

# Teaching Scrum Practices with the Support of Brazilian Startups: an Initiative in Software Engineering Course

Leo Natan Paschoal  and Simone R. S. Souza 

Institute of Mathematics and Computer Sciences – University of São Paulo  
paschoalln@usp.br, srocio@icmc.usp.br

**Resumo** For a long period, software engineering education had been about the transfer of knowledge related to the software development life cycle through the traditional approaches to teaching and learning. In recent years, with the emergence of some pedagogical models, the software engineering community has begun to invest efforts in promoting initiatives that emphasize not only the transfer of knowledge in the classroom but conducting activities and projects that promote effective learning. Therefore, different approaches were incorporated into the teaching of this course. Despite many examples of reports, few initiatives promote practical experiences together with the “real world”. Promote experiences of the “real world” is a challenge highlighted by recent work in this area. We believe that for the context of software engineering, the application of practical approaches must be accompanied by industry partners, who promote a real world view for the application of concepts seen during the course. This paper reports an experience of Software Engineering Education using real projects and industry partnerships, developed using the Scrum. This experience has provided real scenarios to the students, aiming for the learning of agile practices related to the development and management of a software project. The paper reports the students’ feedback on the learning of software engineering in practice, showing the student perspective of this learning experience.

**Keywords:** Scrum · Startups · Software Engineering Education.

## 1 Introdução

Ao longo dos anos, a ideologia por trás do desenvolvimento de software passou por um processo de transição, em que a comunidade de desenvolvimento de software passou a valorizar os indivíduos e interações mais que processos e ferramentas, colaboração com cliente mais que negociação de contratos e responder à mudanças mais que seguir um plano [2]. Ao passo que a Engenharia de Software foi reconhecendo as transformações e o surgimento de diferentes técnicas, métodos e ferramentas, ela foi sendo incorporada aos currículos de referência e às diretrizes de desenvolvimento de cursos de graduação e pós-graduação. Assim, tornou-se um componente curricular obrigatório nos cursos de graduação [11].

De modo geral, a disciplina de Engenharia de Software busca oferecer aos alunos a capacidade de compreender o desenvolvimento do software desde a fase inicial até a implantação do produto. Apesar de ser uma disciplina essencial para o aluno obter o entendimento necessário para produção de um software com qualidade, os alunos gastam uma quantidade significativa de tempo estudando processos de desenvolvimento de software, a fim de compreender e decidir se um determinado processo pode funcionar para um determinado projeto ou não [4].

Ainda que o ensino de Engenharia de Software deva enfatizar princípios e reconhecer os conceitos estáveis e duradouros da disciplina, os aspectos do estado de prática (*i.e.*, tecnologia, ferramentas e métodos) não podem ser ignorados [8]. Outra questão que deve ser levada em conta é a dissociação entre teoria e prática [5]. Conforme Fox e Patterson [6], a articulação entre teoria e prática é essencial para que iniciantes sejam capazes de compreender a natureza e as inter-relações entre as diversas atividades do desenvolvimento de software. Estudar princípios sobre livros didáticos e até mesmo fazer exercícios não é um “aprendizado real” sem uma experiência prática [8]. No entanto, a prática de Engenharia de Software em um segmento da indústria não é bem suportada pelas disciplinas de cursos de graduação [1].

Estudos recentes descrevem que a maioria dos recém-formados em Engenharia de Software muitas vezes enfrenta dificuldades ao iniciar suas carreiras profissionais, em decorrência do desalinhamento das habilidades aprendidas em sua formação universitária com o que é necessário na indústria [7]. Em virtude disso, nos últimos anos alguns esforços realizados por professores de Engenharia de Software foram sendo reportados, de modo que o aluno tenha uma formação sólida tanto do ponto de vista conceitual quanto prático.

Para apoiar o aprendizado prático de Engenharia de Software, algumas abordagens de ensino têm sido utilizadas, como *capstone projects*, *flipped classroom*, *game based learning*, *problem based learning*, *project based learning*, dentre outras. Ainda que algumas iniciativas tenham sido estimuladas e realizadas, estudos descrevem que é difícil reproduzir em sala de aula, ambientes muito próximos àqueles que os alunos enfrentarão ao adentrarem na indústria de software [8]. Dentre as dificuldades que são reportadas, Ghezzi e Mandrioli [8] mencionam que os projetos de software precisam ter um escopo reduzido, porque os alunos precisam conciliar com atividades de outras disciplinas. Outro aspecto a ser considerado é o conhecimento técnico dos alunos sobre programação, redes de computadores e ferramentas de apoio ao desenvolvimento de software [16]. De fato, construir uma iniciativa educacional que reflita e acompanhe a prática industrial não é trivial, mas precisa ser realizada [3].

Para contornar os desafios imbricados ao ensino de práticas de Engenharia de Software, um grupo de professores da Universidade de São Paulo vem incluindo estratégias de ensino diferentes do tradicional nessa disciplina, que envolvem a participação de clientes reais, projetos do “mundo real” e a execução de práticas do Scrum, dado que o Scrum é provavelmente um dos métodos de desenvolvimento mais prestigiados pela atual indústria de software. Neste artigo é reportada uma iniciativa realizada durante o ensino de Engenharia de Software

no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo – ICMC/USP. Esta iniciativa buscou oferecer experiências práticas aos alunos do curso de bacharelado em ciências de computação, a partir do aprendizado do conteúdo por meio de uma metodologia teórico-prática conduzida através de uma parceria estabelecida com um grupo de *startups* da cidade de São Carlos – SP, Brasil. Essa metodologia envolve o aprendizado teórico do conteúdo e a realização de práticas de desenvolvimento de projetos de software por meio de práticas utilizadas por essas *startups* para produção de sistemas de software. Para o desenvolvimento dos projetos, os alunos receberam treinamento das *startups* sobre práticas do método Scrum, foram supervisionados por representantes dessas *startups* e tiveram que propor soluções de software reais.

Para apresentar o relato de experiência, este artigo foi organizado da seguinte forma: a Seção 2 apresenta uma visão geral sobre o Scrum e suas práticas. A iniciativa é apresentada na Seção 3. Um *feedback* emitido pelos alunos sobre a experiência é descrito na Seção 4. Por fim, as conclusões e algumas perspectivas de estudos futuros são discutidos na Seção 5.

## 2 O Método Scrum

O desenvolvimento de software passou por diversas transações, movendo-se de um período em que não existiam mecanismos de apoio ao estabelecimento sistemático de software, chegando ao ponto em que modelos de processos são produzidos para atender o desenvolvimento de sistemas com características específicas, como os jogos eletrônicos [10]. Apesar de modelos de desenvolvimento de software estarem sendo produzidos até hoje para atender às necessidades de uma indústria do software cada vez mais heterogênea, muitos dos sistemas de software vêm sendo produzidos através de metodologias ágeis, como o Scrum.

De acordo com Schwaber [15] o Scrum é um processo de desenvolvimento que tem foco no gerenciamento do desenvolvimento iterativo. Ele vai ao encontro do manifesto ágil e é adequado para ambientes complexos em que as equipes precisam reagir e se adaptar rapidamente a novas situações no sistema (*e.g.*, mudanças de requisitos e/ou de tecnologia) [9]. Pode ser visto como um modelo ou método de desenvolvimento de software que foi estabelecido para lidar com problemas complexos, ajudando a fornecer produtos de maneira produtiva e criativa [17].

O modelo é constituído basicamente por três fases: **(i) planejamento geral**, em que são estabelecidos os objetivos do projeto, é feita a conceituação e análise para definição de um *backlog*<sup>1</sup>, assim como, a estimativa de seu custo; **(ii) ciclos de *sprint***, em que são desenvolvidos artefatos que incrementam o sistema e geram valores ao cliente; **(iii) fase de encerramento do projeto** [15]. Vale salientar que um ciclo de *sprint* envolve a seleção de um item do *backlog*, o desenvolvimento da funcionalidade pela equipe de Scrum, a revisão e o teste dessa funcionalidade.

<sup>1</sup> Conjunto de funcionalidade que precisam ser implementadas.

Em relação à equipe de Scrum, ela é composta por um número definido de membros, preferencialmente inferior a dez [13]. Apesar do número limite de membros, um projeto pode ser desenvolvido por várias equipes. Os membros de uma equipe podem desempenhar determinados papéis - a saber: *product owners*, *scrum master* e *development team* [17].

- *Product owner*: O *product owner* é responsável por gerenciar o *backlog* do projeto, identificar e priorizar corretamente os itens do *backlog* para atingir as metas definidas, garantir que o *backlog* do projeto esteja claro e transparente para todos os membros da equipe [9].
- *Scrum master*: O *scrum master* é o líder da equipe de Scrum e tem a responsabilidade de garantir que sua equipe tenha compreendido as atividades necessárias que precisam ser realizadas e a competência de assegurar que essas atividades estejam sendo executadas/realizadas [9]. Além disso, o *scrum master* é responsável pela aderência dos membros do time aos valores do Scrum e princípios ágeis.
- *Development team*: A equipe de desenvolvimento (*development team*), por sua vez, é representada pelos profissionais que são responsáveis por implementar e avaliar as funcionalidades implementadas [9]. Esses papéis foram definidos com o propósito de facilitar o gerenciamento das *sprints*<sup>2</sup> [14].

Uma *sprint* tem um tempo de duração que pode variar, apesar de existirem recomendações para que ela dure entre duas até quatro semanas. Durante a realização de uma *sprint*, alguns eventos são previstos: reunião de planejamento da *sprint* (*sprint planning*), reunião diária (*daily Scrum*), reunião de revisão da *sprint* (*sprint review*) e retrospectiva da *sprint* (*sprint retrospective*). Cada um desses eventos será descrito a seguir:

- *Sprint planning*: Essa reunião é realizada para decidir o que será implementado em uma *sprint* e como o trabalho será realizado pela equipe. Durante a reunião, o cliente deve mostrar à equipe o *backlog* do produto com os itens classificados de acordo com sua prioridade, explicando o que deve ser feito em cada item de trabalho. A equipe precisa estimar o esforço necessário para implementar os itens e fazer um acordo com cliente sobre o que será entregue no final da *sprint*, considerando a ordem de prioridade estabelecida pelo proprietário do produto. Essa reunião envolve a decisão de tarefas que serão realizadas pelos membros da equipe, em que cada membro da equipe seleciona a quantidade de trabalho que cada um se compromete a desenvolver e entregar no final da *sprint* e o tempo (em horas) necessário para concluí-la é estimado.
- *Daily Scrum*: Todos os membros da equipe devem realizar uma reunião diária para obter uma visão geral do projeto, descobrir novas dependências, atender às necessidades individuais dos membros da equipe e ajustar o plano de trabalho em tempo real à necessidade do dia a dia. Na reunião diária, os membros da equipe são convidados a responder três perguntas fundamentais: (I) O que foi realizado no dia anterior à reunião? (II) O que será realizado hoje? (III) Existe algum obstáculo para realizar o trabalho?. Idealmente, essas reuniões devem ser realizadas pela manhã, permitindo o estabelecimento de prioridades de trabalho para o dia.

---

<sup>2</sup> É um período delimitado de um mês ou menos, durante o qual uma versão potencialmente utilizável do produto é criada.

- *Sprint review*: Essa reunião é realizada no final de cada *sprint* para demonstrar o trabalho realizado durante o período. Nesse sentido, a equipe deve demonstrar os recursos implementados para o cliente. O cliente deve verificar se os recursos estão de acordo com o que ele espera e, quando necessário, sugerir modificações.
- *Sprint retrospective*: Essa reunião é realizada imediatamente após a revisão da *sprint*. Nesse momento, a equipe de Scrum deve discutir o que está funcionando e o que não está funcionando no projeto e chegar a um acordo sobre as mudanças necessárias para a próximo *sprint*.

Esses eventos são reconhecidos na pesquisa de Qumer e Henderson-Sellers [13] como práticas ágeis do Scrum. As práticas ágeis são sugeridas pelos métodos de desenvolvimento para atingir seus objetivos específicos [12]. Cada modelo de desenvolvimento de software tem suas próprias práticas, apesar de alguns complementarem suas práticas com práticas oriundas de outros métodos. No método XP, por exemplo, a programação por pares é uma prática necessária para atingir o objetivo do método [13]. Em razão dos modelos possuírem suas próprias práticas, quando eles são ensinados para alunos e ou profissionais que estão recebendo treinamento, as práticas precisam ser exteriorizadas.

### 3 Iniciativa

Após o estudo sobre a execução de projetos similares ao reportado neste artigo, em particular por meio dos relatos compartilhados por Fioravanti *et al.* [5] e Souza *et al.* [16], decidiu-se por incluir na disciplina a participação de *startups* e definiu-se um projeto guarda-chuva, a partir do qual cada equipe de alunos iria propor uma solução de software. Dessa maneira, os alunos tiveram a oportunidade de desenvolver seu potencial criativo, definindo quais seriam suas soluções e, com o apoio das *startups*, definindo o escopo de seus projetos. De maneira simplificada, o projeto guarda-chuva envolve a construção de uma plataforma de apoio para estudantes que chegam a São Carlos, com aplicativos de apoio para auxílio a encontrar moradia, para conhecer a cidade, conhecer a universidade, entre outras ideias que partiram dos próprios alunos.

#### 3.1 Abordagem de Ensino

A disciplina foi ministrada para os alunos do curso de Ciências de Computação, no 1º semestre de 2018, para 57 estudantes. Essa disciplina é obrigatória, possui quatro créditos-aula (60 horas-aula no semestre, divididas em duas aulas semanais). O objetivo dela é oferecer uma visão geral dos processos de desenvolvimento de software e as técnicas e/ou práticas que podem ser utilizadas em cada fase do ciclo de vida do software.

A metodologia foi definida de modo que todo o conteúdo teórico da disciplina fosse ministrada antes do início do desenvolvimento dos projetos. Buscou-se fazer com que o aluno tivesse uma visão geral e abrangente da Engenharia de Software e de seus processos, métodos e ferramentas antes da aplicação prática. Assim, a disciplina foi fracionada em duas partes:

- **Aulas tradicionais:** compreende a primeira etapa da disciplina, em que os alunos assistiam as aulas teóricas, realizavam exercícios (durante a aula e fora da sala de aula) e participavam de discussões sobre o conteúdo.
- **Aulas em parceria com a indústria:** compreende a segunda etapa da disciplina, em que os alunos desenvolviam o projeto prático: recebiam treinamento por um grupo de *startups*, planejavam *backlog* do produto com os clientes, interagiram com clientes reais em cada início e final de uma *sprint*, apresentavam funcionalidades desenvolvidas para seus clientes, realizavam reuniões e (re) planejavam entregas.

### 3.2 Projetos do “Mundo Real”

Após o aprendizado teórico na disciplina, os alunos foram apresentados aos projetos que deveriam ser realizados. Nesse sentido, a professora da disciplina optou por estabelecer uma parceria com a Arquivêi<sup>3</sup>, uma *startup* da cidade de São Carlos. Os sócios dessa organização aceitaram colaborar com o ensino prático da disciplina e propuseram o envolvimento de outras *startups* da cidade de São Carlos, em especial aquelas que estão incubadas no ONOVOLAB<sup>4</sup>. Assim, estabeleceu-se a parceria com a Liber<sup>5</sup>.

Para o contexto desta disciplina, uma reunião com um grupo de *startups* interessadas foi realizada, visando apresentar a proposta de promover um ambiente facilitador aos alunos de Engenharia de Software, de modo que eles pudessem conviver com a prática de desenvolvimento de projetos de software. Para tanto, inicialmente foi definido que os alunos iriam se envolver com práticas do Scrum. Na sequência foi definido o problema que seria considerado nos projetos desenvolvidos, optando-se pela geração de aplicativos e sistemas de apoio para os alunos que chegam para estudar na cidade. Esse tema foi escolhido porque a maioria dos estudantes de São Carlos vem de outros locais do país e precisam de material de apoio para conhecer a cidade e as universidades que possuem sede na cidade.

Com base nessa ideia, os alunos tiveram um tempo para definirem seus projetos. Um total de nove propostas foram feitas pelos estudantes (descritas na Tabela 1). Todos os projetos possuem um caráter social e seriam desenvolvidos tendo como base a filosofia de software livre. Desse modo, os projetos não visam retorno financeiro.

Após definidos os projetos, foram estabelecidos quem seriam os clientes de cada um, com base nas *startups* envolvidas. No período, os alunos tiveram a possibilidade de visitar as *startups*, solicitar treinamento de alguma tecnologia necessária para o desenvolvimento dos seus projetos e utilizar a infraestrutura necessária dessas organizações para hospedagem das plataformas. Além disso,

---

<sup>3</sup> Mais informações disponíveis em: <<https://arquivêi.com.br/>>.

<sup>4</sup> Um campus de inovação que busca oferecer um ambiente para o desenvolvimento de projetos inovadores, apoiando pessoas, empresas e o Brasil. Mais informações pode ser obtidas em: <<https://www.onovolab.com/>>.

<sup>5</sup> Mais informações disponíveis em <<https://www.libercapital.com.br/>>.

os clientes colaboraram com a delimitação dos projetos e estabelecimento do escopo, de modo que fossem viáveis de serem desenvolvidos no tempo disponível (2 meses).

**Tabela 1.** Descrição dos projetos

#	Projeto	Descrição do projeto	Tipo de aplicação
1	#deSCubra	Uma plataforma com o objetivo de apresentar aos usuários a cidade de São Carlos em um escopo histórico, cultural e informativo.	Web
2	Entretenibit	Uma plataforma para que usuários conheçam eventos culturais de São Carlos e recebem e-mails sobre esses eventos.	Web
3	BusCarlos	Uma plataforma que permite que usuários de transporte circular de São Carlos consigam fazer busca por linhas do transporte e pontos de paradas.	Web
4	SanConnect	Uma plataforma de divulgação de micro-eventos de modo a integrar pessoas, ajudar a fazer novas amizades, com interesses em comum.	Web
5	Roomates	Uma plataforma que visa facilitar a divisão de contas em um grupo de amigos, que possivelmente dividem uma moradia, permitindo o gerenciamento de contas e gastos.	Web/ Mobile
6	SddsR\$	A plataforma tem por objetivo auxiliar um novo estudante de São Carlos a gerenciar seu dinheiro, tendo conhecimento das bolsas de projetos (de iniciação científica, de extensão, de cultura ou de ensino) disponíveis nas instituições públicas de ensino da cidade.	Web
7	House Trinder	A plataforma tem o objetivo de auxiliar novos alunos na busca de moradia em São Carlos assim como, ajudar a encontrar potenciais colegas de quarto que possuam interesses em comum, de maneira similar ao Tinder.	Mobile
8	Morar, Morei	A plataforma tem a finalidade de facilitar a busca por informações relacionadas à moradia (repúblicas, room mate e apartamentos) e as respectivas regiões da cidade, unificando anúncios de moradia de mobiliarias da cidade, similar ao Buscapé.	Web
9	SancaServ	Um site que reúne anúncios de empregadas domésticas, permitindo que um usuário logado ao site possa filtrar parâmetros de seu interesse, obtendo assim o contato que mais se adéque às suas necessidades.	Web

### 3.3 Desenvolvimento dos Projetos

O desenvolvimento seguiu as práticas que caracterizam o Scrum, definindo-se: *scrum master*, *scrum teams*, *product backlog*, *sprint backlog*, *sprint planning meeting*, *daily Scrum meeting* e *sprint review*. Para que os alunos conseguissem adotar com precisão essas práticas, um treinamento foi proposto pelas *startups*. Nesse treinamento, os alunos tiveram oportunidade de conhecer como as *startups* trabalhavam na prática. Eles também puderam solucionar dúvidas com os profissionais que ministravam o treinamento. O treinamento foi no ICMC e os alunos foram convidados a visitar as *startups* para acompanhar como as equipes trabalhavam nos projetos reais e executavam na prática o Scrum.

Foram necessárias algumas adaptações nas práticas, pois os alunos não tinham dedicação exclusiva ao projeto da disciplina. Assim, decidiu-se que as reuniões diárias, nos dias que não havia aula, seria feito da maneira que fosse mais conveniente para a equipe, podendo ser por meio de aplicativo de mensagens, por exemplo. As reuniões de retrospectiva foram realizadas sempre em sala de

aula. As reuniões de iteração com clientes, em sua maioria, foram feitas fora do ICMC (*i.e.*, nas *startups* ou por videoconferência). O papel do *scrum master* e da equipe de desenvolvimento permaneceu o mesmo que é descrito na literatura (mencionados na Seção 2). Em relação às *sprints*, foi definido que elas teriam duração de uma semana.

Como os alunos já haviam aprendido os conceitos teóricos, o tempo em sala de aula foi dedicado para solução de dúvidas, realização de reuniões, implementação e acompanhamento dos projetos. Salienta-se que para acompanhar o desenvolvimento dos projetos, a professora da disciplina solicitou a entrega de alguns artefatos que eram produzidos pelos alunos nas *sprints*. A Tabela 2 contém detalhes dos artefatos que eram utilizados para acompanhar o andamento dos projetos.

**Tabela 2.** Artefatos a serem entregues pelas equipes de cada projeto. Adaptado de Souza *et al.* [16].

Artefato	Descrição
Planejamento da entrevista	Documento com a descrição do roteiro a ser seguido durante a entrevista com o cliente.
Plano de projeto	Documento contendo descrição do escopo e objetivos do projeto, matriz de riscos, organização da equipe para o desenvolvimento, cronograma, recursos utilizados, resultados esperados e recursos para monitoramento e acompanhamento do projeto.
Plano de teste	Documento contendo uma descrição sobre o que e como vai ser testado, os testes a serem aplicados, prazos e os recursos alocados.
<i>Backlog</i> do produto	Construção da lista contendo todas as funcionalidades desejadas para o produto e descrição das funcionalidades na forma de histórias de usuário.
Detalhamento de cada <i>sprint</i>	Construção da lista de funcionalidades do <i>sprint</i> , em ordem de prioridade.
Relatório de execução dos testes	Resultados da execução dos testes, com histórico de defeitos encontrados e corrigidos.

## 4 Percepção dos Alunos

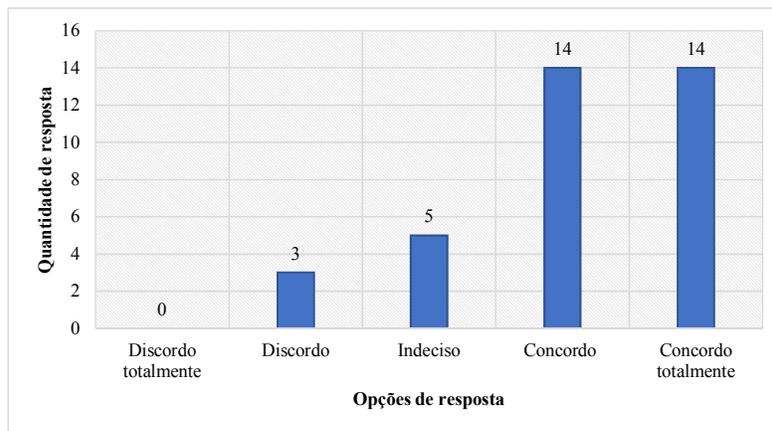
Para ter uma visão geral sobre a percepção dos alunos a respeito da metodologia teórico-prática adotada na disciplina, um questionário foi preparado, contendo sete itens no total, sendo que seis eram obrigatórios. Os itens são apresentadas na Tabela 3. O formulário foi construído na ferramenta Google Forms<sup>6</sup> e foi encaminhado para envio ao final do semestre, não contabilizando como atividade avaliativa da disciplina. Um total de 36 estudantes responderam ao questionário.

Em relação ao primeiro item, observa-se que a maioria dos respondentes avaliou como adequado o Scrum para os projetos adotados na disciplina (Figura 1). Cinco participantes declararam que ficaram indecisos. Por último, três alunos acreditaram que o modelo não foi adequado para os projetos. As respostas desses três alunos também não proveem de participantes de um mesmo grupo.

<sup>6</sup> Mais informações disponíveis em <<https://www.google.com/forms/about/>>.

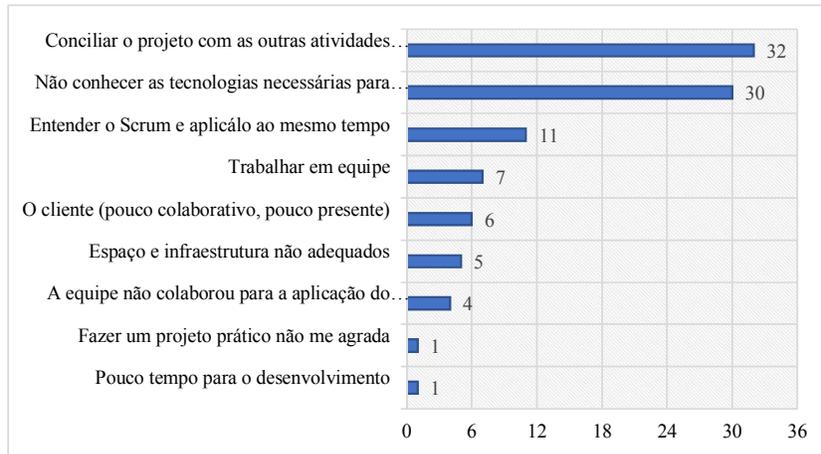
**Tabela 3.** Questionário de coleta de feedback

Itens	Opções de resposta
(1) O Scrum foi adequado ao desenvolvimento do projeto?	Escala Likert
(2) Quais foram as maiores dificuldades? (o aluno pode selecionar diversas opções)	<input type="checkbox"/> Entender o Scrum e aplicá-lo ao mesmo tempo; <input type="checkbox"/> Não conhecer as tecnologias necessárias para o desenvolvimento do projeto; <input type="checkbox"/> O cliente (pouco colaborativo, pouco presente); <input type="checkbox"/> Conciliar o projeto com as outras atividades do semestre; <input type="checkbox"/> A equipe não colaborou para a aplicação do Scrum; <input type="checkbox"/> Trabalhar em equipe; <input type="checkbox"/> Fazer um projeto prático não me agrada; <input type="checkbox"/> Outro (detalhar):
(3) Você sente que aprendeu melhor Engenharia de Software com a prática?	<input type="radio"/> Sim <input type="radio"/> Não <input type="radio"/> Não sei opinar
(4) Quais foram suas expectativas em participar de um projeto envolvendo <i>startups</i> ?	Resposta aberta
(5) Descreva sobre a sua percepção em relação a participação dos sócios das <i>startups</i>	Resposta aberta
(6) Apresente sugestões para melhorar a abordagem usada em sala de aula	Resposta aberta
(7) Se desejar, comente outros aspectos sobre a sua experiência ao desenvolver o projeto e forneça sugestões adicionais	Resposta aberta

**Figura 1.** Percepção sobre adequação do Scrum aos projetos

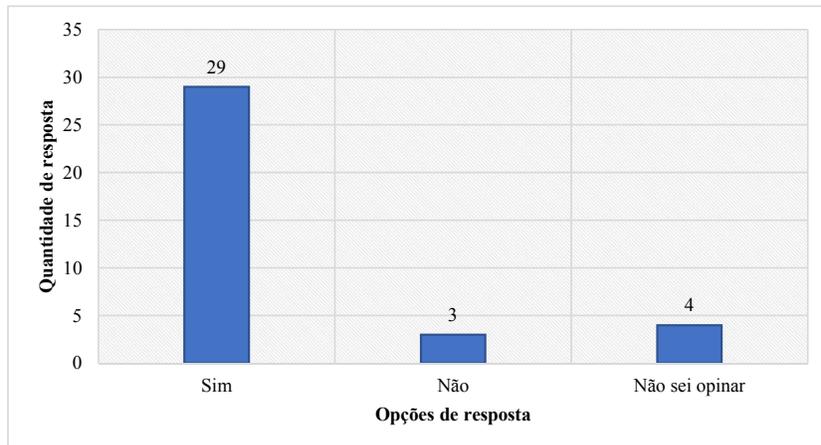
Os alunos puderam selecionar e indicar as maiores dificuldades que eles tiveram ao longo da realização dos projetos. Foi possível observar que a maior dificuldade foi conciliar as atividades do projeto com as demais atividades que eles faziam ao longo do semestre. Outro item que recebeu mais de 50% das respostas dos participantes se refere à falta de conhecimento sobre as tecnologias necessárias para a realização do projeto (Figura 2). Salienta-se que

durante o planejamento da iniciativa já se tinha imaginado que os alunos teriam dificuldades em se organizar com as demais disciplinas em que eles estavam matriculados e com a tecnologia utilizada. Por conta disso, quando a parceria foi estabelecida foi necessário definir projetos que não seriam tão complexo e, para os mitigar problemas associados às tecnologias, as *startups* ficaram disponíveis para oferecer apoio e treinamento.



**Figura 2.** Principais dificuldades dos alunos

O terceiro item buscou compreender se os alunos ao final da disciplina acreditaram que conseguiram aprender melhor o conteúdo com a prática de projetos. A Figura 3 representa as respostas emitidas pelos respondentes. Foi possível observar que a maioria dos estudantes indicou que acredita que conseguiu aprender o conteúdo com precisão.



**Figura 3.** Visão geral sobre a sensação de aprendizado

No quarto item, os alunos relataram suas expectativas em participar do projeto. Uma sumarização das respostas é apresentada na Tabela 4. Conforme é possível observar, a maioria das respostas sobre as expectativas dos alunos estão concentradas na oportunidade criada para eles se aproximarem da realidade do mercado de trabalho e aplicarem conhecimento para solucionar problemas reais.

**Tabela 4.** Expectativas dos estudantes

Feedback dos alunos	Número de ocorrência
Aproximar-se mais da realidade do mercado de trabalho e entender como funciona o trabalho nessas <i>startups</i>	11
A possibilidade de aplicar nosso conhecimento para resolver problemas reais	10
Nada a declarar	6
Tive boas expectativas em aprender com profissionais que já possuem experiência	5
Não criei grandes expectativas com a iniciativa	4

Os alunos também foram questionados sobre suas percepções quanto ao apoio oferecido pelos profissionais das *startups* (item 5). Conforme é possível observar na Tabela 5, os alunos indicaram que gostaram. Alguns alunos acreditaram que foi significativa para realização do projeto até o final, outros indicaram que foi importante para apoiar na atividade de teste. Apesar disso, alguns alunos acreditaram que a participação deles não foi importante. É possível que esses alunos não tenham tido tanta dificuldade ao compreender o que precisavam fazer e/ou até mesmo não tenho gostado de ter que sair em determinados momentos do ICMC para solucionar suas dúvidas nas sedes das *startups*.

**Tabela 5.** Percepção em relação à participação dos profissionais das *startups*

Feedback dos alunos	Número de ocorrência
Eles foram muito prestativos, sempre dedicando boa parte de seu tempo a ajudar nos projetos	14
A participação deles no projeto não foi muito significativa	5
Gostei da iniciativa deles e das aulas que eles foram falar, porém senti uma distância durante o período de desenvolvimento do projeto	5
Eles ajuda muito em relação a dar dicas, ideias e solucionar os defeitos no código	4
Foram de grande auxílio e contribuíram bastante para o produto final	4
Não tenho nada a declarar	4

Os alunos foram convidados a apresentar sugestões de melhorias para a iniciativa realizada. Um resumo das sugestões foi realizado e está disponível na Tabela 6. De modo geral, alguns alunos relataram que o projeto poderia começar no início do semestre e ser executado em paralelo às aulas teóricas. Alguns alunos também indicaram que seria interessante realizar o projeto em parceria com outras disciplina.

**Tabela 6.** Um resumo das sugestões de melhoria

Feedback dos alunos	Número de ocorrência
Não tem sugestões de melhoria.	13
O projeto poderia começar no início da disciplina, uma vez que o tempo é pouco para realizar atividades do projeto e conciliar atividades de outras disciplinas.	10
O projeto poderia ser realizado em conjunto com outras disciplinas como desenvolvimento <i>web</i> .	3
Focar em projetos de escopos menores e mais simples.	3
Adotar um conjunto padrão de ferramentas (de desenvolvimento) e gestão de projetos.	2
A temática do projeto poderia ser outra.	2
Os projetos deveriam ser definidos para desenvolvermos soluções que temos conhecimento para desenvolver. Nestes projetos, a maioria dos alunos teve que aprender outras linguagem de programação.	1
Eu acho que a participação das <i>startups</i> poderia ter sido reduzida. Sinto que eles focaram muito em coisas técnicas.	1
A professora da disciplina poderia ter ensinado as tecnologias necessárias para realização do trabalho e/ou indicado materiais sobre o assunto.	1

Por fim, no último item do questionário, os alunos puderam fazer comentários e emitir depoimentos. Os comentários estão listadas na Tabela 7. Conforme é possível perceber, os alunos que aceitaram responder apresentaram elogios e algumas críticas. Em relação às críticas, observa-se que elas têm relação com a necessidade dos alunos terem tido que aprender a trabalhar com tecnologias que eles ainda não tinham estudado. Apesar das críticas, a maioria dos alunos consideraram proveitosa a experiência.

**Tabela 7.** Comentários emitidos pelos alunos

Feedback dos alunos
(1) As reuniões realizadas em sala de aula foram muito interessante para discutirmos avanços em relação aos <i>sprints</i> e fornecer feedbacks necessários para se discutir possíveis pontos de melhoria de cada integrante da equipe.
(2) A proposta do trabalho me interessou, e também os integrantes do meu grupo. No entanto, a maneira como o projeto foi abordado dificultou a conciliação com outras atividades da disciplina.
(3) Eu queria deixar um elogio para a ideia do projeto final. Acredito que a experiência da prática oferece um aprendizado que nunca teria obtido se não fosse por esse projeto (ainda mais Engenharia de Software, que todo mundo fala que é a disciplina mais “chata”). Eu gostei bastante de trabalhar nesse projeto com meu grupo, e pôr a mão na massa! Acho que aprendi muito não só sobre Engenharia de Software, mas também sobre como é uma rotina de desenvolvimento de um produto. Então espero que os próximos oferecimentos dessa disciplina continuem com esse diferencial também!
(4) O escopo definido como um projeto <i>web</i> dinâmico não foi uma boa escolha visto que a maioria dos alunos ainda não tinha feito a disciplina de <i>web</i> . Logo, além desenvolver o projeto junto com todos os problemas já normais da graduação, tínhamos que correr atrás de aprender programação <i>web</i> e banco de dados. Eu acho que a disciplina de Engenharia de Software acabou ficando em segundo plano quando o grupo teve que aprender tantas outras coisas.
(5) Foi um projeto que o resultado ficou tão legal que é até válido como portfólio. Eu acho que seria interessante se a disciplina fosse ensinada após desenvolvimento <i>web</i> .
(6) Acredito que o projeto foi uma ótima forma de aprender Engenharia de Software na prática, porém senti, especialmente no final de semestre, um enorme peso. Tal fato se deve pelo tamanho do projeto em concomitância com atividades de outras disciplinas e a dificuldade de se aprender diversas tecnologias em curto espaço de tempo, que levou na frustração de não conseguir terminar como queríamos o projeto.

## 5 Conclusões

Este artigo descreve um relato sobre uma experiência realizada na disciplina de Engenharia de Software. A experiência foi conduzida em parceria com um grupo de *startups* da cidade de São Carlos. As *startups* ofereceram treinamento aos alunos sobre práticas ágeis utilizadas no estabelecimento de soluções de software, bem como sobre tecnologias que os alunos ainda não haviam tido contato anteriormente. O propósito do projeto foi estabelecer um cenário em que os alunos teriam que praticar o Scrum e interagir com clientes reais.

Por meio do artigo, tentou-se demonstrar que é possível criar oportunidades mais próximas do “mundo real” para que os alunos tenham uma visão não apenas teórica ou simulada da Engenharia de Software. Em particular, pôde-se capacitar os alunos a utilizarem o Scrum. Apesar do cenário proposto ter sido de grande serventia para estimular a prática da Engenharia de Software, durante o desenvolvimento dos projetos notou-se que os alunos tiveram que enfrentar alguns desafios, como o aprendizado de tecnologias que não estavam acostumados a utilizar. Além disso, muitos dos participantes não tinham aprendido programação *web*, outros estudantes estavam matriculados em muitas disciplinas no semestre e alegaram sobrecarga de trabalho. Embora alguns problemas tenham surgido, foi possível contorná-lo pela flexibilidade do Scrum e auxílio das *startups* que apoiaram a iniciativa. O aprendizado de novas tecnologias normalmente ocorre por meio de oportunidades em conhecê-las e aplicá-las em novos projetos. Assim, acreditamos que os projetos oportunizaram esse aprendizado aos estudantes, apesar dos desafios e dificuldades.

Ainda que a iniciativa adotada nesta experiência tenha sido inspirada nas experiências da professora que ministra a disciplina, problemas diferentes foram observados e reportados pelos alunos. Assim, acredita-se que melhorias precisam ser feitas na forma como a experiência foi conduzida. Portanto, como continuidade deste trabalho, pretende-se evoluir a iniciativa de modo que seja possível evitar problemas como sobrecarga de trabalho nos alunos e dificuldades associadas às tecnologias necessárias para o desenvolvimento dos projetos.

## Agradecimentos

Os autores gostariam de agradecer a CAPES - Código de Financiamento 001, a FAPESP (Processos 2018/26636-2, 2017/10941-8 e 2013/07375-0) e ao CNPq (Processo 312922/2018-3) pelo apoio financeiro.

## Referências

1. Bass, M.: Software engineering education in the new world: What needs to change? In: International Conference on Software Engineering Education and Training. pp. 213–221 (2016)
2. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al.: Manifesto for agile software development (2001)
3. Chatley, R., Field, T.: Lean learning - applying lean techniques to improve software engineering education. In: International Conference on Software Engineering. pp. 117–126 (2017)
4. Devadiga, N.M.: Software engineering education: Converging with the startup industry. In: Conference on Software Engineering Education and Training. pp. 192–196 (2017)
5. Fioravanti, M.L., Sena, B., Paschoal, L.N., Silva, L.R., Allian, A.P., Nakagawa, E.Y., Souza, S.R., Isotani, S., Barbosa, E.F.: Integrating project based learning and project management for software engineering teaching: An experience report. In: ACM Technical Symposium on Computer Science Education. pp. 806–811 (2018)
6. Fox, A., Patterson, D.: Crossing the software education chasm. *Communications of the ACM* **55**(5), 44–49 (2012)
7. Garousi, V., Giray, G., Tuzun, E., Catal, C., Felderer, M.: Closing the gap between software engineering education and industrial needs. *IEEE Software* pp. 1–11 (2019)
8. Ghezzi, C., Mandrioli, D.: The challenges of software engineering education. In: Inverardi, P., Jazayeri, M. (eds.) *Software Engineering Education in the Modern Age*. pp. 115–127. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
9. Gonçalves, L.: Scrum. *Controlling & Management Review* **62**(4), 40–42 (2018)
10. Liu, J.W., Ho, C.Y., Chang, J.Y., Tsai, J.C.A.: The role of sprint planning and feedback in game development projects: Implications for game quality. *Journal of Systems and Software* **154**, 79 – 91 (2019)
11. Mead, N.R., Garlan, D., Shaw, M.: Half a century of software engineering education: The cmu exemplar. *IEEE Software* **35**(5), 25–31 (2018)
12. Nafchi, M.Z., Zulzalil, H., Gandomani, T.J.: On the current agile assessment methods and approaches. In: *Malaysian Software Engineering Conference*. pp. 251–254 (2014)
13. Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology* **50**(4), 280 – 295 (2008)
14. Rubin, K.S.: *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley (2012)
15. Schwaber, K.: *Agile project management with Scrum*. Microsoft press (2004)
16. Souza, S.R.S., Oliveira, B.H., Grillo, F., Cico, C.: Construção de plataformas digitais durante o ensino de engenharia de software: um relato de experiência. In: *Fórum de Educação em Engenharia de Software*. pp. 13–22 (2016)
17. Sutherland, J., Viktorov, A., Blount, J., Puntikov, N.: Distributed scrum: Agile project management with outsourced development teams. In: *Annual Hawaii International Conference on System Sciences*. pp. 274–274 (2007)