

Metodologia Ágil Scrum em uma Disciplina de Engenharia de Software

Breno Antonivaldo Lessa Andrade¹, Moara Sousa Brito¹, Crescencio Lima¹

¹ Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) - BA – Brasil

{breno.lessa.andrade, moarabritto, crescencio}@gmail.com

Abstract. *The Agile software development stands out mainly for prioritizing functionality development through executable code instead of the extensive written documentation allowing quick responses to changes. Due to this context, this paper aims to report the implementation of the Scrum agile methodology in a classroom environment, with all the project execution process. Finally, we present an experience that simulated a real situation, common in development companies, to facilitate the course content learning.*

Resumo. *As metodologias ágeis de desenvolvimento de software destacam-se principalmente por darem prioridade ao desenvolvimento de funcionalidades através de código executável ao invés da produção de extensa documentação escrita, apresentando respostas rápidas às mudanças. Devido a este contexto, este trabalho busca relatar a aplicação da metodologia ágil Scrum em um ambiente de sala de aula, apresentando todo o processo de execução do projeto. Obtendo ao final uma experiência que simulou uma situação real, comum nas empresas de desenvolvimento, facilitando assim o aprendizado do conteúdo da disciplina.*

1. Introdução

As metodologias ágeis de desenvolvimento de *software* se destacam dos processos de desenvolvimento tradicionais devido, principalmente, ao fato de darem prioridade ao desenvolvimento de funcionalidades através de código executável ao invés da produção de extensa documentação escrita, e ainda, respostas rápidas às mudanças e colaboração com o cliente ao invés de seguir planos rígidos e negociações contratuais [Leal 2009]. De acordo com uma pesquisa realizada por Ambler (2006), após a coleta de informações de um grande número de profissionais que utilizavam diferentes métodos, tanto ágeis quanto tradicionais, percebeu-se que métodos ágeis realmente melhoram os resultados dos projetos em termos de qualidade, satisfação do cliente e produtividade, sem aumentar seu custo de forma significativa.

Um dos principais métodos de gerenciamento ágil de projetos de software é o *Scrum* [Schwaber 2004]. Segundo Sousa *et al.* (2015), o gerenciamento com *Scrum* implica na utilização de um framework de procedimentos e artefatos que conduzam aos objetivos da auto organização das equipes de desenvolvimento.

Tal metodologia pode ser utilizada no ambiente de sala de aula, no processo de ensino aprendizagem, uma vez que Prikladnicki e Audy (2008) sugerem que abordagens alternativas podem ajudar os alunos a aprender de maneira mais efetiva, tendo como

exemplos abordagens focadas nos alunos, a partir da substituição de aulas expositivas por discussão de casos práticos, dinâmicas de grupos, ou a execução de um projeto do início ao fim pelos alunos.

De acordo com este cenário, este artigo relata a experiência da utilização da Metodologia ágil *Scrum* no ambiente de sala de aula em um projeto de desenvolvimento de *software* na disciplina de Engenharia de *Software*.

O restante do artigo está organizado da seguinte maneira: na seção 2 são abordados os trabalhos correlatos. Na seção 3 são apresentados os principais mecanismos utilizados no desenvolvimento do projeto. Na seção 4 o artigo é concluído e os trabalhos futuros são apresentados.

2. Trabalhos Correlatos

Meireles e Bonifácio (2015) relatam a experiência prática de alunos de graduação no aprendizado da disciplina de Engenharia de *Software*, utilizando como ferramenta metodológica a combinação de metodologia de ensino baseada em problema (PBL) e o uso de *Scrum* para construção de projetos reais, aplicado para dispositivos móveis, com o objetivo de contribuir para a criação de um ambiente motivador, tornando o processo de aprendizagem mais dinâmico, colaborativo e prazeroso. O trabalho desenvolvido pelos autores apresenta forte relação de semelhança com o relato de experiência apresentado neste artigo, uma vez que foca no ensino da Engenharia de Software atrelada ao desenvolvimento de produtos, buscando assim, fornecer aos estudantes experiências reais. Em contrapartida os trabalhos se diferem, uma vez que Meireles e Bonifácio (2015) relataram que os estudantes se dividiram em 04 (quatro) equipes, onde cada uma delas possuiu a liberdade de escolher o tema da aplicação móvel. Já neste presente trabalho, os estudantes mesmos separados em equipes uniram esforços para desenvolver uma ferramenta maior, realizando um trabalho colaborativo, além disso, uma série de testes de software foram realizados para avaliar a qualidade do projeto desenvolvido.

Carvalho e Mello (2012) apresentam o resultado de uma pesquisa realizada em uma pequena empresa de base tecnológica, na qual, o objetivo do trabalho foi analisar a implantação do *Scrum* nos projetos de desenvolvimento de produtos de *software*, além de compreender e mensurar o impacto desta implantação na empresa. Os autores perceberam que com a adoção do método, melhorou a comunicação e aumentou a motivação do time, diminuiu o custo, o tempo e o risco do projeto e aumentou a produtividade da equipe. Já Branco Coelho e Pires (2008) apresentam a experiência no uso de *Scrum* em projetos *offshore*, ressaltando as vantagens percebidas.

Siller e Braga (2013), dentro da temática da Engenharia de *Software*, apresentada um *software*, mais precisamente um jogo educacional, para ensino e prática do *Scrum*, mostrando ao aluno como o processo funciona na prática. O processo de desenvolvimento foi norteado pela metodologia INTERA que segundo os autores, é específica para o desenvolvimento de objetos de aprendizagem.

Por fim, é possível perceber que existem diversos trabalhos que buscam promover nos estudantes a experiência de um processo de desenvolvimento, mas poucos abordam especificamente as atividades realizadas no ambiente acadêmico como meio de

aprendizagem, ao mesmo tempo em que as etapas presentes em um ambiente de produção são criteriosamente seguidas e relacionadas ao conteúdo teórico da sala de aula. Dessa forma, no presente artigo será relatada a experiência em utilizar a metodologia *Scrum* como auxílio no processo de ensino e aprendizagem no ambiente de sala de aula.

3. Utilização da Metodologia Ágil *Scrum*

3.1. Contextualização

Scrum é uma abordagem ágil para o desenvolvimento de produtos e serviços [Rubin 2012]. De acordo com Fadel e Silveira (2010), essa abordagem não define uma técnica específica para o desenvolvimento de *software* durante a etapa de implementação. Ela se concentra em descrever como os membros da equipe devem trabalhar para produzir um sistema flexível, num ambiente de mudanças constantes. Nela existem 03 (três) papéis importantes e bem definidos por Wazlawick (2013) da seguinte forma: (i) *Scrum Master* é o facilitador e solucionador de conflitos; (ii) *Product Owner*, é a pessoa responsável pelo projeto em si; (iii) *Scrum team*, é a equipe de desenvolvimento, onde todos interagem para desenvolver o produto em conjunto.

O processo do *Scrum*, conforme pode ser visto na Figura 1, inicia-se pelo *Product Backlog*, onde contém todas as funcionalidades definidas com o cliente/usuário, melhorias e correções de defeitos a serem desenvolvidas no projeto [Oliveira e Lima 2011]. Com base nessa lista, é planejada a *Sprint* que será executada. *Sprints* são iterações durante as quais são implementados conjuntos de itens do *Product Backlog* (lista de funcionalidades desejadas), podendo variar a sua duração, de 02 (duas) a 04 (quatro) semanas. No início da *Sprint*, o *Product Owner*, o *Scrum Master* e o time participam da reunião de planejamento, que é composta por duas partes. Na primeira parte participam o *Product Owner*, o *Scrum Master* e o time com a finalidade de selecionar as tarefas (*Backlog* do Produto selecionado) que farão parte do *Backlog* da *Sprint* conforme priorização do *Product Owner*. Na segunda parte, participam o *Scrum Master* e o time para subdividir as tarefas com o intuito de melhor entendê-las para facilitar a implementação de cada uma delas [Oliveira e Lima 2011].

Durante a execução da *Sprint*, o *Scrum Master* e o time realizam reuniões diárias (*Daily Scrum Meeting*) para acompanharem o andamento do projeto. Ao término da *Sprint*, o time realiza a apresentação do incremento do produto desenvolvido para o *Product Owner*, que por sua vez, valida se o que foi solicitado corresponde ao que foi desenvolvido. Assim, a *Sprint* é finalizada iniciando-se um novo ciclo até concluir o produto final [Oliveira e Lima 2011].

3.2. Execução do Projeto

Para o desenvolvimento do projeto principal, a turma de 20 (vinte) pessoas foi dividida em 03 (três) times. A divisão das equipes aconteceu através da afinidade entre os alunos

verificada através de análise SWOT¹. Um questionário com perguntas relacionadas aos pontos fortes e fracos, oportunidades e ameaças foi preenchido por cada aluno.

Cada time trabalhou com módulos diferentes do software principal. A fim de possibilitar o desenvolvimento paralelo entre os times, foram criados ambientes de desenvolvimento para cada equipe utilizando a ferramenta GitHub (2016) e, esses módulos foram integrados, realizando a sincronização destes ambientes em um projeto único. Um integrante de cada time ficou responsável por esta tarefa. Houve uma ordem preestabelecida para ocorrer tal sincronização e, ao final do processo, o ambiente principal estava com todas as alterações.

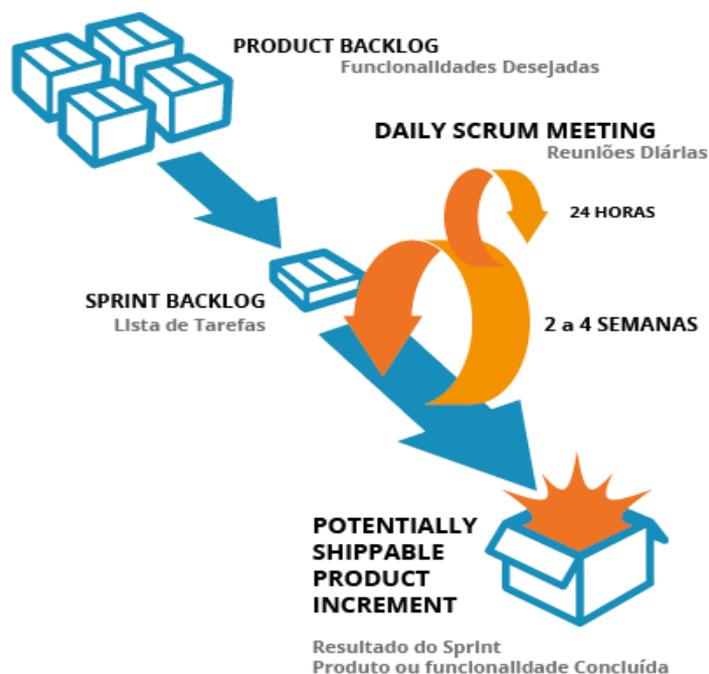


Figura 1: Funcionamento do Scrum.

Fonte: BRQ [BRQ 2016].

A fase inicial do projeto deu-se através de pesquisas e estudos realizados a partir de documentos, capítulos de livros e teses de mestrado relacionados com o projeto. Posterior à fase de entendimento do projeto, foi necessária uma reunião com o docente, que durante o projeto assumiu os papéis de *Product Owner* e cliente do projeto. Nesta reunião foi realizada uma entrevista a respeito dos objetivos e requisitos desejados na ferramenta, assim como, foram sanadas as dúvidas das equipes.

A partir dos requisitos elencados na entrevista, cada time ficou responsável em criar o seu *Product Backlog* de acordo as pesquisas realizadas, recorrendo para isso, às planilhas do Google Docs (2016), por possibilitar o acesso de todos ao mesmo tempo, e o mais importante, possibilitar o acesso de qualquer lugar independente do dispositivo de acesso.

¹ SWOT é a sigla dos termos ingleses *Strengths*, *Weaknesses*, *Opportunities* e *Threats* que consiste em uma metodologia bastante popular no âmbito empresarial.

As colunas que compõem a planilha são: “*desejo*” - requisito solicitado pelo cliente; “*de forma de...*” - coluna responsável por armazenar uma justificativa ou explicação do requisito; “*Tipo*” - classificação do tipo do requisito, onde o mesmo poderia ser uma característica do sistema ou uma apresentação a ser realizada; “*Importância*” - recebe uma pontuação de acordo o grau de urgência ou importância de uma solicitação, permitindo identificar uma ordem de implementação; “*Status*” - responsável por armazenar o status em que se encontra uma implementação, podendo sofrer alterações entre: TO DO (a ser feito), WIP (em andamento) e DONE (concluído); “*Sprint Estimada*” e “*Sprint Feita*” - o primeiro armazena o número da *Sprint* em que a equipe espera cumprir o requisito e o segundo armazena o número real da *Sprint* em que realmente foi atendida a solicitação; “*Responsável*” - informa o responsável por cumprir determinado item, por fim; “*Notas*” - armazena possíveis sugestões e informações adicionais.

Após a análise e documentação dos requisitos, foi iniciado os ciclos de *Sprints*. Para isso, antes de se iniciar uma *Sprint*, os times selecionavam os itens que deveriam ser implementados, recorrendo assim, ao *Sprint Backlog*. Além disso, a cada nova implementação concretizada ou não, atualizava-se o status de andamento do requisito listado e, caso o mesmo não fosse implementado a tempo do fim da *Sprint*, o mesmo deveria ser alocado para a próxima *Sprint Backlog*.

As funcionalidades selecionadas em cada *Sprint Backlog* possuíam um tempo para serem desenvolvidas e entregues ao cliente, sendo definido no início da disciplina o prazo máximo de 15 (quinze) dias para cada *Sprint*. Essa duração foi escolhida devido ao tempo disponível para execução da disciplina, ao mesmo tempo em que facilitaria o acompanhamento do mesmo. Houve apenas um momento em que o prazo estipulado não foi atendido, devido uma greve ocorrida, adotando então, a medida de estender a duração da *Sprint* até a volta às aulas.

Para cada equipe foi eleito um *Scrum Master*, no entanto esse cargo foi rotativo, ou seja, em cada *Sprint* a responsabilidade era compartilhada, sendo eleito um novo integrante para assumir o cargo, possibilitando assim que todos os alunos tivessem a oportunidade de participar dessa experiência.

Essa tática de rotatividade utilizada contribuiu de forma positiva para o desenvolvimento do projeto, levando em consideração que inicialmente alguns membros do time não possuíam tanto comprometimento como os demais. Entretanto, ser *Scrum Master* aumentou o nível de comprometimento, pois era de sua responsabilidade ser a interface de comunicação do time com o *Product Owner*, ou seja, todas as críticas, orientações e questionamentos eram direcionados para o *Scrum Master* e esse deveria repassar para o restante do time, eliminando os impedimentos e acompanhando as atividades do grupo. Porém, vale ressaltar que alguns integrantes tiveram dificuldade ao assumir o papel de *Scrum Master*, na maioria das vezes por não possuírem um perfil de comunicação, mas essa dificuldade foi superada com ajuda da equipe.

Durante o andamento da *Sprint*, cada equipe deveria realizar reuniões diárias (*Daily Scrum Meeting*) a fim de disseminar o conhecimento e atualizar sobre o andamento do projeto. Estas reuniões aconteceram de duas formas, presencial e virtual. Na forma presencial, aproveitou-se das reuniões semanais para estabelecer as comunicações. Além disso, foi necessário recorrer aos intervalos entre as aulas da

disciplina e do espaço reservado pelo professor para discussão e atualização do andamento. Na forma virtual, utilizou-se recursos de videoconferência e chat, recorrendo às ferramentas Skype (2016), Hangouts (2016) e Facebook (2016).

Além das reuniões necessárias para manter a equipe alinhada, foi utilizado o Trello (2016). Este gerenciador de atividades foi utilizado para substituir a *dashboard* do *Scrum* e permitiu que cada time pudesse controlar de forma eficaz as tarefas, estabelecendo uma organização, ao mesmo tempo em que era possível classificar as atividades a serem realizadas em “*Tarefas não iniciadas*”, “*Em progresso*” e “*Concluídas*”, conforme pode ser visto na Figura 2.

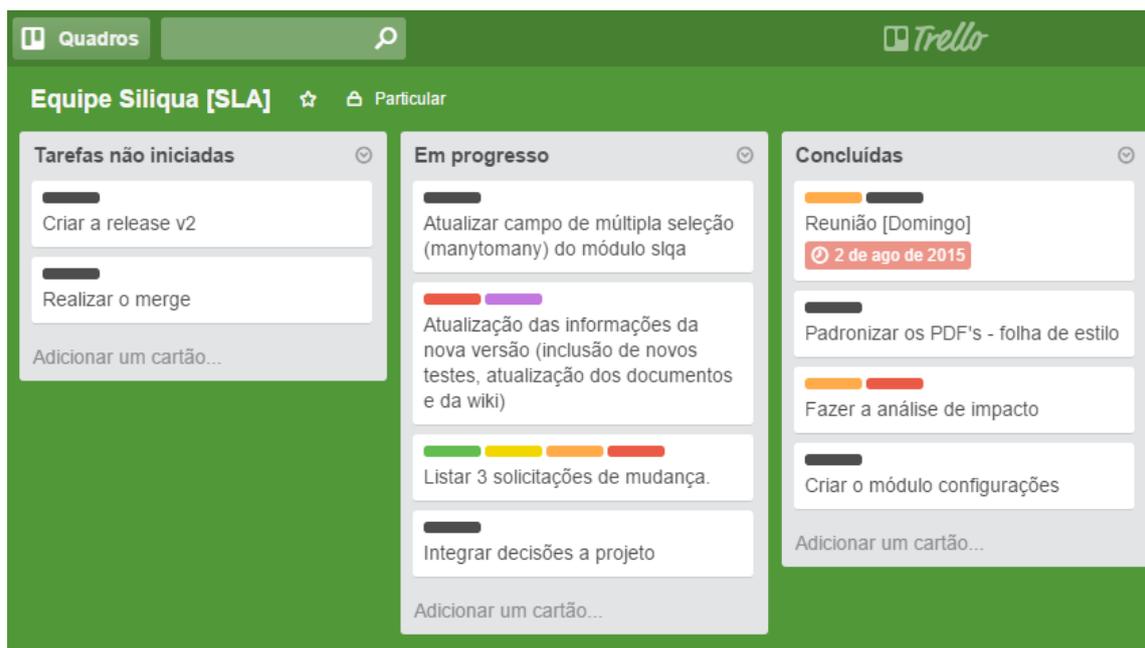


Figura 2: Trello 2016.

Fonte: Próprio Autor

Após o término de cada *Sprint*, as atividades concluídas eram reorganizadas e agrupadas de acordo a numeração da *Sprint*, mantendo assim, um histórico, conforme a Figura 3.



Figura 3: Histórico no Trello 2016.

Fonte: Próprio Autor

Ao final de cada *Sprint*, as equipes realizaram atividades de validação, executando testes de unidade, integração, desempenho e de sistema, obtendo resultados que eram registrados através de planilhas. No final da *Sprint* era necessário realizar apresentações ao *Product Owner*, demonstrando assim, as funcionalidades implementadas e, conseqüentemente, definindo as próximas atividades.

Ao serem finalizadas todas as etapas de desenvolvimento dos três times e estando válidos todos os requisitos levantados, procedeu-se a junção dos três módulos em um único sistema, unificando e criando uma versão estável da ferramenta desenvolvida, que foi apresentado como trabalho final da disciplina. Sendo assim, houve uma reformulação das equipes, onde os alunos passaram a trabalhar de acordo com o seu perfil.

Desta forma, os 20 (vinte) alunos foram divididos em 03 (três) novos times, na qual, cada integrante escolheu a área de maior domínio e interesse, renovando o comprometimento por parte dos estudantes e permitindo que as atividades fossem concluídas de forma mais rápida. No entanto, houve dificuldade em definir o tempo necessário para desenvolvimento das atividades por cada integrante, uma vez que, com a criação de novos times e o fato dos integrantes nunca terem trabalhado em conjunto, os mesmos necessitaram se adaptar com a forma de trabalho de cada um. Além disso, cada integrante possuía um horário de disponibilidade e, ritmo de trabalho o que também exigiu certo tempo para adaptação. As equipes montadas foram organizadas da seguinte forma: desenvolvimento, testes e documentação.

Durante todo o projeto, gerenciou-se os entregáveis utilizando o GitHub, sendo escolhido por ser gratuito e possuir funcionalidades que permitem a documentação, gerenciamento de código, repositório de fonte aberto e possibilitar a criação de grupos de colaboradores, dessa forma o código fonte, os resultados dos testes e toda a

documentação da ferramenta estão disponíveis no repositório que pode ser acessado publicamente no endereço X².

4. Conclusões e Trabalhos Futuros

O ambiente da sala de aula precisa de novas abordagens de forma a permitir a modernização do aprendizado e garantir que o professor e os estudantes desenvolvam novas experiências. Pois, a partir de métodos reais aplicados no mercado de trabalho, gera-se uma vivência de mundo que não pode ser obtida através das práticas habituais no ambiente acadêmico.

A aplicação prática do método *Scrum* aliado ao processo de desenvolvimento apresentou resultados positivos e negativos. Os positivos resultaram em um desenvolvimento característico das empresas de desenvolvimento, obtendo um software implementado de maneira rápida e satisfatória, promovendo momentos de aprendizagem ao mesmo tempo em que se desenvolvia o projeto, simulando assim, situações reais. Os pontos negativos não estão relacionados à metodologia, mas sim a falta de prática e conhecimentos por partes dos discentes envolvidos na abordagem. Entre os pontos de crítica levantados, estão os curtos prazos para os entregáveis, uma vez que os estudantes relataram estar acostumados com períodos e ritmos maiores para entrega de trabalhos; além disso, houve problemas relacionados ao manuseio da ferramenta GITHUB, uma vez que os times envolvidos não apresentavam conhecimento técnico da mesma, recorrendo para isso a estudo e reuniões conjuntas entre as equipes.

Por fim, é visto que a aplicação prática da metodologia ágil *Scrum* em um ambiente de sala de aula foi bem sucedida, apresentando um sistema funcional, ao mesmo tempo em que ressaltou nos alunos envolvidos um senso de comprometimento durante todo o processo.

Como trabalhos futuros, pretende-se aplicar o *Scrum* em uma nova turma de estudantes ao mesmo tempo, novos conceitos da Engenharia de Software são aprendidos, possibilitando uma análise mais aprofundada.

5. Agradecimento

Nossos agradecimentos aos demais estudantes que participaram de todo o processo de desenvolvimento utilizando o *Scrum*.

Os autores gostariam de agradecer à Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPGI) do IFBA pelo apoio financeiro ao desenvolvimento desta pesquisa (Edital BP003-04/2014/PRPGI).

Referências

Ambler, Scott W (2006). Agile Adoption Rate Survey Results: March 2006. Disponível em: <<http://www.ambysoft.com/surveys/agileMarch2006.html>>. Acesso em: 26 fev. 2016.

² Devido a restrição BLIND REVIEW, foi necessário omitir esta informação já que possui informações da instituição de ensino.

- Branco, Jônathas Diógenes Castelo; COELHO, Ciro Carneiro; PIRES, Carlo Giovano S.(2008) Utilizando Scrum em projetos off-shore. In: I Congresso Tecnológico Infobrasil, Fortaleza.
- BRQ. (2016) Metodologias Ágeis de desenvolvimento de software, <http://www.brq.com/metodologias-ageis/>.
- Carvalho, Bernardo Vasconcelos de; MELLO, Carlos Henrique Pereira.(2012). Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. *Gestão & Produção: Scielo*, São Carlos, v. 19, n. 3, p.557-573.
- Facebook. (2016). Disponível em: <https://www.facebook.com/>. Acessado em: 16 de fevereiro de 2016.
- Fadel A. C. e Silveira, H. da M. (2010) Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean. 26 f. Monografia (Especialização) - Curso de Faculdade de Tecnologia, Unicamp – Universidade Estadual de Campinas, Limeira.
- Github. (2016) Disponível em: <https://github.com/>. Acessado em: 16 de fevereiro de 2016.
- Google Docs. (2016). Disponível em: <http://docs.google.com/>. Acessado em: 16 de Fevereiro de 2016.
- Hangouts. (2016). Disponível em: <https://hangouts.google.com/>. Acessado em: 16 de fevereiro de 2016.
- Leal, I. (2009) “Requisitos de Metodologias de Teste de Software para Processos Ágeis”. Outubro.
- Meireles, M. A. C. e Bonifacio, B. A. (2015). Uso de Métodos Ágeis e Aprendizagem Baseada em Problema no Ensino de Engenharia de Software: Um Relato de Experiência. In Anais do XXVI Simpósio Brasileiro de Informática na Educação (XXVI SBIE). p.180-189.
- Oliveira, E. e Lima, R. (2011) Estado da Arte Sobre o Uso do Scrum em Ambientes de Desenvolvimento Distribuído de Software. *Revista de Sistemas e Computação*, Salvador, v. 1, n. 2, p.106-119.
- Prikladnicki, Rafael; AUDY, Jorge.(2008) Desenvolvimento distribuído de software. Rio de Janeiro: Elsevier.
- Rubin, K. S. (2012) *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. AddisonWesley Professional. 504p.
- Schwaber K. (2004) *Agile Project management with Scrum*. Microsoft Press.
- Sille, F. e Braga, J. A. (2013) Software Educacional para Prática do Scrum. In II CBIE – Workshops (WCBIE). pp. 152-161.
- Sousa, Hanna Tátila et al. (2015). Apoio Automatizado ao Planejamento de Sprints em Projetos Scrum. In Anais XI Simpósio Brasileiro de Sistemas de Informação. SBC.

Skype. (2016). Disponível em: <http://www.skype.com>. Acessado em: 16 de fevereiro de 2016.

Trello. (2016). Disponível em: <https://trello.com/>. Acessado em: 16 de fevereiro de 2016.

Wazlawick, R. S. (2013) Engenharia de Software: Conceitos e Práticas. Elsevier, Rio de Janeiro.