

# Relato de Experiência no Desenvolvimento do GradEdu Utilizando a Metodologia Scrum

Fabio R. Pereira Filho<sup>1</sup>, Gregory V. C. Silva<sup>1</sup>, Iago G. de Meira<sup>1</sup>,  
Frank W. C. de Oliveira<sup>1</sup>, Marcelo F. Terenciani<sup>1</sup>

<sup>1</sup>Campus Paranavaí – Instituto Federal do Paraná (IFPR)  
87703-536 – PR – Paranavaí

{20200015459,20220006626,20220009931}@estudantes.ifpr.edu.br,

{frank.willian,marcelo.terenciani}@ifpr.edu.br

## 1. Introdução

Para formar novos profissionais competentes no ramo da Engenharia de *Software*, é interessante que eles tenham algum grau de preparo para lidar com os modelos e *frameworks* de desenvolvimento utilizados profissionalmente. Dentre esses, podemos citar o *Scrum*, focado na entrega contínua e incremental entre *Sprints* (períodos de entrega) e na auto-gestão de times pequenos separados entre *Product Owner* (PO – representa os *stakeholders* e define e prioriza o escopo do projeto), *Scrum Master* (gerencia o projeto e aplica os princípios do *Scrum*) e desenvolvedores [Schwaber and Sutherland 2020].

Ao longo deste ano letivo, as disciplinas de Construção de Software e Projeto Integrador II aplicaram uma adaptação do *Scrum* em um ambiente controlado para que os estudantes possam ter a experiência de estar em um time de desenvolvimento sem as consequências negativas dos erros cometidos. Cada equipe encontrou um cliente real com um problema que poderia ser solucionado com um projeto de *software* e, através do PO, levantou as informações necessárias para definir um escopo e requisitos para serem desenvolvidos ao longo das *Sprints*.

O grupo autor deste resumo definiu como escopo o projeto *GradEdu*, um sistema desenvolvido para o Colégio Estadual Enira Moraes Ribeiro, localizado em Paranavaí - PR. O propósito do sistema é auxiliar a secretaria do colégio na organização da grade de horários, oferecendo uma interface intuitiva para que os usuários – a secretaria – possam gerenciar os horários de cada turma, vinculando disciplinas e professores. Neste trabalho, contaremos sobre a experiência do grupo assim como as técnicas e tecnologias empregadas durante o desenvolvimento do projeto.

## 2. Técnicas e Tecnologias

### 2.1. Técnicas

Além do já mencionado *Scrum*, utilizamos outras metodologias de desenvolvimento, como o quadro *Kanban* que, apesar de não ser uma exclusividade do desenvolvimento de *software*, é bastante utilizado para organizar tarefas em cada *Sprint*. Como padrão da disciplina, foram definidas e adotadas pelo nosso grupo 6 colunas para representar diferentes fases das tarefas ao longo do desenvolvimento, sendo estas:

1. **Product Backlog:** Tarefas pendentes do produto;

2. ***Sprint Backlog***: Tarefas pendentes da *Sprint*;
3. **Em Desenvolvimento**: Tarefas em desenvolvimento no momento;
4. **Revisão do *Scrum Master***: Tarefas que aguardam a aprovação do *Scrum Master*;
5. **Aguardando Avaliação Docente**: Tarefas aprovadas pelo *Scrum Master* que guardam avaliação dos professores ao fim da *Sprint*, e
6. **Entregas *Sprint***: Tarefas concluídas.

Para a definição das tarefas, utilizamos as histórias de usuário, que são úteis para determinar tarefas simples e que sigam o princípio de atomicidade – uma tarefa com um único objetivo e com critérios de avaliação voltados a esse objetivo. Além disso, as histórias de usuário ajudam a identificar o perfil dos *stakeholders* interessados naquela tarefa – com a seção ”COMO”–, a tarefa em si – com a seção ”QUERO”– e a justificativa para a existência dessa tarefa – com a seção ”PARA”; isso abre margem para que a equipe discuta a prioridade das tarefas ao longo do desenvolvimento do projeto.

## 2.2. Tecnologias

Quanto às tecnologias, começaremos mencionando o Git, que é uma ferramenta de versionamento de código que permite a recuperação e atualização dos códigos desenvolvidos, sendo fundamental em projetos em equipe e nos quais se demanda um controle facilitado dos códigos desenvolvidos [Chacon and Straub 2014]. Além do Git, utilizamos o GitHub para: formar uma organização; armazenar os nossos repositórios de código e histórico de atualizações, conhecidas como *commits*, e criar as histórias de usuário e organizá-las no quadro *Kanban*.

Para facilitar o desenvolvimento do projeto, decidimos adotar as tecnologias utilizadas na disciplina de Programação *Web*: *React* no *front-end* e *Spring* no *back-end*. O *React* é uma biblioteca *Javascript* utilizada para o desenvolvimento da interface de usuário por meio de componentes que formam as páginas, bem que a lógica de programação, estrutura e estilo que opera sobre esses componentes. Já o *Spring* é um *framework* de *back-end* desenvolvido para as linguagens Java e Kotlin e responsável por lidar com a lógica do negócio e se comunicar com o *front-end* por protocolos HTTP.

## 3. Experiência da Equipe

Discutidas as tecnologias e técnicas adotadas e os motivos pelos quais o fizemos, resta apenas descrever a experiência da equipe no desenvolvimento do projeto. Apesar de termos definido bastante cedo quais tecnologias e técnicas utilizaríamos, o desconhecimento de como utilizá-las apropriadamente prejudicou o desempenho da equipe. Foram cometidos erros de delimitação do escopo e da priorização de tarefas nas primeiras *Sprints* e isso causou atrasos que impactaram todas as *Sprints* posteriores.

Pela falta de clareza sobre o escopo do projeto, a equipe teve diversos momentos de ociosidade nas primeiras *Sprints*, o que foi um erro conjunto do PO – que cometeu o erro em si – e do *Scrum Master* – que não se atentou às possíveis falhas de comunicação ao analisar as propostas feitas pelo PO – e ambos cometeram esses erros pela sua inexperiência em cargos de liderança que exijam comunicação. Esta avaliação está de acordo com as análises da própria equipe durante as *Sprint Retrospectives* [Schwaber and Sutherland 2020].

Esses erros, no entanto, permitiram a aplicação do conceito de melhoria contínua a cada iteração do *Scrum*. A equipe compreendeu, após as primeiras duas *Sprints* e graças às retrospectivas, quais eram os pontos fortes e fracos de cada um dos integrantes da equipe, assim como as limitações que a equipe enfrentava como um todo. Isso permitiu que o *Scrum Master* e o PO aprimorassem suas habilidades de comunicação com a equipe, uns com os outros e, no caso do PO, com os *stakeholders*.

Contudo, as primeiras retrospectivas revelaram outro problema ainda mais preocupante que as falhas de comunicação: o tamanho da equipe. Apesar do *Scrum* ser aplicável em times pequenos, um time muito pequeno há de enfrentar sobrecarga de tarefas durante as *Sprints*. Desde o início, a equipe contava com apenas 4 integrantes, e logo nas primeiras *Sprints* um dos integrantes mostrou dificuldades de comunicação e certa falta de compromisso com as atividades, deixando de contribuir inteiramente com o desenvolvimento, o que faz com que o *Scrum Master* tivesse que atuar também como desenvolvedor.

Ao longo das *Sprints*, apesar das dificuldades, o time começou a se adaptar e adotar métodos que tornassem a produtividade maior e mais simples ao mesmo tempo, sendo um dos maiores destaque a refatoração do código fonte no *front-end*, que facilitou o trabalho nele. Além disso, PO, *Scrum Master* e desenvolvedores passaram a manter comunicação mais frequente e a prestar assistência uns aos outros, conversando sobre os rumos a se tomar em cada *Sprint* e definindo metas plausíveis com as condições de trabalho.

#### 4. Conclusão

A implementação do *Scrum* no desenvolvimento do *GradEdu* nos proporcionou uma experiência rica em aprendizados. A metodologia, com seus ciclos curtos de entrega, transparência, adaptação e melhoria contínua, foi desafiadora no início, especialmente pela inexperiência da equipe com as tecnologias, técnicas e com a dinâmica ágil como um todo. No entanto, ao longo do projeto, começamos a nos adaptar ao ritmo ágil proporcionado pelas técnicas e tecnologias adotadas.

Com o progresso feito até agora, estamos mais capacitados para continuar o desenvolvimento do *GradEdu*, e dessa vez de forma mais eficiente. Através das retrospectivas, da comunicação rotineira e da entrega contínua de tarefas que agregam valor ao projeto, a capacidade produtiva da equipe evoluiu e tende a continuar evoluindo desde que ela continue adotando essas boas práticas eficientemente. E o ambiente proporcionado pelas disciplinas, como bem dito na introdução, é o ideal para desenvolver essas habilidades.

#### Referências

- Chacon, S. and Straub, B. (2014). *Pro Git*. Apress, 2 edition. Disponível em: <https://git-scm.com/book/pt-br/v2>. Acesso em: 13 nov. 2024.
- Schwaber, K. and Sutherland, J. (2020). O guia do scrum: O guia definitivo para o scrum: As regras do jogo. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>. Acesso em: 13 nov. 2024.