

# Elementos de colaboração nos métodos ágeis de desenvolvimento de software

Carlos Eduardo Stefani<sup>1</sup>, Marcelo Duduchi<sup>1</sup>

<sup>1</sup>Centro Paula Souza (CPS)  
Rua dos Bandeirantes, 169 –Bom Retiro, São Paulo –SP –Brasil

cafani@gmail.com, mduduchi@gmail.com

***Abstract.** This study identifies collaboration elements in most used agile methods of software development. The elements were grouped and classified into collaboration requirements. Furthermore, were related with collaboration maturity requirements established by ColabMM. As a result, were found 26 groups of elements explicitly presents at the studied agile methods and few gaps in collaboration requirements. Was found also relationship between agile methods elements and collaboration maturity requirements.*

***Resumo.** Este estudo identifica elementos de colaboração nos métodos ágeis de desenvolvimento de software mais usados, agrupa-os e classifica-os em requisitos de colaboração e relaciona-os a requisitos de maturidade em colaboração estabelecidos no ColabMM. Como resultado, encontra 26 grupos de elementos explicitamente presentes nos métodos ágeis estudados e algumas lacunas de cada método em relação aos requisitos de colaboração, além de relações entre os elementos de colaboração dos métodos ágeis e requisitos de maturidade em colaboração.*

## 1. INTRODUÇÃO

No mundo atual o software se tornou onipresente, principalmente em ambientes corporativos e colaborativos. De softwares para tomada de decisão nos níveis mais altos da hierarquia, até planilhas eletrônicas e comunicadores instantâneos nos níveis mais operacionais há intensa utilização de software. Atualmente o software é utilizado como produto e veículo para distribuir o produto, assumindo duplo papel. Os processos de desenvolvimento de software passaram por grandes transformações nas últimas décadas, passando a ser cada vez mais iterativos e colaborativos. Ao passo que estas transformações facilitaram o processo, também trouxeram maior complexidade aos softwares desenvolvidos e maior valor na sua utilidade nos negócios e nas organizações (PRESSMAN, 2016).

De acordo com informações de Digital.Ai (2021), a adoção de métodos ágeis tem aumentado ao longo do tempo, passando de 80% em 2011 para 94% em 2021. Parte das contribuições dos métodos ágeis é o incentivo à comunicação e colaboração. Quanto melhor for a colaboração entre os membros das equipes de desenvolvimento, patrocinadores, clientes e os demais envolvidos, mais bem direcionados serão os esforços para a produção do software, mais fluida será a resposta contínua às mudanças, maior será a capacidade de produzir software alinhado às necessidades de negócio e de inovação e, por fim, maior será a qualidade do software aferida sob a ótica do usuário (BECK et

al., 2001; HIGHSMITH E COCKBURN, 2001; COCKBURN, 2006; JACOBSON, 2002; PRESSMAN, 2016).

Indaga-se, então, a presença explícita de elementos colaborativos nos processos atuais de desenvolvimento de software que, notadamente, utiliza-se de métodos ágeis e se é possível vinculá-los a níveis de maturidade em colaboração. Neste contexto, formula-se a questão de pesquisa: *Quais são os elementos colaborativos presentes nos métodos ágeis de desenvolvimento de software, como podemos classificá-los e qual sua relação com níveis de maturidade em colaboração?*

## **2. FUNDAMENTAÇÃO TEÓRICA**

Esta seção apresenta fundamentação em relação a Engenharia de Software, métodos ágeis, colaboração e modelos de maturidade em colaboração.

### **2.1. Engenharia de software e métodos ágeis**

Softwares são “programas executáveis em um computador de qualquer porte ou arquitetura, conteúdos apresentados (...) e informações descritivas tanto na forma impressa como na virtual (...)” (PRESSMAN, 2016, p. 29). A presença do software é tamanha que o mundo moderno não poderia existir sem ele. A maioria dos dispositivos eletrônicos inclui um computador e um software que o controla (SOMMERVILLE, 2011).

O software é viabilizador na distribuição do produto mais importante da nossa era: a informação. Ao longo dos últimos cinquenta anos, passou a ter disponíveis importantes avanços na capacidade dos computadores em que executam, resultando em utilização mais ampla em sistemas computacionais cada vez mais sofisticados e complexos. Atualmente, ele está profundamente incorporado em quase todos os aspectos de nossas vidas (PRESSMAN, 2016).

O desenvolvimento de software é o processo que vai da sua concepção à disponibilização para uso. Envolve a identificação do problema ou oportunidade, entendimento de suas regras, programação e implantação (STAIR e REYNOLDS, 2015). Segundo Pressman (2016) é um processo de aprendizagem social iterativo que incorpora conhecimento, podendo ser essencialmente prescritivo, em que suas etapas são bem definidas, ou flexível, com maior adaptabilidade às mudanças e pessoas.

Os processos de desenvolvimento de software podem ser avaliados na tentativa de assegurar a conformidade com critérios básicos tidos como de sucesso. Para isto, existem abordagens como SCAMPI (Standart CMMI Assessment Method For Process Improvement), que fornece com um conjunto de cinco etapas para avaliação: início, diagnóstico, estabelecimento, atuação e aprendizado, CBA IPI (CMM – Based Appraisal for Internal Process Improvement), que avalia a maturidade relativa de uma organização de software, SPICE (ISO/IEC 15504), que define requisitos para avaliação do processo de software e ISO 9001:2000 para software, que é aplicável a qualquer organização que deseje aperfeiçoar a qualidade de seus produtos, incluindo softwares (PRESSMAN, 2016). Cabe também destacar o programa de Melhoria do Processo de Software Brasileiro (MPS.BR), que tem por objetivo melhorar a capacidade de desenvolvimento de software, serviços e as práticas de gestão de recursos humanos na indústria de software.

Para melhor organizar os processos de software, foram adotados modelos. Modelo de processo de software é uma representação simplificada de um processo para desenvolvimento de software que tem por objetivo definir de forma geral o funcionamento do processo sob uma perspectiva particular (SOMMERVILLE, 2011). Os modelos de processos de desenvolvimento de software evoluíram dos mais prescritivos aos mais flexíveis, sendo ambos adotados na atualidade visando adequar-se às necessidades de cada projeto.

Os modelos clássicos, também conhecidos como prescritivos, foram os primeiros a surgir. Segundo Pressman (2016, p. 58) “modelos de processo prescritivo foram propostos para trazer ordem ao caos existente na área de desenvolvimento de software”. Segundo Cockburn (2006), os modelos prescritivos tinham como falha a baixa consideração sobre as fraquezas humanas. Consideravam que as pessoas envolvidas no processo agiriam da forma em que o processo era definido e os desvios deveriam ser ajustados principalmente por meio da disciplina. O autor destaca, ainda, que nem sempre o ser humano age de forma consistente e que as metodologias com disciplina rigorosa são, portanto, frágeis.

Highsmith e Cockburn (2001) enfatizam que os métodos ágeis são capazes de promover abordagem dinâmica e inovativa como solução para abraçar as mudanças em um mundo turbulento e tecnológico. Segundo Rigby, Sutherland e Takeuchi (2016), são mais efetivos e fáceis de implementar quando envolvem inovação: o problema é complexo, soluções existentes não são conhecidas e os requisitos provavelmente mudarão, o trabalho pode ser modularizado, há colaboração e feedback rápidos com usuários finais e, neste contexto, times criativos possuem maior chance de sucesso.

Os métodos ágeis têm por base, num contexto geral, o chamado “Manifesto Ágil”. De acordo com Boehm (2002), foi uma iniciativa de praticantes dos métodos emergentes de engenharia de software mencionados. Nele, são priorizados “indivíduos e interações acima de processos e ferramentas, software operacional acima de documentação completa, colaboração dos clientes acima de negociação contratual e respostas a mudanças acima de seguir um plano” (BECK et al., 2001).

De acordo com Digital.Ai (2021), os métodos ágeis mais utilizados são Scrum, Extreme Programming (XP) e Kanban. Também há grande utilização de métodos híbridos, que combinam mais de um método ágil. Estes métodos foram, portanto, enfatizados nesta pesquisa.

O Scrum é um framework para solução de problemas complexos e adaptativos para entregar produtos com o maior valor possível, de forma ao mesmo tempo produtiva e criativa (SCHWABER E SUTHERLAND, 2017). O Extreme Programming (XP) é um método ágil que busca conciliar fatores humanos com produtividade, de forma adaptativa e incremental. Ele distribui as atividades de planejamento, análise e projeto no decorrer da construção de software. Sua principal característica é a promoção de mudanças sociais nas relações das pessoas no desenvolvimento de software (BECK, 2004). O Kanban é uma ferramenta para controlar a cadeia logística de produção utilizada pelo Lean Manufacturing que visa atingir a produção *Just-in-time* (sob demanda). O método Kanban, escrito em maiúsculo, foi uma iniciativa de Anderson (2010) em transportar a ferramenta para os processos de desenvolvimento de software. O Kanban também pode ser definido como um conjunto de conceitos, princípios, práticas, técnicas e ferramentas

para gerenciar o processo de desenvolvimento de produto com ênfase na entrega contínua de valor aos clientes, com aprendizado e melhoria contínua (AL-BAIK e MILLER, 2015).

Mesmo mais de vinte anos após sua adoção, os métodos ágeis e suas práticas ainda são estudados em pesquisas teóricas e empíricas, como em Al-Saqqa et al. (2020), Dudhat e Ali Abbasi (2021) e Strode, Dingsøyr e Lindsjorn (2022).

A partir de informações de Schwaber e Sutherland (2017) acerca do Scrum, Beck (1999) e Beck (2004) sobre XP e Al-Baik e Miller (2015), tratando de Kanban, elaborou-se o Quadro 1, que contém os principais elementos dos métodos ágeis Scrum, XP e Kanban. O quadro visa mostrar uma visão geral de cada um dos métodos ágeis, com ênfase em suas características colaborativas que são objeto deste trabalho.

**Quadro 1. Principais elementos dos métodos ágeis Scrum, XP e Kanban**

Categoria	Elementos Scrum	Elementos XP	Elementos Kanban
<b>Participantes</b>	<ul style="list-style-type: none"> <li>Product Owner, Time de Desenvolvimento, Scrum Master, Stakeholders</li> </ul>	<ul style="list-style-type: none"> <li>Testadores, Projetistas de interação, Arquitetos de software, Gerentes de projeto, Gerentes de produto, Gerentes executivos, Escritores de manuais, Usuários, Programadores, Recursos humanos (trabalho individual e trabalho social), Papéis fixos, mas não rígidos</li> </ul>	<ul style="list-style-type: none"> <li>Membros do time, gerência e clientes</li> </ul>
<b>Eventos</b>	<ul style="list-style-type: none"> <li>Planejamento da Sprint</li> <li>Reuniões diárias</li> <li>Revisões da Sprint</li> <li>Retrospectivas da Sprint</li> </ul>	<ul style="list-style-type: none"> <li>Reuniões semanais de revisão</li> <li>Reuniões trimestrais de planejamento</li> </ul>	<ul style="list-style-type: none"> <li>Reuniões de planejamento e estimativas</li> <li>Reuniões para discutir restrições</li> </ul>
<b>Artefatos</b>	<ul style="list-style-type: none"> <li>Incremento de software</li> <li>Backlog do Produto</li> <li>Backlog da Sprint</li> </ul>	<ul style="list-style-type: none"> <li>Código, Teste, Histórias de usuário, Manuais</li> </ul>	<ul style="list-style-type: none"> <li>Quadro Kanban</li> </ul>
<b>Características</b>	<ul style="list-style-type: none"> <li>Destaca a importância do empirismo como fonte de aprendizado contínuo da equipe</li> <li>Projetado para aperfeiçoar flexibilidade, criatividade e produtividade</li> <li>Divide trabalho em intervalos de tempo chamados Sprints</li> <li>Uma Sprint representa um período igual ou inferior a um mês em que um incremento de produto potencialmente liberável é criado</li> <li>O objetivo do grupo é o objetivo da Sprint</li> <li>Interações externas com a equipe são auxiliadas pelo Scrum Master</li> </ul>	<ul style="list-style-type: none"> <li>Entregas pequenas</li> <li>Projeto simples / Código simples (sem duplicidades)</li> <li>Ênfase em testes (unitários e funcionais)</li> <li>Refatoração de código</li> <li>Programação em par</li> <li>Integração contínua (de novos códigos ao já construído)</li> <li>Propriedade de código coletiva (não existe um único "dono")</li> <li>Cliente presente todo o tempo</li> <li>Horas de trabalho limitadas (horas extras reduzidas)</li> <li>Espaço físico de trabalho aberto</li> </ul>	<ul style="list-style-type: none"> <li>Sistema puxado</li> <li>Fila de priorização</li> <li>Crítérios de inclusão</li> <li>Trabalho em progresso</li> <li>Crítérios para item pronto</li> <li>Item revertido (retornou etapa)</li> <li>Aprendizado validado</li> <li>Cycle time/Lead time (tempo do início ao fim da produção ou entrega)</li> <li>Ferramenta de medição de desempenho</li> <li>Identificação de gargalos</li> <li>Estoque</li> <li>Identificação de desperdícios</li> <li>Colaboração do time</li> <li>Avatar (representação visual do responsável pela tarefa)</li> <li>Definição de políticas (regras e guias do que deve ser feito e como)</li> </ul>
<b>Regras</b>	<ul style="list-style-type: none"> <li>Fundamentos: Transparência, Inspeção dos artefatos e progresso do trabalho, Adaptação</li> <li>Valores: comprometimento, coragem, foco, abertura e respeito</li> <li>Entrega: iterativa e incremental (maximizando feedback)</li> <li>Feedback do mercado, feedback dos stakeholders, feedback do Product Owner</li> <li>Equipes auto-organizáveis (escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidas por outros de fora da equipe).</li> <li>Equipes multifuncionais (possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe)</li> <li>Equipes pequenas (de 3 a 9 pessoas)</li> <li>Equipe deve estabelecer critérios de qualidade</li> </ul>	<ul style="list-style-type: none"> <li>Valores: Comunicação, Simplicidade, Feedback, Coragem, Respeito</li> <li>Princípios: Humanidade, Economia, Benefício mútuo, Autossimilaridade (o todo e partes isoladas possuem estruturas semelhantes), Aprimoramento, Diversidade, Reflexão, Fluxo contínuo, Problemas em oportunidades, Redundância (nos testes), Falhar para aprender, Qualidade, Passos pequenos, Aceitação de Responsabilidade</li> <li>Práticas primárias: Sentar-se junto, Time completo (multidisciplinar), Espaço de trabalho informativo (sobre o projeto), Trabalho Energizado (trabalhar a quantidade de horas em que há produtividade), Programação em par, Histórias de usuário, Ciclo semanal (revisão), Ciclo trimestral (planejamento), Folga (tempo livre do projeto), Ten-minute build, Integração contínua, Teste primeiro, Projeto incremental</li> <li>Práticas Secundárias: Envolvimento real do cliente, Desenvolvimento incremental, Continuidade da equipe (não separar integrantes de uma equipe efetiva), encorajar equipes (quanto aumentarem a produtividade), Buscar causa raiz de defeitos, Código compartilhado (sem único dono), Os principais artefatos são código e teste, Repositório único de código, Desenvolvimento diário (entrega contínua), Contrato de escopo variável, Privilegia sistemas pay-per-use (pague por uso)</li> <li>Teste antes, sempre e automaticamente</li> </ul>	<ul style="list-style-type: none"> <li>Limitar WIP (Work in progress / trabalho em andamento)</li> <li>Equipe deve colaborar para término de tarefas do quadro</li> <li>Equipe deve definir políticas</li> </ul>

Fonte: Resultados da pesquisa

Observa-se que estão presentes diversas características colaborativas, como equipes auto-organizadas, reuniões, feedbacks, objetivo do grupo, quadros visuais, entre outros.

## **2.2. Colaboração**

Segundo Vreede e Briggs (2005), colaboração significa unir esforços para atingir um objetivo do grupo. Isto não significa que os membros do grupo concordem ou se sintam felizes com o objetivo, mas que, ao menos, por qualquer razão, trabalhem para a sua obtenção.

O desenvolvimento de software atual comumente é realizado por equipes, sendo necessário que seus membros colaborem entre si e com os demais envolvidos para atingir seu objetivo (PRESSMAN, 2016). A colaboração está explicitamente presente em um dos itens do manifesto ágil: colaboração com o cliente (BECK et al., 2001). Colaborar com o cliente significa que todos os envolvidos – patrocinadores, clientes, usuários e desenvolvedores – estão no mesmo time (HIGHSMITH e COCKBURN, 2001).

Segundo Jacobson (2002), uma equipe ágil reconhece que as capacidades individuais somadas à capacidade de colaborar são fatores principais de sucesso de um projeto. Pressman (2016, p. 83) ressalta que agilidade vai além de resposta a mudanças: “incentiva a estruturação e as atitudes em equipe que tornam a comunicação mais fácil (entre membros da equipe, entre o pessoal ligado à tecnologia e o pessoal da área comercial, entre os engenheiros de software e seus gerentes)”. A colaboração pode estar presente desde o processo de ideação e construção de conhecimento até a implementação em inovação e pode ocorrer por alianças estratégicas, por meio de cooperação e coordenação (GULATI et al., 2012).

Existem diversas teorias e modelos de colaboração, que mostram o porquê e como as pessoas trabalham em grupo. A Teoria dos jogos é uma das mais antigas, cujo foco são cenários de tomada de decisões estratégicas. Nela, são evidenciadas como decisões individuais afetam o resultado individual e do grupo. Além disso, estão presentes conceitos como auto interesse, matriz de ganho, incentivos e jogos de soma não zero (PIMENTEL e FUKS, 2011).

Para o Modelo 3C, colaboração é a combinação entre comunicação, coordenação e cooperação. Comunicação representa a troca de informações e mensagens entre as pessoas, coordenação é o gerenciamento das pessoas, atividades e recursos e cooperação é o compartilhamento do espaço de trabalho (FUKS et al., 2005). No Modelo 3C, a comunicação é feita de e para todos os integrantes do grupo, sendo voltada para a ação. As pessoas se comunicam ao mesmo tempo em que tomam decisões. Os membros do grupo se auto coordenam ao organizar suas atividades e lidam com conflitos de forma a otimizar a atuação do grupo, cooperando com o trabalho uns dos outros e utilizando comunicação. Durante a cooperação normalmente surgem questões que exigem comunicação e coordenação. As três atividades, portanto, se complementam e auto relacionam (PIMENTEL e FUKS, 2011).

Vreede, Massey e Briggs (2009) trazem o termo Engenharia da Colaboração, que trata a colaboração de forma mais holística, envolvendo tecnologias e guias para padrões de colaboração. Ela é especialmente útil quando trata de tarefas colaborativas de missão

crítica. No desenvolvimento de software, destaca negociação e especificação de requisitos, teste de usabilidade, inspeção dos requisitos e inspeção de código.

### 2.3. Modelos de maturidade em colaboração

Existem alguns modelos propostos de maturidade em colaboração. Entre eles, destacam-se: The Fast Reactive Extended Enterprise-Capability Assessment Framework (FREE-CAF) (LEBRUN et al., 1998); Collaboration Maturity Grid (CMG) (FRASER, FARRUKH E GREGORY, 2003); SAP AG (RAMASUBBU, KRISHNAN E KOMPALLI, 2005); Col-MM (BOUGHZALA e VREEDE, 2012) e ColabMM (MAGDALENO, 2006).

O modelo ColabMM (MAGDALENO, 2006) se mostrou mais aderente aos objetivos deste trabalho, pois trata da colaboração no contexto de processos organizacionais. Este relaciona maturidade em colaboração com conceitos de modelagem de processos de negócios e propõe um método para avaliar a maturidade em colaboração. Para desenvolver o método, os autores utilizaram tanto métodos de avaliação de maturidade de outras áreas do conhecimento, quanto modelos de modelagem de processos de negócio. No ColabMM são estabelecidos requisitos para atingir níveis de maturidade em colaboração que partem do Casual, em que nenhuma prática colaborativa está explícita, passando por planejado e perceptivo, até chegar a reflexivo, momento em que se observa a aplicação de diversas práticas.

As práticas definidas pelo ColabMM partem da proposição de requisitos de colaboração definidos por áreas de *Computer Supported Cooperative Work* (CSCW), cujas pesquisas objetivam a construção de sistemas colaborativos, e que os agrupam em Comunicação, Coordenação, Memória e Percepção. Estes requisitos estão aderentes ao modelo 3C de colaboração (FUKS et al., 2005), já discutido, dividindo a cooperação em dois subgrupos: Memória e Percepção.

Cada uma das práticas de colaboração do modelo ColabMM pode ser considerada requisito para atingimento do nível de maturidade correspondente e possuem, ainda, características específicas, apresentadas no Quadro 2. Observa-se que cada característica geral possui uma ou mais características específicas. De forma geral, no Nível 1 (Casual) a organização ainda não considera a colaboração de forma explícita, esperando que ocorra naturalmente. O Nível 2 (Planejado) inclui, principalmente, aspectos de planejamento da colaboração. No Nível 3 (Perceptivo), os membros já reconhecem suas responsabilidades e atividades que precisam executar para alcance dos objetivos do grupo. No Nível 4 (Reflexivo) a colaboração já acontece de forma plena, tendo, os membros do grupo, percepção do valor gerado pelo trabalho individual e coletivo e preocupação em divulgar o trabalho para o restante da organização (MAGDALENO, 2006).

**Quadro 2 - Classificação Detalhada das Práticas de Colaboração de acordo com os Aspectos de Apoio a Colaboração e os Níveis de Maturidade. Fonte: Adaptado de Magdaleno (2006)**

Nível	Aspecto	Prática	Característica
1 – Casual	Nenhum	Nenhuma	Nenhuma
2 – Planejado	Coordenação	Planejamento do trabalho em grupo	A coordenação é centralizada em um líder que planeja o trabalho e distribui as tarefas
	Percepção	Percepção social	Grupos de trabalho formalmente estabelecidos
	Comunicação	Planejamento da comunicação	Existe um plano de como realizar a comunicação com os intervenientes
	Memória	Integração dos produtos individuais	Os artefatos são integrados de forma a se complementarem como parte de um todo
3 – Perceptivo	Coordenação	Acompanhamento do trabalho	Todos conhecem as suas tarefas e as executam sem necessidade de distribuição por um líder
	Percepção	Percepção do processo	Os membros compreendem o processo de trabalho, seus objetivos, os passos necessários e possuem os conhecimentos necessários para execução das tarefas
	Comunicação	Distribuição das informações	Os membros do grupo conseguem cumprir o plano de comunicação, tendo acesso às informações necessárias para o cumprimento de suas tarefas
	Memória	Compartilhamento de conhecimento explícito	Estão definidos os artefatos necessários, que são compartilhados de forma explícita com o grupo
4 – Reflexivo	Coordenação	Avaliação	Existe <i>feedback</i> sobre resultado individual e do grupo no que diz respeito a qualidade dos produtos e do processo colaborativo adotado
	Percepção	Percepção da colaboração	Participantes possuem clareza de como colaborar e de como as atividades interagem
	Comunicação	Encerramento	Os membros capturam lições aprendidas, avaliam forças e fraquezas, extraem ideias para melhorias, celebram os resultados e os divulgam para a organização
	Memória	Compartilhamento de conhecimento tácito	O conhecimento tácito é compartilhado em forma de ideias, opiniões e experiências

Fonte: Resultados da pesquisa

Segundo Magdaleno (2006), as organizações possuem dificuldade em planejar e explicitar a colaboração nas atividades diárias do trabalho. O ColabMM possui grande preocupação com a explicitação da colaboração nas organizações, buscando tanto a definição mais clara dos processos colaborativos, quanto a divulgação destes processos entre os envolvidos.

## 2.4 Trabalhos correlatos

Fontana et al. (2014) destaca que os modelos de maturidade em desenvolvimento de software, como CMMI-DEV e ISO/IEC 15504 são voltados a processos, enquanto os métodos ágeis são voltados mais a pessoas do que processos. Conclui que nos métodos ágeis a definição de maturidade está menos relacionada a definições de processo e análises quantitativas e mais relacionada a capacidades mais subjetivas, como colaboração, comunicação, comprometimento, cuidado, compartilhamento e auto-organização.

Gren, Torkar e Feldt (2015) busca identificar se maturidade e agilidade estão conectadas, por meio de uma survey respondida por 65 participantes e conclui que equipes mais maduras também são equipes que praticam mais a agilidade e que, portanto, os métodos ágeis podem ser melhor executados se considerarem o desenvolvimento do grupo como fator de sucesso.

Magdaleno et al. (2015) descreve uma abordagem de otimização para determinar o melhor processo para um projeto específico de desenvolvimento de software que leva em consideração a colaboração entre membros por meio de algoritmos heurísticos e realiza um estudo em uma grande companhia de petróleo brasileira.

No contexto da pandemia do COVID-19, Diirr, Classe e Santos (2021) realizam uma *Survey* com profissionais, professores e estudantes que atuam em diversas áreas de TI sobre o apoio das Tecnologias de Informação e Comunicação (TICs) à colaboração, tendo como resultado amplo auxílio das ferramentas em todas as dimensões

colaborativas. Lisbôa et al. (2021) discutem sobre a adaptação necessária ao trabalho remoto no desenvolvimento de software durante a pandemia e fazem sugestões para melhoria da colaboração.

Observa-se que são escassos os estudos que tratam de colaboração no próprio desenvolvimento de software (Neiva et al., 2016). Apesar de existirem pesquisas recentes que tangenciam a proposta desta, não foi encontrado similar ao objeto desta pesquisa.

### **3. MÉTODO**

Utilizando por base as classificações sugeridas por Cauchick-Miguel (2018), esta pesquisa pode ser classificada quanto à sua natureza, como pesquisa teórico-conceitual, pois realiza discussões conceituais a partir da literatura. Quanto aos seus objetivos, classifica-se como pesquisa descritiva, uma vez que busca descrever os elementos encontrados. Sua abordagem é qualitativa, pela ênfase na interpretação dentro de um contexto, sendo suas variáveis categóricas e classificatórias. Quanto ao método, utiliza pesquisa bibliográfica e análise de contexto para identificar os elementos de colaboração relevantes ao processo de desenvolvimento de software.

Os procedimentos de pesquisa foram divididos em duas etapas: Pesquisa bibliográfica e Organização e Classificação, detalhadas a seguir.

#### **3.1. Pesquisa bibliográfica**

A pesquisa bibliográfica visou buscar na literatura embasamento teórico acerca do contexto em que a pesquisa está inserida, fundamentos teóricos dos assuntos abordados e trabalhos correlatos existentes. Para tanto, foram adotadas as seguintes tarefas:

Tarefa 1: Busca na literatura sobre desenvolvimento de software e colaboração: nesta tarefa buscou-se contextualizar os aspectos conceituais em que esta pesquisa está inserida, que culminou na abordagem do software e o que o diferencia de outros produtos, as formas que ele pode ser desenvolvido e a abordagem por métodos ágeis. Além disto, conceituou-se a colaboração e os aspectos que a envolvem;

Tarefa 2: Busca na literatura sobre elementos de colaboração em métodos ágeis: identificou-se, a partir da literatura, os elementos colaborativos presentes nos métodos ágeis mais utilizados. Os métodos híbridos, que são compostos por mais de um método ágil, competiram, em quantidade de utilização, com os métodos ágeis puros, porém, a partir da premissa de que os métodos híbridos são formados pelos elementos dos métodos que os compuseram, optou-se por estudar a literatura a partir dos métodos ágeis originais;

Tarefa 3: Busca na literatura sobre maturidade em colaboração: nesta tarefa, buscou-se na literatura pesquisas sobre modelos que tratam de aspectos de maturidade na colaboração e selecionou-se o mais viável para compor o objeto deste trabalho, detalhando seus requisitos de maturidade.

Para ambas as tarefas 2 e 3, buscou-se na base Scopus artigos que continham os termos “*agile*”, para incluir somente documentos que falem de métodos ágeis ou agilidade, “*team*”, para incluir somente os que tivessem alguma relação com os times, “*software development*” ou “*software project*” ou “*software engineering*” ou “*system development*” ou “*system project*” ou “*system engineering*”, para encontrar artigos que falem do processo de desenvolvimento de software, e “*collaborat\**”, para encontrar os

documentos que abordassem colaboração, e foram selecionados para análise os artigos mais aderentes ao assunto desta pesquisa.

### **3.2. Organização e classificação**

Tarefa 4: Classificação elementos colaborativos dos métodos ágeis: nesta tarefa buscou-se organizar e classificar os elementos encontrados na literatura utilizando análise de contexto. Primeiramente os autores buscaram na literatura original dos métodos Scrum, XP e Kanban quais elementos de colaboração estavam explícitos. Enquanto os elementos eram identificados foram também sendo identificadas categorias que facilitassem seu agrupamento, resultado nas categorias participantes, eventos, artefatos, características e regras. Os elementos encontrados em cada método foram, então, classificados nas categorias. Além disso, procedeu-se ao agrupamento dos itens semelhantes para identificar a presença em mais de um método ágil. Os elementos de colaboração encontrados nos métodos ágeis foram, por fim, classificados pelos mesmos critérios de CSCW utilizados no ColabMM (coordenação, comunicação, memória e percepção);

Tarefa 5: Referência cruzada colaboração métodos ágeis e maturidade em colaboração: buscou-se estabelecer uma relação entre os elementos de colaboração dos métodos ágeis com os elementos de maturidade em colaboração. Para tanto, elaborou-se uma matriz que relaciona os requisitos de colaboração do ColabMM com os 26 elementos dos métodos ágeis.

Ressalta-se que as tarefas 4 e 5 foram executadas e revistas pelos autores de forma colaborativa.

## **4. RESULTADOS E DISCUSSÃO**

A seguir são apresentados os resultados divididos em duas seções: Classificação elementos colaborativos dos métodos ágeis e Referência cruzada colaboração métodos ágeis e maturidade em colaboração.

### **4.1. Classificação elementos colaborativos dos métodos ágeis**

Como já discutido, a colaboração é intrínseca aos métodos ágeis de desenvolvimento de software, tendo em vista suas características flexíveis e interativas. Nesta seção, organiza-se, de acordo com o encontrado na literatura, os elementos colaborativos presentes nos métodos mais utilizados, classificando-os de acordo com as teorias apresentadas.

Obteve-se 26 itens a partir do agrupamento por similaridade dos elementos de colaboração encontrados na literatura, obtendo-se o Quadro 3. Observa-se que enquanto alguns elementos são específicos de um método ágil, como reuniões de retrospectiva (Scrum), histórias de usuário (XP) e reuniões a qualquer tempo (Kanban), outros são compartilhados entre dois ou mais métodos, como reuniões de planejamento (Scrum, XP e Kanban), lista de incrementos do produto (Scrum, XP e Kanban), quadro visual (XP e Kanban), reuniões de revisão do trabalho (Scrum e XP), entre outros.

### Quadro 3 – Elementos de colaboração em métodos ágeis

Elemento de colaboração	Scrum	XP	Kanban
Quadro visual		X	X
Repositório único de código		X	
Programação em par		X	
Decisões compartilhadas entre membros	X		
Reuniões de Planejamento	X	X	X
Reuniões diárias (progresso do trabalho)	X		
Reuniões semanais (progresso do trabalho)		X	
Reuniões a qualquer tempo (restrições)			X
Reuniões de revisão do trabalho	X	X	
Reuniões de retrospectiva/reflexão	X		
Feedback do mercado	X		
Feedback dos stakeholders	X	X	
Integrantes da equipe no mesmo espaço físico (sentar-se junto, espaço físico aberto)		X	
Cliente presente em tempo integral		X	
Equipe estável (sem muitas mudanças de membros)		X	
Identificação de desperdícios			X
Melhoria contínua	X	X	X
Definição de políticas (regras e guias do que deve ser feito e como)			X
Lista de incrementos do produto (backlog do produto, backlog da sprint, lista de histórias de usuário, fila de priorização)	X	X	X
Estabelecimento de marcos de acompanhamento (término da sprint, término do ciclo semanal)	X	X	
Priorização da qualidade (Estabelecimento de critérios de qualidade, testes)	X	X	
Time multidisciplinar/multifuncional (possui todos os conhecimentos necessários para o projeto)	X	X	
Auto-organização (própria equipe decide como organizar o trabalho)	X	X	
Empoderamento	X		
Transparência	X		
Desierarquização	X		

Fonte: Resultados da pesquisa

O Quadro 4 apresenta o resultado compilado da busca por elementos e práticas de colaboração, que relaciona os métodos ágeis encontrados em Schwaber e Sutherland (2017), Beck (1999), Beck (2004) e Al-Baik e Miller (2015) com a presença explícita dos elementos de colaboração em sua composição, agrupados pelos mesmos aspectos adotados nas práticas de colaboração do modelo ColabMM, já apresentados.

### Quadro 4– Elementos e práticas de colaboração em métodos ágeis por aspecto colaborativo

Método ágil	Coordenação	Comunicação	Memória	Percepção
<b>Scrum</b>	<ul style="list-style-type: none"> <li>▪Auto-organização</li> <li>▪Reuniões de Planejamento por <i>Sprint</i></li> <li>▪<i>Backlog</i> do Produto</li> <li>▪<i>Backlog</i> da <i>Sprint</i></li> </ul>	<ul style="list-style-type: none"> <li>▪Reuniões diárias</li> <li>▪Reuniões de Revisão</li> </ul>		<ul style="list-style-type: none"> <li>▪Reuniões de Retrospectiva</li> </ul>
<b>XP</b>	<ul style="list-style-type: none"> <li>▪Auto-organização</li> <li>▪Lista de histórias do usuário</li> <li>▪Reuniões trimestrais de planejamento</li> </ul>	<ul style="list-style-type: none"> <li>▪Reuniões semanais de revisão</li> <li>▪Programação em par</li> <li>▪Ambiente informativo do trabalho sendo realizado</li> <li>▪Cliente presente todo tempo</li> <li>▪Sentar-se junto</li> <li>▪Espaço de trabalho aberto</li> <li>▪Quadro visual</li> </ul>	<ul style="list-style-type: none"> <li>▪Histórias de usuário</li> <li>▪Código</li> <li>▪Teste</li> <li>▪Manuais</li> </ul>	
<b>Kanban</b>	<ul style="list-style-type: none"> <li>▪Quadro visual</li> <li>▪Fila de priorização</li> <li>▪Trabalho em progresso</li> <li>▪Ferramenta de medição de desempenho</li> <li>▪Definição de políticas</li> </ul>	<ul style="list-style-type: none"> <li>▪Quadro visual</li> <li>▪Identificação de gargalos</li> <li>▪Avatar</li> </ul>		<ul style="list-style-type: none"> <li>▪Identificação de desperdícios</li> <li>▪Aprendizado validado</li> </ul>

Fonte: Resultados da pesquisa

Observa-se que existem lacunas de elementos de colaboração em cada um dos métodos estudados, sendo eles, então, complementares. Enquanto Scrum e Kanban não possuem elementos e práticas no aspecto memória, mas possui nos demais aspectos, o método XP possui elementos em memória, mas não possui em Percepção. Este resultado teórico mostra uma possível motivação para adoção de métodos híbridos, corroborando com o resultado empírico encontrado em Digital.Ai (2021).

## 4.2. Referência cruzada colaboração métodos ágeis e maturidade em colaboração

Buscou-se estabelecer uma relação entre os elementos de colaboração dos métodos ágeis com os elementos de maturidade em colaboração, tendo como resultado o Quadro 5. Foi possível associar os elementos de agilidade e maturidade quase em sua totalidade, sendo que apenas para os elementos 12 (Feedback dos stakeholders, inclusive Product Owner e Product Manager) e 16 (Identificação de desperdícios) não foi encontrada possibilidade de associação.

Observa-se que diversos elementos colaborativos de agilidade puderam ser associados a muitos requisitos de maturidade em colaboração, como os elementos 6 (Reuniões diárias: progresso do trabalho), 7 (Reuniões semanais: progresso do trabalho) e 25 (Transparência), que tiveram mais de 15 correspondências cada. No sentido oposto também pode-se observar o mesmo fato, ou seja, diversos elementos de maturidade puderam ser associados a muitos elementos de agilidade, como os elementos “Auto-organização” e “As ideias, opiniões e experiências são compartilhadas sem restrições pelos membros do grupo”. Isto mostra que, utilizando-se como fator de ligação a colaboração, há forte relação entre os assuntos.

**Quadro 5 – Relação de elementos de agilidade com elementos de maturidade**

Nível	Requisito	Elemento Agilidade																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
1 – Casual	Nenhum																												
	Coordenação centralizada em um líder / define papeis e distribui responsabilidades															X	X			X					X				
2 – Planejado	Grupo de trabalho formalmente estabelecido														X	X	X						X						
	Existe um plano de como realizar a comunicação	X					X	X	X		X	X								X								X	
3 – Perceptivo	Artefatos são integrados com o todo	X								X										X	X	X							
	Auto-organização: integrantes conhecem as tarefas e se auto-organizam para executá-las	X		X	X	X	X	X	X	X	X					X			X	X	X	X		X	X	X		X	
	Processo de trabalho, objetivos e passos necessários são conhecidos pelos membros	X				X	X	X												X	X	X	X					X	
	Conhecimentos necessários para execução das tarefas são conhecidos pelos membros					X	X	X												X			X					X	
	Plano de comunicação é executado	X					X	X	X	X	X										X	X						X	
	Membros possuem acesso às informações necessárias	X									X				X	X				X	X							X	
	Artefatos esperados em cada tarefa estão definidos					X														X	X		X					X	
	Artefatos são compartilhados de forma explícita com o grupo	X					X	X	X	X											X	X						X	
	Existe <i>feedback</i> sobre resultado individual acerca da qualidade dos produtos						X	X	X	X	X				X		X					X						X	
	Existe <i>feedback</i> sobre resultado individual acerca do processo colaborativo						X	X	X	X	X				X		X											X	
4 – Reflexivo	Existe <i>feedback</i> sobre resultado do grupo acerca da qualidade dos produtos					X	X	X	X	X				X		X					X							X	
	Existe <i>feedback</i> sobre resultado do grupo acerca do processo colaborativo					X	X	X	X	X				X		X												X	
	Membros possuem clareza de como as atividades interagem	X		X		X	X	X								X				X					X			X	
	Membros refletem sobre lições aprendidas						X	X	X		X									X								X	
	Membros avaliam forças e fraquezas						X	X	X	X	X									X								X	
	Membros extraem ideias para melhorias						X	X	X	X	X									X								X	
	Membros celebram os resultados do grupo						X	X	X	X	X																	X	
	O resultado do grupo é divulgado para a organização									X	X	X									X							X	
	As ideias, opiniões e experiências são compartilhadas sem restrições pelos membros do grupo			X	X	X	X	X	X	X	X					X				X						X	X	X	X

**Elementos Agilidade**

- |  |   |
|--|---|
| <ol style="list-style-type: none"> <li>1. Quadro visual</li> <li>2. Repositório único de código (Git e semelhantes)</li> <li>3. Programação em par</li> <li>4. Decisões compartilhadas entre membros</li> <li>5. Reuniões de Planejamento</li> <li>6. Reuniões diárias (progresso do trabalho)</li> <li>7. Reuniões semanais (progresso do trabalho)</li> <li>8. Reuniões a qualquer tempo (restrições)</li> <li>9. Reuniões de revisão do trabalho</li> <li>10. Reuniões de retrospectiva/reflexão</li> <li>11. Feedback do mercado</li> <li>12. Feedback dos stakeholders (inclusive Product Owner e Product Manager)</li> <li>13. Integrantes da equipe no mesmo espaço físico (sentar-se junto, espaço físico aberto)</li> </ol> | <ol style="list-style-type: none"> <li>14. Cliente presente em tempo integral</li> <li>15. Equipe estável (sem muitas mudanças de membros)</li> <li>16. Identificação de desperdícios</li> <li>17. Melhoria contínua</li> <li>18. Definição de políticas (regras e guias do que deve ser feito e como)</li> <li>19. Escopo/Requisitos/Lista de incrementos do produto</li> <li>20. Estabelecimento de marcos de acompanhamento (término da sprint, término do ciclo semanal)</li> <li>21. Estabelecimento de critérios de qualidade, testes, Priorização da qualidade</li> <li>22. Time multidisciplinar/multifuncional (possui todos os conhecimentos necessários para o projeto)</li> <li>23. Auto-organização (própria equipe decide como organizar o trabalho)</li> <li>24. Empoderamento</li> <li>25. Transparência</li> <li>26. Desierarquização</li> </ol> |
|--|---|

Fonte: Resultados da pesquisa

Todavia, não é possível indicar os elementos de agilidade como necessários para requisitos de maturidade uma vez que os primeiros contribuem para o atingimento do

segundo, mas não são obrigatórios, pois o requisito de maturidade pode ser atendido por outros elementos de agilidade. Os elementos de agilidade, porém, podem ser encarados como possibilidades que contribuirão para a equipe atender determinado requisito de maturidade. Por exemplo, a utilização de Quadro Visual (elemento de agilidade) contribui para que os membros tenham melhor clareza de como as atividades interagem (elemento de maturidade), porém, existem outros elementos que podem contribuir para alcançar esta clareza, como programação em par e reuniões de planejamento.

Mesmo estando todos os elementos de agilidade associados a um requisito de maturidade, ainda não é possível afirmar, apenas por este fato, que se atendeu o requisito de maturidade. Tomando-se o mesmo exemplo, mesmo que a equipe utilize quadro visual, programação em par, reuniões de planejamento e os demais elementos, é possível, ainda assim, que os membros não possuam clareza de como as atividades interagem.

Propôs-se então, uma associação entre os elementos de colaboração em métodos ágeis e requisitos de maturidade em colaboração. Porém, conclui-se que estes dois assuntos estão relacionados, mas não correlacionados.

## **5. CONCLUSÃO**

Foram encontrados diversos elementos colaborativos nos métodos ágeis, corroborando com a literatura que sugere a presença de colaboração nestes métodos. Os elementos encontrados foram classificados e agrupados em 26, sendo que alguns dos elementos, após agrupamento, estão presentes em mais de um deles. Classificou-se os elementos nos aspectos de colaboração coordenação, comunicação, memória e percepção, resultando em diversos elementos colaborativos em cada aspecto. Porém, foram observadas lacunas: enquanto Scrum e Kanban não possuem elementos e práticas no aspecto memória, possuindo nos demais aspectos, o método XP possui elementos em memória, mas não possui em percepção. Este resultado teórico mostra uma possível motivação para adoção de métodos híbridos, corroborando com o resultado empírico encontrado em Digital.Ai (2021).

Acerca da maturidade, não foi encontrada pesquisa que trate especificamente de maturidade em colaboração no processo de desenvolvimento ágil de software, porém o modelo ColabMM (MAGDALENO, 2006) mostrou-se capaz de atender diversos processos corporativos, inclusive este. Buscou-se, então, relacionar os requisitos de maturidade do ColabMM com os elementos colaborativos presentes nos métodos ágeis estudados, o que foi possível em parte. Encontrou-se associação entre os elementos de colaboração em métodos ágeis e requisitos de maturidade em colaboração. Porém, conclui-se que estes dois assuntos estão relacionados, mas não correlacionados, uma vez que não foi possível estabelecer uma relação causal entre os elementos de colaboração dos métodos ágeis e o atingimento dos níveis de maturidade em colaboração.

Este estudo contribui para a área ao identificar os elementos colaborativos presentes nos métodos ágeis mais utilizados, classificando-os, agrupando-os e relacionando-os com requisitos de maturidade em colaboração. Ele está limitado, todavia, à literatura original dos métodos ágeis e a experiência dos autores no assunto, utilizada na classificação dos elementos. Entretanto, seu resultado permite a realização de estudos empíricos que utilizem os elementos encontrados para avaliar a colaboração em equipes que utilizem métodos ágeis de desenvolvimento de software.

## REFERÊNCIAS

- AL-BAIK, O.; MILLER, J. The kanban approach, between agility and leanness: a systematic review. *Empiric Software Eng.* n. 20. p. 1861-1897. New York, 2015.
- AL-SAQQA, SAMAR, SAMER SAWALHA, HIBA ABDELNABI. "Agile software development: Methodologies and trends." *International Journal of Interactive Mobile Technologies* 14.11. 2020.
- ANDERSON, D. J. *Kanban: successful evolutionary change for your technology business.* Blue Hole Press. Seattle, 2010.
- BECK et al. Manifesto para desenvolvimento ágil de software. 2001. Disponível em: <<http://agilemanifesto.org>>, acesso em 09/06/2018.
- BECK, K. Embracing change with extreme programming. *Computer*, 32(10), 70–77. 1999.
- BECK, K. *Extreme programming explained: embrace change.* 2nd ed. 2004.
- BOEHM, B. Get ready for agile methods, with care. *Computer*, vol. 35, no. 1. IEEE Computer Society. Janeiro, 2002.
- BOUGHZALA, I.; VREEDE, G. A Collaboration Maturity Model: Development and Exploratory Application, *Proceedings of the 45th Hawaii International Conference on System Sciences.* Hawaii, 2012.
- CAUCHICK-MIGUEL, P. A. (Coord.). *Metodologia de pesquisa em engenharia de produção e gestão de operações.* 3ª ed. Rio de Janeiro: Elsevier, 2018.
- COCKBURN, A. *Agile Software development: the cooperative game.* 2nd edition. Addison-Wesley. 2006.
- DIGITAL.AI. 15th Annual State of Agile Report. 2021. Disponível em: <<https://itnove.com/wp-content/uploads/2021/07/15th-state-of-agile-report.pdf>>. Acesso em 20/03/2022.
- DIIRR, B; CLASSE, T. M.; SANTOS, P. S. M. Uso de TICs para Apoio`a Colaboração Durante a COVID-19. *Simpósio Brasileiro de Sistemas Colaborativos (SBSC) 2021, Evento Online.* Porto Alegre: Sociedade Brasileira de Computação, 2021. DOI: <https://doi.org/10.5753/sbsc.2021.16016>
- DUDHAT, A., & ALI ABBASI, M. Discussion of Agile Software Development Methodology and its Relevance to Software Engineering. *ADI Journal on Recent Innovation*, 3(1), 105–114. 2021.
- FRASER, P.; FARRUKH, C.; GREGORY, M. Managing product development collaborations – A process maturity approach. *Proceedings of the Institution of Mechanical Engineers*, 217, 11, pg. 1499-1519. 2003.
- FONTANA, R. et al. Processes versus people: How should agile software development maturity be defined? 2014.
- FUKS, H. et al. Applying the 3C model to groupware development. *International Journal of Cooperative Information Systems.* Vol. 14. 2005.

- GREN, L.; TORRAR, R.; FELDT, R. Group maturity and agility, are they connected? - A survey study. 2015 41st Euromicro Conference on Software Engineering and Advanced Applications. Funchal, 2015.
- GULATI et al. The Two Facets of Collaboration: Cooperation and Coordination in Strategic Alliances. *Academy of Management Annals* 6. Harvard Business School. Estados Unidos, 2012.
- HIGHSMITH, J. E COCKBURN, A. Agile software development: the business of innovation. *Computer*, vol. 34, no. 9, pp. 120-127. IEEE Computer Society. Setembro, 2001.
- JACOBSON, I. A Resounding Yes to Agile Process – But Also More. *Cutter IT Journal*, v. 15, n. 1, p. 18-24. 2002.
- LEBRUN, J. et al. *Capability Assessment Framework – CAF, Volume 1*. 1998.
- LISBÔA, D. A. S.; ROCHA, T. A.; MACHADO, L. S.; CALDEIRA, C. M.; SOUZA, C. R. B. Um Estudo Observacional sobre as Adaptações ao Trabalho Remoto no contexto da Pandemia de COVID-19. *Simpósio Brasileiro de Sistemas Colaborativos (SBSC) 2021, Evento Online*. Porto Alegre: Sociedade Brasileira de Computação, 2021. DOI: <https://doi.org/10.5753/sbsc.2021.16024>.
- MAGDALENO, A. *Explicitando a Colaboração em Organizações através da Modelagem de Processos de Negócio*. Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2006.
- MAGDALENO, A. et al. Collaboration optimization in software process composition. *Journal of Systems and Software*. Volume 103. Pages 452-466. 2015.
- NEIVA et al. Interoperability Requirement to Enhance Collaboration in Software Product Lines: A Systematic Mapping. *13º Simpósio Brasileiro de Sistemas Colaborativos*. SBC. Rio Grande do Sul, 2016.
- PIMENTEL, M.; FUKS, H. *Sistemas Colaborativos*. Elsevier. Rio de Janeiro, 2011.
- PRESSMAN, R. S. *Engenharia de software: uma abordagem profissional*. 8ª edição. Porto Alegre: AMGH, 2016.
- RAMASUBBU, N.; KRISHNAN, M.; KOMPALLI, P. A process maturity framework for managing distributed development, *IEEE Software*, May/June, p. 80-86. 2005.
- RIGBY, D. K.; SUTHERLAND, J.; TAKEUCHI, H. Embracing Agile: How to Master the Process That's Transforming Management. *Harvard Business Review* 94, no. 5. Maio, 2016.
- SCHWABER, K.; SUTHERLAND, J. *Guia do Scrum: um guia definitivo para o Scrum: As regras do jogo*. 2017. Disponível em: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>, acesso em 04/11/2019.
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. Pearson Prentice Hall. São Paulo, 2011.
- STAIR, R. M; REYNOLDS, G. W. *Princípios de Sistemas de Informação*. Tradução da 11ª edição norte americana. Cengage Learning. São Paulo, 2015.

- STRODE, D., DINGSØYR, T. & LINDSJORN, Y. A teamwork effectiveness model for agile software development. *Empir Software Eng* 27, 56. 2022.
- VREEDE, G.J. DE; BRIGGS, R.O. Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. *Proceedings of the Hawaii International Conference on System Science*. IEEE Computer Society Press. Los Alamitos, 2005.
- VREEDE, G.; MASSEY, A. P.; BRIGGS, R. O. Collaboration Engineering: Foundations and Opportunities. *Journal of the Association of Information Systems*. v. 10, article 2. 2009.