

# Utilização do SCRUM no gerenciamento de projetos: Uma revisão da literatura

Glauber Andrade Botelho

## RESUMO

Nos dias atuais, é essencial um nível de produtividade alto para se manter no mercado. Nesse contexto, as metodologias ágeis propõem que a gestão de projetos seja realizada com enfoque nos indivíduos e interações, nas respostas rápidas à mudanças e na colaboração do cliente para o sucesso do projeto. O SCRUM é um framework baseado na metodologia ágil largamente utilizado em projetos de desenvolvimento de software. Neste trabalho, foi realizada uma revisão da literatura a fim de identificar quais os rumos tomados pelas pesquisas relacionadas ao SCRUM.

Palavras-chave: SCRUM, Metodologia Ágil, Gerenciamento de Projetos

## 1 INTRODUÇÃO

Devido ao avanço da tecnologia e da globalização, as relações de trabalho sofreram transformações no século XXI. O desenvolvimento tecnológico encurtou distâncias e tornou o relacionamento interpessoal mais rápido e dinâmico. Neste cenário, a competitividade entre as empresas atinge uma escala global. Diante desta situação, fica em evidência a importância do investimento na gestão de pessoas. A criticidade do gerenciamento de pessoas pode ser facilmente percebida quando se pensa no impacto que a perda de produtividade ou a não otimização de recursos pode ser determinante para o fracasso de uma organização (BRESSANO, 2011).

Apesar de ser o grande propulsor do encurtamento de distâncias e do aumento na dinâmica das relações interpessoais, o avanço tecnológico ainda esbarra em alguns obstáculos. Um desses obstáculos é a dificuldade encontrada no processo de desenvolvimento software. Pode-se enumerar diversos fatores que tornam o desenvolvimento de software algo complexo, dentre eles estão:

- analisar requisitos: a elicitação (ou levantamento) de requisitos é a fase onde a equipe responsável pelo desenvolvimento do sistema verifica, junto aos demais *stakeholders*<sup>1</sup>, quais as expectativas que se tem em relação ao software a ser desenvolvido. Exemplos de questões respondidas nesta etapa: o objetivo do software, o que deve estar presente no software, o que não deve estar presente no software. A dificuldade encontrada nesse ponto

---

1 Partes interessadas no projeto.

reside em dois fatores principais: i) muitas vezes os *stakeholders* ainda não tem certeza do que esperam do software; ii) a equipe responsável pelo desenvolvimento, por uma falha de comunicação, interpretam de forma equivocada os requisitos levantados.

- estimar o software: em qualquer projeto, não importa qual seja a sua natureza, é necessário que existam estimativas. Não se pode avaliar o que não pode ser mensurado de alguma maneira. Embora existam na Engenharia de Software técnicas e métricas que auxiliem na estimativa de características de um software, a realização dessa tarefa não é algo simples e requer, entre outras coisas maturidade e bom senso da equipe responsável pela execução do projeto.
- mudanças no decorrer da execução do projeto: esse ponto se relaciona diretamente ao levantamento de requisitos. Se a elicitação não é feita de forma correta no início do projeto, inevitavelmente vão ocorrer mudanças no escopo do projeto durante a sua execução e, por consequência, atrasos no cronograma do projeto.

A fim de minimizar as dificuldades encontradas no desenvolvimento de software, ocorreu, em 1968, a Conferência da OTAN sobre Engenharia de Software (*NATO Software Engineering Conference*). Esta conferência tinha como objetivo o estabelecimento de práticas que tornassem o desenvolvimento de software um processo melhor definido (GONÇALVES, 2004). Atualmente, existem diversos processos de desenvolvimento de software, entretanto, apesar do grande número, existem atividades que são fundamentais a todos esses processos (SOMMERVILLE, 2003):

- Especificação de Software: na especificação, são levantados os requisitos do software. É nesta fase que são definidas as funcionalidades e as restrições do software a ser desenvolvido.
- Projeto e Implementação de Software: na fase de projeto e implementação, o software é produzido seguindo as especificações obtidas na primeira fase do ciclo.
- Validação de Software: após a implementação do software, é realizada uma validação a fim de garantir que nenhum ponto especificado tenha sido

esquecido na fase de projeto e implementação.

- Evolução de Software: a fase de evolução visa a adequação do software a eventuais novas necessidades do cliente.

Mesmo com a existência de diversos processos de desenvolvimento de software, muitas empresas o desenvolvimento de software ocorre sem que seja utilizado processo algum:

Muitas organizações desenvolvem software sem usar nenhum processo. Geralmente isso ocorre porque os processos tradicionais não são adequados às realidades das organizações. Em particular, as organizações pequenas e médias não possuem recursos suficientes para adotar o uso de processos pesados. Por esta razão, muitas organizações não utilizam nenhum processo. O resultado desta falta de sistematização na produção de software é a baixa qualidade do produto final, além de dificultar a entrega do software nos prazos e custos pre definidos e inviabilizar a futura evolução do software. (SOARES, 2004)

Apesar de muitas empresas trabalharem sem utilizar processos de desenvolvimento de software, existem alternativas para empresas cujas realidades não são adequadas para o uso de processos pesados (orientados à documentação). Exemplos dessas alternativas são as metodologias ágeis.

Conforme apresentados em (SOARES, 2004), as metodologias ágeis são baseadas em conceitos dos Manifesto Ágil, os conceitos chave deste Manifesto são:

- Indivíduos e Interações ao invés de processos e ferramentas;
- Software executável ao invés de documentação;
- Colaboração do cliente ao invés de negociação de contratos;
- Respostas rápidas a mudanças ao invés de seguir planos.

Dentre as diversas metodologias ágeis de gestão de projetos, pode-se destacar o SCRUM, “que consiste na aplicação de práticas de gerenciamento durante o ciclo de produção (sprint), com o objetivo de promover ao time a execução do seu trabalho seguindo a filosofia do PDCA (*Plan, Do, Check, Act*)” (BRESSANO, 2011).

## **1.1 Objetivo Geral**

Executar protocolo de revisão sistemática a fim de conhecer o estado da arte da pesquisa relacionada ao framework SCRUM.

## **1.2 Objetivos Específicos**

Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar revisão bibliográfica sobre o framework SCRUM;
- Elaborar protocolo de revisão sistemática;
- Executar protocolo de revisão sistemática;
- Analisar os resultados obtidos.

## **2 REVISÃO DA LITERATURA**

As primeiras metodologias utilizadas no desenvolvimento de software focavam na documentação e planejamento do software antes da sua implementação. Essa preocupação era decorrente da dificuldade que se tinha para realizar modificações e correções em um sistema. Essa dificuldade é compreensível quando se considera as limitações dos primeiros computadores e a ausência de ferramentas como depuradores e analisadores de código (SOARES, 2004).

Dentre as metodologias tradicionais, o principal modelo de desenvolvimento é o Clássico (ou Cascata), nesse modelo, o desenvolvimento de software ocorre de maneira sequencial e sistêmica. A fase inicial do processo é o levantamento das necessidades do cliente, após isso, seguem-se as fases de planejamento, modelagem, construção, implantação e suporte (PRESSMAN, 2001). Na Figura 1 é apresentado um fluxograma do modelo Cascata.

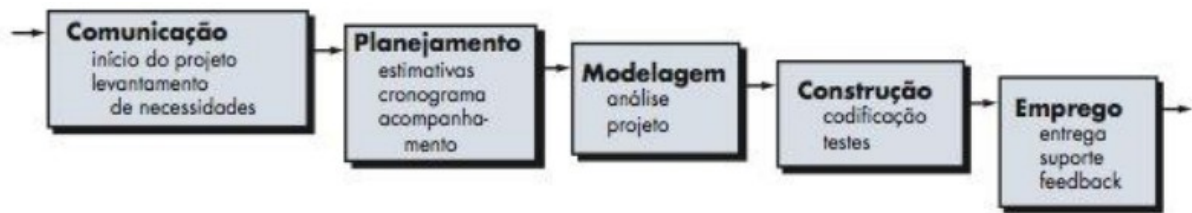


Figura 1: Fluxograma do Modelo Cascata (Extraído de (PRESSMAN, 2001))

Em (PRESSMAN, 2001), são apontados três problemas principais inerentes ao modelo Cascata, são eles:

- O fluxo proposto pelo modelo raramente é seguido em projetos reais;
- Existe uma dificuldade por parte do cliente para estabelecer com clareza, no início do projeto, os requisitos do sistema. Para que o modelo cascata funcione adequadamente, é necessário que os requisitos estejam bem definidos desde o início, visto que modificações no planejamento do software são custosas ao se trabalhar com esse processo de desenvolvimento;
- Não é liberada uma versão operacional do software até que o ciclo de desenvolvimento esteja próximo do seu fim. Por isso, é necessária uma paciência, muitas vezes inexistente por parte do cliente, para que ele possa ver um resultado concreto.

Além destes pontos, percebeu-se que, ao trabalhar com uma abordagem onde uma fase não pode ser iniciada até que a anterior seja concluída, o modelo Clássico conduz a estados de bloqueio, onde muitas vezes um profissional fica ocioso aguardando que um colega finalize as suas atribuições para que ele possa iniciar o seu trabalho. Considerando o dinamismo do mercado nos dias atuais, o ideal é que ociosidade dê lugar ao aumento de produtividade. Pensando nesse aumento de produtividade, muitas vezes são aplicadas metodologias ágeis no desenvolvimento de software.

## 2.1 Manifesto Ágil

As metodologias ágeis são baseadas no Manifesto Ágil. Segundo (SOARES, 2004), no Manifesto Ágil não há uma rejeição aos processos e ferramentas, à documentação, à negociação de contratos ou ao planejamento. O que ocorre é que

nas metodologias ágeis estes elementos ficam em segundo plano quando comparados aos indivíduos e às interações, ao fato do software estar ou não executável e às respostas rápidas às mudanças. Tudo isso faz com que as metodologias baseadas no Manifesto Ágil se adéquem melhor à realidade de pequenas e médias organizações. Os doze princípios do Manifesto Ágil<sup>2</sup> são apresentados a seguir:

- 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.
- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- 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.
- Software funcional é a medida primária de progresso.
- 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-

---

2 Disponível em <http://www.manifestoagil.com.br/principios.html>

organizáveis.

- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

### **2.1.2 SCRUM**

Dentre os diversos processos baseados no Manifesto Ágil está o SCRUM, um framework utilizado no gerenciamento de projetos complexos baseado em práticas que objetivam deixar tudo visível durante a execução do projeto. Desta forma, os membros da equipe sabem exatamente o que está acontecendo e fazem rapidamente os ajustes necessários para manter a execução do projeto na direção correta (SCHWABER, 2004). O SCRUM é baseado em iterações (Sprints). No início de cada iteração, a equipe verifica os requisitos a serem implementados, consideram os recursos disponíveis e avaliam as suas capacidades e habilidades. Após isso, coletivamente, a equipe determina como executar as tarefas escolhidas para aquela iteração, adaptando a sua forma de trabalho às dificuldades e surpresas que aparecem com o tempo. À medida que desafios inerentes à execução do projeto surgem, a equipe encontra maneiras de resolvê-los e escolhe o melhor caminho para fazer isso. Este processo criativo é responsável pelo nível de produtividade obtido com o uso do SCRUM (SCHWABER, 2004).

Para que o SCRUM funcione, são definidos três papéis principais: O dono do produto, a equipe e o ScrumMaster. O dono do produto é responsável por representar todos aqueles que tem interesse no projeto e nos seus resultados. A equipe é responsável por executar o projeto. A equipe é se gerencia e se organiza, e é responsável por encontrar maneiras de fazer com que os requisitos do projeto sejam cumpridos, além disso, a equipe é responsável por coordenar o seu próprio trabalho a fim de progredir na execução do projeto a cada iteração. O ScrumMaster é o responsável pelo processo do SCRUM, por ensinar o SCRUM a todos envolvidos no projeto, por implementar o SCRUM de tal forma que o processo seja adequado à cultura da organização e por garantir que o fluxo do SCRUM seja seguido. O fluxo do SCRUM é apresentado na Figura 2.

De acordo com (SCHWABER, 2004), cada Sprint é iniciada com o seu planejamento, onde o dono do produto e a equipe se reúnem para discutir o que

deve ser feito na próxima Sprint, as tarefas a serem executadas são selecionadas a partir do backlog do produto, o critério para escolher estas tarefas a prioridade de cada uma, o dono do produto diz à equipe o que é desejado para aquela Sprint e a equipe diz ao dono do produto o quanto do que é desejado é possível entregar na Sprint.

Depois de decidido o que será entregue, a equipe planeja a Sprint. No decorrer da Sprint, a equipe faz reuniões diárias que tem por objetivo manter a equipe informada sobre o andamento do projeto, nessas reuniões cada membro da equipe informa ao restante da equipe o que foi feito no dia anterior e qual o planejamento para o próximo dia. Além disso, nas reuniões diárias são informados possíveis elementos ou fatores que estejam impedindo o trabalho de um ou mais membros da equipe.

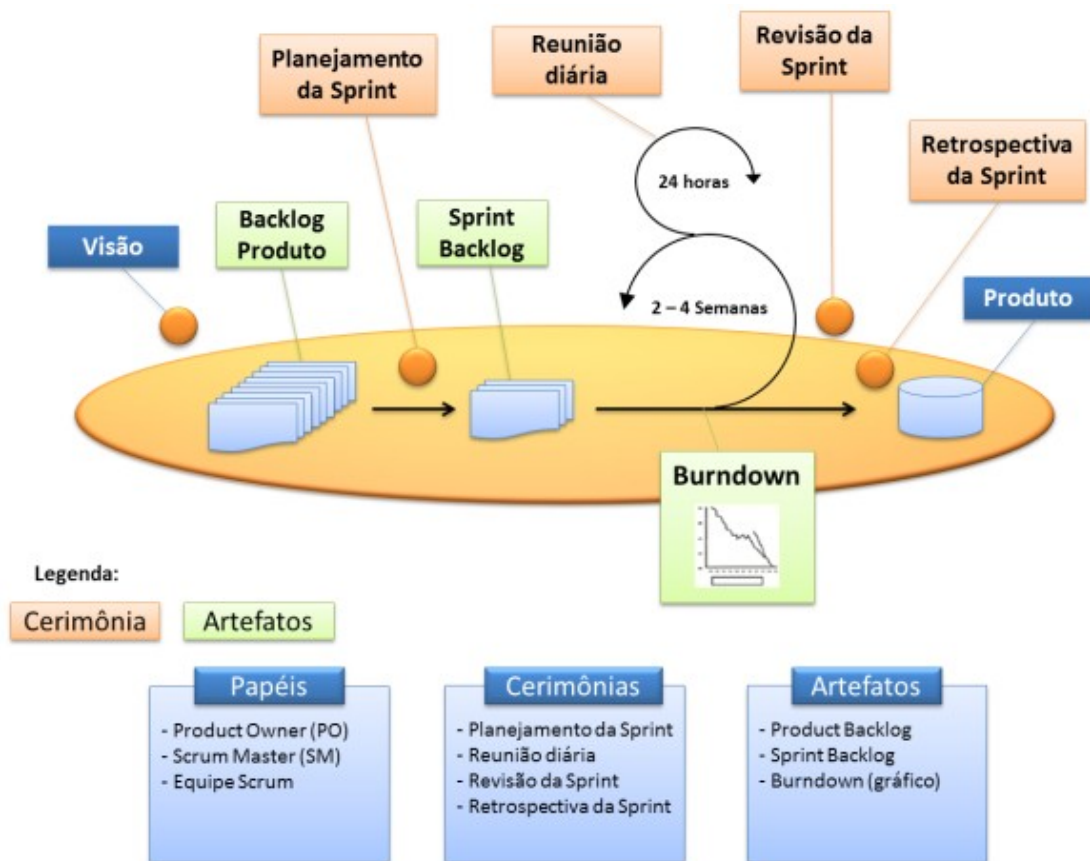


Figura 2: A engrenagem do SCRUM (ETTINGER, 2011)

Ao fim da Sprint, é realizada uma reunião de revisão da Sprint onde a equipe apresenta ao dono do produto o que foi realizado durante a Sprint. Após a reunião



de revisão, o ScrumMaster faz uma retrospectiva da Sprint, onde a equipe é encorajada a revisar os processos e práticas do SCRUM, a fim de ser mais eficaz na próxima Sprint.

### **3 METODOLOGIA**

Pesquisa teórica, descritiva e explicativa, bibliográfica. Nesta pesquisa foi executado um protocolo de revisão sistemática, com o objetivo de identificar para quais direções a pesquisa sobre o framework SCRUM estão sendo direcionadas. Após a execução do protocolo, os artigos encontrados foram analisados quantitativa e qualitativamente.

#### **3.1 Protocolo de Revisão Sistemática**

Nesta subseção é descrito o protocolo utilizado na realização da revisão da literatura, cujo objetivo foi identificar em quais direções seguem as pesquisas relacionadas ao framework SCRUM.

##### **3.1.1 Questão de pesquisa**

Foi definida uma pergunta a ser respondida na busca pelos trabalhos.

- Quais têm sido os esforços dos pesquisadores para desenvolver a pesquisa relacionada ao framework SCRUM?

##### **3.1.2 Critérios de inclusão e exclusão**

Abaixo são citados critérios de inclusão e exclusão dos trabalhos encontrados através das buscas. Estes critérios foram utilizados com o objetivo de evitar a seleção de trabalhos que apresentem pouca ou nenhuma contribuição para resolver a questão de pesquisa apresentada anteriormente.

- Só serão considerados na revisão artigos publicados entre 2000 e 2015;
- Outros trabalhos de revisão sistemática não serão considerados;

- Os trabalhos devem estar disponíveis gratuitamente na web;
- Os trabalhos devem estar escritos em Português, Inglês ou Espanhol;

### **3.1.3 Motores de busca**

O *IEEE Xplore Digital Library* foi escolhido como motor de busca, considerando a sua importância na área de computação, área esta onde o framework SCRUM é largamente utilizado. O *IEEE Xplore Digital Library* objetiva a busca de periódicos, jornais e conferências do IEEE.

O Portal de Periódicos da Coordenação de aperfeiçoamento de Pessoal de Nível Superior (CAPES) é uma biblioteca virtual que reúne e disponibiliza a instituições de ensino e pesquisa no Brasil publicações relacionadas a pesquisa das mais diversas áreas do conhecimento.

### **3.1.4 Termo de busca**

Foi utilizada a ferramenta de busca avançada do IEEE Xplore Digital Library e a pesquisa foi feita considerando os metadados dos documentos (título, palavras chave, etc). No portal Periódicos CAPES foi realizada uma busca simples por tema, e os resultados foram restringidos ao tópico *Computer Science*. Para conduzir a pesquisa, foi utilizado o seguinte termo de busca:

- S1: SCRUM

## **4 RESULTADOS E DISCUSSÃO**

Após a aplicação dos métodos de seleção, previamente definidos no protocolo apresentado na subseção 3.1, foram selecionados cinco artigos. Na Tabela 1, podem ser vistas as filtragens utilizadas na seleção dos artigos. Na Tabela 1, na linha do cabeçalho, a coluna Métodos do protocolo indica as fases de execução do protocolo, a coluna S1 indica a quantidade de artigos encontrados em cada fase do protocolo quando o termo de busca S1 foi utilizado.

**Tabela 1** – Resultados da quantidade de artigos selecionados pelos métodos do protocolo

<b>Métodos do protocolo</b>	<b>S1</b>
Após a consulta do termo de busca	26
Após critérios de inclusão e exclusão	25
Após leitura do resumo	5

Os artigos selecionados foram (MALLER; OCHOA e SILVA, 2005), (SUTHERLAND; HARRISON e RIDDLE, 2014), (BEHM *et al*, 2013), (TOMANEK e KLIMA, 2015) e (QURASHI e QURESHI, 2014).

Em (MALLER; OCHOA e SILVA, 2005), os autores propõem o uso de uma metodologia ágil em um ambiente regido pelo Modelo de Capacidade e Maturidade (CMM, do inglês *Capacity and Maturity Model*). O CMM é um modelo que descreve princípios e práticas de processos de software amadurecidos, dentre os benefícios da sua aplicação estão o aumento da qualidade do produto e redução considerável dos tempos de entrega e dos custos do projeto. Os autores viram na GSG Motorola Argentina (empresa avaliada como CMM Nível 5) a oportunidade de experimentar a aplicação de metodologias ágeis em projetos onde os prazos de entrega são curtos e os requisitos são instáveis ou mudam com frequência. Os autores concluem que a metodologia Xp@SCRUM tem muitas propriedades que facilitam a sua integração dentro de uma organização que alcançou o nível 4 ou 5 do CMM. Segundo os autores, não existe nenhuma incompatibilidade entre o Xp@SCRUM e os níveis 4 e 5 do CMM, além disso, destaca-se que as propriedades do Xp@SCRUM podem ser aplicadas a um nível organizacional, permitindo que os processos e práticas ágeis estejam disponíveis em toda a organização, como determina o CMM.

Em (SUTHERLAND; HARRISON e RIDDLE, 2014), um padrão SCRUM é uma solução reutilizável para problemas que ocorrem comumente ao utilizar o framework SCRUM. Os autores ressaltam que, apesar de ter uma estrutura simples e ser desenvolvido com o objetivo de ajudar as equipes, o SCRUM não resolve todos os problemas. No trabalho, 9 padrões SCRUM são elencados, juntos, estes padrões compõem o que os autores chamam de Linguagem de Padrões para Equipes Hyper-produtivas. Os autores apontam que a implementação e a execução dos 9 padrões aumentam drasticamente a habilidade da equipe de terminar uma Sprint mais cedo e, por consequência, permite que a equipe assuma mais itens do

backlog do produto, aumentando, assim, a velocidade da equipe. Além disso, ressalta-se que finalizar uma Sprint em um tempo mais curto, faz com que a equipe se sinta mais confiante sobre a sua capacidade de cumprir as Sprints.

Em (BEHM *et al*, 2013), os autores buscam verificar como as práticas de comunicação são exploradas em um projeto onde é feito uso do SCRUM e a equipe se encontra distribuída geograficamente. Com o avanço da tecnologia e das ferramentas de comunicação, o Desenvolvimento de Software Global é uma prática cada vez mais comum. Nesse cenário, existem dois pontos-chave a serem observados para que a comunicação ocorra bem: o primeiro ponto é que a comunicação presencial é limitada, o segundo ponto é que se deve ter atenção aos contrastes culturais, a depender do grau de diversidade existente na equipe. Os autores observaram uma equipe distribuída em três locais diferentes e concluíram que os membros da equipe sempre trabalhavam mais às vésperas de uma reunião de equipe e que o desempenho da equipe cresceu à medida que o número de e-mails enviados foi reduzido.

Em (TOMANEK e KLIMA, 2015), os autores propõem o acoplamento de rotinas de teste de software ao fluxo normal do SCRUM. Como o desenvolvimento de software com o SCRUM é realizado de forma incremental, caso os testes sejam automatizados, os autores sugerem a implementação dos requisitos de segurança e dos requisitos necessários para os testes durante o início do desenvolvimento e, a partir daí, selecionar e priorizar os demais requisitos do backlog, que poderão ser testados continuamente nos próximos sprints. Se os testes forem realizados de forma manual, a sugestão dos autores é que os testes sejam realizados não a cada sprint, mas apenas quando o dono do produto decidir publicar o software em um ambiente de produção.

Em (QURASHI e QURESHI, 2014), o foco dos autores são equipes grandes. Sob o argumento de que o framework SCRUM não foi projetado para ser aplicado junto a equipes numerosas, os autores propõem uma abordagem chamada SCRUM de SCRUMs. Nessa abordagem, a equipe é dividida em equipes menores, nas quais o SCRUM poderia ser aplicado adequadamente. Para que a técnica funcione a contento é necessário que a comunicação entre as equipes seja otimizada, é necessário que a dependência entre as equipes seja reduzida e que o sistema

funcione após a integração das partes produzidas por cada equipe.

## **5 CONCLUSÕES**

Neste trabalho, foi realizada uma revisão da literatura visando identificar em quais direções as pesquisas sobre o framework SCRUM – framework baseado na metodologia ágil e largamente utilizado no desenvolvimento de software – vêm sendo desenvolvidas. A fim de atingir o objetivo, primeiramente, foi realizada uma revisão bibliográfica sobre o SCRUM; após isso, foi elaborado um protocolo de revisão sistemática utilizado para guiar a revisão. Após a execução do protocolo, foram encontrados cinco artigos, em um deles, o esforço é concentrado em verificar a adequabilidade do SCRUM em um ambiente CMM; em outro artigo, o foco é aumentar a produtividade de uma equipe que já utiliza o framework SCRUM; no terceiro artigo analisado, os autores buscam verificar como as práticas de comunicação são exploradas em um projeto onde o SCRUM é utilizado por uma equipe geograficamente distribuída; os autores do penúltimo artigo analisado propõem o acoplamento de rotinas de teste de software ao fluxo normal do SCRUM; por fim, o último artigo é focado na utilização do SCRUM por equipes numerosas.

Ao fim da realização deste trabalho, ficou clara a diversidade de caminhos através nos quais podem ser desenvolvidas pesquisas relacionadas ao SCRUM, percebe-se que existem pesquisas voltadas tanto para a adição do SCRUM em modelos já consolidados presentes nas organizações, assim como existem pesquisas que objetivam investigar pontos que podem ser melhorados no SCRUM para que o framework possa ser aplicado em cenários onde originalmente os resultados dessa aplicação não seriam satisfatórios. Com a realização desta pesquisa, foi possível perceber em quais direções caminham as pesquisas relacionadas ao SCRUM, entretanto, apesar de atingir o objetivo geral do trabalho, é importante salientar que a revisão sistemática foi realizada utilizando apenas duas bases de dados. Diante disso, deve-se considerar que a quantidade de trabalhos selecionados não representa uma parcela significativa, se considerada a quantidade total de trabalhos referentes ao framework SCRUM encontrados na literatura. Entretanto, mesmo considerando apenas o IEEE Xplore Digital Library e o portal

Periódicos CAPES na revisão sistemática, é perceptível a preocupação que se tem em melhorar ainda mais os resultados obtidos pelo SCRUM.

No futuro, pode-se realizar uma revisão sistemática onde são consideradas mais fontes de busca e, assim, perceber em quais outros cenários a pesquisa sobre o SCRUM vem evoluindo.

## **ABSTRACT**

Nowadays, it is essential a high level of productivity to stay in the market. In this context, agile methodologies propose that project management is carried out with a focus on individuals and interactions, in quick response to changes and customer collaboration for project success. The Scrum is a framework based on agile methodology widely used in software development projects. In this work, a literature review was conducted to identify the direction taken by research related to SCRUM.

Keywords: SCRUM, Agile Methodology, Project Management

## **REFERÊNCIAS**

BEHM, Benjamin et al. Communication Practices in a Distributed Scrum Project. *arXiv preprint arXiv:1308.2260*, 2013.

BRESSANO, A. Como Formar Equipes de Alto Desempenho com AGILE. *Congresso Nacional de Gerenciamento de Projetos (2011)*. 2011.

ETTINGER, D. *A Engrenagem do Scrum*. Disponível em <http://danielettinger.com/2011/04/06/a-engrenagem-do-scrum/>. Acessado em 13 de Agosto de 2016.

GONÇALVES, A. *A Crise do Software*. Disponível em <http://www.portaleducacao.com.br/informatica/artigos/55859/a-crise-do-software>. Acessado em 13 de Agosto de 2016.

MALLER, P; OCHOA, C.; SILVA, J. Agilizando el Proceso de Producción de Software en un Entorno CMM de Nivel 5. *IEEE Latin America Transactions*, v. 3, n. 1, 2005.

PRESSMAN, R. *Engenharia de Software*. McGraw-Hill, 2001.

QURASHI, Saja Al; QURESHI, M. Scrum of scrums solution for large size teams using scrum methodology. *arXiv preprint arXiv:1408.6142*, 2014.

SCHWABER, K. *Agile Software Development with Scrum*. Editora Microsoft Press. 192P, 2004.

SOARES, M. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. *INFOCOMP (UFLA. Impresso)* , v. 3, p. 8-13, 2004.

SOMMERVILLE, I. *Engenharia de Software*. Editora Addison-Wesley. 592p, 2003.

SUTHERLAND, J; HARRISON, N; RIDDLE, J. Teams that Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams. *47th Hawaii International Conference on System Science*. 2014.

TOMANEK, Martin; KLIMA, Tomas. Penetration Testing in Agile Software Development Projects. *arXiv preprint arXiv:1504.00942*, 2015.