



Software de Gestão de Projetos baseada em Metodologia Ágil

Project Management Software based on Agile Methodology

Aureliano Soares Aragão Junior

UNICEPLAC – Centro Universitário do Planalto Central Aparecido dos Santos
aragaojr.aureliano@gmail.com

Marcos Rafael Mota Campinho

UNICEPLAC – Centro Universitário do Planalto Central Aparecido dos Santos
marcosrafaelbrother@gmail.com

Leonardo Alves da Costa

UNICEPLAC – Centro Universitário do Planalto Central Aparecido dos Santos
leonardo.costa@uniceplac.edu.br

Resumo

Metodologias ágeis surgiram como formas alternativas aos modelos tradicionais de desenvolvimentos de software, como o modelo cascata, e para seu complemento surgiu o framework Scrum. Porém uma das maiores dificuldades que os gestores passam é que maioria das ferramentas de gestão não foram desenvolvidas especificamente para esse *framework*. Sendo assim, este trabalho apresenta um software desenvolvido com foco no método ágil em conjunto com framework Scrum para auxiliar na gestão de projetos, tendo como principais funcionalidades o gerenciamento do *Product Backlog*, *Sprint Backlog*, tarefas e *releases* de um projeto. Comprovando sua eficácia e validando as funcionalidades por meio de comparação com uma planilha, desenvolvida para gestão de projetos que utilizem o *framework* Scrum.

Palavras chave: Scrum. Metodologia ágil. Gerenciamento de projetos. Modelo cascata.

Abstract

Agile methodologies emerged as alternative to traditional models of software development, such as the cascade model, and to complement these method the Scrum framework. However, one of the major difficulties faced by managers is that most management tools have not been developed specifically for this framework. This work presents a software developed with focus on the agile method with Scrum framework to assist in the management of projects, having as main functionalities the management of Product Backlog, Sprint Backlog, tasks and releases of a project. Verifying its effectiveness and validating the functionality by means of a comparison with a spreadsheet, developed for project management using the Scrum framework.

Keywords: Scrum. Agile Method. Project management. Waterfall model.



Introdução

O uso de metodologias ágeis no processo de desenvolvimento de software vem crescendo nos últimos anos conforme apontado pela VersionOne (2017). Cerca de 60% das empresas de software já utilizam esse tipo de metodologia nas equipes de desenvolvimento. Dentre os diversos frameworks ágeis o mais utilizado é o Scrum é definido

[...] como um framework ágil, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos, tendo como principal objetivo uma abordagem iterativa e incremental para entregar valor, ao cliente, com frequência, consequentemente reduzindo os riscos dos projetos. (Sabbagh, 2013)

O *framework* Scrum está presente em 56% das equipes de desenvolvimento ágil e se combinado com *frameworks* híbridos atinge a marca de 70% de uso (VersionOne, 2018), sendo o mais utilizado devido aos grandes benefícios trazidos aos projetos de desenvolvimento de software, tais como: aumento da produtividade, redução de desperdício, visibilidade de progresso do projeto, fácil adaptação à mudança atrelada a uma redução dos riscos e entregas frequentes agregando valor ao negócio do cliente (Sabbagh, 2013). Essas características permitem que projetos ágeis, sobretudo o Scrum, tenham um alto nível de sucesso. Segundo a VersionOne (2018), no ano de 2018, projetos ágeis apresentam até 98% de sucesso, 55% dos projetos são entregues sem atraso, aumento de 47% na qualidade do software e de 57% na satisfação dos clientes.

Com o crescimento da utilização de Scrum, surgiram dificuldades para gerenciar os times e as atividades realizadas pela equipe envolvida no projeto (Prado, 2013). A VersionOne (2018) ainda demonstra que uma das maiores resistências das empresas para adoção de metodologias ágeis, sobretudo o Scrum, é a falta de ferramentas adequadas ao auxílio da gestão. Sendo assim é necessária uma ferramenta intuitiva e efetiva para auxiliar a gestão de projeto.

Neste contexto, o objetivo deste trabalho é desenvolver uma nova ferramenta para gestão de projetos baseada em Scrum, verificando sua eficácia comparando as funcionalidades desenvolvidas com outra ferramenta já utilizada no mercado.

Metodologia Ágil

A Metodologia Ágil surgiu como um método alternativo para o modelo cascata de desenvolvimento de software. Este modelo, de acordo com Pressman (2011), possui uma abordagem sequencial e sistemática para o desenvolvimento de software, ou seja, ao utilizar o este todas as atividades de desenvolvimento são feitas de modo sequencial e para se iniciar uma nova etapa é necessário a conclusão, por total, da tarefa anterior.

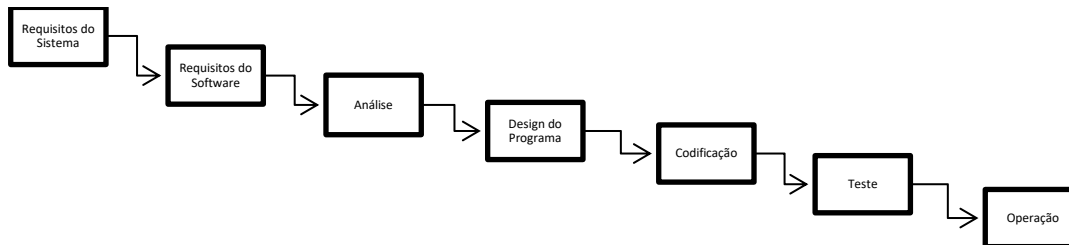
Este modelo é caracterizado pelo detalhamento do projeto focando em identificar logo no início tudo o que será necessário desenvolver, gerando um processo longo de criação de documentos o que acaba tornando os projetos extremamente resistentes à mudança. Porém, suas características não foram tão benéficas no processo de gerenciamento de software. Segundo Pressman (2011), o desenvolvimento de software não segue um fluxo sequencial, na maioria dos casos, pois está suscetível a mudanças durante seu desenvolvimento, sobretudo porque é difícil para o cliente transparecer logo no início do projeto suas necessidades o que traz incerteza em relação ao levantamento, inicial, de requisitos e se atenderá suas reais necessidades. Neste modelo o cliente só irá ver o software operacional no fim do projeto, o que traz muitos riscos.

O modelo cascata tem bem definidos seus processos e sequência. Começando pelo levantamento das necessidades do solicitante e, logo após, a fase de planejamento do que será desenvolvido. Com sua conclusão é construída a modelagem do que será implementado e parte-se

para o desenvolvimento do software, a implantação e por fim o suporte ao software após sua conclusão (Pressman, 2011).

A Figura 1 demonstra as etapas do modelo cascata:

Figura 1 – O modelo cascata e suas etapas.



Fonte: ROYCE, Dr. Winston W. Royce. *Managing the Development of large software systems*. 1970 p. 2

Dr. Winston Royce, criador do modelo cascata, já afirmava em 1970 que esse conceito trazia riscos ao projeto, pois a etapa de testes já se encontra no final do projeto. Essa abordagem ajudou a eliminar muitas das dificuldades encontradas na época, passando a ser a base para o desenvolvimento de software (Boehm, 1998). Contudo, com o aumento da complexidade dos sistemas, este modelo passou a apresentar limitações.

Boehm (1998) aponta que a principal dificuldade que o modelo cascata trouxe é a ênfase, exacerbada, na elaboração de documentações complexas para os projetos, com isso as especificações de requisitos passaram a ser difíceis de entender gerando uma grande quantidade de código que não agregava valor ao cliente.

Essas limitações, impostas pelo excesso de documentação, também causam impacto direto no prazo e custo do projeto, uma vez que, o processo de desenvolvimento de software ocorre num ritmo acelerado e sujeito a mudanças, logo ele só deve ser utilizado em situações onde os requisitos não sofrerão mudanças e seguirão de forma linear até sua finalização (Pressman, 2011).

Como a grande maioria dos projetos de software não possuem essa característica, no ano de 2001, um grupo composto de dezessete pessoas especialistas em desenvolvimento de software se reuniram com intuito de criar uma nova metodologia que melhorasse o resultado das equipes de desenvolvimento, assim como o nível de satisfação dos clientes (Highsmith, et al., 2001). Partindo do princípio de uma abordagem baseada em experiências anteriores, essa forma passou a ser o modo mais produtivo e eficaz de se desenvolver softwares. Esse movimento se caracterizou como o Manifesto Ágil.

Desta forma, esse Manifesto tem como premissa valorizar as pessoas e suas interações, uma vez que quem gera o produto, ou software, são pessoas que possuem características únicas e são essas individualidades que afetam diretamente a qualidade do software (Highsmith, et al., 2001).

Existem doze princípios que devem ser seguidos no uso das metodologias ágeis de acordo com Highsmith (2001), a saber:

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.



- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Scrum

O Scrum teve início no começo de 1990, sendo desenvolvido inicialmente para o gerenciamento e desenvolvimento de produtos. De acordo com Schwaber e Sutherland (2017) Scrum é um *framework*, definido como uma estrutura básica de suporte de um sistema, conceito ou texto (Oxford, 2019), dentro do qual é possível empregar vários processos ou técnicas, isto é, o *framework* pode ser integrado a várias outras técnicas para melhor se adaptar ao ambiente de trabalho e a equipe. Baseado no empirismo, ou seja, o Scrum é formado principalmente pela tomada de decisões e na experiência, tornando a documentação do software uma complementação ao invés de uma necessidade.

O processo empírico é composto por três pilares:

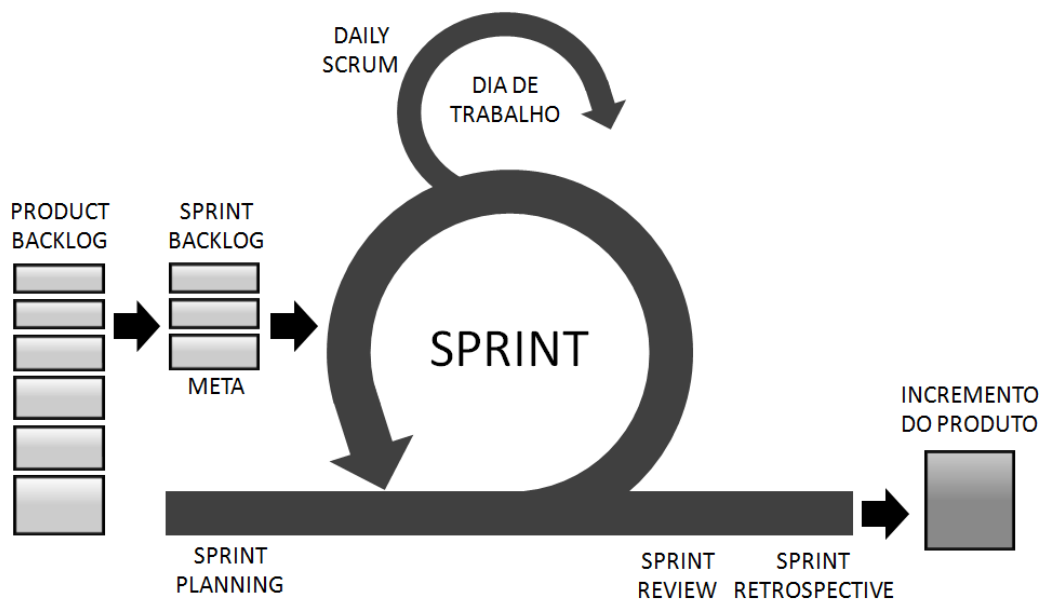
- **Transparência:** Os artefatos devem estar visíveis aos responsáveis pelo resultado para que tenham o mesmo entendimento.
- **Inspeção:** O time deve ser responsável por inspecionar os artefatos e progresso.
- **Adaptação:** Na inspeção dos artefatos, se algum dos elementos não se adequem de acordo com a transparência é necessário que seja ajustado, sendo assim o ajuste deve ser realizado o mais breve possível.

A aplicação do Scrum em projetos pode trazer muitos benefícios, alguns dos citados por Sabbagh (2013) é a redução de riscos de insucesso, entregar valor mais rápido e desde cedo, e lidar com as mudanças de escopo, transformando-as em vantagem competitiva. Segundo a VersionOne (2018), projetos ágeis apresentam 98% de sucesso, demonstrando o desempenho que o *framework* pode trazer para o seu projeto.

Diferente do modelo cascata, a utilização do Scrum no projeto possibilita pequenas entregas desde seu início, trazendo retorno ao investimento do cliente e possibilitando mudanças ou adições de acordo com seu *feedback*. Naquele modelo, não é possível fazer entrega em pequenas partes, pois é preciso que a etapa anterior esteja concluída para iniciar a seguinte, fazendo com que a entrega do produto seja no fim de todas as etapas. A Figura 2 apresenta como funciona o ciclo do Scrum.



Figura 2 – O ciclo do Scrum



Fonte: Sabbagh, Rafael. Scrum Gestão Ágil para Projetos de Sucesso. 2013 p. 42

Um projeto Scrum começa com o *Product Owner* (PO), responsável por maximizar o valor do produto e o trabalho do time de desenvolvimento (Schwaber e Sutherland, 2017), definindo quais são as atividades a serem desenvolvidas e colocando-as no *Product Backlog*. Tendo conhecimento dessas atividades inicia-se a *Sprint*, que são as atividades realizadas em período determinado que resultam no incremento do produto (Schwaber e Sutherland, 2017), com o PO em conjunto com o time desenvolvimento escolhendo os itens do *Product Backlog* que serão realizados para gerar um incremento ao produto, a partir desses itens é criado o *Sprint Backlog* e inicia-se as atividades da *Sprint*.

Concluindo as atividades, ou esgotando o tempo da *Sprint*, ocorre o *Sprint Review* que é a inspeção do incremento e adaptação do *Product Backlog* caso seja necessário. Nessa etapa são definidos quais itens estão prontos, são discutidos problemas que ocorreram na *Sprint* e na execução das tarefas e como eles foram solucionados, além disso também são definidas novas prioridades e o que deve ser feito na próxima *Sprint* (Schwaber e Sutherland, 2017).

Em seguida é realizada a *Sprint Retrospective* os times fazem uma revisão geral de seu desempenho avaliando como foi o seu relacionamento, quais processos devem ser alterados ou melhorados e com base nessa revisão é criado um plano de melhorias da equipe de desenvolvimento, no ponto de vista pessoal e não técnico, que devem ser implementado já na próxima *Sprint* (Schwaber e Sutherland, 2017).

Os projetos realizados com Scrum são restritos ao time que iniciou o projeto, que são encarregados por sua realização, validação e testes com o propósito de garantir que as funcionalidades estejam conforme os requisitos acordados com o cliente. Sabbagh (2013) diz que não existem times externos responsáveis por avaliar ou garantir a qualidade do produto gerado. Isso torna o próprio time responsável pela sua qualidade do produto, sendo necessária uma validação contínua no projeto e em suas tarefas, para que não ocorra atrasos nas entregas.



Desenvolvimento

Materiais e Métodos

No desenvolvimento deste trabalho foi utilizada a linguagem de programação Java em conjunto com o *framework* JavaServer Faces (JSF), que é a interface padrão para o desenvolvimento de aplicação *web* (Oracle, 2019). A escolha da linguagem Java foi devido ao indicador da TIOBE *Programming Community* (2019), aponta o Java como a linguagem de programação mais bem ranqueada, desde 2015, para o desenvolvimento de software de acordo com as tendências estratégicas, popularidade e aplicação em novos projetos no ambiente corporativo.

Para o levantamento das funcionalidades do sistema foram utilizados os conceitos do Guia Scrum de Ken Schwaber e Jeff Sutherland, em sua versão traduzida para o português do Brasil e distribuída em 2017, em conjunto com os conceitos apresentados por Sabbagh em seu livro Scrum: Gestão Ágil para Projetos de Sucesso, publicado em 2013. Baseado na interpretação e debate dos conceitos foram definidas como principais funcionalidades do sistema a criação, deleção, manutenção e visualização de projetos, *sprints*, itens das *sprints* e *releases*. Estes itens em conjunto permitem o gerenciamento de projetos baseados no *framework* Scrum.

Tendo conhecimento das principais necessidades para o desenvolvimento do sistema aplicou-se a técnica de *brainstorming* para definição de quais campos, itens e demais métodos se tornam necessários para que o sistema atenda aos objetivos propostos. Com essas informações a prototipação do sistema foi iniciada, possibilitando uma visão inicial do comportamento e de que maneira a ferramenta apresentará as informações aos usuários.

As funcionalidades do sistema desenvolvido serão comparadas com uma planilha eletrônica desenvolvida para facilitar a gestão de projetos que utiliza o *framework* Scrum. Esta ferramenta foi escolhida baseando-se no estudo realizado pela VersionOne, em 2019, que apontou o Microsoft Excel como a segunda ferramenta mais utilizada, com 48%, para o gerenciamento de projetos, e o Google Docs com 19%. No mesmo estudo é demonstrado que 29% dos participantes recomendam o uso do Excel, o que reforça a necessidade de outra ferramenta para auxiliar no gerenciamento do projeto. A planilha escolhida para comparação foi obtida pesquisando em sites de buscas, utilizando como palavras-chave: planilha scrum excel template. Analisando os resultados obtidos, baseado nos conceitos Scrum, a planilha que melhor auxilia no gerenciamento de projetos Scrum foi a desenvolvida pela Platinum Edge, empresa especializada em realizar treinamentos sobre Scrum, e disponibilizada de forma gratuita para utilização.

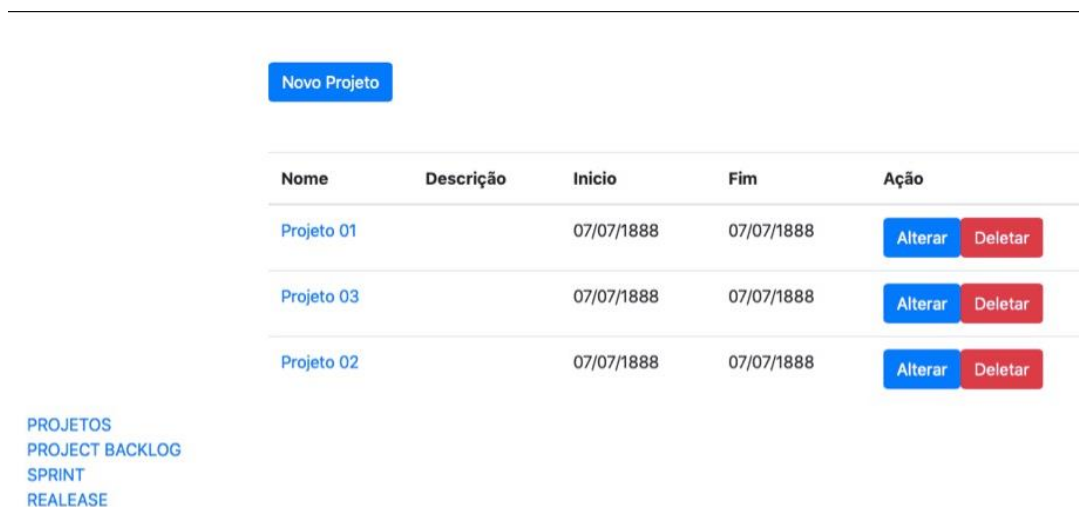
Resultados

O sistema desenvolvido foi baseado nos conceitos de Scrum, sendo divididos em e quatros funcionalidades centrais:

- Gerenciamento de *backlog*
 - Gerenciamento de *sprint*
 - Gerenciamento de itens
 - Gerenciamento de *releases*
-

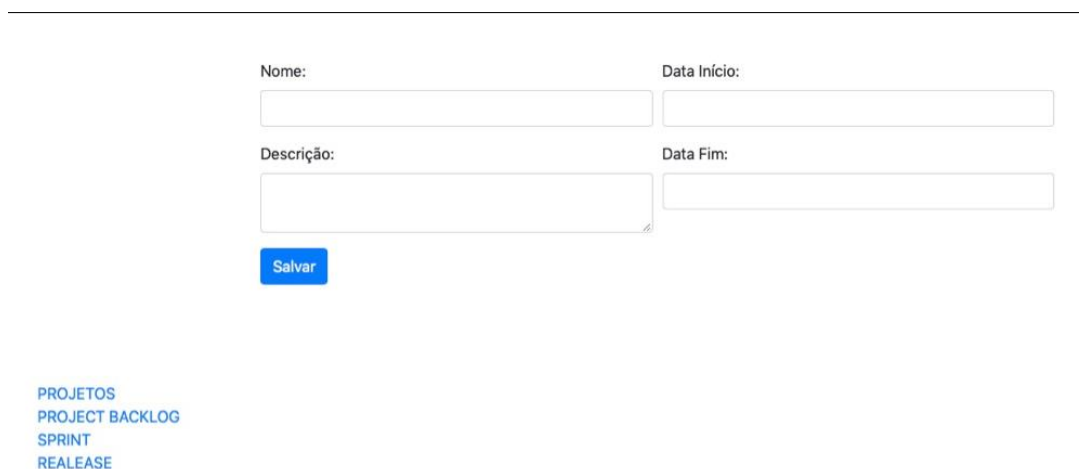


Figura 3 – Tela de início.



Após a abertura do sistema, é apresentado para o usuário todos os projetos cadastrados. Sendo possível gerenciá-los, criando, editando ou deletando um projeto existente. É preciso que exista ao menos um projeto para continuar utilizando as outras funcionalidades do sistema. Na figura 4, é possível ver a figura de cadastro de projeto.

Figura 4 – Tela de cadastro do projeto.



Ao selecionar um projeto, o usuário é levado para a tela de *Project Backlog*, ver figura 5. Nesse menu, é exibido as funcionalidades necessárias para a elaboração do projeto. É possível criar



itens, deletar ou editar. A partir dos itens desse menu, será criado as sprints. Na Figura 6, é possível ver a representação da tela de cadastro de itens do *Project Backlog*.

Figura 5 – Tela de *Project Backlog*.

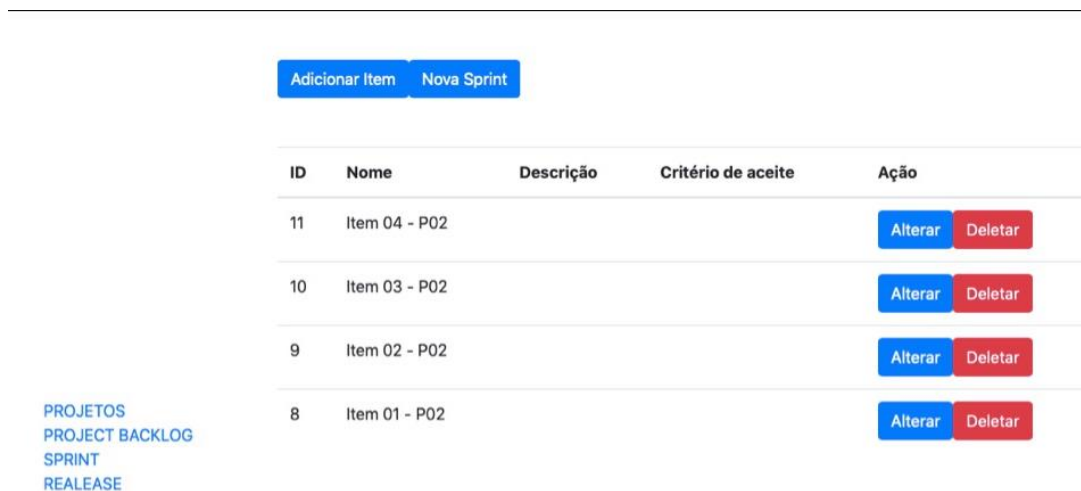
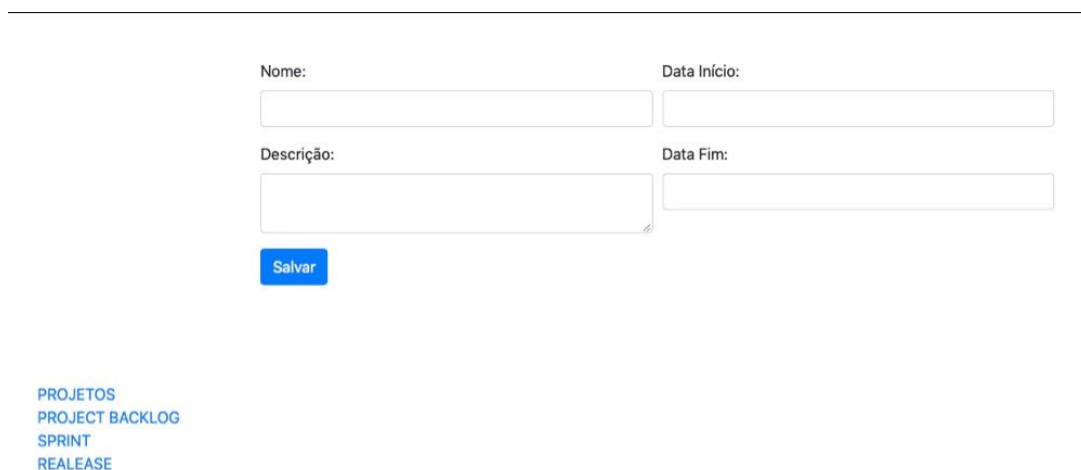


Figura 6 – Tela de cadastro do *Project Backlog*.



Ao criar uma Sprint é preciso que tenham algum item já criado para compô-la, pois uma Sprint é composta pelos itens do *Project Backlog*, que devem ser selecionados no momento do cadastro da Sprint. Os itens exibidos serão apenas aqueles que não estejam associados a uma Sprint já existente. Nos detalhes da Sprint, conforme imagem 9, é possível cadastrar a situação para os itens, para que possa ter uma noção do andamento da sprints. Caso tenha algum problema



para a realização dos requisitos pode inserir um impedimento, para que seja dada uma atenção e encontrar uma solução.

Figura 7 – Tela de Sprint.

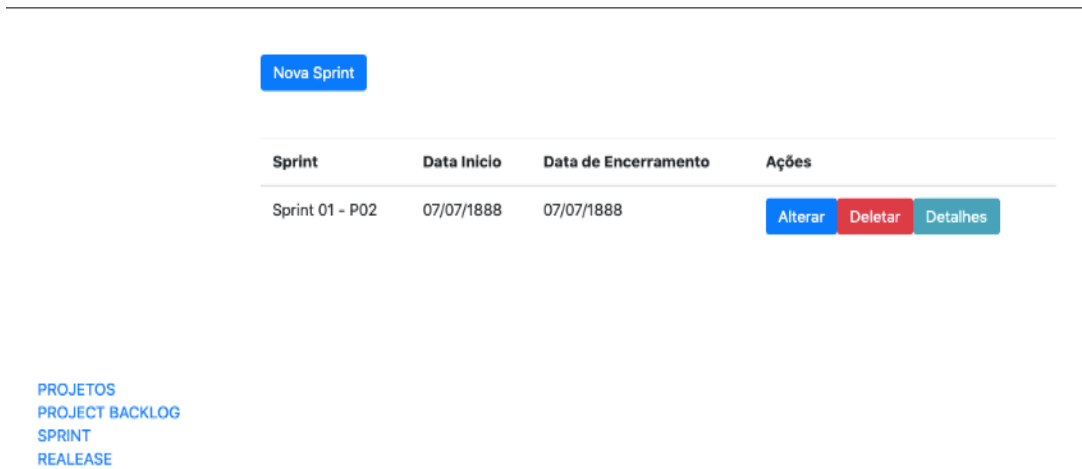


Figura 8 – Tela de cadastro da Sprint.

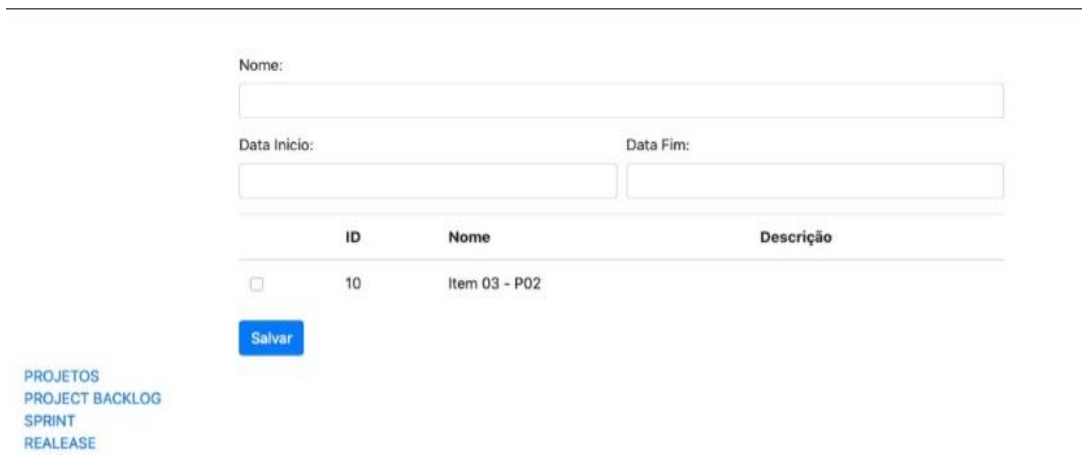




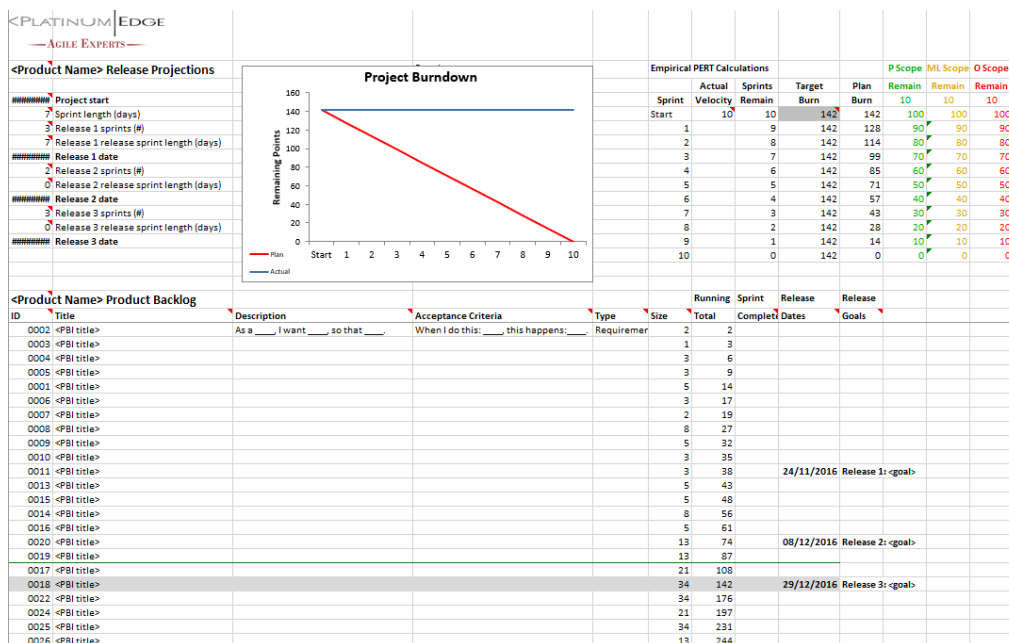
Figura 9 – Tela de detalhes da Sprint.

ID	Nome	Descrição	Status	Impedimento	Ação
11	Item 04 - P02		A fazer	<input checked="" type="checkbox"/>	Atualizar Deletar
9	Item 02 - P02		A fazer	<input checked="" type="checkbox"/>	Atualizar Deletar
8	Item 01 - P02		A fazer	<input checked="" type="checkbox"/>	Atualizar Deletar

PROJETOS
 PROJECT BACKLOG
 SPRINT
 REALEASE

Ao compararmos as funcionalidades do sistema com as apresentadas na planilha desenvolvida pela Platinum Edge, a principal diferença está na forma e clareza com que as informações são apresentadas. Ao contrapor a forma como o *Project Backlog* é apresentado no sistema desenvolvido, ver figura 5, as informações estão dispostas de modo mais claro e objetivo ao apresentado na planilha, ver Figura 10.

Figura 10 – Lista de tarefas.

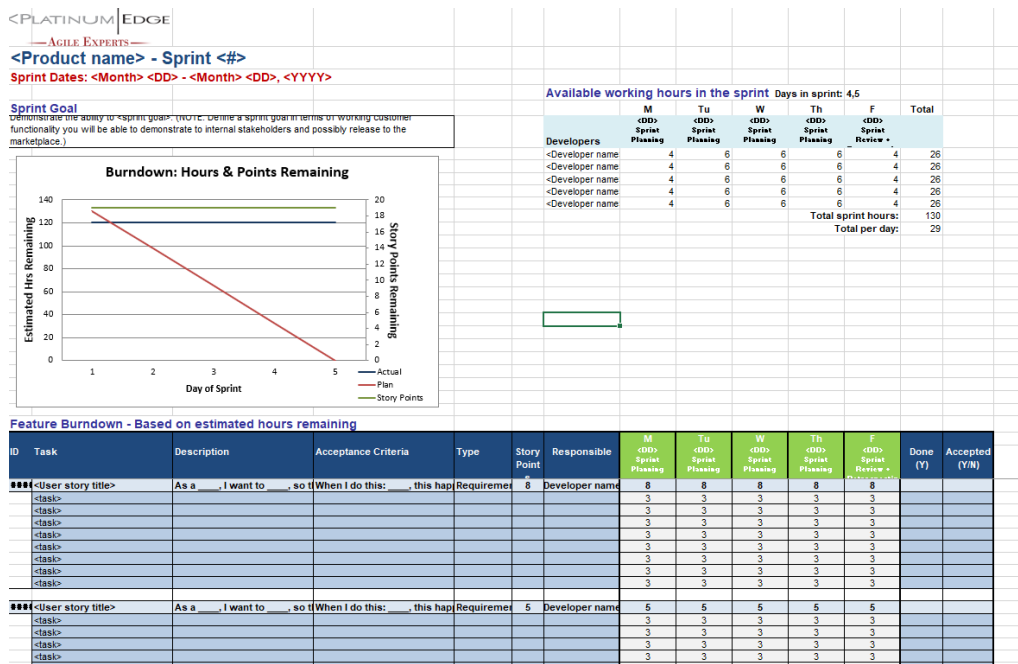


As informações apresentadas na planilha não permitem uma fácil interpretação e leitura das tarefas, o que *Product Backlog* deve proporcionar. Informações referentes a *release* não são pertinentes nesse momento. Porém a planilha apresenta funções a que não foram contempladas, em destaque temos o tamanho da tarefa que permite um melhor dimensionamento do esforço necessário para a execução da tarefa.

O mesmo ocorre com o *Sprint Backlog*, ver Figura 10. As informações mais pertinentes para a equipe de desenvolvimento nesse caso são o status da atividade assim como a sua descrição, o que não está tão claro e de fácil identificação na planilha. Já a ferramenta desenvolvida apresenta exatamente essas informações de forma objetiva, ver Figura 9.



Figura 11 – Sprint Backlog da planilha.



Quanto as informações referentes aos itens a serem desenvolvidos tanto a planilha quanto o sistema apresentam as informações de modos semelhantes, conforme mostrado nas figuras 9 e 11. Já as *releases* são apresentadas de forma completamente diferentes no sistema e na planilha. Na planilha as informações estão juntas com o *Product Backlog*. No sistema primeiro tem-se a visão de quais *releases* fazem parte do projeto e ao detalhar a *release* que se vê as suas respectivas *Sprints* e tarefas, facilitando o entendimento dos dados e permitindo que as tarefas não possuam um id sequencial e estejam ordenadas, já que a relação delas com a *release* se dará pela *Sprint*.

A planilha apresenta gráficos que facilitam o acompanhamento do trabalho desenvolvido. Na planilha os gráficos são baseados no *sizing* das tarefas e no tempo em que serão desenvolvidas.

Figura 12 – Tela de Release.





Discussões

Baseado nas funcionalidades apresentadas, nos resultados, o objetivo de desenvolver uma ferramenta de gestão baseada nas práticas ágeis e Scrum foi atingindo. Uma vez que, conforme proposto, a ferramenta permite a gestão e o acompanhamento do projeto, de acordo com os conceitos Scrum, definidos por Schwaber e Sutherland. Para o desenvolvimento da ferramenta optou-se por design simples e funcional, permitindo maior eficiência e adaptabilidade dos usuários, o que facilita a gestão do projeto.

Um dos principais aspectos do Scrum está presente na ferramenta, que é seguir o processo empírico. Este processo é garantido pela visibilidade das tarefas para todos os membros envolvidos permitindo aos envolvidos no processo de desenvolvimento validar e realizar ajustes necessários durante sua execução, ver Figura 5.

A ferramenta permite o gerenciamento do Product e Sprint Backlog, que são um dos principais artefatos do ciclo Scrum, ver Figura 2, uma vez que eles compreendem as tarefas que devem ser executadas. O bom gerenciamento destes itens facilita o sucesso do projeto. Conforme a Figura 5, todos os itens estão de fácil visualização e selecionar um Sprint apenas os itens que dela fazem parte são exibidos na Figura 9 possibilitando que a equipe tenha de forma clara em quais as tarefas necessitam ser feitas.

Uma das funcionalidades que o sistema não possui e é possível realizar na planilha é o *sizing* das tarefas e Sprint. Essa funcionalidade não está presente no sistema porque seu uso requer um nível maior de maturidade e conhecimento das práticas Scrum, além de não ser obrigatório, e sua implementação poderia aumentar a complexidade de entendimento da ferramenta, dificultando o seu uso.

A maior diferença encontrada entre o sistema e a planilha está na forma como as *releases* são apresentadas. No modo demonstrado pela planilha o maior problema está na forma como é mostrada, já que se as tarefas não estejam ordenadas pelo id sequencial fica difícil saber a qual *release* pertence determinada tarefa ou *Sprint*. Já na ferramenta desenvolvida foi criada uma opção exclusiva para o acompanhamento do *release*.

Considerações finais

A utilização da Metodologia Ágil com o *framework* Scrum vem sendo bastante utilizadas por equipes de desenvolvimento com o passar dos anos. Algumas ferramentas do mercado não são voltadas exclusivamente para a utilização desse *framework*, sendo necessário realizar alterações para melhor se adaptar as ferramentas de mercado. Com a utilização de uma ferramenta adequada para o gerenciamento de projetos baseados em Scrum, o controle do que está sendo feito se torna mais eficiente. Dessa forma existe um maior controle sobre o tempo, e no que a equipe está trabalhando, de forma simplificada, facilitando a gestão.

O principal objetivo do artigo é o desenvolvimento de uma ferramenta, que facilite a gestão de projetos baseado no *framework* Scrum. Conforme explanado nos resultados e discussão podemos concluir que o sistema desenvolvido facilita a gestão, principalmente daqueles que possuem pouca maturidade e experiência no uso de Scrum. Pois seu uso está intuitivo e as principais informações estão visíveis de forma clara e objetiva.

As funcionalidades desenvolvidas foram validadas e consolidadas na comparação com planilha. Forma apontados como principais vantagens a simplicidade e facilidade com que as informações estão a disposição do usuário, de modo que os membros envolvidos no projeto conseguem



visualizar o que deve ser desenvolvido. Também foram verificadas que algumas das soluções adotadas possuem ampla vantagem em relação a planilha utilizada na comparação destacando-se as informações referentes as *releases*. Porém a planilha também apresenta funcionalidades que não estão presentes no sistema, como o *sizing* das tarefas e da Sprint e a presença de gráficos.

Como melhorias para a ferramenta e projeto futuro, percebeu-se que uso de gráficos facilita o acompanhamento do projeto, permitindo uma visão mais clara e rápida do progresso do trabalho. Do mesmo o desenvolvimento de usuários para acesso ao sistema permitira um melhor controle, fazendo com que cada um, além de todas as informações do projeto, verificasse qual tarefa é de sua responsabilidade.

Com essas novas funcionalidades sendo implementadas é possível validar o sistema com outras ferramentas mais robustas e maduras do que uma planilha eletrônica. Além dessa comparação com outras ferramentas é interessante que a ferramenta seja testada por um profissional da área permitindo que novas funcionalidades possam ser implementadas e que as atuais possam tenham sua eficácia comprovada com base na vivência profissional.

Referências

BOEHM, Barry W. *A Spiral Model of Software Development and Enhancement*. TRW Defense Systems Group, The Institute of Electrical and Engineers (IEEE), 1998.

BORGES JÚNIOR, Alex N. e KAMEI, Fernando K. **A Utilização de Práticas Scrum no Desenvolvimento de Software com Equipes Grandes e Distribuídas: um Relato de Experiência**. Disponível em: < <http://www.lbd.dcc.ufmg.br/colecoes/wdds/2011/005.pdf>>. Acesso em: fev. 2019.

DE LIMA, Felipe Augusto Pastore. **Gerenciamento de Projetos de Software com SCRUM**. Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná (UTFPR). Medianeira, 2011. Disponível em: < http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/586/1/MD_CO-ADS_2011_1_02.pdf>. Acesso em: mar. 2019.

HIGHSMITH, Jim, et al. **Manifesto Ágil**. 2001. Disponível em: < <https://www.manifestoagil.com.br/>>. Acesso em mai. 2019.

LOPES, Cristiana Sofia da Silva. **Scrum para Ambientes de Software Distribuídos: Análise Crítica e Estudo de Casos**. Dissertação de Mestrado, Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia e Gestão de Sistemas de Informação. Universidade do Minho, Braga, Portugal. Outubro de 2014. Disponível em: < <http://repositorium.sdum.uminho.pt/handle/1822/35261>>. Acesso em: mar. 2019.

OLIVEIRA, Juliano e SOUZA, Adriana. **Adaptando o Scrum par evolução de um ERP de grande porte com múltiplas equipes concorrentes**. Instituto de Informática – Universidade Federal de Goiás, Goiânia-GO. 2012.

ORACLE. **JSP Specifications**. Disponível em: < <https://javaee.github.io/javaxserverfaces-spec/>>. Acesso em: abr. 2019.

OXFORD. **Framework**. Disponível em:< <https://en.oxforddictionaries.com/definition/framework>>. Acesso em: mai. 2019.

PLATINUM EDGE. **Planilha Scrum**. Disponível em: < https://platinumedge.com/sites/default/files/public/platinumedge_pb-sb-templates_excel.xlsx>. Acesso em mai.2019



PRADO, Gustavo Somadossi e FERRARI, Fabiano Cutigi. **Avaliação do Impacto do Desenvolvimento Distribuído de Software em um Projeto Adotando o Scrum: Um Estudo Comparativo**. Revista T.I.S Tecnologias Infraestrutura Software, vol. 2, nº 3, p. 201-210, 2013. Disponível em: < <http://www.revistatis.dc.ufscar.br/index.php/revista/article/view/64/59>>. Acesso em: mar. 2019.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7ª edição. Porto Alegre-RS: AMGH Editora Ltda., 2011.

ROYCE, Dr. Winston W. *Managing the development of large software systems*. *Proceedings, IEEE Wescon, The Institute of Electrical and Engineers (IEEE)*, p. 1-9, 1970.

SABBAGH, Rafael. **Scrum – Gestão Ágil para Projetos de Sucesso**. Casa do Código. São Paulo, 2013.

SATPATHY, Tridibesh. **Um guia para o Conhecimento em Scrum (Guia Sbok™)**. SCRUMstudy™, Estados Unidos, Arizona, 3ª edição, 2017.

SCHAWABER, Ken e SUTHERLAND, Jeff. **Guia do Scrum – Um guia definitivo para o Scrum: As regras do Jogo**. Disponível em: <<https://www.scrumguides.org/docs/scrum-guide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: abr. 2019.

SILVA, Adriana F. e SILVEIRA, Francisca Raquel de V. **Uma abordagem sobre engenharia de requisitos para metodologia ágil SCRUM**. Anais Eletrônico, IX Encontro Unificado de Computação, ENUCOMP 2016, Teresina-PI, v.01, n.01, 2016.

TIOBE Programming Community Index. Disponível em: <<https://www.tiobe.com/tiobe-index/>>. Acesso em: abr. 2019.

VERSIONONE. 12th Annual State of Agile Report. 2018. Disponível em: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>>. Acesso em: mar. 2019.

VERSIONONE. 13th Annual State of Agile Report. 2019. Disponível em: <<https://www.state-ofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>>. Acesso em: jun. 2019.
