



# UMA SOLUÇÃO ALGORÍTMICA PARA PROJETOS DE TI UTILIZANDO METODOLOGIA ÁGIL SCRUM: UM ESTUDO DE CASO

Recebido: 08/08/2018

Aprovado: 10/01/2019

<sup>1</sup> Daiane de Almeida Oliveira

<sup>2</sup> Irapuan Glória Júnior



## Resumo

As metodologias ágeis estão sendo empregadas no desenvolvimento de software nas empresas, por apresentarem uma forma moderna e simplificada, tornando as equipes de desenvolvimento mais flexíveis e auto gerenciáveis, além de permitindo que as tarefas sejam definidas e amplamente discutidas antes de serem executadas. O objetivo desta pesquisa é a solução computacional de um sequenciamento de tarefas com a criação de um algoritmo para auxiliar o time de desenvolvimento nas definições das atividades a serem desenvolvidas por meio da metodologia ágil Scrum. A metodologia utilizada foi de estudo de caso único e coleta de dados junto aos gestores e desenvolvedores ligados ao time Scrum. O resultado foi a prototipação de um sistema e a criação do algoritmo. As contribuições para a prática são a utilização do algoritmo e protótipo para auxiliar os gestores de projetos de TI a melhoria da seleção de atividades para as Sprints. A contribuição teórica é apresentar formas alternativas de escolha das atividades que estão ausentes no Scrum Guide.

**Palavras-chave:** Scrum. Metodologia Ágil. Desenvolvimento. Projetos de TI. Algoritmo.

<sup>1</sup>Faculdade de Teologia e Ciências – Fatec, Itu, São Paulo, (Brasil). E-mail: [oliveira.daiane13@yahoo.com.br](mailto:oliveira.daiane13@yahoo.com.br) Orcid id: <https://orcid.org/0000-0002-0166-2634>

<sup>2</sup> Faculdade de Teologia e Ciências - Fatec Itu, São Paulo, (Brasil). E-mail: [ijunior@ndsgn.com.br](mailto:ijunior@ndsgn.com.br) Orcid id: <https://orcid.org/0000-0003-2973-3470>

## **AN ALGORITHMIC SOLUTION FOR IT PROJECTS USING AGILE SCRUM METHODOLOGY: A CASE STUDY**

### **Abstract**

Agile methodologies are being used in software development in companies because they present a modern and simplified form, making development teams more flexible and self-manageable, and allowing the tasks to be defined and widely discussed before they are executed. The objective of this research is the computational solution of a task sequencing with the creation of an algorithm to help the development team in the definitions of the activities to be developed through the agile Scrum methodology. The methodology used was a single case study and data collection with the managers and developers connected to the Scrum team. The result was the prototyping of a system and the creation of the algorithm. The contributions to practice are the use of the algorithm and prototype to help IT project managers improve the selection of activities for the Sprints. The theoretical contribution is to present alternative ways of choosing activities that are absent in the Scrum Guide.

**Keywords:** Scrum. Agile Methodology. Development. IT Projects. Algorithm.

## Introdução

No Brasil as metodologias de desenvolvimento ágil têm gerado grande entusiasmo entre seus usuários e na comunidade acadêmica. Destacam-se, principalmente, os aspectos relacionados às melhorias nos resultados que a empresa de software deseja obter, como o aprimoramento de seus processos internos e de sua estrutura organizacional (Varaschim, 2009).

Uma dessas metodologias é o *Scrum*: um *framework* para desenvolver, entregar e manter produtos complexos, dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível (Sutherland & Schwaber, 2013), documentação reduzida (Varaschim, 2009) e robusto o suficiente para desenvolver e manter produtos complexos por meio de funções, eventos, artefatos e regras que os mantêm integrados (Sutherland & Schwaber, 2013).

É possível encontrar o Scrum sendo aplicado em projetos de TI devido a possibilidade de empregar equipes auto organizadas (Dingsøyr, Dyba, & Moe, 2010), com até 12 membros que possuem alternância nos cargos e funções (Sutherland & Schwaber, 2013) que promove a continuação das atividades, mesmo com *turnover* (Glória Júnior, Oliveira, & Chaves, 2014).

Em uma empresa de desenvolvimento de sistemas, denominada Empresa-Alpha, relatou aos autores que havia dificuldades na escolha e priorização das tarefas a serem realizadas no momento da criação da lista do *Sprint Backlog*.

Diante deste problema, o presente relato técnico atuou na solução da questão de pesquisa: "Como determinar o sequenciamento de tarefas para o desenvolvimento de software utilizando metodologia ágil?". Possui como objetivos: (1) Identificar dos critérios utilizados pelos desenvolvedores de software para definir a forma de sequenciar o desenvolvimento do mesmo, através das Sprints, como elas são determinadas; e (2) Apresentar uma solução algorítmica para o sequenciamento de tarefas aplicado na metodologia *Scrum*, para definir as *Sprints* a partir do *Sprint Backlog*.

A pesquisa utilizou a metodologia de estudo de caso único (Yin, 2015), com coleta de dados junto aos time *Scrum* para determinar os requisitos para a determinação das escolhas das atividades. Este relato possui a seção 2 com o referencial teórico dos eixos que compõem este trabalho. Na seção 3 é apresentada a metodologia utilizada.

A seção 4 apresenta os resultados obtidos. A conclusão e considerações finais são feitas na seção 5.

## Referencial Teórico

### Projetos de TI

Projetos são um conjunto de atividades temporárias, realizadas em grupo, destinadas a produzir um produto, serviço ou resultado único (PMI, 2017).

Os projetos de TI são aqueles que empregam tecnologia em seus resultados (Pressman, 2014; Sommerville, 2015).

Em todos os projetos há riscos (Kerzner, 2010; PMI, 2017), mas os que possuem dependência tecnológica possuem maior número de riscos (Silveira, Russo, Júnior, & Sbragia, 2018).

Para desenvolver um sistema, é necessário efetuar o levantamento de requisitos junto aos *Stakeholders* (Sommerville, 2015), que são as pessoas envolvidas no projeto (PMI, 2017), e que indicarão as necessidades a serem atendidas pelo software a ser desenvolvido ou adquirido pela empresa (Pressman, 2014).

Os principais papéis encontrados em projetos de TI são: (1) Programadores, responsáveis por escrever testes e código, automatizar processos tediosos e melhorar o design do sistema; (2) Analistas, que trabalham com o cliente e com os analistas de negócio para escrever testes de aceitação automatizados, definindo os cenários de sucesso e erro de cada história; e, (3) Gerentes de TI, que facilitam a comunicação dentro do time, removendo empecilhos e coordenando a comunicação com pessoas externas à equipe do projeto, como fornecedores, clientes externos ou o resto da organização (Sato, 2007).

Os sistemas e seus requisitos são representados no padrão *Unified Modeling Language* (UML) aos outros desenvolvedores e demais *stakeholders*, sendo os principais artefatos o Diagramas de Caso de Uso que apresenta os processos contemplados e o Diagrama de Atividades com o detalhamento dos processos mais próximo ao código computacional (OMG, 2015).

## Scrum

Os métodos ágeis, como o *Scrum*, são provenientes do manifesto ágil e são caracterizados por entregas rápidas e constantes, além de equipes auto gerenciadas, trocas constantes de papéis e aproximação com o cliente (Beck et al., 2001).

O ciclo *Scrum*, conforme Figura 1, é iniciado com o *Product Backlog* que possui a lista de todas as atividades a serem realizadas, em seguida são selecionadas as atividades que farão parte do *Sprint Backlog* que serão divididas em 4 ciclos de aproximadamente uma semana cada um (Buckl et al., 2011).

O *Scrum* é uma estrutura para desenvolver e manter produtos complexos por meio de funções, eventos, artefatos e regras que os mantêm integrados (Sutherland & Schwaber, 2013), com os seguintes papéis (Sutherland & Schwaber, 2013):

✓ **Time Scrum.** São auto organizáveis e multifuncionais, escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros de fora do Time. É composto pelo *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*;



projetos de TI que utilizavam a metodologia ágil *Scrum*. Foi realizado no período de março/2019 a abril/2019;

✓ **Análise do Questionário.** Foram identificados os processos e decisões que deveriam ser contemplados na solução;

✓ **Construção do Algoritmo.** Foram criados diagramas em UML que representavam os processos para a solução computacional.

### Perfil dos Entrevistados

Na Tabela 1 foram selecionados os funcionários da empresa envolvidos em projetos de TI que aplicam a metodologia ágil *Scrum*, denominados Respondentes (R).

Tabela 1 - Respondentes

Respondentes	Função
R1	Programador Júnior
R2	Programador Java Pleno
R3	Analista de Testes
R4	Programador Java Pleno
R5	Programador Java Pleno
R6	Programador Java Pleno
R7	Programador Java Trainee
R8	Analista de Projetos
R9	Consultor TI Especializado
R10	Programador Júnior

### Aplicação Prática

#### Identificação dos Requisitos

Os critérios utilizados no processo de escolha realizados pelos profissionais de TI foram obtidos por meio das entrevistas resultando na Tabela 2. Foi identificado que havia a verificação das pendências da *Sprint* anterior (**REQ01**) e colocadas como prioridade máxima a serem realizadas na próxima *Sprint*, conforme mencionada pela maioria dos respondentes, como o R2: "...é analisado se esta atividade foi finalizada, e caso não tenha sido, ela passa para a próxima *Sprint*...".

A decisão de cancelamento da *Sprint* (**REQ02**) ou mesmo a decisão de cancelamento de uma atividade da *Sprint* (**REQ03**) que seriam as atividades que não fazem mais sentido serem feitas de acordo com as circunstâncias, como atrasos de entrega e mudança de tecnologia, como foi mencionado pelo R4: "...durante a reunião, caso a tarefa não tenha sido finalizada, ela passa para a próxima *Sprint* com prioridade maior, a não ser que o *Product Owner* decida que essa tarefa não deve ser continuada...".

A Definição de "Pronto" da *Sprint* (**REQ04**), quando todas as atividades da *Sprint* são finalizadas, revisadas e validadas, são classificadas como incremento do produto, foi mencionada pelo R9: "...ao final da *Sprint*, se ela foi finalizada, é arquivada como 'realizada com sucesso'...".

Diante dos requisitos identificados, o algoritmo foi elaborado para ser utilizado no momento da criação de uma nova *Sprint*.

Tabela 2 - Requisitos

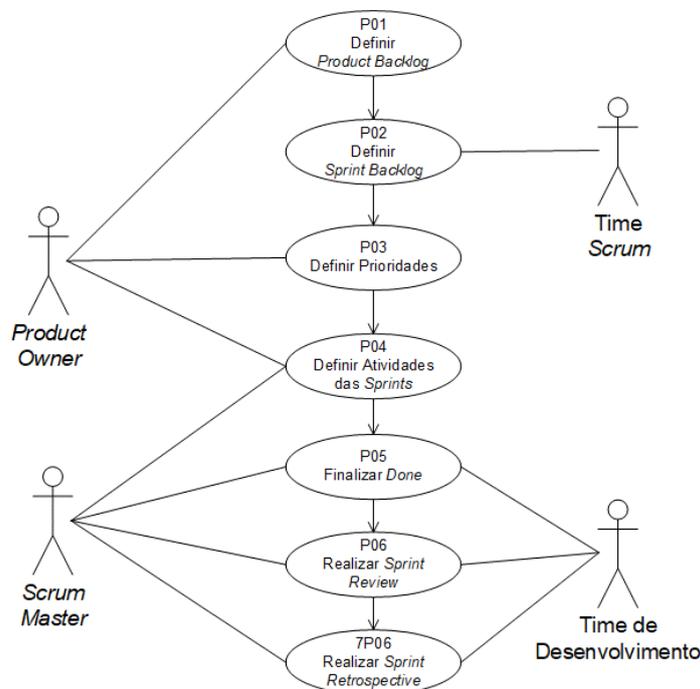
#	Requisito	Respondentes
REQ01	Verificação das pendências da <i>Sprint</i> anterior	R1; R2; R3; R5; R6; R8; R10; R11; R12; R13
REQ02	Decisão de cancelamento da <i>Sprint</i>	R4
REQ03	Decisão de cancelamento de uma atividade da <i>Sprint</i>	R4
REQ04	Definição de "PRONTO" da <i>Sprint</i>	R9

### Solução Algorítmica

A diagramação do sistema foi realizada por meio da UML com o software VISIO ([www.microsoft.com.br](http://www.microsoft.com.br)) com a representação dos Diagramas de Caso de Uso e de Atividades. O Diagrama de Caso de Uso apresentado na

Figura 2 contém os processos identificados, a partir da análise das respostas obtidas, com seus atores e relacionamentos. Os atores são o *Product Owner*, *Times Scrum*, *Time de Desenvolvimento* e *Scrum Master*.

Figura 2 - Diagrama de Caso de Uso Scrum



O ciclo é iniciado pelo processo de definir *Product Backlog* (P01) que consiste em uma lista de todas as atividades necessárias ao produto e que evolui à medida em que são feitas as entregas, sempre buscando deixá-lo apropriado, competitivo e útil. Apenas o *Product Owner* é responsável por gerenciar o *Product Backlog*, isso inclui, seu conteúdo, visibilidade e compreensão.

O processo de definir *Sprint Backlog* (P02) compreende os itens do *Product Backlog* selecionados para a *Sprint* e a forma como eles serão construídos durante a *Sprint* pelo Time *Scrum*, nesta fase o *Product Owner* ajuda a tornar claro os itens do *Product Backlog* selecionados e nas decisões conflituosas, é nessa fase também que o Time explica ao *Product Owner* e *Scrum Master* como pretendem trabalhar.

Em relação às prioridades, o processo definir prioridades (P03) determina, a partir da lista de tarefas da *Sprint*, que o Time *Scrum* e o *Scrum Master* irão estabelecer as prioridades de cada tarefa.

A escolha das atividades está presente no processo definir atividades da *Sprint* (P04) que, a partir das prioridades identificadas em cada tarefa a ser feita, são organizadas por ordem crescente de prioridade.

No processo de finalização do *Done* (P05) as atividades previstas na *Sprint* são finalizadas pelo time de desenvolvimento gerando o "Pronto".

Os processos de *Sprint Review* (P06) e *Restrospective Sprint* (P07) encerram o ciclo do *Scrum* com a verificação dos status de conclusão das tarefas e a criação de lições aprendidas feita pelo Time de desenvolvimento e *Scrum Master*.

A decisão de quais as atividades e sua priorização correspondem aos processos P02 a P04 e a solução algorítmica foi criada para solucionar esses processos. O diagrama de atividade apresentado na Figura 3 demonstram as ações a serem realizadas.

A tela do sistema, apresentada na Figura 4, indica a situação dos dados ao acessar a opção do **Gerador de Sprint** que indica a nova *Sprint*. No exemplo ao acessar a tela é atribuído o número "10" para a próxima *Sprint*, as atividades pendentes do ciclo anterior e as atividades novas selecionadas.

Ao clicar em **Gerar Prioridades**, será aberta a Figura 5, gerando algumas interações inerentes às atividades selecionadas quando necessitar de algum pré-requisito não especificado na lista anterior. Em seguida as atividades são transferidas para o grid à direita, indicando nome da atividade e grau de prioridade.

Caso alguma atividade precise ser cancelada, basta marcar o *Checkbox* ao lado esquerdo da atividade e clicar em Cancelar atividades. Por fim, o usuário deve clicar em **Gerar Sprint** para que os dados sejam efetivados.

Figura 3 – Diagrama de Atividades

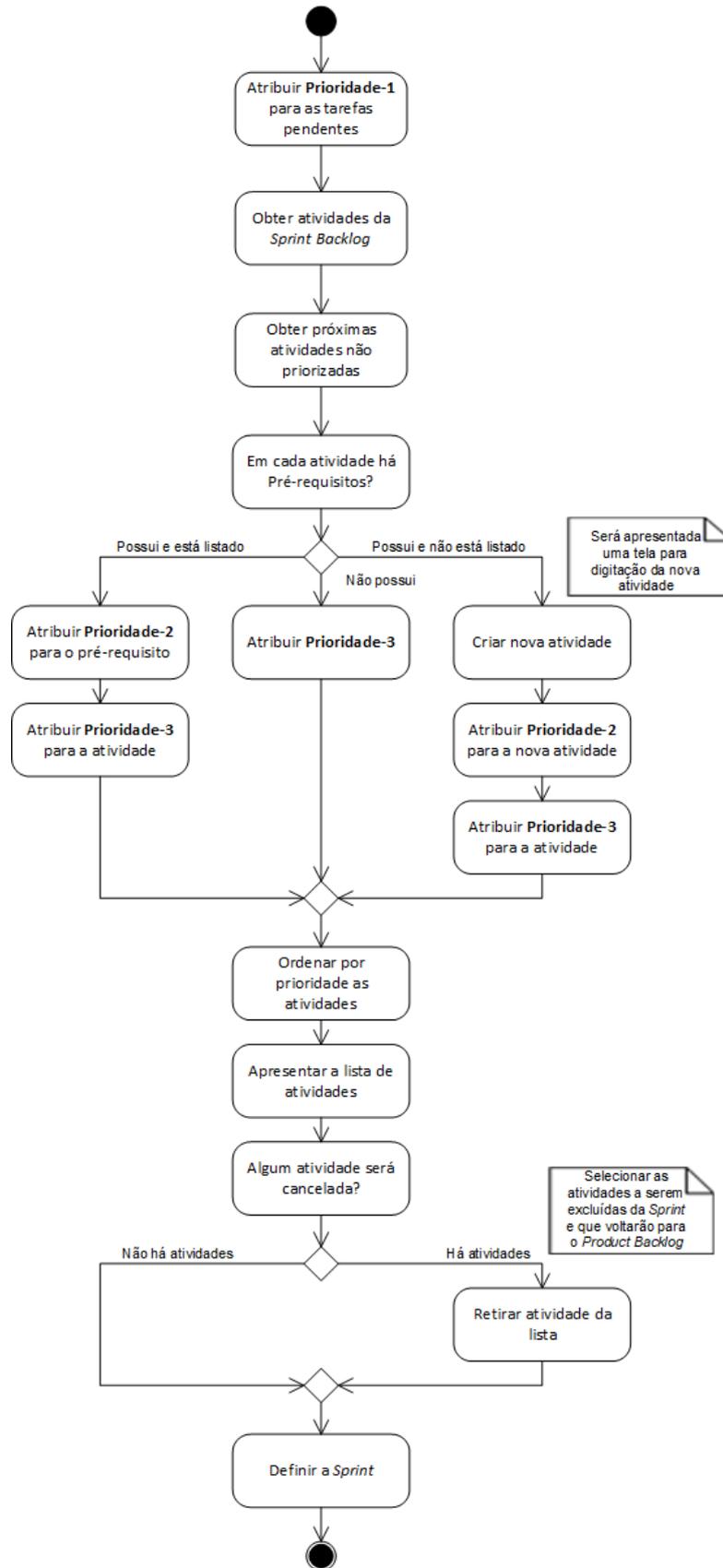
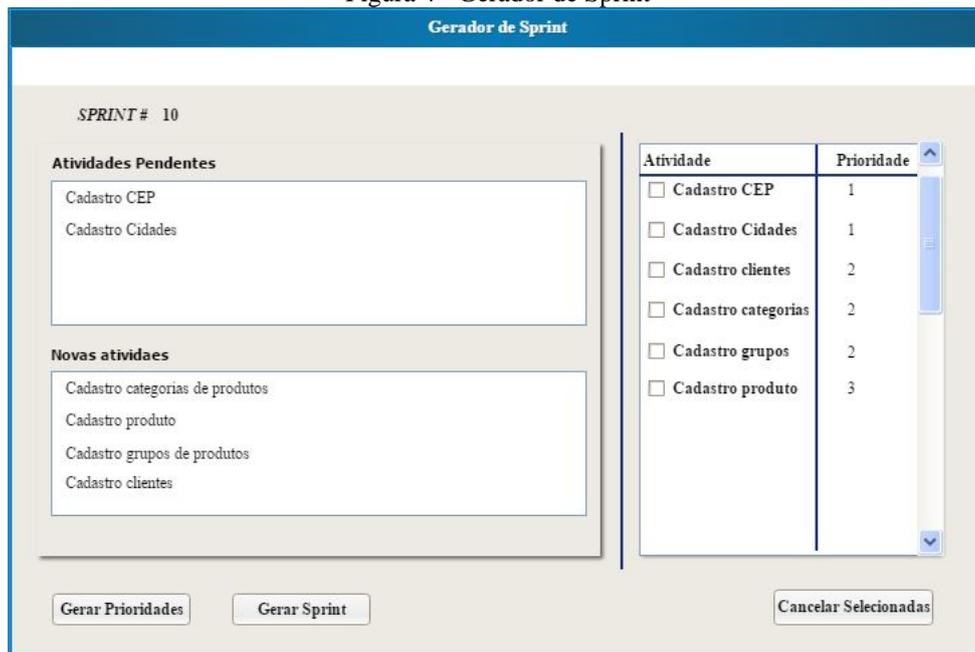
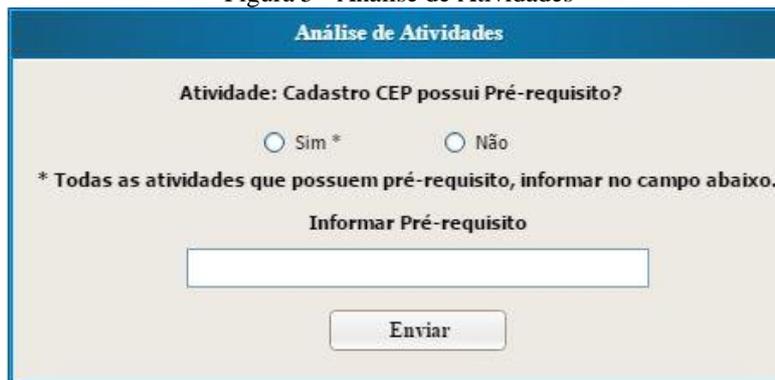


Figura 4 - Gerador de Sprint



Atividade	Prioridade
<input type="checkbox"/> Cadastro CEP	1
<input type="checkbox"/> Cadastro Cidades	1
<input type="checkbox"/> Cadastro clientes	2
<input type="checkbox"/> Cadastro categorias	2
<input type="checkbox"/> Cadastro grupos	2
<input type="checkbox"/> Cadastro produto	3

Figura 5 - Análise de Atividades



Atividade: Cadastro CEP possui Pré-requisito?

Sim \*     Não

\* Todas as atividades que possuem pré-requisito, informar no campo abaixo.

Informar Pré-requisito

Enviar

### Análise e Discussão dos Resultados

A solução computacional apresentada, juntamente com o algoritmo criado, corresponde a uma ferramenta que poderá corroborar com a escolha e criação das atividades, mas poderá ser efetiva de acordo com a experiência do time de desenvolvimento que estiver utilizando.

O algoritmo está inserido no diagrama de caso de uso nos processos definir prioridades (P3) e definir atividades das *Sprints* (P4), sendo os outros processos naturais ao ciclo de vida da metodologia ágil Scrum (Sutherland & Schwaber, 2013).

### Conclusões

O Scrum está em ascensão entre as empresas de desenvolvimento de software, devido à promessa de entregas rápidas, constantes e de qualidade pelo time de desenvolvimento e cada vez mais os times Scrum estão diante de desafios na escolha das atividades a serem realizadas.

O resultado desta pesquisa culminou em um algoritmo que propicia a análise das atividades das *Sprints* a serem criadas, bem como em um protótipo para ser implementado que irá corroborar com as equipes de desenvolvimento *Scrum*.

A limitação deste trabalho é relacionada à ausência do desenvolvimento do protótipo e, conseqüentemente, da sua implantação na Empresa-Alpha. As contribuições para a prática estão na apresentação de um protótipo com um algoritmo que pode auxiliar os gestores de projetos de TI na escolha de atividades para as *Sprints*. A contribuição teórica está na apresentação de formas alternativas de escolha das atividades que estão ausentes no *Scrum Guide*. Futuros trabalhos incluem o desenvolvimento do protótipo e medição do desempenho do algoritmo criado.

## Referências

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., et al. (2001). Manifesto for agile software development.

Buckl, S., Matthes, F., Monahov, I., Roth, S., Schulz, C., & Schweda, C. M. (2011). Towards an agile design of the enterprise architecture management function. *2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops* (p. 322–329). IEEE.

Dingsøyr, T., Dyba, T., & Moe, N. B. (2010). Agile software development: an introduction and overview. *Agile Software Development*, 1–13.

Gil, A. C. (2008). *Métodos e técnicas de pesquisa social* (6<sup>o</sup> ed). São Paulo: Atlas.

Glória Júnior, I., Oliveira, R., & Chaves, M. (2014). A Proposal for Using Web 2.0 Technologies in Scrum.

Kerzner, H. (2010). *Gestão de projetos: as melhores práticas*. Porto Alegre: Bookman.

OMG. (2015). *Object Management Group - Unified Modeling Language 2.5*. United States: OMG. Recuperado de <https://www.omg.org/spec/UML/2.5/PDF>

PMI. (2017). *Project Management Body of Knowledge Guide*. Pennsylvania: Four Campus Boulevard.

Pressman, R. (2014). *Software Engineering: A Practitioner's Approach* (8<sup>o</sup> ed). New York: McGraw-Hill Education.

Sato, D. T. (2007). Uso eficaz de métricas em métodos ágeis de desenvolvimento de software. *Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo*, 139.

Silveira, F. F., Russo, R. de F. M., Júnior, I. G., & Sbragia, R. (2018). Systematic Review of Risks in Domestic and Global IT Projects. *Journal of Global Information Management (JGIM)*, 26(1), 20–40.

Sommerville, I. (2015). *Software Engineering* (10<sup>o</sup> ed). India: Pearson.

Sutherland, J., & Schwaber, K. (2013). The definitive guide to scrum: The rules of the game. Recuperado de [www.scrum.org](http://www.scrum.org)

Theóphilo, C. R., & Martins, G. de A. (2009). Metodologia da investigação científica para ciências sociais aplicadas. *São Paulo: Atlas*, 2(104–119), 25.

Varaschim, J. D. (2009). Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet. *Monografia em Ciência da Computação do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro*.

Yin, R. K. (2015). *Estudo de caso: planejamento e métodos*. Porto Alegre: Bookman.

## **Apêndice A - Protocolo de Entrevistas**

O protocolo de entrevistas aplicado aos respondentes possui os seguintes passos:

Passo 1. Realizar a introdução sobre o pesquisador e a pesquisa a ser feita.

Passo 2. Descrever como será conduzida a entrevista

Passo 3. Ressaltar a preocupação com a confidencialidade e privacidade dos entrevistados.

Passo 4. Iniciar as questões:

(Q01). Há quanto tempo você trabalha com Scrum?

(Q02). As atividades a serem executadas em todo o sistema ficam em um repositório chamado Product Backlog. A partir do Product Backlog criado, como são escolhidas as atividades para as próximas 4 semanas (Sprint Backlog) que serão distribuídas em Sprints?

(Q03). Uma vez definida as atividades da Sprint Backlog, como são escolhidas as atividades e definidas a sequência de elaboração da Sprint?

(Q04). Ao final de uma Sprint, a atividade é repassada para a próxima Sprint? Ela recebe algum tratamento especial?

Passo 5. Finalizar a entrevista.

Para citar este artigo:

Oliveira, D., & Glória Júnior, I. (2019). Uma Solução Algorítmica para Projetos de TI Utilizando Metodologia Ágil Scrum: Um Estudo de Caso. *Revista IPTEC*, 7(1), 80-91.  
<https://doi.org/10.5585/iptec.v7i1.170>