



A Utilização do Scrum como Recurso Educacional no Processo de Aprendizagem em Engenharia de Software

Carlos Alexandre Gouvea da Silva¹; Edson Leonardo dos Santos²; Lucilene Moreira Angelo³; Mary Anne da Cruz Siqueira de Oliveira⁴; Rafael Veiga de Moraes⁵

¹carlos.gouvea@ufpr.br, UFPR, Brasil

²edson.santos@pr.senai.br, UFPR, Brasil

³lucilene.angelo@pr.senai.br, SENAI-PR, Brasil

⁴mary.oliveira@pucpr.edu.br, SENAI-PR, Brasil

⁵veiga.moraes@pucpr.edu.br, SENAI-PR, Brasil

Resumo

O desenvolvimento e a formação técnica na Educação Profissional Tecnológica constituem um conjunto de métodos e ferramentas educacionais pedagógicas que permitem o aprendizado, exercício e a prática dos alunos em muitas instituições de ensino. Com esse propósito, projetos interdisciplinares buscam contribuir para uma melhor aprendizagem desses indivíduos partindo da construção teórica à prática de forma mais rápida e eficiente. Em cursos do eixo da Informática ou Computação, a utilização de métodos ágeis e projetos reais vêm sendo empregada como forma de melhorar e aprimorar o aprendizado dos alunos durante a sua formação profissional. Este artigo apresenta o relato positivo da utilização de método ágil Scrum em um projeto com cliente real numa prática interdisciplinar envolvendo alunos do Curso Técnico em Informática em uma Instituição de Ensino Profissionalizante privada na cidade de Curitiba, PR. Resultados mostraram que a realização de projetos interdisciplinares com o Scrum, permitiu uma prática de desenvolvimento de *software* real onde o aluno pôde conhecer o funcionamento de processos ágeis e os desafios que consistem na entrega contínua de produtos de *software* funcional bem como os desafios na satisfação dos clientes.

Palavras-chave: Aprendizagem Profissional, Metodologias Ágeis, Projetos Interdisciplinares, Scrum.

Abstract

The development and technical training in technical education is a set of methods and pedagogical educational tools that enable learning, exercise and practice of students in many educational institutions. To this end, interdisciplinary projects seek to contribute to better learning of these individuals starting from the theoretical construction to practice more quickly and efficiently. In courses of computer or computing, the use of agile methods and real projects are being employed in order to improve and enhance student learning during their professional training. This article presents the positive report of the use of agile method Scrum in a project with real customer an interdisciplinary practice involving students of Technical Course in Computer by a private vocational educational institution in the city of Curitiba, PR. Results showed that the realization of interdisciplinary projects with Scrum, allowed a real *software* development practice where the student could understand the operation of agile processes and challenges involving the continuous delivery of functional *software* products as well as the challenges in meeting the customers.

Keywords: Interdisciplinary Projects, Methodologies Agile, Professional Learning, Scrum.

Resumen

El desarrollo y la formación técnica en la educación profesional tecnológica constituyen un conjunto de métodos y herramientas pedagógicas educacionales que permiten el aprendizaje, el ejercicio y la práctica de los estudiantes en muchas instituciones educativas. Con este fin, los proyectos interdisciplinarios buscan contribuir a un mejor aprendizaje de estos individuos a partir de la construcción teórica a la práctica más rápida y eficaz. En cursos afines al de Informática o Computación, se están empleando el uso de métodos ágiles y proyectos reales con el fin de mejorar y potenciar el aprendizaje del estudiante durante su formación profesional. En este artículo se presenta un informe positivo del uso del método Scrum ágil en un proyecto con el cliente real de una práctica interdisciplinaria que involucra a los estudiantes del curso Técnico en Informática en un centro de enseñanza profesional privada en la ciudad de Curitiba, PR. Los resultados mostraron que la realización de proyectos interdisciplinarios con Scrum, permitió una práctica de desarrollo de *software* real donde el estudiante pudo comprender el funcionamiento de los procesos ágiles y los desafíos que implican la entrega continua de productos de *software* funcional, así como los desafíos relacionados con la satisfacción de los clientes.

Palabras claves: Aprendizaje profesional, Metodologías Ágiles, Proyectos Interdisciplinarios, Scrum.

1. Introdução

Desde os anos 90 quando o Brasil encontrava-se em um momento de grande modificação da economia, os brasileiros viam a sua formação como uma forma de obter melhores oportunidades profissionais. Assim, foram criadas diversas políticas de incentivo, inserção e estímulo a uma formação orientada a um mercado deficitário de mão de obra qualificada em diferentes áreas e tecnologias [1]. As políticas criadas a partir deste período visaram suprir as novas necessidades da classe trabalhadora a partir das mudanças ocorridas no mundo do trabalho [2]. Até os dias atuais, a qualificação profissional é porta de entrada para muitos jovens que tentam inserir-se no mercado e por profissionais que buscam por melhores qualificações e oportunidades.

Os desafios inerentes à Educação Profissional estão ligados a alguns fatores essenciais como financiamento, formação de professores, ambientes adequados de aprendizagem, educação como fator de inclusão social e principalmente como estratégia de desenvolvimento [3]. Esse desenvolvimento é associado a práticas educacionais que permitem uma formação de qualidade dos profissionais enquanto em salas de aula e/ou práticas acadêmicas que permitam a ampliação do conhecimento desses indivíduos. Em instituições educacionais, métodos são aplicados como forma de melhorar o aprendizado individual e coletivo de seus indivíduos, como na Aprendizagem Significativa sugerida pelo SENAI (Serviço Nacional de Aprendizagem Industrial) [4].

Na Aprendizagem Significativa, os conhecimentos prévios do aluno, ao interagir com os novos conhecimentos, vão sofrendo mudanças, adquirindo novos significados e diferenciando-se progressivamente. Diferentemente, na Aprendizagem Mecânica e Repetitiva, o aluno não consegue articular os conhecimentos já construídos com as novas informações. Ao não construir significado, o aluno pode limitar-se à memorização do conteúdo e encontrar dificuldades no processo de ensino e aprendizagem [4].

A construção da aprendizagem e do conhecimento é realizada a partir de meios nos quais os alunos possam absorver os conceitos teóricos inerentes a um tema de estudo buscando aplicá-los na resolução e idealização de soluções de problemas ou melhorias em processos, produtos ou tecnologias. Esse aprendizado pode ser construído a partir de pesquisas, trabalhos em grupo, visitas técnicas, rodas ou grupos de discussão, provas orais ou escritas, seminários e apresentações, situações problema, projetos, entre outras.

Em relação aos projetos como mediador da absorção de conhecimento, Prado (2003) esclarece que:

Na pedagogia de projetos, o aluno aprende no processo de produzir, de levantar dúvidas, de pesquisar e de criar relações, que incentivam novas buscas, descobertas, compreensões e reconstruções de conhecimento. E, portanto, o papel do professor deixa de ser aquele que ensina por meio da transmissão de informações - que tem como centro do processo a atuação do professor -, para criar situações de aprendizagem cujo foco incide sobre as relações que se estabelecem neste

processo, cabendo ao professor realizar as mediações necessárias para que o aluno possa encontrar sentido naquilo que está aprendendo, a partir das relações criadas nessas situações[5].

A realização de projetos como mediação não pode ser visto meramente como mecanismo unilateral do aprendizado, mas sim como uma parte de um conjunto de mecanismos, sendo o professor uma parte crucial desse processo. Sobre a relação do professor nesse processo Rocha, Sabino e Acipreste (2015) dizem que:

[...] há a necessidade de propostas pedagógicas que se alinhem aos aspectos de colaboração, compartilhamento, autonomia e experimentação. Com a orientação do professor, os alunos podem experimentar situações em que são chamados à participação efetiva, com a articulação de ideias entre os colegas, possibilitando o desenvolvimento de processos mentais, habilidades cognitivas e de relacionamento social [6].

Conciliada aos métodos de aprendizagem e seus mecanismos de intermediação do conhecimento, o conceito de interdisciplinaridade vem com o objetivo de melhorar a construção dos preceitos da Educação Profissional. A interdisciplinaridade visa superar a fragmentação e o caráter de especialização do conhecimento, comumente presente no modelo disciplinar e desconectado da formação atual. Esse panorama é decorrente da estruturação fragmentada dos currículos, lógica funcional e racionalista que o poder público e a iniciativa privada impõem na organização de seus currículos [7].

Em cursos de formação voltados a Informática e a Computação, a grande variedade de ferramentas e técnicas existentes permitem uma interação e imersão mais efetiva entre alunos, professores e as tecnologias da área. O desenvolvimento ágil é um conceito utilizado no mercado de trabalho, com objetivo de maximizar os resultados e na obtenção de melhores níveis de qualidade de produtos de *software*. Métodos ágeis, como o Scrum, vêm sendo utilizados como um recurso didático para a prática e desenvolvimento do aprendizado acadêmico de alunos na realização de projetos e estudos relacionados.

Apresentamos como objetivos deste artigo expor os resultados da utilização do Scrum, como método ágil de desenvolvimento, em uma turma de Curso Técnico em Informática de uma Instituição de Educação Profissional privada da cidade de Curitiba. Os alunos foram desafiados a construir um *software* funcional de um cliente real, a partir de um projeto interdisciplinar envolvendo diferentes disciplinas do curso com duração de aproximadamente três meses. Resultados mostraram que os alunos após a utilização de métodos ágeis na construção de *software* desenvolveram capacidades essenciais para a sua formação técnica como: interpretação e leitura de processos de desenvolvimento, escolha e aplicação de ferramentas ágeis, definição e controle de processos a partir de ferramentas gerenciais, tomada de decisões e resolução de conflitos, estimação de prazos e entrega de versões intermediárias, autocontrole e gerenciamento na ocorrência de problemas, trabalho em equipe e análise sistemática das etapas que compreendem o processo de desenvolvimento de aplicações de *software*.

2. Fundamentação Teórica

2.1. Métodos Ágeis

O desenvolvimento de *software* a partir dos anos 60 começou a modificar os diferentes padrões de relacionamento entre o ser humano e as tecnologias computacionais inerentes à época. Essa interação acarretou no aumento pela procura e necessidade de ferramentas computacionais em diversos segmentos, gerando assim ao longo dos anos o aparecimento de processos de desenvolvimento de *software* convencionais como o cascata, o evolutivo, espiral e outros [8].

A necessidade por melhores modelos de desenvolvimento de *software* onde fossem influenciados variáveis como os desenvolvedores, a comunicação e a interação entre os envolvidos, a necessidade de documentação, os critérios de qualidade ou o gerenciamento de projetos fez aparecer o conceito de Desenvolvimento Ágil de *software*, que apresentou grandes vantagens aos processos convencionais [9]. Esse novo conceito surgiu a partir de uma conferência, chamada Aliança Ágil, onde foram definidos doze princípios ágeis a serem seguidos e que na prática fornecem um novo modo de desenvolvimento orientado a entregas rápidas de *software* com

qualidade e maximização do trabalho realizado [10]. Dentre os principais métodos ágeis de desenvolvimento de *software* se destacam o XP (*Extreme Programming*), ou programação extrema, e o Scrum [11]. Mesmo apresentando grandes avanços e vantagens, o XP não foi usado no desenvolvimento desse trabalho de pesquisa, ficando somente restrito ao uso do Scrum. A Aliança Ágil foi criada em fevereiro de 2001 em Utah, EUA com a participação de dezessete especialistas e profissionais da área, que se reuniram para a definição de um denominador comum a respeito de Desenvolvimento Ágil de *Softwares*. O XP foi criado no final dos anos noventa nos Estados Unidos, o XP é uma metodologia de desenvolvimento ágil empregada em projetos para equipes pequenas e média onde os requisitos não necessitam estar bem definidos e que podem ser modificados ao longa da construção dos *software*.

2.2. Scrum

O método de desenvolvimento ágil Scrum é utilizado amplamente por equipes de desenvolvimento em todo o mundo com objetivo de maximizar as tarefas e a produtividade em um projeto de *software* com foco na qualidade de produtos de sistemas. O Scrum foi criado em 1993 por Ken Schwaber e Jeff Sutherland e constitui um conjunto de práticas e técnicas utilizadas por anos, sendo inicialmente apresentada em 1995 na OOPSLA (*Object-Oriented Programming, Systems, Languages Applications*) e aprimorada ao longo dos anos [12]. O OOPSLA é uma conferência de pesquisa realizada anualmente pela Association for Computing Machinery (ACM). Tem como principal objetivo a difusão de trabalhos que tratem de qualquer aspecto o desenvolvimento de *software* como: requisitos, modelagem, prototipagem, design, implementação, geração, análise, verificação, validação, teste, manutenção, reutilização, substituição e linguagens de programação.

O Scrum é considerado um framework de simples inspeção e adaptação com três grupos de responsabilidade, três momentos de atividades e três artefatos, sendo:

- Responsáveis: *Product Owner*, Scrum Master e *Scrum Team*, e suas responsabilidades são indicadas no Quadro 1;
- Momentos: *Sprint Planning*, *Sprint Review* e *Daily Scrum*; e
- Artefatos: *Product Backlog*, *Sprint Backlog* e *Burndown*

O ciclo do Scrum tem o seu desenvolvimento e desempenho baseados em iterações realizadas ao longo de todo o desenvolvimento do projeto. A Figura 1 ilustra o ciclo de desenvolvimento Scrum, com as suas principais atividades e envolvidos.

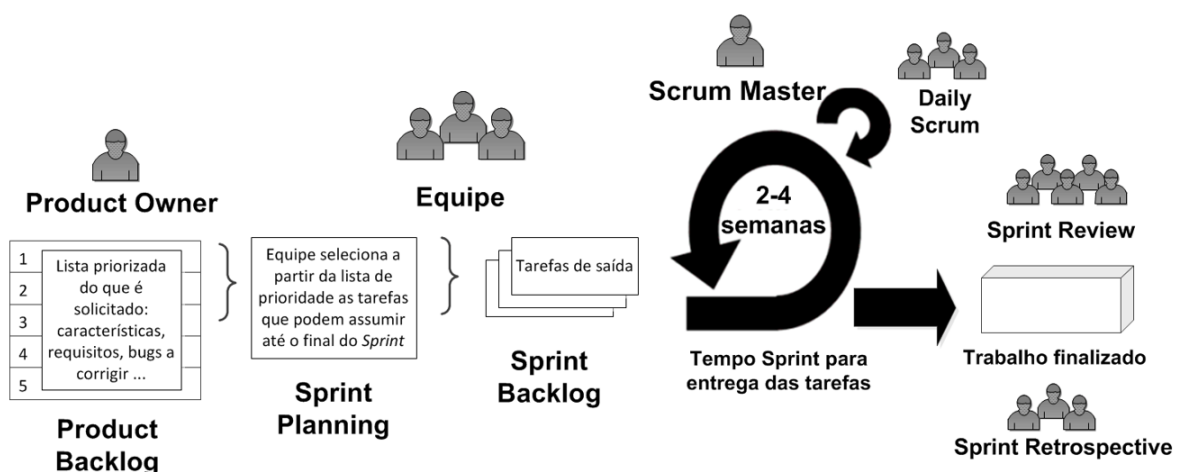


Figura 1. Ciclo desenvolvimento do Scrum.

Quadro 1. Responsabilidades dos envolvidos no projeto, na visão do Scrum [13].

Responsável	Responsabilidade
Product Owner	Responsável por elaborar todas as entradas em que o produto deve ser construído, a partir do cliente ou do usuário final do produto, bem como de membros da equipe e das partes interessadas, e traduzi-los em uma visão do produto. Em alguns casos, o Product Owner e o cliente são uma mesma pessoa.
Scrum Master	Encarregado de fazer o que for necessário para ajudar a equipe ser bem sucedida. O Scrum Master não é o gerente da equipe, e sim ele ou ela serve à equipe, ajudando remover as barreiras para o sucesso da equipe, facilitando reuniões, apoiando e estimulando a prática do Scrum. Algumas equipes terão alguém dedicado totalmente ao papel de Scrum Master, enquanto outros terão um membro da equipe para desempenhar esse papel. O Scrum Master e o Product Owner, provavelmente, não devem ser o mesmo indivíduo; às vezes, o Scrum Master pode ser chamado para analisar uma possível retirada ou inserção de tarefas pelo Product Owner (por exemplo, se eles tentam introduzir novas exigências no meio de um Sprint). E ao contrário de um gerente de projeto, o Scrum Master não diz às pessoas o que fazer ou atribui tarefas - que facilitam o processo, para permitir que a equipe organize e se gerencie.
Scrum Team	Constrói o produto que o cliente vai consumir. A equipe em Scrum é normalmente de cinco a dez pessoas, embora equipes tão grande quanto 15 e tão pequeno como 3 comumente relatam benefícios. A equipe deve incluir todos os conhecimentos necessários para entregar o projeto concluído - assim, por exemplo, a equipe para um projeto de <i>software</i> pode incluir programadores, designers de interface, testadores e pesquisadores. Eles constroem o produto, mas também proporcionam contribuições e ideias para o Product Owner sobre como tornar o produto tão bom quanto ele pode ser. Os membros da equipe também podem mudar de um Sprint para o próximo, mas que também reduz a produtividade da equipe.

Tal desenvolvimento baseado em iterações é construído em períodos chamados de Sprints que podem ser de 2 a 4 semanas e as tarefas realizadas são listadas no Product Backlog. O Product Backlog é mantido pelo Product Owner e pode ser alterado por ele sempre que necessário e também contém uma lista de itens prioritizados onde inclui tudo o que deve ser desenvolvido como forma de agregar valor ao produto final. Dentre as principais prioridades podem ser mencionadas as implementações de requisitos funcionais ou não funcionais, testes de sistemas, implantações e integrações, correções de bugs entre outros.

Antes do início de cada Sprint é realizado um Sprint Planning com a presença do Product Owner e da equipe. A equipe seleciona os itens de trabalhos a serem desenvolvidos no Sprint, considerando como critério os itens de maior prioridade. Em alguns casos o Product Owner pode sugerir itens que apresentaram maior valor ao produto final. A quantidade de itens a serem negociados com o Product Owner é estimada a partir da capacidade total da equipe dentro do Sprint. A equipe nunca deve assumir um compromisso que não pode entregar ao final desse período. Para manter o foco e objetivo nesse momento, o Scrum Master pode ser solicitado para interceder. Os itens definidos no Sprint Planning formam o Sprint Backlog, que contém os itens que serão desenvolvidos, e que não deverão ser alterados até o final do Sprint.

Durante o desenvolvimento do Sprint são realizadas reuniões diárias de Daily Scrum. Essas reuniões, geralmente, são organizadas pelo Scrum Master em horários fixos e não podem exceder o tempo de 15 minutos. Esses encontros têm como objetivo manter a equipe atualizada quanto ao progresso de trabalho e que podem ser medidos a partir da utilização de ferramentas específicas como o Kanban e o Burndown [14] [15]. Esta atualização da equipe é feita a partir de questionários realizados oralmente sobre o andamento do projeto, como:

- O que já foi feito desde a última Daily Scrum?
- O que será desenvolvido até a próxima reunião?

- Há alguma situação problema que deve ser gerenciada?

O Sprint é finalizado no prazo estipulado e em hipótese alguma pode ser finalizado antecipadamente ou postergado. Depois de finalizado, é realizado o Sprint Review, ou revisão do Sprint, onde a equipe entrega o produto testado e revisado a partir de uma demonstração prática. Neste momento o Product Owner realiza uma inspeção no produto e verifica se os itens foram desenvolvidos de acordo com o especificado. É também realizada uma retrospectiva do Sprint, onde são analisados os pontos positivos e situações a serem corrigidas para o Sprint futuro, melhorando assim a experiência de todos. O ciclo do Scrum é repetido até que todos os itens do Product Backlog sejam atendidos ou finalizados e o produto final tenha sido aceito e aprovado pelo cliente.

2.3. Trabalhos Relacionados

A formação profissional tecnológica parte do princípio da formação técnico-tecnológica de profissionais inerente aos conceitos teórico/prático nas diferentes áreas e tecnologias existentes. Neste contexto, as Tecnologias de Informação relacionadas à Informática e Computação representam um grande potencial na formação desses profissionais. Assim, demonstraremos nesta seção algumas iniciativas relacionadas a projetos interdisciplinares em cursos de Informática e a utilização de Métodos Ágeis na Formação Técnica e Profissional de alunos.

Projetos interdisciplinares permitem uma interação entre professores e alunos como prática de aprendizagem que busca o envolvimento de diferentes disciplinas, tecnologias e desafios de aprendizagem. O estímulo à interdisciplinaridade em ambientes acadêmicos traz como benefícios a imersão direta dos estudantes em práticas reais que desenvolvem capacidades técnicas específicas e pessoais dos alunos. Cunha et. al. (2008) apresentam uma abordagem da realização de trabalho interdisciplinar em projetos acadêmicos no curso de Pós-Graduação em Engenharia Eletrônica e Computação do ITA (Instituto Tecnológico de Aeronáutica). Os alunos foram envolvidos em três disciplinas de desenvolvimento e validação de *software*, sendo que foi utilizado uma versão customizada do RUP (*Rational Unified Process*) como processo de trabalho e desenvolvimento. Resultados posteriores mostraram que as aplicações de problemas reais em sala de aula, através de estudos de caso, permitiram que os acadêmicos tivessem uma percepção positiva mais próxima das situações que ocorrem no mercado de trabalho ficando mais adaptados as diferentes situações reais que lá ocorrem [16]. O RUP é uma metodologia criada pela empresa Rational Software Corporation que viabiliza que grandes e médios projetos de software possam ser desenvolvidos e bem sucedidos. Utiliza uma abordagem da orientação a objetos em sua concepção e é projetado e documentado utilizando a notação UML (do inglês *Unified Modeling Language*).

A imersão de ferramentas de desenvolvimento ágil em ambientes acadêmicos vem sendo empregada como forma de estimular a difusão ampla no aprendizado em salas de aula. Em Silva, Barbosa e Carvalho (2016) o Scrum é utilizado na avaliação de Ensino a Distância, onde alunos foram envolvidos na elaboração e gerência de histórias de usuários. Neste trabalho não foram abordados aspectos de desenvolvimento de *software*, contudo a imersão a métodos ágeis permitiu um aprofundamento maior nas disciplinas que abordam esse tema [17]. Um estudo da adoção de métodos ágeis em Instituição Pública é proposto por Melo e Ferreira (2010), contudo esse trabalho transcorreu fora de ambiente acadêmico e não indicou a utilização do Scrum [18]. Metodologias Ágeis, como o XP, também são utilizadas por Kasperavicius et. al. (2008) no ensino de desenvolvimento de jogos digitais no incentivo aos alunos na utilização de processos práticos e vantajosos, sendo seus resultados indicados através do desempenho e dedicação dos alunos na utilização dessas metodologias [19].

Em Rocha, Sabino e Acipreste (2015) o Scrum é utilizado como prática pedagógica no aprendizado da disciplina de Engenharia de *software* em um Curso Técnico em Informática de uma Escola Profissionalizante na região nordeste do Brasil. Este trabalho teve como objetivo estimular os alunos às práticas ágeis no conceito teórico. Foram utilizados questionários para que os alunos avaliassem o entendimento do Scrum e a reação à uma prática onde os Sprints eram realizados em minutos a partir de atividades simples que pudessem ser realizadas. Os autores não desenvolveram nenhum produto de *software* durante a realização desse trabalho. Seus principais aspectos observados foram o trabalho em equipe entre os alunos, autogerenciamento,

desenvolvimento de visão sistêmica e compartilhamento de ideias e dúvidas [6].

O eduScrum é uma ferramenta proposta por Borges, Schmitt e Nakle (2014) a partir do Scrum e teve como objetivo, o de promover projetos de Aprendizagem Colaborativa em ambientes educacionais que auxiliasse na coordenação de atividades de alunos [20]. O Scrum é utilizado por Andrade et. al. (2016) no desenvolvimento de *software* distribuído, no processo de ensino aprendizagem da disciplina de Engenharia de *software* [21]. Assim como por Meireles e Bonifácio (2015) que utilizaram também o Scrum como prática de aprendizagem no desenvolvimento de aplicativos móveis em engenharia de *software* [22].

Observa-se que nos trabalhos citados anteriormente ocorre a imersão de Métodos Ágeis de desenvolvimento no meio acadêmico. Essa imersão e abordagem é acompanhada de vantagens e benefícios no ensino aprendizagem dos acadêmicos de diferentes cursos de Sistemas de Informação. Contudo, a baixa abordagem na utilização de projetos reais não contempla a prática da vivência profissional em projetos de sistemas, inclusive quando o *software* não é construído e o cliente não existe. Outra característica é a falta da utilização de ferramentas gráficas de controle e gerenciamento do progresso de desenvolvimento.

3. Metodologia

Com o objetivo de promover práticas educacionais de aprendizagem e estímulo aos alunos de Cursos Profissionalizantes na Área de Tecnologia de Informação ao desenvolvimento de sistemas, foi proposto um plano interdisciplinar de desenvolvimento de projeto de um *software* a partir de um cliente real. Esse cliente era de um Setor de Serviços Tecnológicos e de Inovação da instituição onde foi realizado esse trabalho e que utilizava uma planilha eletrônica para realizar um questionário com o objetivo de obter o perfil de empreendedores de projetos de uma *Start Up*. A demanda do cliente consistiu na necessidade de migração dessa planilha para um site em uma plataforma web. *Start Up* está relacionado ao ato de começar algo, sendo nos últimos anos relacionada a empresas e companhias que estão no começo de suas atividades com o objetivo de explorar atividades e segmentos inovadores do mercado.

No início do primeiro semestre de 2016, na disciplina de Fundamentos de Engenharia de *software* do curso Técnico em Informática, os alunos foram apresentados a conceitos sobre Metodologias de Desenvolvimento Ágil. Dentre as diferentes metodologias existentes, foi explicado detalhadamente o Scrum. Em seguida, foi realizado uma reunião com a turma e o cliente que apresentou os requisitos funcionais e não funcionais do *software* proposto, fornecendo o modelo em planilha que foi utilizado como referência. Neste momento, os alunos tiveram a oportunidade de extrair o máximo de informações importantes e inerentes ao sistema solicitado.

Apresentado a demanda do *software* a ser desenvolvido, os alunos foram divididos em três grupos de trabalho (A, B e C), onde cada um seria responsável por desenvolver uma versão diferente do sistema web do cliente. Em cada grupo, os alunos foram orientados a se organizarem quanto as suas responsabilidades, atribuições e resultados esperados de cada um. Como o Scrum trabalha em ciclos de desenvolvimento chamados de Sprints, definimos que cada ciclo teriam em média duas semanas para realização.

A partir desse escopo inicial, foram definidos os seguintes papéis no projeto já os relacionando com as respectivas funções no Scrum, como mostrado no Quadro 2.

O professor líder, responsável pela disciplina de Fundamentos de Engenharia de *software*, teve como responsabilidade realizar a demonstração do Scrum para os alunos. Já os demais professores, responsáveis pelas disciplinas de Técnicas de Programação Front End e Back End, Banco de Dados e Análise e Projeto de Sistemas, orientaram os alunos quanto ao desenvolvimento do *software*. Um fluxograma simplificado dessa prática educacional é apresentado na Figura 2 e descreve sucintamente as etapas de sua realização.

Quadro 2. Responsabilidades dos envolvidos no projeto na visão do Scrum [13].

Envolvido	Responsabilidade	Relação Scrum
Professor Líder	Atuar como mediador da Metodologia Ágil, auxiliando no esclarecimento de dúvidas e orientações para mantê-los dentro do Scrum. Criar, atualizar e manter o Product Backlog a partir das necessidades do cliente e quando necessário ser capaz de identificar novas necessidades para a equipe de trabalho.	Scrum Master e Product Owner
Demais professores	Intermediar o conhecimento técnico necessário nas Tecnologias de Informação para construção do projeto, incluindo: Linguagem e Lógica de Programação, Análise e Modelagem de Sistemas, Testes e verificação de <i>software</i> .	Não se aplica
Alunos	Formam a equipe de desenvolvimento do <i>software</i> e a partir das competências técnicas desenvolvidas em cada disciplina deverão: extrair requisitos, modelar, programar, testar e validar o <i>software</i> .	Scrum Team
Cliente	Solicitar, descrever, validar a demanda de <i>software</i> solicitado e participar das reuniões de melhoria do sistema.	Não se aplica

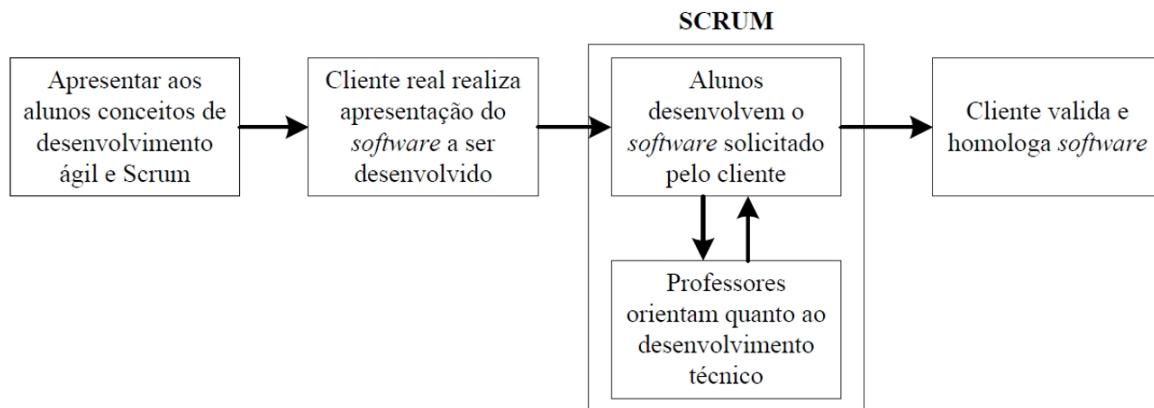


Figura 2. Ciclo de realização da prática educacional com os acadêmicos.

Para o gerenciamento do Product Backlog, controle dos Sprints e acompanhamento das atividades foi utilizado uma versão livre de um framework web, chamado Visual Studio Online (VSO), que permitiu realizar essa atividade. O VSO é uma ferramenta de colaboração para trabalho em equipes de desenvolvimento de *software* e que oferecem um conjunto de instrumentos que funcionam com qualquer IDE (Interface Development Environment) ou editor, para que as equipes possam trabalhar de forma efetiva em projetos de todos os formatos e tamanhos [23]. Esta ferramenta ainda permitiu realizar o compartilhamento coletivo de informações de maneira informatizada como códigos fonte, controle de versões, a atualização do quadro Kanban e o gráfico Burndown, integração contínua em diferentes linguagens e criação de Product Backlog e Sprint.

Utilizando o VSO foi criado o Product Backlog das tarefas que foram realizadas. As tarefas foram identificadas a partir das necessidades do cliente do *software* e descritas por grupos de trabalho. Em cada atividade foi atribuída uma prioridade de realização e estimada a quantidade em horas do tempo necessário para executá-la. Os grupos de trabalho visavam classificar as atividades quanto a sua aplicação no desenvolvimento do projeto. Os grupos foram identificados como descritos no Quadro 3.

Quadro 3. Responsabilidades dos envolvidos no projeto na visão do Scrum [13].

Grupo	Descrição atividades do grupo
Engenharia de Requisitos	Extrair e identificar os requisitos funcionais e não funcionais
Modelagem de Sistemas	Desenvolver diagramas em UML - Caso de Uso, Classes, Sequência, Atividades e Estado
Implementação de Banco de Dados	Desenvolver o respectivo banco de dados relacional
Codificação/Implementação - Back End	Implementar estrutura lógica comportamental interna
Codificação/Implementação - Front End	Implementar telas de interface seguindo critérios de qualidade de usabilidade
Documentação	Desenvolver documentação de especificação - Requisitos, Visão, Arquitetura e Plano de Teste
Teste de <i>software</i>	Implementar casos de teste e executar testes de unidade e usabilidade
Implantação e Integração	Acoplar as partes do projeto e publicar

Antes do início de cada Sprint, o professor líder se reuniu com cada equipe em um Sprint Planning e definiu o Sprint Backlog das atividades a serem desenvolvidas. A definição das tarefas que foram executadas em cada Sprint foi obtida a partir da capacidade da equipe, sendo obtida da Equação (1). Onde C é a capacidade total da equipe em horas para o Sprint planejado: nD é o número de dias totais do Sprint e qH é a quantidade de horas diária disponibilizada por todos os envolvidos no projeto.

$$C = nD \cdot qH \quad (1)$$

Para definir as tarefas do Sprint, a somatória delas não pôde ser superior a capacidade (C) em horas da equipe de desenvolvimento. Esse acordo entre as partes em todos os Sprints foi essencial para que os alunos não assumissem mais tarefas daquilo que eles poderiam realizar evitando assim trabalhos não realizados por falta de tempo ou baixa qualidade de implementação decorrente do pouco tempo disponível. Após iniciado e durante o período de execução dos Sprints os alunos realizaram as atividades ao qual tinham definido anteriormente. Para cada dia de trabalho realizado eles atualizavam o estado de cada atividade no quadro do Kanban entre as opções de To Do, In Progress e Done. Esta atualização dos estados eram realizadas durante as reuniões de Daily Scrum entre os alunos de cada equipe.

Ao final de cada Sprint foi realizado o Scrum Review onde foram apresentados os resultados obtidos no período do Sprint e apontado os pontos de melhoria para o próximo. Nessas entregas de partes do *software* o cliente real e o Product Owner participavam de forma a identificar novas necessidades de modificação de requisitos, apontar pontos de correção ou melhoria na usabilidade do *software* e inserção/exclusão de novas funcionalidades. Em seguida, o processo foi repetido até que todo o sistema tivesse sido entregue com todas as funcionalidades solicitadas.

4. Resultados e Discussões

No Visual Studio Online (VSO) inicialmente foram criados os grupos com todos os envolvidos no projeto. Na versão livre do VSO apenas foi permitido o cadastramento de até cinco usuários com perfil de utilização para os recursos da ferramenta. As tarefas do Product Backlog foram classificadas em sete grupos de atuação e o seu total de horas estimadas para realização estão descritas no Quadro 4. Lembrando que para cada grupo existem inúmeras tarefas listadas de forma mais detalhada, que aqui não foram descritas.

Quadro 4. Product Backlog dos grupos de atividades realizadas.

Grupo	Quantidade de tarefas	Quantidade em horas das tarefas
Engenharia de Requisitos	2	8
Modelagem de Sistemas	4	24
Implementação de Banco de Dados	3	14
Codificação/Implementação - Back End	10	70
Codificação/Implementação - Front End	5	35
Documentação	3	9
Teste de <i>software</i>	6	20
Implantação e Integração	2	20
Total	35	200

A partir do Product Backlog foram realizados um total de seis Sprints até a entrega final do projeto. É importante ressaltar que antes do início de cada Sprint foram realizados Sprints Planning do planejamento quanto às atividades que foram realizadas em cada etapa. Como nesta fase de planejamento há a aceitação das atividades quanto à capacidade da equipe, os próprios alunos estimaram a sua capacidade em horas, sendo que a quantidade de horas necessárias para desenvolver as atividades não poderia exceder essa capacidade. A relação de horas estimadas para cada equipe é apresentada no Quadro 5. Essa capacidade é obtida pela quantidade de dias do Sprint multiplicado pela disponibilidade diária de cada integrante da equipe, e que ao final foi o suficiente para o desenvolvimento do projeto. As capacidades em horas se mostraram diferentes entre as equipes devido ao fato de que a quantidade de horas disponível para cada aluno não necessariamente deveriam ser iguais.

Quadro 5. Capacidade em horas por Sprint de cada equipe de trabalho.

Sprint	Equipe A (2 membros)	Equipe B (3 membros)	Equipe C (2 membros)
Sprint 1	33	45	36
Sprint 2	33	45	36
Sprint 3	33	45	45
Sprint 4	33	45	45
Sprint 5	80	77	99
Sprint 6	25	10	10
Total	237	267	271

Ao decorrer de cada Sprint os alunos receberam as aulas da grade curricular regular, sem ônus à ementa prevista, e utilizaram-se do projeto interdisciplinar para realizar os trabalhos referentes à avaliação de cada disciplina. O acompanhamento contínuo do progresso de cada Sprint foi feito a partir das reuniões de Daily Scrum realizadas diariamente, onde os alunos tiveram a oportunidade de verificar os pontos positivos e críticos, se houvessem, até o momento. Para o registro foi usado o Kanban, disponível no Visual Studio Online, de interface simples e de fácil manuseio. Na Figura 3 é apresentado o Kanban final para o Sprint 1 da equipe A. Como esperado, neste Sprint a equipe conseguiu completar todas as tarefas, ficando todas no estado de done.

Além do Kanban, que permitiu um acompanhamento real do estado atual de cada tarefa (to do - a fazer; in progress - sendo feito; done - feito), o Burndown gerou automaticamente o progresso da equipe. Esse progresso permitiu verificar se a equipe estava adiantada, atrasada ou dentro do prazo estimado. Os gráficos de Burndown são apresentados no Gráfico 1 e correspondem aos Sprints executados, também, pela equipe A.

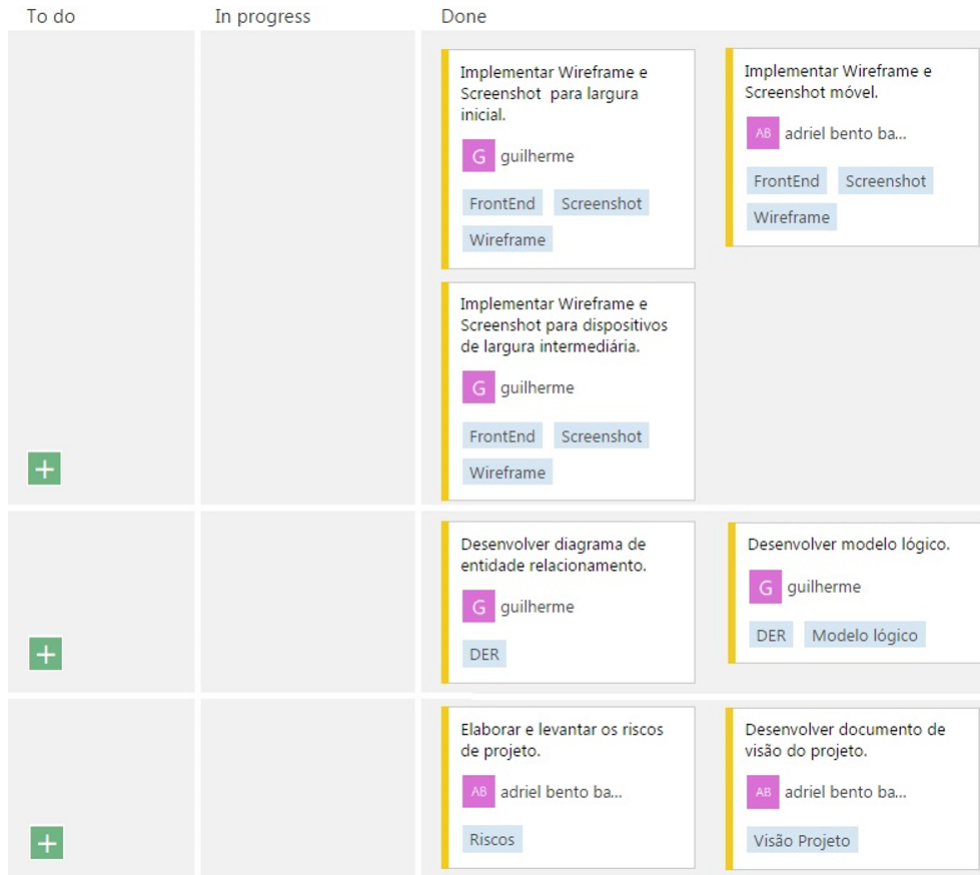
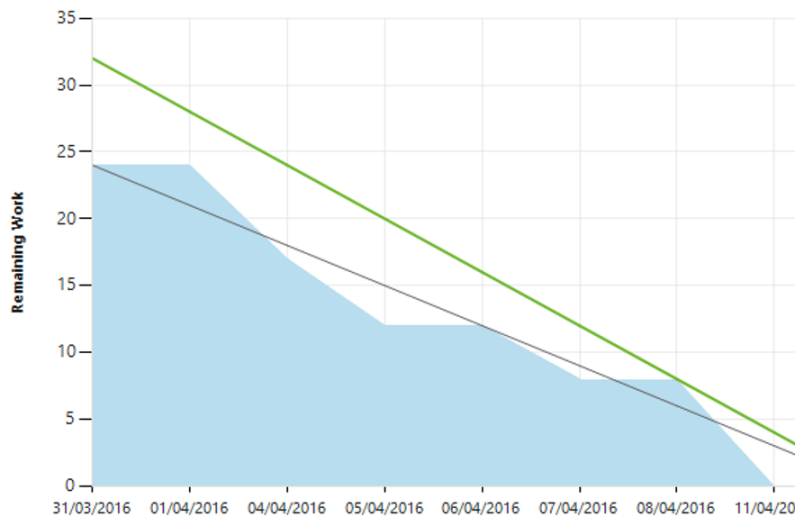
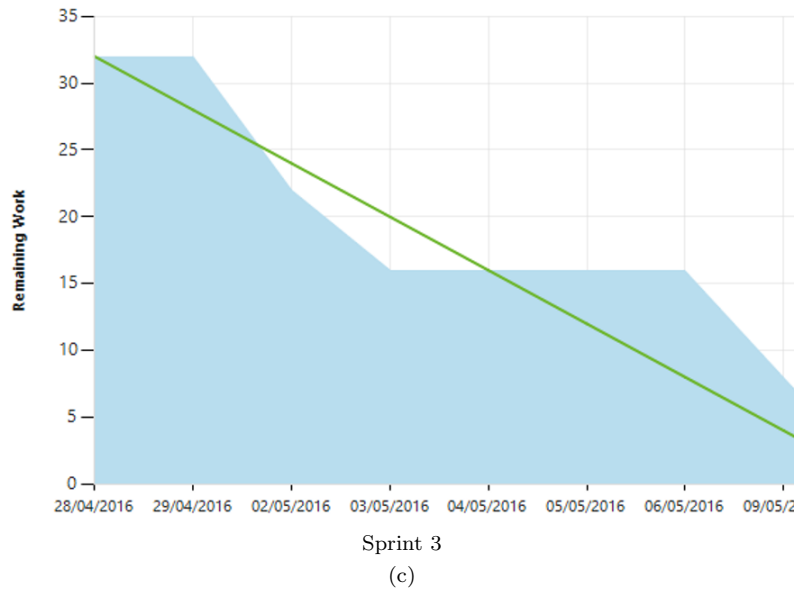
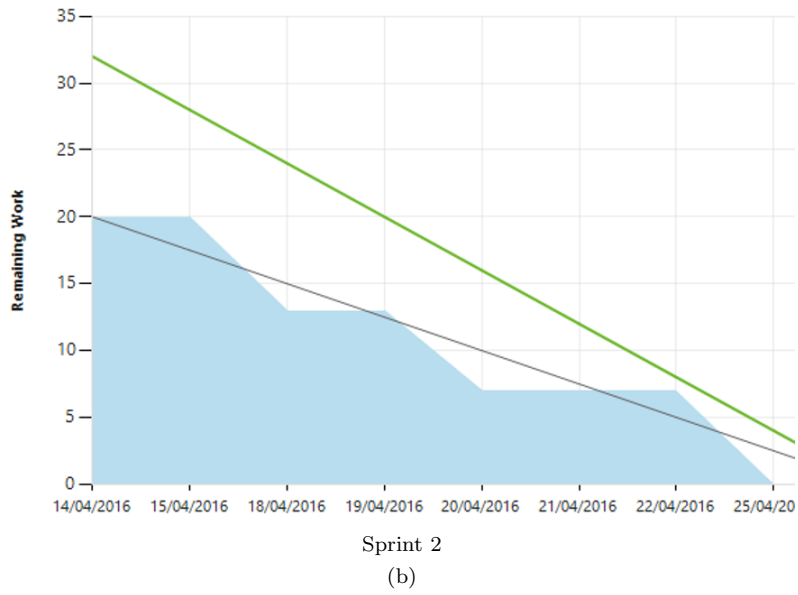


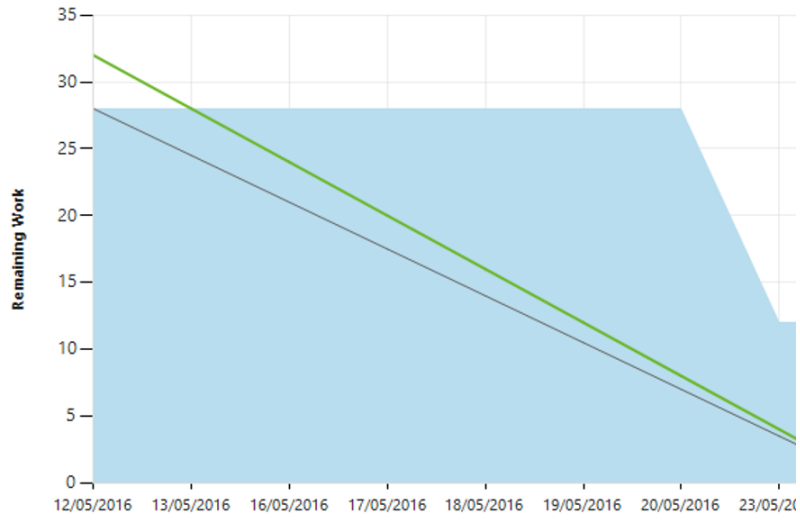
Figura 3. Visão geral do Kanban do Sprint 1 da equipe A.



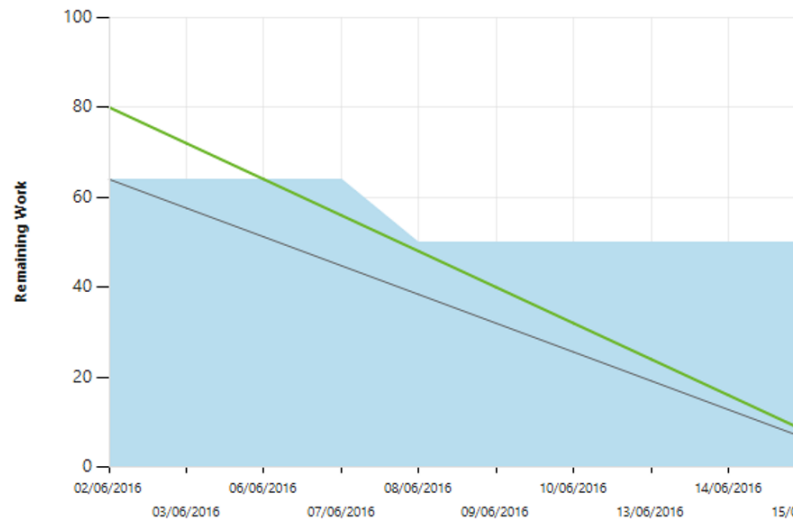
Sprint 1

(a)





Sprint 4
(d)



Sprint 5
(e)

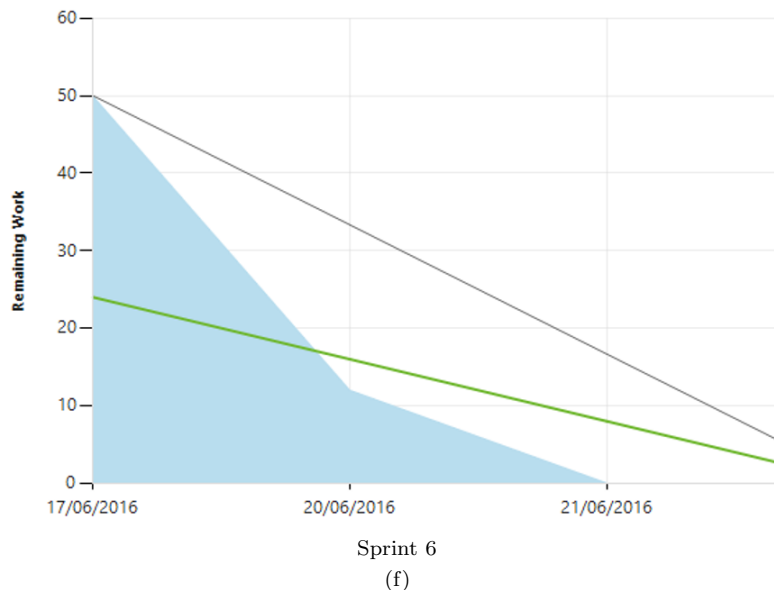


Gráfico 1. Burndown dos Sprints realizados pela equipe A.

Como observado nessas figuras, os Sprints 1, 2 e 3 foram realizados dentro do prazo estimado. No caso dos dois primeiros Sprints, foram terminados mesmo antes do tempo máximo estipulado inicialmente. As linhas na cor cinza correspondem à tendência ideal esperada ao longo do tempo no Sprint, como observado nas Figuras 4c, 4d e 4e houve períodos em que os alunos estiveram atrasados em relação ao tempo esperado. Os atrasos decorreram de alguns problemas ao longo do desenvolvimento que foram tratados e corrigidos. Contudo, ainda observa-se que nas Figuras 4d e 4e algumas atividades não foram finalizadas, decorrente do cancelamento de algumas tarefas que no Sprint seguinte foram realizadas. O último Sprint foi mais curto com o objetivo de finalizar as tarefas pendentes e também para a realização de ajustes necessários no sistema a ser entregue. Tanto o Kanban ou Burndown das equipes B e C não foram apresentados pois não apresentaram grandes variações em relação a equipe A.

Foi realizado a entrega e o fechamento de cada Sprint em conjunto e presença da Scrum Team, Product Owner e o cliente. Nesta fase, as versões foram sendo apresentadas e o feedback utilizado para melhorias ou adaptações nos requisitos de *software*.

5. Conclusões

A utilização de Processos Ágeis de desenvolvimento vem se destacando desde os anos 90 como uma abordagem simplificada, rápida e maximizada no desenvolvimento de *software*. Dentre uma das técnicas, o Scrum, destaca-se como um método de inúmeras vantagens e de resultados positivos. No meio acadêmico, a aprendizagem e conhecimento desses métodos não se faz apenas como conteúdo previsto nos planos de cursos de diferentes níveis como técnicos, graduação e pós-graduação, mas também como forma de colocar o aluno em contato com métodos utilizados e aplicados no mercado de trabalho profissional tecnológico.

Observa-se também que diferentes abordagens de trabalho com alunos nas Instituições de Ensino, permitem um melhor entendimento das diferentes situações que o aluno poderá encontrar durante a sua carreira no mercado de trabalho. Os trabalhos, projetos e práticas acadêmicas realizadas atualmente permitem, por parte dos alunos, uma análise sistemática e direta do impacto da utilização de processos ágeis no desenvolvimento de *software*.

Neste artigo apresentamos uma experiência de sucesso da utilização de Métodos Ágeis, o Scrum, no desenvolvimento de um projeto e cliente reais. Esta experiência obteve ótimos resultados quando empregado em uma prática interdisciplinar, onde todos os professores de quatro disciplinas e alunos se envolveram. Foi utilizada para auxílio ao gerenciamento e controle das etapas do desenvolvimento uma ferramenta web (Visual Studio Online), que permitiu a criação e atualização das atividades, análise do desempenho contínuo através do Kanban e Burndown e registro e acompanhamento das informações por todos os envolvidos.

Assim, a imersão desses alunos em situações reais com ferramentas e metodologias utilizadas no mercado, permite uma aprendizagem mais efetiva em relação às práticas voltadas à Formação Técnica Profissional Tecnológica dos alunos em Cursos de Informática ou Computação. Ainda cabe de investigação, e sugerido como trabalhos futuros, a utilização de outros processos ágeis, como o XP (programação extrema) em projetos de longa duração e com grupos maiores de trabalhos.

Referências

- [1] C. J. Ferretti, “Formação profissional e reforma do ensino técnico no Brasil: anos 90”. *Educação Sociedade*, SciELO Brasil, vol. 18, no. 59, pp. 225–269, 1997.
- [2] A. Z. Kuenzer, “A educação profissional nos anos 2000: a dimensão subordinada das políticas de inclusão”. *Educação e Sociedade*, SciELO Brasil, vol. 27, no. 96, pp. 877–910, 2006.
- [3] J. Moll, “Educação Profissional e Tecnológica no Brasil Contemporâneo: desafios, tensões e possibilidades”. *Artmed Editora*, 2009.
- [4] S. D. Nacional, “Metodologia SENAI de Educação Profissional”. Brasília, DF, Brasil: *Departamento Nacional*, 2013.
- [5] M. E. B. B. Prado, “Pedagogia de projetos: fundamentos e implicações”. BRASIL. Ministério da Educação. *Pedagogia de Projetos e Integração de Mídias*. MEC: Brasília, pp. 12–17, 2003.
- [6] F. G. Rocha, R. F. Sabino e R. H. L. Acipreste, “A metodologia Scrum como mobilizadora da prática pedagógica: Um olhar sobre a engenharia de software”. in: *Fórum de educação em engenharia de software*, Belo Horizonte, 2015.
- [7] J. S. Thiesen, “A interdisciplinaridade como um movimento articulador no processo ensino aprendizagem”. *Revista brasileira de educação, Associação Nacional de Pós-Graduação e Pesquisa em Educação*, vol. 13, no. 39, pp. 545, 2008.
- [8] D. T. Sato, “Uso eficaz de métricas em métodos ágeis de desenvolvimento de software”. Dissertação de Mestrado em Ciências - Instituto de Matemática e Estatística, Universidade de São Paulo, 29 Junho 2007, São Paulo, SP.
- [9] M. S. Soares, “Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software”. *INFOCOMP Journal of Computer Science*, vol. 3, no. 2, pp. 8–13, 2004.
- [10] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. M. Ilor, K. Schwaber, J. Sutherland e D. Thomas, “Manifesto para o Desenvolvimento Ágil de Software”. 2016. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em: 7 julho 2016.
- [11] M. S. Soares, “Metodologias ágeis extreme programming e scrum para o desenvolvimento de software”. *Revista Eletrônica de Sistemas de Informação*, vol. 3, no. 1, 2004.
- [12] W. V. Lima, “Scrum no Brasil”. *Dissertação de Especialização em Engenharia de Projeto de Software*, Universidade do Sul de Santa Catarina, 27 de Junho de 2011, Florianópolis, SC.

- [13] J. Sutherland e K. Schwaber, “The scrum papers: Nuts, bolts, and origins of an agile process”. *ScrumInc*, 2007.
- [14] H. Kniberg e M. Skarin, “Kanban and Scrum-making the most of both”. *InfoQ.com C4Media Inc.*, 2010.
- [15] D. V. S. Silva, F. A. O. Santos e P. S. Neto, “Os benefícios do uso de Kanban na gerência de projetos de manutenção de software”, in *VIII Simpósio brasileiro de sistemas de informação - Trilhas Técnicas*, São Paulo, 2012.
- [16] A. M. Cunha, G. B. Silva, J. A. Monte-Mor, M. A. P. Domiciano e R. G. Vieira, “Estudo de caso abrangendo o ensino interdisciplinar de engenharia de software”, in *Fórum de educação em engenharia de software*, Campinas-SP, 2008.
- [17] V. B. Silva, M. W. Barbosa e L. A. C. Carvalho, “Experiências do ensino a distância do gerenciamento ágil de projetos com scrum e apoio de uma ferramenta para gerência de histórias de usuário”. *Revista Brasileira de Computação Aplicada*, vol. 8, no. 1, pp. 2–13, 2016.
- [18] C. O. Melo e G. R. Ferreira, “Adoção de métodos ágeis em uma instituição pública de grande porte. Um estudo de caso”, in *Workshop brasileiro de métodos ágeis*, Porto Alegre, 2010.
- [19] L. C. C. Kasperavicius, L. N. M. Bezerra, L. Silva e I. F. Silveira, “Ensino de desenvolvimento de jogos digitais baseado em metodologias ágeis: o projeto primeira habilitação”, in *XXVIII Congresso da SBC*, Belém do Pará, 2008, pp. 89–98.
- [20] K. S. Borges, M. A. R. Schmitt e S. M. Nakle, “Eduscrum projetos de aprendizagem colaborativa baseados em scrum”. *RENOTE - Revista Novas Tecnologias na Educação*, vol. 12, no. 1, 2014.
- [21] B. A. L. Andrade, M. S. Brito, A. S. Sampaio, I. R. Costa, D. L. Santos e C. L. Neto, “Aplicando e adaptando a metodologia ágil scrum no processo de ensino e aprendizagem de engenharia de software baseado no desenvolvimento com equipes distribuídas”, in *XVI ERBASE*, Maceió, 2016.
- [22] M. C. Meireles e B. Bonifácio, “Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de software: Um relato de experiência”, in: *Anais do simpósio brasileiro de informática na educação*, Maceió, 2015, pp. 180–189.
- [23] Microsoft. Visual Studio Online. 2016. Disponível em: <<https://www.visualstudio.com>>. Acesso em: 1 fevereiro 2016.