvimento, foram realizadas algumas reuniões presenciais com a participação do Time de Apoio, equipes de desenvolvimento e do *Product Owner*. Nessas reuniões foram definidos o escopo do projeto e especificações do processo, de forma a alinhar e padronizar as atividades realizadas pelas equipes. O Time de Apoio foi responsável por um conjunto de artefatos elaborados na fase preliminar do projeto: plano do projeto, plano de qualidade e documento de especificação do processo.

Em cada *Sprint*, o Time de Apoio era responsável por definir os itens de *Backlog* que seriam implementados, além de, em conjunto com o *Product Owner*, definir um valor de negócio para cada item, que variava de 0 à 100 e indicava sua importância e prioridade. Depois, os itens eram atribuídos às equipes e registrados na ferramenta *Mantis*⁴. Cada equipe era responsável por dividir suas atividades em tarefas menores, onde cada membro da equipe possuía autonomia na escolha da atividade na qual ficaria responsável. As equipes usaram a técnica de *Planning Poker* para estimar os esforços para as atividades, durante as reuniões de *Sprint Planning Meeting 2*. Para realizar as rodadas de estimativas de *Planning Poker* de forma distribuída, as equipes utilizaram ferramentas instantâneas de comunicação e um aplicativo específico disponível na web⁵.

No início de cada ciclo quinzenal de desenvolvimento (*Sprint*), eram realizadas presencialmente as *Sprint Plannings*, *Sprint Review* e *Sprint Retrospective*. Cada equipe apresentava atividades desenvolvidas, os pontos positivos e negativos ocorridos durante a *Sprint*, as dificuldades, lições aprendidas e perspectivas futuras, de forma que fosse planejado o próximo ciclo de desenvolvimento, como também o ciclo de integração para que os protótipos gerados fossem analisados e validados pelo *Product Owner*.

Cada iteração resultava em incrementos de funcionalidades do sistema. Dentro das equipes, a integração dos itens desenvolvidos foi realizada de forma contínua. No entanto, a integração dos artefatos gerados por cada equipe, responsabilidade do Time de Apoio, não foi realizada de forma contínua, gerando problemas como acúmulos de artefatos de diferentes equipes, dificuldades na integração e atrasos na entrega de protótipos. No final, a integração foi realizada gradativamente pelo Time de Apoio e por membros das equipes, que alinharam as últimas versões produzidas por cada equipe para que o produto final fosse gerado, testado e depois validado pelo *Product Owner*.

3.3. Gerenciamento do Projeto

Como citado anteriormente, foi adotado um processo ágil, baseado nas práticas do *Scrum*, para o gerenciamento e desenvolvimento do sistema The SeC. Apesar da complexidade do projeto, envolvendo equipes grandes e distribuídas, o seu gerenciamento foi eficiente e produtivo, pelo fato de ter dividido os participantes em times menores, assim foi possível trabalhar da melhor maneira as práticas *Scrum* dentro de cada equipe. Além disso, a criação de um Time de Apoio, formado pelos alunos mais experientes foi fundamental para gerenciar as atividades realizadas entre as equipes.

⁴ Mantis é uma ferramenta Web que realiza o controle de atividades no desenvolvimento de software.

⁵ <http://www.planningpoker.com/>

O gerenciamento dentro de cada equipe era realizado em conjunto por todos os membros, caracterizando equipes autogerenciáveis, não possuindo assim um gerente de projeto que determine as atividades de cada membro da equipe, mas sim cada membro possui autonomia na escolha de suas tarefas. Cada equipe possuía um *Scrum Master*, responsável por gerenciar conflitos dentro de sua equipe, remover os impedimentos durante o processo e interagir com os outros *Scrum Masters* e com o Time de Apoio.

Visando manter uma comunicação constante e transparência do andamento do processo, cada equipe realizava reuniões diárias (*Daily Scrum*). Por envolver equipes distribuídas, essas reuniões eram realizadas virtualmente, através de comunicadores instantâneos como *Messenger*, *Skype* e *Google Wave*, onde cada membro deveria expor as atividades desenvolvidas, os impedimentos e as próximas tarefas a serem realizadas. O *Scrum Master* do time ficava responsável por anotar os impedimentos.

O gerenciamento das equipes foi realizado pelo Time de Apoio, com o auxílio dos professores da disciplina. Mesmo com as equipes interagindo à distância, eram necessárias reuniões presenciais, que aconteciam semanalmente, para uma validação e avaliação conjunta do desenvolvimento. Além disso, o Time de Apoio era responsável por promover o *Scrum* de *Scrums* com os *Scrum Masters* das equipes para contatar dúvidas, expor situações, além de solicitar um suporte para o desenvolvimento. Esse gerenciamento do Time de Apoio ajudou a sincronizar as atividades entre as equipes.

3.4. Dificuldades e Adaptações do Projeto

Dificuldades foram observadas ao longo do desenvolvimento, muitos deles devido ao fato do projeto envolver equipes com muitos integrantes e distribuídas geograficamente, além de ter prazos curtos. As principais dificuldades foram:

- **Gerenciar Equipes Distribuídas:** gerenciar várias equipes de desenvolvimento compostas por pessoas desconhecidas e localizadas em diferentes cidades, principalmente por problemas de comunicação decorrentes da distância física;
- **Gestão de Atividades:** por envolver equipes distribuídas, tornou-se difícil acompanhar e validar todas as atividades realizadas em cada equipe, além das dificuldades para estimar tempo necessários para executar uma dada atividade;
- **Falta de domínio da plataforma utilizada:** a falta de domínio na plataforma Salesforce resultou em dificuldades para as equipes no início do projeto;
- **Ferramentas de Comunicação:** a falta de ferramentas para o suporte ao processo distribuído fez com que as equipes tivessem problemas de comunicação, esbarrando na ineficiência das tecnologias de comunicação, como o *Skype*, que não suporta muitos usuários ao mesmo tempo em uma conferência;
- **Reuniões Diárias:** atividade complexa por envolver equipes distribuídas, além das ferramentas de comunicação não fornecerem suporte adequado e das dificuldades de estabelecer um horário de reunião compatível para todos;
- **Integração:** gerenciar e integrar, de forma contínua, vários artefatos gerados por várias equipes foi uma atividade complexa, tanto pela abordagem distribuída, quanto pela dificuldade da plataforma no suporte a este recurso. Com isso, ocorreram atrasos na entrega de alguns protótipos do produto.

Com o objetivo de diminuir os problemas encontrados e utilizar da melhor maneira às práticas do *Scrum* no desenvolvimento e gerenciamento do projeto, foram realizadas algumas adaptações:

- A divisão dos integrantes em equipes menores tornou mais eficiente o gerenciamento do projeto, podendo ser implantada de maneira mais eficaz as práticas do *Scrum* dentro de cada equipe. Além disso, a utilização de um time de apoio ajudou acompanhar e validar as atividades realizadas em cada equipe;
- Para facilitar e tornar mais precisa a estimativa de esforço em cada atividade, foi utilizada a técnica do *Planning Poker*, tanto pelo Time de Apoio para atribuir os itens de *Backlog* para as equipes, como dentro de cada equipe, na divisão das atividades entre seus membros. Dentro das equipes, as rodadas de estimativas de *Planning Poker* de forma distribuída foram feitas através de ferramentas instantâneas de comunicação e aplicativos específicos disponíveis na web;
- Diante da falta de domínio e experiência de parte dos integrantes do projeto em relação à plataforma Salesforce, foi realizado um treinamento, de forma presencial, através de workshops, ocorrendo o nivelamento dos participantes;
- Visando manter o espírito de equipe, o comprometimento e a interação dentro de cada equipe, eram realizadas várias reuniões virtuais na semana, através de comunicadores instantâneos como *Messenger*, além do uso de listas de discussões. Para manter o comprometimento com as atividades, cada integrante deveria escolher sua atividade e estimar o prazo de conclusão;

4. Resultados e Lições Aprendidas

No estudo de caso relatado neste trabalho, foi utilizado um processo baseado nas práticas *Scrum*, sendo fundamental sua adaptação no contexto do projeto. Mesmo com todas as dificuldades, os resultados obtidos foram positivos, sendo listados a seguir:

- **Desenvolvimento eficiente e produtivo:** através da adaptação das práticas ágeis do *Scrum* no contexto do projeto, as equipes tiveram um desenvolvimento eficiente e produtivo, com o foco na entrega contínua do produto;
- **Construção de um negócio baseado em *cloud computing*:** o resultado final do projeto foi a implementação de um sistema de compras e vendas coletivas na Web através de uma plataforma nas nuvens – Salesforce;
- **Maturidade da equipe de desenvolvimento:** maior conhecimento em relação a desenvolvimento e gerenciamento ágil de projetos de software;
- **Conhecimento da plataforma Salesforce:** utilização de uma plataforma que propõe o desenvolvimento de soluções colaborativas nas nuvens.

As experiências vivenciadas neste projeto foram abrangentes no sentido de adaptar e utilizar práticas ágeis do *Scrum* em projetos de software com equipes grandes e distribuídas. Algumas lições aprendidas são apresentadas a seguir:

- Divisão dos integrantes em equipes menores, trabalhando da melhor maneira as práticas do *Scrum* em cada equipe, proporcionando um desenvolvimento mais dinâmico e produtivo, além de facilitar o gerenciamento do projeto;
- Equipes heterogêneas, facilitando a atribuição de atividades de acordo com o perfil e experiência de cada integrante. Para isso, foram aplicados questionários para verificar habilidades e experiências individuais;
- Equipes autogerenciáveis, fazendo os integrantes agirem de maneira proativa e em busca constante por resultados;
- Manter o comprometimento e a motivação, fatores ainda mais importantes em ambientes de DDS. Para isso, cada time realizava várias reuniões virtuais;

- Arquitetura modular, diminuindo as dependências de atividades entre as equipes;
- Definição do Time de Apoio foi fundamental para controlar todo o processo de desenvolvimento, pois sua consultoria foi uma presença constante, orientando e avaliando os artefatos gerados, além de resolver os conflitos existentes;
- Presença constante do *Product Owner*, esclarecendo e validando os requisitos;
- Ferramentas de comunicação ainda não suportam o trabalho ágil e distribuído;

5. Considerações Finais

Com a evolução do mercado de software, em conjunto com as vantagens competitivas do desenvolvimento distribuído, além da procura por baixos custos e curtos prazos de tempo, ocorreu um crescimento do número de empresas que adotam abordagens ágeis e distribuídas, conseguindo muitos benefícios, no entanto, enfrentando alguns desafios.

Pesquisar novas formas de adaptar metodologias de desenvolvimento de software às realidades de equipes e projetos é uma maneira interessante de trazer novos processos para diferentes abordagens. Este trabalho apresentou uma experiência de utilização de práticas ágeis do *Scrum* em projeto distribuído de software, mostrando que é possível adaptar boas práticas em diferentes contextos de projeto. Foram relatadas as dificuldades encontradas, assim como eventuais soluções para amenizar tais problemas.

Assim, o trabalho tem como contribuição apresentar os principais problemas enfrentados com a utilização do *Scrum* no DDS, além de abordar possíveis soluções que podem ser adotadas por equipes distribuídas que utilizam metodologias ágeis. É fundamental que experiências práticas, como a apresentada neste trabalho, sejam realizadas e colocadas à disposição, buscando resultados mais relevantes para a comunidade de software.

Referências

- Audy, J. and Prikładnicki, R. (2008) “Desenvolvimento Distribuído de Software”, Editora Elsevier.
- Carmel, E., Global Software Teams – Collaborating Across Borders and Time-Zones. EUA, Prentice-Hall, 1999.
- Ferreira, A. I. F., Cerqueira, R., Santos, G., et al. (2006) “ISO 9001:2000, MPS.BR Nível F e CMMI Nível 3: Uma Estratégia de Melhoria de Processos na BL Informática”, V SBQS, pp. 375-382, Vila Velha, Brasil.
- Prikładnicki, R. (2003) “MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software”, Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Brasil.
- Rico, D. F., Sayani, H. H., and Sone, S. (2009) “The Business Value of Agile Software Methods: Maximizing Roi With Just-in-time Processes and Documentation”, J. Ross Publishing.
- Schwaber, K. (2004) “Agile Project Management with Scrum”, Microsoft Press.
- Sutherland, J., Viktorov, A., Blount, J., Puntikov, N. (2007) "Distributed Scrum: Agile Project Management with Outsourced Development Team", HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii.