

CAS 2.0: Evolução e Automação do Checklist de Avaliação do Scrum para Projetos de Software

Thiago F. V. da Cunha^{1,3}, Ismayle de S. Santos^{2,3,a}, Alysson A. de Macedo^{2,3}, Alberto H. M. T. Monteiro³, Bruno S. Aragão^{2,3}, Rossana M. C. Andrade^{2,3,b}

¹ Dataprev
Fortaleza, Ceará, Brasil

²Departamento de Computação - Universidade Federal do Ceará (UFC)
Fortaleza, Ceará, Brasil

³Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Fortaleza, Ceará, Brasil

{thiagoferraz,ismaylesantos,alyssonmacedo,bruno,rossana}@great.ufc.br

Resumo. *Este trabalho propõe uma evolução e automação de um checklist já existente, denominado Checklist de Avaliação do Scrum (CAS), componente do Agile DMAIC, que é um método para avaliar e melhorar o uso do Scrum em projetos de software. Com o CAS, é possível avaliar a abrangência do Scrum nos projetos e identificar melhorias. A evolução apresentada neste artigo visa ampliar a inspeção de princípios e práticas ágeis, e foi realizada de forma sistemática. A automação possibilita que o CAS consolide a visão das equipes sobre o Scrum e forneça resultados em tempo real, a partir de um preenchimento online. Para avaliar o CAS 2.0, ele foi aplicado em quinze projetos reais e seus resultados são também discutidos neste artigo.*

Abstract. *This paper proposes an evolution and automation of an existing checklist, called Scrum Assessment Checklist (CAS), component of Agile DMAIC, which is a method to evaluate and improve the use of Scrum in software projects. With CAS, one can assess the Scrum coverage in projects and identify improvements. The evolution presented in this article aims to increase the inspection of agile principles and practices, and was carried out systematically. The automation enables the CAS to consolidate teams' vision about the Scrum and provide real-time results from an online filling. To evaluate the CAS 2.0, it was applied in fifteen real projects and their results are also discussed in this article.*

1. Introdução

As metodologias ágeis objetivam acelerar o desenvolvimento de *software* a partir da melhoria contínua dos processos, gerando benefícios na comunicação e interação da equipe, organização, diminuição de falhas, respostas mais rápidas às mudanças e aumento da produtividade do time de desenvolvimento [Schwaber 2007].

^a Bolsista de doutorado da CAPES

^b Bolsista do CNPq de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora (DT) 2

O *Scrum* [Sutherland e Schwaber 2016] é uma das metodologias ágeis mais utilizadas no desenvolvimento de software e tem contribuído na melhoria da gestão dos projetos, aumentando a probabilidade de sucesso das equipes [VersionOne 2016]. Criada em 1996, essa metodologia adota uma abordagem empírica e pode ser utilizada em projetos complexos com, por exemplo, uma grande quantidade de produtos diferentes, uso simultâneo de várias tecnologias, muitas restrições tecnológicas e incertezas associadas [Melo et al. 2012]. Entretanto, para que os projetos usufruam dos principais benefícios do *Scrum*, como a melhoria da produtividade e qualidade dos produtos desenvolvidos, ele precisa ser adequadamente instanciado de acordo com o contexto de cada projeto, organização e cliente. Além disso, essa instanciação precisa evoluir (e.g., revisar e ajustar os tamanhos da equipe e do *sprint*, os papéis dos envolvidos, a forma como a equipe realiza os eventos e utiliza os artefatos do *Scrum*) durante a execução dos projetos. Surge, portanto, a necessidade de gerenciar a adoção ou uso do *Scrum* para a evolução contínua e controlada do processo ágil.

Nesse cenário, o *Agile DMAIC* [Cunha e Andrade 2014a] é um método que possibilita às equipes gerenciarem a adoção ou uso do *Scrum*, promovendo uma gestão quantitativa do uso de seus princípios e práticas, bem como a identificação e o tratamento de causas de problemas para a melhoria contínua da agilidade.

Um dos componentes do *Agile DMAIC* é o *Checklist* de Avaliação do *Scrum* (CAS), utilizado para coletar a percepção das equipes sobre o uso do *Scrum*. Sua primeira versão, elaborada com base apenas no *Nokia Test* [Vode e Sutherland 2008] e no *Scrum Checklist* [Kniberg 2009], não abrange algumas práticas importantes do *Scrum* (e.g., definição dos objetivos do *sprint*, revisão e atualização da definição de “pronto”, atenção do time à qualidade do *software* desenvolvido) que estão no Guia do *Scrum* [Sutherland e Schwaber 2016], uma das principais referências sobre essa metodologia. Além disso, para facilitar o reuso do CAS, ele precisa ser automatizado, permitindo o seu preenchimento *online*.

Este trabalho propõe, então, uma evolução e automação do *Checklist* de Avaliação do *Scrum* (CAS) [Cunha e Andrade 2014a], componente do *Agile DMAIC*. Essa evolução visa ampliar a inspeção de princípios e práticas ágeis, e foi realizada de forma sistemática, considerando o Guia do *Scrum* e outros sete *checklists* de avaliação do uso do *Scrum*. A automação possibilita que o CAS consolide a visão das equipes sobre o *Scrum* e forneça resultados em tempo real, a partir de um preenchimento *online*. Para avaliar a nova versão do CAS, ela foi aplicada em quinze projetos reais e seus resultados, que também são discutidos neste artigo, foram positivos.

Para apresentar o CAS 2.0, este artigo está estruturado em mais seis seções. Os trabalhos relacionados ao uso de *checklists* de avaliação do *Scrum* são discutidos na Seção 2. O *Agile DMAIC* é apresentado na Seção 3 e a Seção 4 descreve o processo de elaboração do novo *checklist*. A Seção 5 apresenta o CAS 2.0, detalhando o seu objetivo, público alvo, conteúdo e a forma adequada de sua utilização, bem como um comparativo das versões desse *checklist* e a arquitetura de sua automação. Os resultados da aplicação do CAS 2.0 em projetos reais são discutidos na Seção 6 e, finalmente, na Seção 7, são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Para encontrar os trabalhos relacionados foi realizada uma busca na literatura por publicações sobre os seguintes tópicos: (i) *checklists* de avaliação do *Scrum*; e (ii) conformidade de processos de desenvolvimento de *software*. Como locais de busca, foram utilizadas as bases ACM DL Library¹, *Web of Science*², *Scopus*³ e os anais do Simpósio Brasileiro de Qualidade de Software (SBQS). Adicionalmente, o Google⁴ e Google Scholar⁵.

A maioria das ferramentas de avaliação de processos ágeis fornece um *checklist* de avaliação e uma análise das respostas de seu preenchimento, e foca em uma ou mais metodologias ágeis [Jalali et al. 2014]. Nas buscas por trabalhos relacionados à avaliação do *Scrum*, foram identificados dois trabalhos [Park et al. 2016, Cunha e Andrade 2014a] que propõem métodos de avaliação do *Scrum* que utilizam *checklists*.

Cunha e Andrade [Cunha e Andrade 2014a] apresentam o *Agile DMAIC* (descrito na Seção 3), um método que possibilita avaliar e melhorar o uso do *Scrum*, fornecendo medições qualitativas e quantitativas de sua aplicação nos projetos. Para isso, os autores propõem a primeira versão do *CAS*, um *checklist* elaborado a partir do *Nokia Test* [Vode e Sutherland 2008] e do *Scrum Checklist* [Kniberg 2009]. Park et al. [Park et al. 2016] descrevem o *Scrum* através do *Essence*, um padrão para elaboração e melhoria de métodos de engenharia de *software*. Os autores propõem mapeamentos entre as categorias do *Essence* e os artefatos, eventos e papéis do *Scrum*. Através desses mapeamentos, são especificados *checklists* para avaliar o progresso dos projetos. Uma desvantagem é a necessidade de entendimento do *Essence*, mesmo que superficial. Além disso, apenas um dos *checklists* é apresentado (sobre o planejamento de *releases*).

Quanto aos *checklists* publicados em páginas *online*, foram identificados os seguintes: (i) *checklist* da firma de consultoria *Total Programme Control* [TPC 2016]; (ii) *checklist* de Maharana [Maharana 2015]; (iii) *checklist* de Lomonaco e Lapointe [Lomonaco e Lapointe 2013]; (iv) *InfoQ Scrum Checklist* [Gloger 2012]; (v) *Scrum Checklist* [Kniberg 2009]; e (vi) *Nokia Test* [Vode e Sutherland 2008].

O comparativo dos *checklists* supracitados é apresentado na Tabela 1. A maioria dos *checklists* possui uma baixa cobertura (abrangência) dos princípios e práticas do *Scrum* [TPC 2016], [Maharana 2015], [Lomonaco e Lapointe 2013], [Gloger 2012] e [Vode e Sutherland 2008]). Contudo, as maiores limitações dizem respeito a consolidação do preenchimento pela equipe e a apresentação de estudos de caso avaliando os *checklists*, identificados apenas no *CAS* [Cunha e Andrade 2014a], e a exibição de resultados em tempo real, implementada por apenas três *checklists* ([TPC 2016], [Lomonaco e Lapointe 2013], e [Vode e Sutherland 2008]).

Além disso, existem trabalhos que não apresentam um *checklist*, mas estão relacionados à detecção de problemas em processos. Zazworka et al. [Zazworka et al. 2010] apresentam uma abordagem genérica para detectar não conformidades com o processo, baseada em dados fornecidos pelo analista de conformidade do processo e

¹ <http://dl.acm.org/>

² <https://webofknowledge.com/>

³ <https://www.scopus.com/>

⁴ <https://google.com/>

⁵ <http://scholar.google.com.br/>

informações capturadas por um sistema de controle de versão. Khelladi *et al.* [Khelladi et al. 2015] apresentam um conjunto de sete restrições formalizando as melhores práticas do *Scrum*. Essas restrições podem ser verificadas em um modelo de processo utilizado. Perkusich *et al.* [Perkusich et al. 2015] apresentam um processo de avaliação da aplicação do *Scrum* com redes *Bayesianas*, que possuem como entradas métricas e evidências sobre o uso do *Scrum*. A compilação dessas entradas, na rede *Bayesiana* definida, permite identificar as deficiências para reflexão da equipe e definição de planos de ações durante as retrospectivas dos *sprints*.

Tabela 1. Comparativo dos Checklists

Checklists	Abrangência	Resultado em Tempo Real	Consolidação do Preenchimento	Estudo de Caso
TCP 2016	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Maharana 2015	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cunha e Andrade 2014a (CAS)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Lomonaco e Lapointe 2013	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gloger 2012 (InfoQ Scrum Checklist)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kninberg 2009 (Scrum Checklist)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vode e Sutherland 2008 (Nokia Test)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/> Atende	<input checked="" type="checkbox"/> Atende parcialmente	<input type="checkbox"/> Não atende	

As principais desvantagens dessas abordagens são a cobertura dos elementos do *Scrum*, que é baixa no trabalho de Khelladi *et al.*, e o esforço e conhecimento adicional para realizar a verificação da conformidade com o processo, como é o caso dos trabalhos de Zazworka *et al.* e Perkusich *et al.*.

Nesse contexto, este artigo aborda essas lacunas existentes, propondo o CAS 2.0, que é uma evolução do *checklist* apresentado em [Cunha e Andrade 2014a]. Essa nova versão foi elaborada de forma sistemática, considerando o Guia do *Scrum* e os *checklists* elencados nesta seção. O processo de elaboração do CAS 2.0, o CAS 2.0 e a avaliação de sua aplicação em projetos reais são apresentados e discutidos nas seções 4, 5 e 6, respectivamente.

3. Agile DMAIC

Agile DMAIC [Cunha e Andrade 2014a] é um método para avaliação e melhoria do uso de princípios e práticas do *Scrum*, que pode ser usado por equipes iniciantes ou experientes, na adoção e na melhoria da utilização dessa metodologia.

As principais contribuições do *Agile DMAIC* são: (i) possibilitar uma gestão quantitativa do uso do *Scrum*; (ii) avaliar o uso de seus princípios e práticas; e (iii) identificar e tratar as causas de problemas, considerando o contexto e a necessidade dos projetos e clientes.

O *Agile DMAIC* utiliza princípios e técnicas do *Lean Six Sigma* [Radujkovic et al. 2014] e foca na execução, de forma iterativa e incremental, de uma simplificação do ciclo de melhoria *DMAIC*. Essa execução é realizada em paralelo ao uso do *Scrum* e, à medida que os problemas são identificados e tratados, as melhorias são implantadas a cada *sprint*.

O *Agile DMAIC* é composto de três componentes (veja a Figura 1): (i) *DMAIC*, voltado à melhoria de processos, é utilizado na definição de metas, identificação e análise

de causas, implantação e controle de melhorias; (ii) *CAS*, *checklist* utilizado para coletar a percepção das equipes quanto ao uso de princípios e práticas do *Scrum*; e (iii) Índice de Medição do Nível do *Scrum* (*IMS*), que avalia a adoção das práticas do *Scrum*.

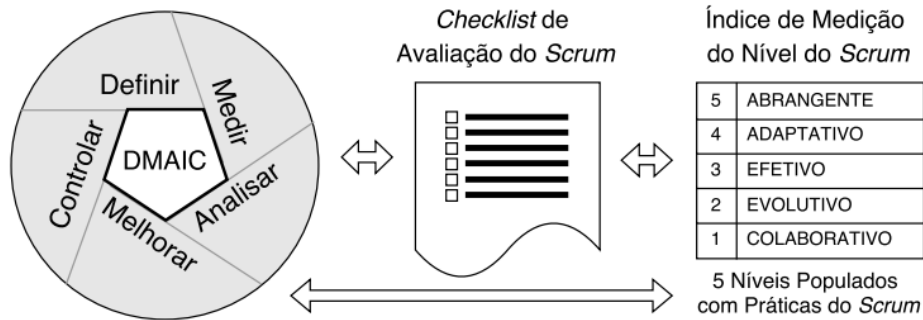


Figura 1. Componentes do Agile DMAIC.

O *Agile DMAIC* se baseia em coletar periodicamente a percepção da equipe sobre o uso do *Scrum* e essa percepção é utilizada na identificação de problemas e na classificação do nível do *Scrum* no projeto. Em seguida, é necessário analisar as causas, definir e implantar ações de melhoria, antecipando os benefícios do *Scrum*. Mais detalhes sobre o *Agile DMAIC* podem ser encontrados em [Cunha e Andrade 2014b].

4. Processo de Elaboração do CAS 2.0

O processo de elaboração da nova versão do *CAS* é apresentado na Figura 2. Esse processo é cíclico, possui quatro etapas e inicia com a identificação e seleção de artefatos sobre o *Scrum*, os quais são os *checklists* elencados na Seção 2 e o Guia do *Scrum*.

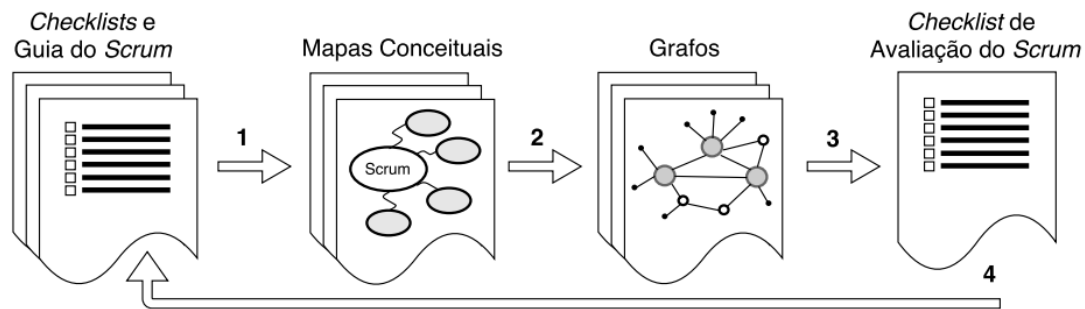


Figura 2. Processo de Elaboração do Checklist de Avaliação do Scrum.

A elaboração de mapas conceituais para os *checklists* e para o Guia do *Scrum* é realizada na etapa 1. Esses mapas são representações gráficas da organização de conceitos e regras, em sua estrutura cognitiva, e identificam e relacionam os elementos do *Scrum* (e.g. artefatos, papéis e eventos). Uma vez que esses mapas podem ser definidos como grafos (com vértices representando os elementos do *Scrum* e arestas, as relações existentes entre esses elementos), a comparação dos conceitos e regras do *Scrum* entre os artefatos pode ser vista como um problema de correspondência de grafos, no qual podem ser utilizadas métricas para medição de suas similaridades [De Souza et al. 2008].

A elaboração dos mapas é realizada por especialistas no *Scrum* e o procedimento manual utilizado é apresentado no Algoritmo 1. Como entradas para esse algoritmo têm-se o artefato a ser analisado e os mapas conceituais já criados. O primeiro passo é a análise

do artefato para identificar as regras e os conceitos sobre o *Scrum*. Se o elemento identificado é uma regra, então ela é documentada em uma planilha junto às entidades associadas a ela. Se a regra já foi analisada antes, seu identificador é o mesmo atribuído anteriormente; do contrário um novo identificador é gerado. Depois disso, a regra é adicionada ao mapa conceitual. Se o elemento for um conceito, ele é simplesmente adicionado ao mapa conceitual.

Algoritmo 1: ELABORAR MAPA CONCEITUAL

Função *ElaboraMapaConceitual* (*artefato*, *mapas*)

```
mapaConceitual = {}  
para cada regra do Scrum ∈ artefato faça  
  r.descricao = regra  
  r.entidades = obterEntidades(regra)  
  se regra == mapa1.r1.descricao ∧ mapa1 ∈ mapas então  
    | r.id = mapa1.r1.id  
  senão  
    | r.id = novo id  
  mapaConceitual.regras = mapaConceitual.regras ∪ r  
para cada conceito do Scrum ∈ artefato faça  
  | mapaConceitual.conceitos = mapaConceitual.conceitos ∪ conceito  
mapas = mapas ∪ mapaConceitual  
retorna mapaConceitual
```

Para cada um dos mapas, documentados em planilhas eletrônicas, são gerados grafos, considerando que: (i) os nós correspondem às entidades; e (ii) as arestas relacionam essas entidades a partir das regras identificadas. A utilização desses grafos é importante para auxiliar na análise e comparação dos artefatos, reduzindo a propensão a falhas durante o processo de elaboração do novo *checklist*.

Em seguida, são realizadas operações de interseção e subtração dos grafos para uma análise das semelhanças e diferenças entre cada artefato. Essa análise é realizada por especialistas no *Scrum*.

Um exemplo de um grafo de interseção entre o Guia do *Scrum* e a nova versão do CAS é apresentado na Figura 3. Os nós representam os elementos do *Scrum* e as arestas contêm os identificadores das regras que relacionam esses elementos. O tamanho do nó é proporcional à quantidade de regras associadas a ele. A nova versão do CAS, apresentada na Seção 5, pode ser consultada para a identificação das regras referenciadas nessa figura, considerando que cada identificador corresponde a um item do *checklist* (e.g., 45 identifica a regra expressa pelo item “[Reunião de Revisão] 45. Feedback é recebido do Product Owner” e no grafo corresponde à aresta entre o Product Owner e a Revisão do Sprint).

O novo *checklist* é então elaborado ou ajustado, na etapa 3, conforme a análise dos grafos, que objetiva consolidar, a partir dos artefatos, as principais entidades e regras sobre o *Scrum*.

Esse *checklist* é então utilizado em projetos reais, na etapa 4, e os *feedbacks* das equipes possibilitam que ele seja aperfeiçoado. Nessa etapa, alguns itens do *checklist* podem ser criticados (e.g., quanto à subjetividade e a possibilidade de obter diferentes respostas para um mesmo item). Na prática, essas críticas são solucionadas através das discussões da equipe sobre o resultado consolidado do preenchimento do *checklist*.

Nessas discussões, eventuais dúvidas são resolvidas e o entendimento sobre cada item tende a ser alinhado, ou melhorias são sugeridas para o *checklist*.



Figura 3. Grafo de Interseção entre o Guia do Scrum e o CAS 2.0.

Após a implantação das melhorias identificadas pelas equipes, o *checklist* pode compor os artefatos da etapa 1 do processo, além de outras fontes sobre o Scrum, e um novo ciclo de melhoria do CAS pode ser realizado. Os algoritmos utilizados na geração e comparação dos grafos, bem como os artefatos, mapas e os próprios grafos gerados neste estudo estão disponíveis em um repositório no *Bitbucket*⁶.

5. Checklist de Avaliação do Scrum (CAS) 2.0

O CAS 2.0⁷ foi desenvolvido a partir do processo descrito na Seção 4, utilizando-se do Guia do Scrum e *checklists* discutidos na Seção 2, com o intuito de consolidar as principais regras do Scrum, bem como solucionar os pontos de melhoria daqueles *checklists*.

Essa nova versão do CAS amplia a inspeção do uso de princípios e práticas ágeis para que as equipes melhorem a identificação e tratamento dos problemas no uso do Scrum. Esse *checklist* continua como um componente do *Agile DMAIC*, método apresentado na Seção 3, e, embora possa ser utilizado separadamente, recomenda-se que seja aplicado junto ao método.

⁶ <https://bitbucket.org/thiagoferraz/comparacaografos>

⁷ <http://www.agiledmaic.com/home>

O CAS 2.0 foi elaborado para ser respondido por toda a equipe *Scrum* (*Product Owner*, *Scrum Master* e Time de Desenvolvimento) a cada iteração, idealmente, momentos antes das reuniões de retrospectiva ou mesmo durante essas reuniões, para que as equipes possam refletir a respeito da evolução do processo ágil, propondo soluções para os principais problemas identificados.

O CAS 2.0 é formado de 70 itens, distribuídos em treze seções, apresentadas na Figura 4. Como exemplo, a seção *Planejamento do Sprint* possui 11 itens, onde o primeiro item visa identificar se o planejamento é realizado a cada iteração. O CAS 2.0 concentra-se em avaliar como são utilizados os eventos e artefatos do *Scrum*. No entanto, ele também abrange a inspeção do uso dos papéis, fundamentos, valores e princípios dessa metodologia.

O objetivo do CAS 2.0 é avaliar quais princípios e práticas do *Scrum* estão sendo utilizados nos projetos. Embora alguns itens do *checklist* indiretamente avaliem a qualidade do produto (e.g., “25. Time atento à qualidade do *software* desenvolvido no *sprint*” e “44. *Software* testado é apresentado”), o CAS 2.0 não inspeciona a qualidade com que os princípios e práticas do *Scrum* são utilizados pelas equipes.

5.1. Comparativo das Versões do Checklist

Durante o uso da primeira versão do CAS em projetos reais, as equipes sinalizaram que as avaliações não haviam identificado alguns problemas importantes (e.g., não atendimento dos objetivos do *sprint*, entregas de *software* com baixa qualidade, longas reuniões de planejamento e acompanhamento). Dessa forma, havia divergências entre as avaliações com a primeira versão do CAS e os resultados dos projetos.

Nesse cenário, as mudanças realizadas para a nova versão do *checklist* objetivaram ampliar a inspeção de princípios e práticas do *Scrum* e suprir as lacunas identificadas. Dentre as alterações, destacam-se as seguintes:

- *Objetivos do Sprint* - 2 novos itens para inspecionar se, durante o planejamento, os objetivos do *sprint* foram definidos pelo time e se, ao final do *sprint*, esses objetivos haviam sido atendidos;
- *Qualidade do Incremento de Software* - 1 novo item para verificar se o time estava atento a qualidade do *software* desenvolvido no *sprint*; e
- *Reuniões de Planejamento, Revisão e Acompanhamento Diário* - 4 novos itens relacionados à realização e duração das reuniões do *Scrum*.

Em resumo, a nova versão do *checklist*, CAS 2.0, manteve a mesma quantidade de seções da primeira versão, com alteração apenas da seção *Plano de Lançamentos* que, por sugestão das equipes envolvidas no estudo de caso, foi renomeada para *Plano de Releases*. Já em relação aos itens do *checklist*, a quantidade de itens aumentou de 65 para 70 e somente 29 dos 65 itens iniciais não sofreram ajustes.

A planilha *Comparativo_Ver soes_CAS.xlsx*, disponível no repositório do *Bitbucket*⁸, apresenta um maior detalhamento das mudanças entre as versões do CAS.

⁸ <https://bitbucket.org/thiagoferraz/comparacaoografos>

VALORES, PRINCÍPIOS E FUNDAMENTOS

<input type="checkbox"/> 01. Time entrega <i>software</i> testado a cada <i>sprint</i>
<input type="checkbox"/> 02. Time entrega o que o <i>Product Owner</i> mais prioriza
<input type="checkbox"/> 03. Processo está continuamente melhorando

PLANEJAMENTO DO SPRINT

<input type="checkbox"/> 04. Há planejamento a cada <i>sprint</i>
<input type="checkbox"/> 05. Duração de 8 horas ou menos
<input type="checkbox"/> 06. Todo o time participa
<input type="checkbox"/> 07. Utiliza <i>Backlog</i> do Produto atualizado
<input type="checkbox"/> 08. <i>Product Owner</i> apresenta itens prioritários do <i>backlog</i>
<input type="checkbox"/> 09. Somente o Time decide quantos itens de <i>backlog</i> cabem no <i>sprint</i>
<input type="checkbox"/> 10. Itens do <i>Backlog</i> do <i>Sprint</i> decompostos em tarefas
<input type="checkbox"/> 11. Somente o Time define e estima tarefas
<input type="checkbox"/> 12. <i>Product Owner</i> disponível quando o Time está estimando
<input type="checkbox"/> 13. Time revisa e atualiza a definição de "pronto"
<input type="checkbox"/> 14. Objetivos do <i>sprint</i> claramente definidos

SPRINT

<input type="checkbox"/> 15. Há iterações definidas
<input type="checkbox"/> 16. Duração de 4 semanas ou menos
<input type="checkbox"/> 17. Datas de início e fim do <i>sprint</i> respeitadas
<input type="checkbox"/> 18. Estimativas atualizadas durante o <i>sprint</i>
<input type="checkbox"/> 19. Testes de aceitação realizados durante o <i>sprint</i>
<input type="checkbox"/> 20. <i>Software</i> desenvolvido no <i>sprint</i> testado durante o próprio <i>sprint</i>
<input type="checkbox"/> 21. <i>Software</i> é implantado em ambiente para validação do <i>Product Owner</i>
<input type="checkbox"/> 22. Alocação de tarefas realizada somente pelo Time
<input type="checkbox"/> 23. Time conhece e respeita a definição de "pronto" do <i>sprint</i>
<input type="checkbox"/> 24. Time não é interrompido nem controlado de fora
<input type="checkbox"/> 25. Time atento à qualidade do <i>software</i> desenvolvido no <i>sprint</i>
<input type="checkbox"/> 26. Velocidade do <i>sprint</i> é medida considerando apenas os itens "prontos"
<input type="checkbox"/> 27. Todos os objetivos do <i>sprint</i> foram alcançados

TIME DE DESENVOLVIMENTO

<input type="checkbox"/> 28. Costuma entregar o que foi prometido
<input type="checkbox"/> 29. Possui as competências necessárias
<input type="checkbox"/> 30. Integrantes não estão dedicados a papéis específicos
<input type="checkbox"/> 31. Possui de 3 a 9 integrantes
<input type="checkbox"/> 32. Conhece os principais impedimentos

BACKLOG DO PRODUTO

<input type="checkbox"/> 33. Há um <i>backlog</i> do produto
<input type="checkbox"/> 34. Priorizado por valor de negócio
<input type="checkbox"/> 35. Time compreende os itens de <i>backlog</i>
<input type="checkbox"/> 36. Itens prioritários grandes decompostos em menores
<input type="checkbox"/> 37. Acessível a todo o time

PRODUCT OWNER

<input type="checkbox"/> 38. Claramente definido
<input type="checkbox"/> 39. Tem conhecimento e autoridade para priorizar
<input type="checkbox"/> 40. Interage com todo o time

REUNIÃO DE REVISÃO

<input type="checkbox"/> 41. Time realiza a reunião de revisão a cada <i>sprint</i>
<input type="checkbox"/> 42. Duração de 4 horas ou menos
<input type="checkbox"/> 43. Todo o time participa
<input type="checkbox"/> 44. <i>Software</i> testado é apresentado
<input type="checkbox"/> 45. <i>Feedback</i> é recebido do <i>Product Owner</i>

RETROSPECTIVA

<input type="checkbox"/> 46. Há retrospectiva ao final do <i>sprint</i>
<input type="checkbox"/> 47. Duração de 3 horas ou menos
<input type="checkbox"/> 48. Time e <i>Scrum Master</i> participam
<input type="checkbox"/> 49. Resulta em propostas de melhorias concretas
<input type="checkbox"/> 50. Algumas propostas são implementadas

PLANO DE RELEASES

<input type="checkbox"/> 51. <i>Product Owner</i> tem plano de <i>releases</i> com base no <i>Backlog</i> do Produto
<input type="checkbox"/> 52. <i>Product Owner</i> usa velocidade do Time para planejar <i>releases</i>
<input type="checkbox"/> 53. Acompanhado e atualizado a cada <i>sprint</i> pelo <i>Product Owner</i>
<input type="checkbox"/> 54. Acessível a todo o time

SCRUM MASTER

<input type="checkbox"/> 55. Claramente definido
<input type="checkbox"/> 56. Focado em remover impedimentos
<input type="checkbox"/> 57. Interage com todo o time
<input type="checkbox"/> 58. Conhece e dissemina os princípios e práticas do <i>Scrum</i>

BACKLOG DO SPRINT

<input type="checkbox"/> 59. Time utiliza <i>backlog</i> do <i>sprint</i>
<input type="checkbox"/> 60. Pertence exclusivamente ao Time
<input type="checkbox"/> 61. Acompanhado e atualizado diariamente pelo Time
<input type="checkbox"/> 62. Acessível a todo o time

REUNIÃO DIÁRIA

<input type="checkbox"/> 63. Há reuniões diárias
<input type="checkbox"/> 64. Duração de 15 minutos ou menos
<input type="checkbox"/> 65. Ocorre no mesmo horário e local
<input type="checkbox"/> 66. Time e <i>Scrum Master</i> participam
<input type="checkbox"/> 67. Problemas e impedimentos são informados

GRÁFICO DE BURNDOWN

<input type="checkbox"/> 68. Time utiliza um gráfico de <i>Burndown</i>
<input type="checkbox"/> 69. Acompanhado e atualizado diariamente pelo Time
<input type="checkbox"/> 70. Acessível a todo o time

Figura 4. CAS 2.0 - Checklist de Avaliação do Scrum.

5.2. Sobre a Automação do Checklist

A automação do CAS 2.0 foi realizada a partir da plataforma *Google Sites*⁹ e está disponível *online* através do endereço <http://www.agiledmaic.com/>. O *Google Sites* possibilita a criação rápida de sites e foi selecionado neste estudo por ser gratuito, de fácil aprendizado e uso, e por fornecer os componentes necessários para a automação.

A automação do CAS 2.0 pode ser dividida em três funcionalidades: (i) Preencher o CAS: implementação realizada com o *Google Forms*, integrado ao *Google Sites*; (ii) Realizar os cálculos de abrangência e nível de maturidade no uso do *Scrum*, os quais são realizados de acordo com a especificação do método *Agile DMAIC*, a partir das fórmulas das planilhas e dados do preenchimento das equipes; e (iii) Consolidar os resultados da equipe e apresentá-los em tempo real, e vale ressaltar que nessa funcionalidade foram utilizados os componentes gráficos *Google Charts* e o *Awesome Table*.

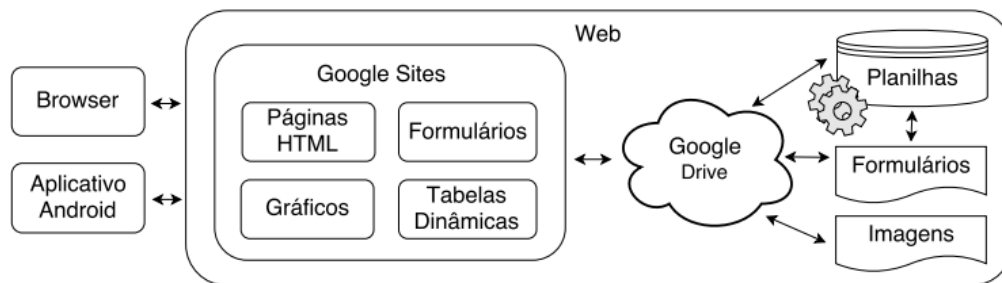


Figura 5. Arquitetura da Automação do CAS 2.0.

A arquitetura da automação do CAS 2.0 é apresentada na Figura 5. O CAS 2.0 pode ser acessado na *Web* através de um *browser* ou pelo aplicativo Android disponível no *Google Play*¹⁰. O *Google Sites* fornece uma variedade de componentes para a criação dos sites. Os principais componentes utilizados foram as páginas *HTML*, formulários, gráficos e um componente para geração de tabelas dinâmicas, *Awesome Table*. Esses componentes, por sua vez, possuem integração com as planilhas, formulários e imagens armazenados e gerenciados na nuvem pelo *Google Drive*¹¹.

6. Aplicando o CAS 2.0 em Projetos Reais

Esta seção apresenta uma avaliação qualitativa da aplicação do CAS 2.0 em projetos reais, na qual avalia-se as informações e também as traduz em números para serem classificadas e analisadas [Severino 2007].

6.1. Objetivo da Avaliação do CAS 2.0

O objetivo da avaliação foi definido com base no *template* de definição de objetivos do *Goal-Question-Metric (GQM)* [Basili et al. 1995]. O objetivo do estudo de caso é analisar o CAS 2.0, com a finalidade de avaliação, no que diz respeito a sua estrutura e utilização, do ponto de vista dos papéis do *Scrum*, no contexto de projetos de desenvolvimento de *software* com o *Scrum*. Para atingir esse objetivo, foram definidas três questões de análise e cinco medidas, conforme são apresentadas na Figura 6 e descritas a seguir:

⁹ <https://apps.google.com/intx/en/products/>

¹⁰ <https://play.google.com/store/apps/details?id=com.ah2.apps.agile2us>

¹¹ <https://www.google.com/drive/>

- **(M1) Clareza e objetividade** - Avalia se o *checklist* usa uma linguagem simples e direta, sendo breve e objetivo na identificação do essencial sobre o uso do *Scrum*;
- **(M2) Ordem e organização das seções e itens** - Avalia se os itens do *checklist* estão organizados em uma ordem lógica, numerados e agrupados por similaridades;
- **(M3) Quantidade de seções e itens** - Avalia o *checklist* quanto ao seu tamanho a partir da quantidade de seções e itens;
- **(M4) Tempo de preenchimento** - Avalia o tempo de preenchimento do *checklist*; e
- **(M5) Importância para identificar problemas no uso do Scrum** - Avalia o *checklist* quanto a sua importância na identificação de problemas no uso do *Scrum*.

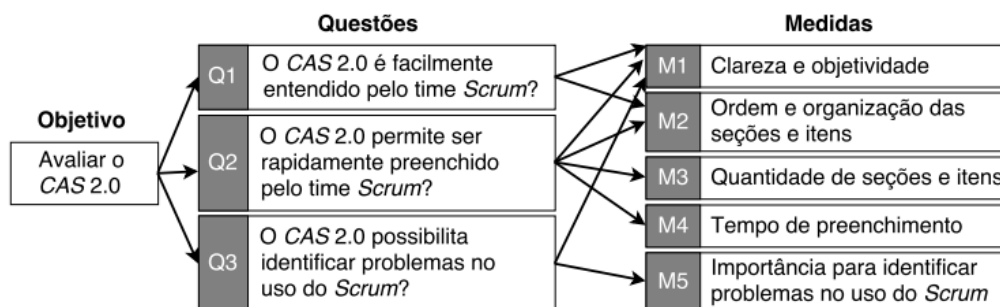


Figura 6. Relacionamento entre as questões e medidas da avaliação.

Sobre o método de coleta, as informações necessárias às medidas foram obtidas com as equipes dos projetos, a partir de um questionário¹² aplicado via *Web*. Esse questionário contém 4 perguntas sobre o perfil dos respondentes, 5 sobre as medidas e 1 pergunta aberta para sugestões de melhorias sobre o *checklist*. Nas questões relacionadas às medidas, os participantes deveriam indicar suas opiniões a partir de um dos seguintes valores: *Péssimo*, *Ruim*, *Regular*, *Bom* ou *Excelente*. O resultado final de cada medida foi obtido aplicando-se a moda dos valores indicados pelos respondentes.

6.2. Projetos Reais do Estudo de Caso

Os projetos selecionados para o estudo de caso foram desenvolvidos em duas instituições públicas e uma privada, que utilizaram o CAS 2.0 durante a adoção do *Scrum* ou na melhoria de sua utilização. Um resumo desses projetos, com informações obtidas dos líderes através de um questionário¹³ aplicado via *Web*, é apresentado na Tabela 2. Por motivo de confidencialidade, a razão social das instituições envolvidas, bem como informações mais detalhadas sobre os projetos não são apresentadas.

As instituições A e C são públicas, enquanto a B é privada. Todas as equipes envolvidas são formadas, em sua maioria, por pesquisadores e desenvolvedores de áreas da computação e engenharia, e a maior parte dos integrantes possui uma alocação integral aos projetos. Na Tabela 2, há projetos de Desenvolvimento de *Software* (D) e de Pesquisa e Desenvolvimento (P&D). A *Duração (Meses)* consiste na duração planejada dos projetos e a *Equipe* representa o tamanho das equipes, considerando o *Product Owner*, *Scrum Master* e o Time de Desenvolvimento. A *Complexidade* dos projetos está relacionada ao uso simultâneo de várias tecnologias, ao elevado número de envolvidos e à necessidade de conhecimentos técnicos especializados.

¹² <http://www.agiledmaic.com/home/feedback-do-checklist>

¹³ <http://www.agiledmaic.com/projetos/novo-projeto>

Tabela 2. Projetos do Estudo de Caso.

Projeto	Instituição	Tipo de Projeto	Duração (Meses)	Equipe	Complexidade
PID_1977	A	D	10	10	Média
PID_2103	A	D	3	5	Baixa
PID_2131	A	D	3	9	Média
PID_2139	A	D	4	10	Média
PID_FV6	A	D	5	10	Alta
PID_PP	A	D	6	5	Média
PID_PR	A	D	6	6	Média
Mobile App	B	D	6	9	Média
Tools - App1	B	P&D	6	8	Alta
Tools - App2	B	P&D	6	6	Média
Tools - App3	B	P&D	6	6	Média
Tools - App4	B	P&D	6	7	Alta
Tools - App5	B	P&D	6	3	Média
Tools - App6	B	P&D	3	5	Alta
SGPR	C	D	7	10	Média

6.3. Resultados e Discussão

As informações apresentadas nesta seção foram obtidas com as equipes dos projetos do estudo de caso. No total, 53 pessoas participaram da avaliação do CAS 2.0, de forma espontânea e anônima. Sobre os respondentes, 75,5% (40) deles eram da instituição A, 13,2% (7) da B e 11,3% (6) da C.



Figura 7. Tempo, Experiência e Papel dos Respondentes do Estudo de Caso.

Os gráficos da Figura 7 apresentam, respectivamente, o tempo, a experiência e o papel dos respondentes com o *Scrum*. Quanto ao uso do *Scrum*, 98,1% dos respondentes já tinham mais de seis meses de experiência e, além disso, 86,8% afirmaram ter uma experiência boa ou excelente com o *Scrum*. Dessa forma, boa parte dos participantes conhecia bem os conceitos e práticas utilizados. Sobre o papel dos respondentes, houve a participação dos três papéis do *Scrum*, sendo 3,8% (2) deles *Product Owner*, 11,3% (6) eram *Scrum Masters* e 84,9% (45) eram do Time de Desenvolvimento.

Em se tratando do CAS, todos os participantes utilizaram o *checklist* uma ou mais vezes, e 52,8% deles, seis ou mais vezes. No que diz respeito ao conteúdo do CAS, os resultados da avaliação das medidas foram positivos (veja a Figura 8):

- (M1) 86,8% avaliaram a clareza e a objetividade do CAS como *Bom* ou *Excelente*. O resultado da moda para M1 foi *Bom* (54,7%);
- (M2) 90,6% consideraram a ordem e a organização dos itens e seções como *Bom* ou *Excelente*. O resultado da moda para M2 foi *Bom* (56,6 %);

- (M3) 81,1% avaliaram o tamanho do *checklist* como *Bom* ou *Excelente*. O resultado da moda para M3 foi *Bom* (58,5%);
- (M4) 86,8% avaliaram o tempo de preenchimento como *Bom* ou *Excelente*. O resultado da moda para M4 foi *Bom* (56,6%); e
- (M5) 88,7% acreditam que o CAS é importante para identificar problemas no uso do *Scrum*, avaliando o *checklist* como *Bom* ou *Excelente*. O resultado da moda para M5 foi *Bom* (54,7%).

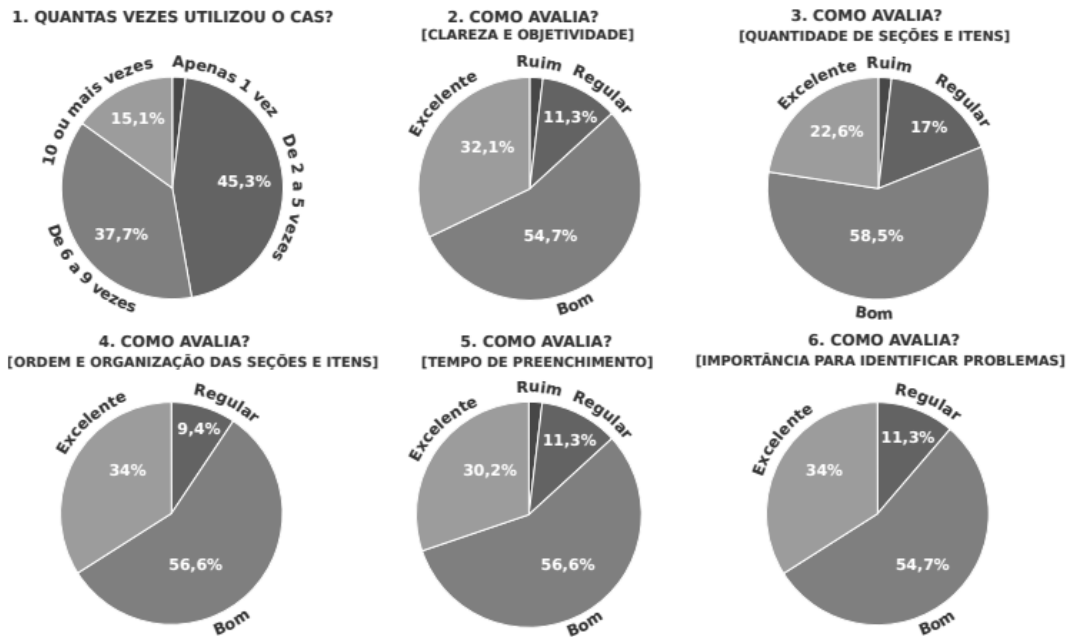


Figura 8. Feedback das Equipes sobre a Nova Versão do CAS.

Com base nos valores das medidas, foi possível responder às questões definidas: (Q1) O CAS 2.0 é de fácil compreensão, pois ele é claro e objetivo (M1 = *Bom*), além de possuir uma boa organização dos itens e seções (M2 = *Bom*); (Q2) O preenchimento do CAS 2.0 pode ser considerado rápido, uma vez que ele é de fácil compreensão, tem uma quantidade satisfatória de itens e seções (M3 = *Bom*) e tem um tempo de preenchimento satisfatório (M4 = *Bom*); e (Q3) O CAS 2.0 é de fácil entendimento (M1 = *bom*) e auxilia na identificação de problemas no uso do *Scrum* (M5 = *Bom*). Outras análises (e.g., uma análise por papéis, experiência no *Scrum*) podem ser encontradas na *Web*¹⁴.

Além disso, como no questionário havia uma questão aberta para sugestões de melhoria, foi possível obter outros *feedbacks* a serem utilizados para a evolução do *checklist*. Os principais comentários são descritos a seguir: (i) “Alguns itens possuem certa relação e isso poderia ser utilizado para ocultar (em tempo real, no preenchimento) os itens desnecessários, diminuindo o esforço do usuário e o tempo de preenchimento”; (ii) “Diminuir a quantidade de seções ou deixá-las de acordo com o nível do time ou do *sprint* a ser avaliado”; (iii) “Adicionar uma descrição do que se busca inspecionar com cada item ou seção”; e (iv) “Permitir que apenas a quantidade informada de integrantes do time preencha o *checklist*, de forma a não alterar a avaliação final”. Conforme supracitado, a maioria dos *feedbacks* visam a otimização do preenchimento do CAS.

¹⁴ <http://www.agiledmaic.com/feedback-sobre-o-checklist>

Outras preocupações dos respondentes estão relacionadas à segurança e usabilidade do *checklist*.

7. Conclusões

Para que os projetos usufruam dos principais benefícios do *Scrum*, como a melhoria na produtividade e qualidade dos produtos desenvolvidos, sua adoção e uso precisam ser adequadamente gerenciados para uma evolução contínua e controlada do processo ágil. Nesse cenário, para avaliar o uso de seus princípios e práticas, uma nova versão do *Checklist* de Avaliação do *Scrum* (CAS) foi proposta neste artigo, elaborada com base no Guia do *Scrum* e em mais sete *checklists*. A principal vantagem dessa nova versão, CAS 2.0, é que, além de melhorar a inspeção de princípios e práticas do *Scrum*, sua automação permite consolidar a visão dos integrantes da equipe sobre o uso do *Scrum* e fornecer resultados em tempo real.

Para elaborar o CAS 2.0, foi definido um processo com quatro etapas: (i) identificação e seleção de artefatos sobre o *Scrum*; (ii) criação dos mapas conceituais com base nas regras e conceitos identificados; (iii) geração de grafos a partir desses mapas; e (iv) consolidação da nova versão do *checklist* através da análise das operações de subtração e interseção dos grafos. Uma vez que esse processo pode ser utilizado na melhoria do CAS, acredita-se que ele pode ser usado para melhorar outros *checklists*.

Esse *checklist* foi aplicado em quinze projetos reais, durante um ano, envolvendo três instituições. Para coletar o *feedback* dos usuários, foi elaborado um questionário *online*, o qual obteve 53 respondentes. De maneira geral, os resultados foram positivos, pois, em todos os critérios avaliados (objetividade, tamanho, organização, tempo de preenchimento e importância para identificar problemas no uso do *Scrum*), a aprovação da nova versão do CAS foi superior a 80%.

Como trabalhos futuros, pretende-se: (i) realizar uma análise comparativa mais aprofundada dos *checklists* de avaliação do *Scrum*, com base na Engenharia de *Software* Experimental; (ii) traduzir o CAS 2.0 para outros idiomas; (iii) aumentar a segurança quanto ao seu preenchimento *online*; e (iv) implementar mecanismos de adaptação do *checklist* para diferentes contextos (e.g., experiência e maturidade da equipe com o *Scrum* e metas de melhoria no uso de princípios e práticas ágeis).

Referências

- Basili, V. R., Caldiera, G., e Rombach, H. D. (1995). Goal question metric paradigm. In *Encyclopedia of Software Engineering*, pages 527–532. John Wiley Sons.
- Cunha, T. F. V. d. e Andrade, R. M. C. (2014a). Agile DMAIC: Um método para avaliar e melhorar o uso do Scrum em projetos de software. In XIII Simpósio Brasileiro de Qualidade de Software (SBQS), pages 121–135.
- Cunha, T. F. V. d. e Andrade, R. M. C. (2014b). SLeSS 2.0: uma evolução da abordagem de integração do Scrum e lean six sigma para aplicações móveis.
- De Souza, F., Boeres, M., Cury, D., De Menezes, C., e Carlesso, G. (2008). An approach to comparison of concept maps represented by graphs.
- Gloger, B. (2012). Infoq scrum checklist. Disponível em: <https://www.infoq.com/minibooks/scrum-checklists>. Acesso em: 23 de Maio de 2016.

- Jalali, S., Wohlin, C., e Angelis, L. (2014). Investigating the applicability of agility assessment surveys: A case study. *Journal of Systems and Software*, 98:172 – 190.
- Khelladi, D.-E., Bendraou, R., Baarir, S., Laurent, Y., e Gervais, M.-P. (2015). A framework to formally verify conformance of a software process to a software method. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC '15*, pages 1518–1525, New York, NY, USA. ACM.
- Kniberg, H. (2009). Scrum Checklist – version 2.0. Disponível em: <http://blog.crisp.se/2009/08/14/henrikkniberg/1250265360000>. Acesso em: 04 de Julho de 2016.
- Lomonaco, F. e Lapointe, C. (2013). Scrum Adherence Index. Disponível em: <https://goo.gl/VdaLpp>. Acesso em: 23 de Maio de 2016.
- Maharana, A. (2015). A Scrum Master's Checklist. Disponível em: <https://www.linkedin.com/pulse/scrum-masters-checklist-anand-maharana>. Acesso em: 23 de Maio de 2016.
- Melo, C. O., Santos, V. A., Corbucci, H., Katayama, E., Goldman, A., e Kon, F. (2012). Métodos ágeis no Brasil: estado da prática em times e organizações. Technical Report RTMAC201203.
- Park, J. S., McMahon, P. E., e Myburgh, B. (2016). Scrum powered by essence. *SIGSOFT Softw. Eng. Notes*, 41(1):1–8.
- Perkusich, M., Soares, G., Almeida, H., e Perkusich, A. (2015). A procedure to detect problems of processes in software development projects using Bayesian networks. *Expert Systems with Applications*, 42(1):437 – 450.
- Radujkovic, M., Vukomanovic, M., Wagner, R., Tenera, A., e Pinto, L. C. (2014). A Lean Six Sigma (LSS) project management improvement model. *Procedia – Social and Behavioral Sciences*, 119:912 – 920.
- Schwaber, K. (2007). *The Enterprise and Scrum*. Microsoft Press.
- Severino, A. J. (2007). *Metodologia do trabalho científico*. Cortez, São Paulo, 23 edition.
- Sutherland, J. e Schwaber, K. (2016). Guia do Scrum. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>. Acesso em: 04 de Agosto de 2016.
- TPC (2016). Agile/Scrum Check List - Total Programme Control Ltd. Disponível em: <http://www.totalprogrammecontrol.com/scrum.php>. Acesso em: 23 de Maio de 2016.
- VersionOne (2016). 10th annual state of agile development survey results. Disponível em: <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>. Acesso em: 04 de Julho de 2016.
- Vode, B. e Sutherland, J. (2008). Scrum but test. Disponível em: <http://antoine.vernois.net/scrumbut/?page=test&lang=en>. Acesso em: 27 de Junho de 2015.
- Zazworka, N., Stapel, K., Knauss, E., Shull, F., Basili, V. R., e Schneider, K. (2010). Are developers complying with the process: An XP study. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '10*, pages 14:1–14:10, New York, NY, USA. ACM.