

VINICIUS CANHISARES MAZUCA

O Papel da Comunicação no Desenvolvimento de Software Ágil:

Uma Revisão Sistemática da Literatura

São Paulo
2018

VINICIUS CANHISARES MAZUCA

O Papel da Comunicação no Desenvolvimento de Software Ágil:

Uma Revisão Sistemática da Literatura

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo – como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software.

Orientadora: Profa. Dra. Denise Lazzeri Gastaldo Bombonatti

São Paulo
2018

Mazuca, Vinicius Canhisares

O Papel da Comunicação no Desenvolvimento de Software Ágil: Uma Revisão Sistemática da Literatura / V. C. Mazuca -- São Paulo, 2018.
64 p.

Monografia (MBA em Tecnologia de Software) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.Abordagens Ágeis 2.XP (eXtreme Programming) 3.Scrum 4.Revisão Sistemática de Literatura I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

Dedico este trabalho aos meus pais, por me influenciarem tanto a crescer
e aos meus irmãos, que são a mim tão preciosos.

AGRADECIMENTOS

Primeiramente, agradeço a Deus pela fortaleza empreendida na dura jornada da maturação espiritual e profissional, sendo ela infinitamente maior que meu mérito.

À minha orientadora e professora Dra. Denise Denise Lazzeri Gastaldo Bombonatti, por sua paciência, estímulo e suporte sempre presentes na condução do trabalho. Sua organização foi inspiradora desde as primeiras aulas que a presenciei conduzindo e considero fundamental para meu crescimento.

À meus pais, irmãos, familiares e amigos que me proporcionaram apoio paciente e amor incondicional.

À todos que colaboraram direta ou indiretamente na execução deste trabalho.

RESUMO

O objetivo deste trabalho foi complementar a revisão sistemática de literatura sobre o papel da comunicação no desenvolvimento de software realizada por Hummel, Rosenkranz e Holten (2013), que considerou referências até ano de 2012/2013. Este trabalho estende o estudo para o período entre 2012/2013 até final de 2017. A intenção é investigar, através da execução de uma SLR – *Systematic Literature Review*, ou Revisão Sistemática da Literatura, em português, se a comunicação continua apresentando a importância detectada pelos autores no desenvolvimento de software ágil e, caso tenha mudado, qual foi a alteração. A SLR foi executada seguindo as mesmas regras aplicadas pelos autores originais. Foram selecionados 131 artigos relevantes às questões de pesquisa e deles discutiu-se a essencialidade da comunicação à condução de projetos de desenvolvimento de software ágeis. Concluiu-se que a comunicação persiste desempenhando um papel fundamental nos projetos de desenvolvimento de software envolvendo abordagens ágeis e as práticas apresentadas no trabalho original com maior relevância apresentaram neste trabalho ainda maior número de resultados.

Palavras-chave: Abordagens Ágeis, XP (eXtreme Programming), Scrum, Revisão Sistemática da Literatura.

ABSTRACT

The purpose of this work is to update the systematic review of the literature about the role of communication in software development by Hummel, Rosenkranz and Holten (2013) from the time they ended the research until 2017. The intention is to investigate, through another SLR – Systematic Literature Review, if the communication continues presenting the role in the agile software development found by the authors and, if it has changed, what was the change. 131 articles were found relevant to research questions, selected, and used to discuss the essentiality of communication to the management of agile software development projects. It was concluded that communication persists playing a fundamental role in software development projects involving agile approaches and the practices presented in the original work with greater relevance presented even more results in this work.

Keywords: Agile Approaches, XP (eXtreme Programming), Scrum, Systematic Literature Review.

SUMÁRIO

| | | |
|-----------|---|-----------|
| 1. | INTRODUÇÃO | 9 |
| 1.1 | MOTIVAÇÕES..... | 9 |
| 1.2 | JUSTIFICATIVA | 12 |
| 1.3 | OBJETIVO..... | 13 |
| 1.4 | METODOLOGIA DO TRABALHO | 14 |
| 1.5 | ORGANIZAÇÃO DO TRABALHO | 14 |
| 2. | INTRODUÇÃO TEÓRICA | 15 |
| 2.1 | ABORDAGENS ÁGEIS..... | 15 |
| 2.1.1 | História do Ágil | 15 |
| 2.1.2 | Scrum | 16 |
| 2.1.3 | XP – Extreme Programming | 21 |
| 2.2 | COMUNICAÇÃO | 24 |
| 2.3 | REVISÃO SISTEMÁTICA DE LITERATURA..... | 26 |
| 3. | A PESQUISA DE HUMMEL, ROSENKRANZ E HOLTEN | 29 |
| 3.1 | INTRODUÇÃO..... | 29 |
| 3.2 | METODOLOGIA DE PESQUISA..... | 29 |
| 3.3 | RESULTADOS DA PESQUISA DE HUMMEL, ROSENKRANZ E HOLTEN | 32 |
| 4. | EXTENSÃO DO TRABALHO DE HUMMEL ROSENKRANZ E HOLTEN | 39 |
| 4.1 | PROTOCOLO DA SLR..... | 39 |
| 4.2 | RESULTADOS DA SLR..... | 45 |
| 5. | CONCLUSÕES | 56 |
| | REFERÊNCIAS..... | 58 |

1. INTRODUÇÃO

Este capítulo apresenta as motivações, justificativa, objetivos e a metodologia para o trabalho.

1.1 MOTIVAÇÕES

A presença de elementos computacionais no cotidiano é quase despercebida devido ao costume, ao passo que, há pouco tempo, não era nem evidente nem concebida, apenas sonhada. Suas origens, contudo, não são tão próximas como aparentam. Fonseca Filho (2007) denominou como a “pré-história” da tecnologia computacional o período que se iniciou com os primeiros registros de escrita numérica dos babilônios em 4000 a.C. e que persistiu até a criação do ábaco e outros dispositivos mecânicos para auxílio nos cálculos numéricos a partir do século XVII d.C. Explicou em sua obra que a relevância do período para a computação digital se deu através de diversos conceitos teóricos originados nesse interim.

Após esse longo período, a evolução da tecnologia de hardware aconteceu com grande velocidade, de forma que o século XX teve a glória de materializar uma diversidade de dispositivos com capacidade de processar informações, criar softwares – que acompanharam fundamentalmente essa evolução – para instruir tais dispositivos e consolidar a arquitetura de computadores de von Neumann que, desde sua proposta em meados de 1940, tem sido utilizada quase sem alterações na computação atual (REGAN, 2008).

Em termos mais concretos, nas últimas 8 décadas a tecnologia computacional evoluiu de aplicações científicas e militares a aplicações comerciais e de grande porte, passando posteriormente a ser construída em aplicativos pessoais nos microcomputadores, para depois utilizar os conceitos de redes simples (em arquiteturas cliente-servidor), e chegando finalmente aos softwares distribuídos, e às redes mais

complexas como *groupware*, internet e cloud, por exemplo. Atualmente se discutem quais os caminhos nos quais a computação ubíqua ou pervasiva segue rumo à internet das coisas (GALEGALE et al., 2016).

Das transições acima, resultou que a ascensão dos computadores do laboratório (limitados ao uso de engenheiros elétricos, em maioria) ao domínio público gerou uma nova profissão: o desenvolvimento de software se tornou uma atividade de muitos e todos os tipos de empresas passaram a contratar programadores (WIRTH, 2008).

A produção de software tornou-se, então, “industrializada”, seu desenvolvimento passou a ser realizado em escala e por diversas e distintas equipes através de componentes reutilizáveis, de forma que o produto de software gerado tem sido comparado a um bem tangível (GREENFIELD, 2004).

Como cada geração de computadores produzida trouxe consigo maior poder de processamento, o mundo se deparou com um ambiente propício para o desenvolvimento de softwares cada vez mais sofisticados e a tecnologia passou a ser aplicada em tarefas ainda mais complicadas, o que facilitou muito a vida dos usuários, mas por outro lado, gerou programas praticamente impossíveis de serem analisados ou desenvolvidos por apenas um indivíduo (WIRTH, 2008). Além disso, o avanço da computação trouxe mudanças sociais, culturais e econômicas, de forma que as indústrias e os negócios passaram a exigir sistemas mais intrincados, ao passo que os programadores careciam de recursos para suprir tais demandas.

Essa tendência potencializou a queda da qualidade e confiabilidade dos softwares, aumento nos excedentes de orçamento, cronograma e conseqüentemente, a insatisfação dos clientes. Deflagrou-se, durante a primeira conferência de engenharia de software da OTAN entre 1968 e 1969 na Alemanha, o período da crise de software (REGAN, 2008), que levou ao nascimento da disciplina de engenharia de software, trazendo metodologias para o processo de desenvolvimento e evitando dispêndios excessivos nos projetos.

Os processos da engenharia de software trouxeram diversos métodos para manufatura, gestão e controle, com princípios que evoluíram nas últimas décadas,

surgindo assim diversas técnicas, ferramentas e abordagens para desenvolvimento de software. Atualmente, destaca-se a abordagem ágil, que se contrapõe em relação à linearidade dos modelos de ciclo de vida tradicionais (*waterfall* ou cascata) em termos de flexibilidade, comunicação, colaboração e pelo fato de situar-se num ambiente produtivo em que se espera evolução e alteração dos requisitos no decorrer do projeto, ao passo que a metodologia tradicional exige requisitos a definidos logo de início.

As abordagens ágeis vieram para incorporar diversas lições práticas aprendidas no desenvolvimento de software durante anos, entre elas, a substituição do controle rigoroso por maior autonomia e flexibilidade, e a aplicação de um modo de reagir rapidamente à mudanças de requisitos através de lançamentos de versões menores, encorajando feedbacks constantes (HUMMEL; ROSENKRANZ; HOLTEN, 2013). Dessa forma, projetos de desenvolvimento de software utilizando a abordagem ágil, tendem a apresentar maiores chances de sucesso e consequentemente, maior satisfação dos clientes.

Dados do CHAOS Summary de 2010, um relatório do StandishGroup que analisa projetos de TI, apresentam maiores taxas de sucesso e menores de falhas em projetos que usam a abordagem ágil em comparação com o modelo de cascata tradicional num período entre 1994 e 2004, conforme apresentado na Figura 1 (CHAOS, 2010).

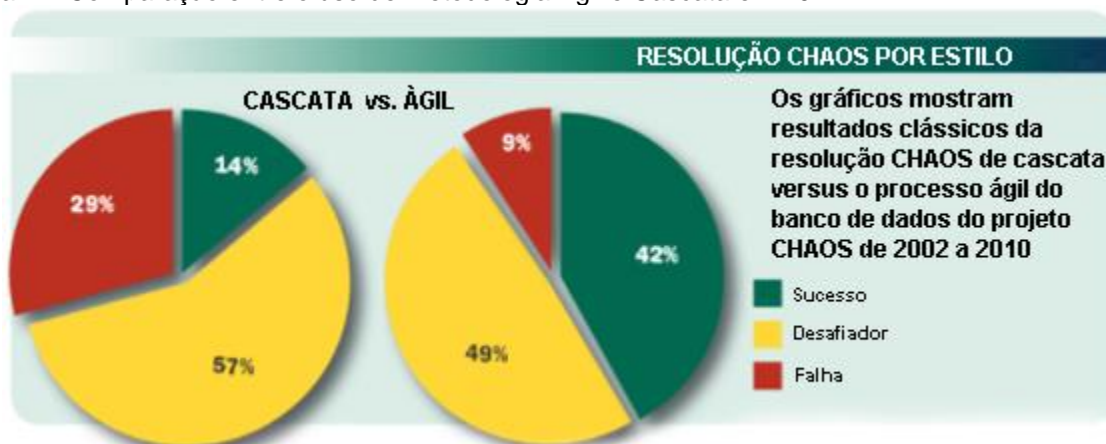
Figura 1 – Comparação entre o uso de metodologia Ágil e Cascata em 2010.



Fonte: Traduzido de CHAOS, 2010.

Em 2011, o relatório CHAOS apresentou um resultado constatando que o sucesso em projetos ágeis é três vezes maior que em projetos que utilizam o tradicional modelo cascata, conforme mostra a Figura 2 (CHAOS, 2011).

Figura 2 – Comparação entre o uso de metodologia Ágil e Cascata em 2011.



Fonte: Traduzido de CHAOS, 2011.

Esses dados permitem a compreensão do porquê, de acordo com Kapitsaki e Christou (2015), as abordagens ágeis têm sido adotadas por muitos líderes de indústrias mundiais, incluindo a Microsoft, IBM, HP, Oracle, Federal Reserve Bank.

Baseando-se nessa tendência e num aspecto tido como fundamental da abordagem ágil, e em qualquer framework utilizado em projetos gerais, este trabalho visa investigar o papel da comunicação, reconhecidamente importante para o sucesso dos projetos, em abordagens ágeis nas pesquisas recentes.

1.2 JUSTIFICATIVA

Uma das principais diferenças entre o desenvolvimento ágil e as abordagens tradicionais é o papel central da comunicação entre os envolvidos no processo de criação do produto de software. Tal proposição pode ser observada tanto nos princípios quanto nas práticas que estão descritos no manifesto Ágil. No entanto, até o ano em que Hummel, Rosenkranz e Holten (2013) realizaram a pesquisa, poucos estudos tentaram explicar a relação entre o desenvolvimento ágil e a comunicação.

A importância desse trabalho é a integração dos diversos estudos referentes ao desenvolvimento ágil que foram publicados desde 2012/2013 para verificar se o papel da comunicação continua sendo crítico e tentar explicar as razões para tanto e identificar

quais alterações foram sensibilizadas durante o período da atualização até o final de 2017.

Além disso vale mencionar que, em comparação com o trabalho original, este trabalho descreve com maior minúcia os processos para a realização da revisão sistemática da literatura, facilitando pesquisas posteriores a serem realizadas por outros autores.

1.3 OBJETIVO

O objetivo deste trabalho é estender a pesquisa sobre o papel da comunicação nas abordagens ágeis até os dias atuais, especificamente considerando as abordagens Scrum e XP, realizada por Hummel, Rosenkranz e Holten (2013), que originalmente consideraram pesquisas publicadas até 2013 (os autores encontraram pesquisas publicadas entre 2000 e 2013). A atualização é feita até o início do ano de 2018. O intuito é avaliar se a comunicação continua sendo um tema relevante nas pesquisas recentes, além de construir um mapa atualizado das práticas ágeis vigentes que possuem maior foco na melhoria da comunicação. As questões de pesquisa deste trabalho podem ser descritas como:

RQ1: Qual é o papel da comunicação em projetos de desenvolvimento de software empregando práticas ágeis?

RQ2: A questão anterior continua sendo um tema frequentemente abordado nas pesquisas realizadas após o período (2012/2013) em que o trabalho original foi realizado?

RQ3: Quais aspectos foram alterados com base na pesquisa original?

1.4 METODOLOGIA DO TRABALHO

Observando que uma extensão de um trabalho original pode ser vista como uma espécie de continuação do mesmo, as metodologias usadas devem ser, no mínimo, semelhantes. Portanto é realizada uma SLR que é um processo metodológico para coletar, selecionar, analisar e integrar os resultados de diversas pesquisas acadêmicas e suas respectivas publicações.

O processo é explicado com maiores detalhes no Capítulo 2. Nos capítulos subsequentes explica-se a forma como a SLR foi empregada no trabalho original e neste.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em cinco capítulos, um anexo e as referências bibliográficas.

O Capítulo 1 apresenta de forma sucinta as motivações, justificativas, objetivos, metodologia e estrutura organizacional do trabalho.

O Capítulo 2 apresenta uma rápida base teórica com as principais referências sobre o histórico da abordagem ágil, as abordagens XP e Scrum, e comunicação.

O Capítulo 3 mostra um resumo da SLR e resultados obtidos pelo trabalho de Hummel, Rosenkranz e Holten.

O Capítulo 4 apresenta os resultados obtidos neste trabalho.

O Capítulo 5 mostra uma breve discussão sobre a realização do trabalho bem como seus principais tópicos conclusivos.

As REFERÊNCIAS mostram uma lista das fontes pesquisadas e usadas como respaldo para a escrita deste trabalho.

2. INTRODUÇÃO TEÓRICA

Uma vez definidos o tema a ser estudado, a metodologia utilizada e as questões levantadas nesse trabalho, detalha-se neste capítulo um referencial teórico que considera o histórico das abordagens ágeis utilizadas na pesquisa, a importância da comunicação e o processo de revisão sistemática da literatura (SLR – *Systematic Literature Review*) da qual se faz uso.

2.1 ABORDAGENS ÁGEIS

Esta seção tem a finalidade de apresentar o histórico das abordagens ágeis e detalhar as duas abordagens utilizadas na pesquisa, o Scrum e o XP.

2.1.1 História do Ágil

Conforme mencionado previamente, as mudanças globais trouxeram, especialmente na década de 90, a evolução de abordagens leves de desenvolvimento de software, como forma de reação àquelas cujos processos são construídos sob regulamentações complexas e sobrecarregadas, influência do tradicional modelo de ciclo de vida cascata, *waterfall*.

Tais abordagens designadas leves traziam consigo a promessa de poucas regras, menos documentações e práticas fáceis de serem seguidas, implicando maior simplicidade ao processo. Dentre as novas metodologias que surgiram nessa época, se encontravam a RAD (*Rapid Application Development*) de 1991, a RUP (*Rational Unified Process*) de 1994, além das abordagens Scrum de 1995 e XP (*Extreme Programming*) de 1996.

O manifesto ágil, documento fundamentador dos valores ágeis e que popularizou a abordagem, foi criado em fevereiro de 2001 num *resort* em Utah, onde 17 profissionais do desenvolvimento de software se encontraram para discutir as diferentes

características dos métodos de desenvolvimento existentes no mercado. Nessa reunião alcançaram um consenso a respeito do conceito de “leveza” nos processos, em contraposição, ao aparente “peso” das formalizações exageradas em documentações e regulamentações.

O resultado se apresentou na forma de quatro valores Ágeis, que atribuem maior ênfase à flexibilidade e colaboração, em oposição à rigidez dos processos e planejamento clássicos, conforme seguem:

- Os indivíduos e suas interações estão acima dos procedimentos e ferramentas;
- O funcionamento do software está acima da documentação abrangente;
- A colaboração com o cliente está sobre a negociação e o contrato;
- A capacidade de resposta à mudança está acima de um planejamento pré-estabelecido;

Vale ressaltar que os valores acima não significam a desvalorização dos processos e documentações, mas representam uma visão mais flexível e informal, promovendo uma maior qualidade de comunicação tanto entre os membros da própria equipe do projeto quanto com os usuários/clientes do produto final, maior facilidade de adaptação às alterações dos requisitos e entregas mais constantes do software funcionando. Adicionalmente, doze princípios de desenvolvimento suportados pelos valores descritos no manifesto foram estabelecidos para pormenorizar a perspectiva “leve” da abordagem.

2.1.2 Scrum

O Scrum nasceu do trabalho de Takeuchi e Nonaka (1986), que propuseram um estilo de gerenciamento de projetos em empresas automotivas com equipes pequenas e multidisciplinares, associando-as à formação “Scrum” nos jogos de *rugby*. A teoria proposta pelos professores foi implementada na Easel Corporation em 1993 e

formalizada por Schwaber (1995), aplicando ao desenvolvimento de software os conceitos Scrum originalmente criados para a indústria fabril.

Conforme o guia de Sutherland e Schwaber (2016), o Scrum pode ser definido como um *framework* que orienta a condução de um time ágil, e que pode ser utilizado para resolver problemas complexos enquanto realizam entregas de forma criativa e produtiva, com o maior valor possível.

O Scrum é fundamentado pela teoria de controle empírico do processo, ou seja, é baseado na crença do aprendizado pela experiência e na tomada de decisão baseado no conhecimento adquirido por ela. Por sua vez, o empirismo no controle de processos é mantido pelos pilares da transparência, inspeção e adaptação. Isso quer dizer que os aspectos significantes do processo devem ser visíveis aos responsáveis pelos resultados, os usuários do Scrum devem inspecionar frequentemente os artefatos e os progressos em relação aos objetivos da sprint para detectar variações indesejadas e, caso um inspetor perceba um ou mais desvios fora dos limites aceitáveis, solicitar ou realizar os respectivos ajustes.

A Figura 2 mostra um pôster contendo os cinco valores Scrum, adicionado posteriormente ao guia Sutherland e Schwaber (2016), complementando e amplificando o poder da abordagem ao promover um suporte para tomadas de decisões.

Figura 2 – Cinco valores Scrum.



Fonte: Traduzido e adaptado de West (2016).

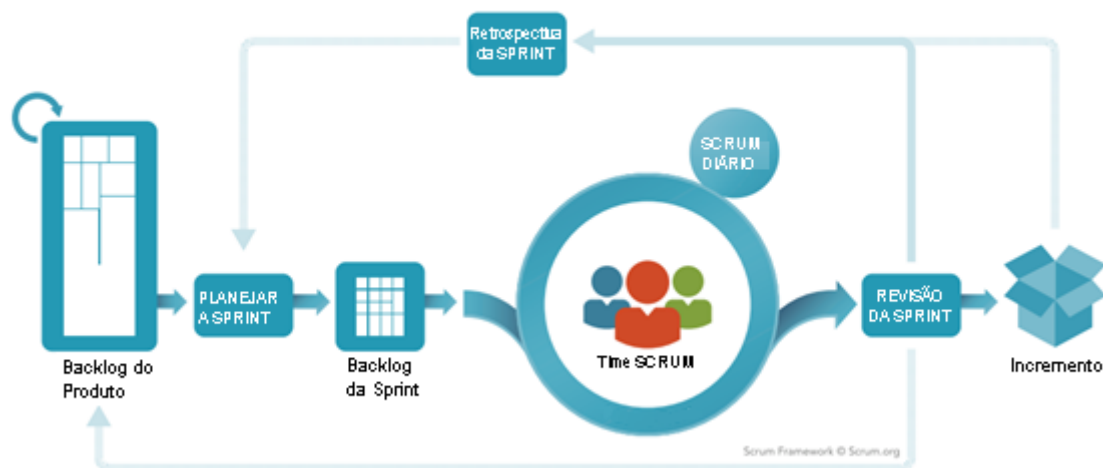
O time Scrum, composto pelos *Product Owner* (dono do produto), Time de Desenvolvimento e *Scrum Master* (Mestre Scrum), é multifuncional e auto organizável, ou seja, independe de outras equipes pois possui todas as competências para completar um trabalho completo e tem autonomia para escolher a melhor forma de completar um trabalho.

Cada membro do time Scrum possui suas funções e papéis que, de forma resumida, podem ser definidas como: o *Product Owner* deve garantir o valor do produto e do trabalho do time de desenvolvimento através do gerenciamento do backlog do produto; o time de desenvolvimento gera as entregas de versões usáveis e incrementa o produto final ao termino de cada *sprint*; o *Scrum Master* é responsável pela garantia da compreensão e aderência às regras do Scrum pelas equipes envolvidas, tanto o time Scrum quanto os de fora.

O Scrum utiliza alguns eventos de durações fixas e determinadas com o intuito de criar rotinas regulares e diminuir reuniões de caráter imprevisto e excepcional. O principal deles é a *sprint*, um *container* para todos os outros eventos, que deve durar no máximo um mês, período em que se cria uma versão incremental do produto, o “pronto” (ou “*done*”, em inglês). As sprints iniciam com uma reunião de planejamento da *sprint*, possuem reuniões diários e Trabalho de Desenvolvimento e finalizam com os eventos de revisão da *sprint* e retrospectiva da *sprint*, marcam o início da próxima *sprint*, com a mesma duração prevista. A Figura 3 representa um diagrama contendo as sequências dos eventos descritos e os artefatos que alteram.

Figura 3 – Diagrama do framework Scrum

FRAMEWORK SCRUM



Fonte: Traduzido e adaptado de Schwaber e Sutherland, 2017.

A reunião de planejamento da *sprint* define de forma colaborativa, com a participação do time, baseado no *backlog* do produto: os objetivos, o incremento e o trabalho a ser realizado para a entrega do incremento ao final dessa *sprint*. Para *sprint* de um mês, define-se o máximo de oito horas para a reunião de Planejamento, podendo ser menor em *sprints* menores, e sua gestão é realizada pelo *Scrum Master*.

As *daily scrum* (reuniões diárias) são realizadas em, no máximo, 15 minutos sempre no mesmo horário e local pelo time de desenvolvimento para sincronizar as atividades realizadas no dia anterior, determinar um plano de ação até a próxima reunião diária e compartilhar obstáculos, inspecionando dessa forma o progresso em relação à meta da *sprint* e definindo possíveis adaptações. Imediatamente após essa reunião, o time de desenvolvimento ou alguns dos membros podem se reunir para discussões mais detalhadas, o que não está especificamente prevista no Scrum, mas cabe nos valores e pilares da abordagem.

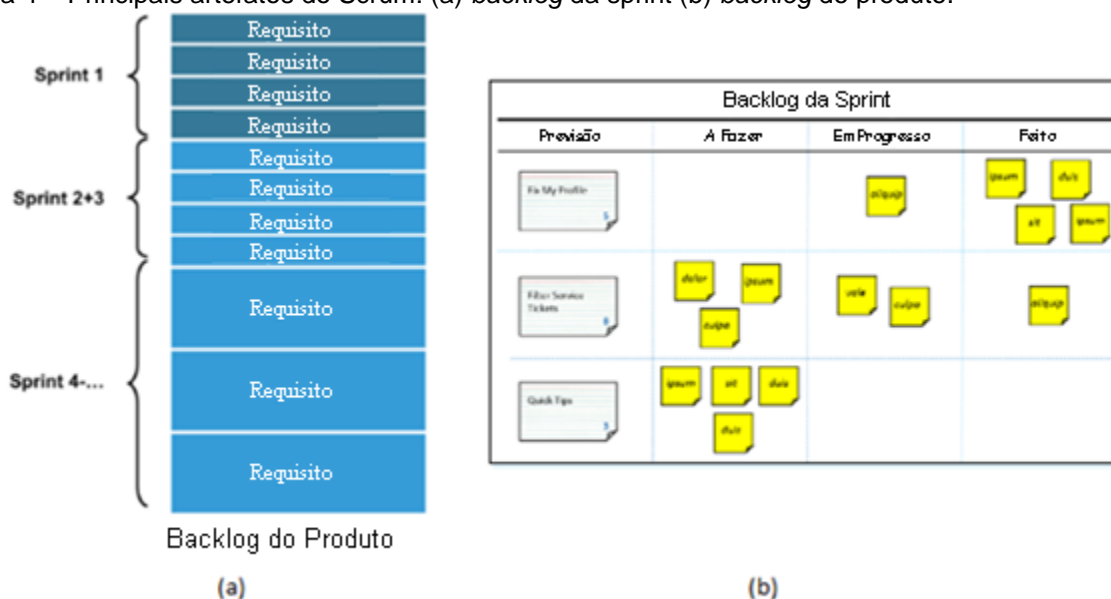
A revisão da *sprint* é uma reunião de até 4 horas que acontece com o time Scrum e os *stakeholders* ao final da *sprint* para discutirem de forma colaborativa o incremento criado e sua relação com o *backlog* do produto, motivando feedbacks para possíveis

adequações, e alinhamento do progresso, orçamento e recursos. O resultado dessa reunião é o *backlog* do produto revisado para a próxima *sprint*.

A retrospectiva da *sprint* é uma reunião de até 3 horas com o time Scrum que tem por objetivo a auto inspeção e a definição de um plano de melhorias sobre o modo que a equipe faz seu trabalho a ser implementado na próxima *sprint*.

Os artefatos do Scrum estão associados à transparência, de forma que apresentam livre oportunidade de inspeção e adaptação. A Figura 4 representa os dois principais artefatos Scrum utilizados pelas equipes como a base para a criação do último artefato, o Incremento.

Figura 4 – Principais artefatos do Scrum: (a) *backlog* da sprint (b) *backlog* do produto.



Fonte: adaptado de Schwaber e Sutherland, 2017.

O *backlog* do produto é uma lista ordenada com todas as características, funções, requisitos, melhorias e correções necessários ao produto, contendo atributos de descrição, ordem, estimativa e valor, e cujos conteúdo, disponibilidade e ordenação cabem ao *Product Owner*. O *backlog* da *sprint* é um conjunto de itens do *backlog* do produto escolhidos para serem entregues no próximo Incremento “Pronto” e o planejamento para atingí-lo, dando visibilidade ao trabalho do time de desenvolvimento. O incremento é um produto utilizável que contempla todos os itens do *backlog* de produto desde o início do projeto até o fim da *sprint* atual.

Scrum-of-scrums é uma técnica usada para escalar scrum (grandes grupos) em que diversos grupos ágeis, ao fim da *daily scrum*, escolhem um *embaixador* para participar de uma reunião diária com outros *embaixadores* para reportarem conclusões, próximos passos e impedimentos. Tal reunião diária recebe o nome de *Scrum-of-scrums* e todos os itens nela tratados serão rastreados através de um *backlog* próprio, de forma a melhorar a coordenação entre times (SUTHERLAND, 2001).

2.1.3 XP – Extreme Programming

O XP foi implementado pela primeira vez em março de 1996 no projeto denominado C3, abreviação para “*Chrysler Comprehensive Compensation*”, cujo objetivo era a unificação de diversos sistemas de folha de pagamento da Chrysler Corporation. Os resultados promoveram experiências tão enriquecedoras que um dos especialistas contratado para integrar a equipe do C3 publicou a primeira versão das práticas que foram usadas no projeto (WELLS, 1999) enquanto o líder do mesmo escreveu um livro explicando a metodologia usada (BECK, 1999) e que se tornou a principal bibliografia sobre XP.

De princípio, Beck (1999) publicou um artigo meramente formal apresentando uma versão inicial do que seriam os valores do XP e algumas formas de começar a incorporar tais ideias no trabalho. Os quatro valores a seguir foram propostos com o reconhecimento de que desenvolvimento de software é um processo humano: comunicação, simplicidade, testes (refere-se ao emprego dos testes unitários e funcionais baseados na perspectiva do cliente) e agressividade (o termo foi substituído posteriormente pelo mais formal “coragem” e refere-se ao incentivo às experiências de novos aprendizados ou desafios).

O autor também sugere, neste primeiro trabalho, cinco práticas que podem facilitar a adoção do XP no serviço. As três primeiras são práticas a serem seguidas pelos desenvolvedores e as duas últimas pela gestão ou liderança:

- Programação em pares: desenvolver o código com outro analista observando a ação e que isso seja feito reciprocamente.
- Testes unitários: escrever um teste antes de iniciar a programação para que o mesmo tenha sucesso somente quando a tarefa estiver completa; deve ser executado a cada compilação.
- Programação letrada (ou literária): escrever um pedaço complexo de código como se fosse uma história, podendo conter textos, figuras e código.
- Histórias: escrever (ou pedir pra um usuário) as próximas 20-50 coisas que seu sistema deve ser capaz de fazer, categorizando-as como funcionalidades que o sistema “deve ter”, “poderia ter” e as que o usuário “gostaria de ter”.
- Iterações: o autor sugere que, no início da semana, cada analista elenque de 2 a 5 atividades que farão e suas respectivas durações estimadas. Ao final da semana, discute-se o progresso.

Nota-se que, apesar do caráter primitivo de suas recomendações, Beck (1999) deflagrou as diretrizes iniciais do XP. No ano seguinte, publicou a primeira edição do livro que propõe de modo mais formal o XP baseando-se principalmente em quatro valores (adaptados do artigo anterior) e em doze práticas que acabaram mostrando-se de difícil adoção, pois algumas delas não eram necessariamente adequadas a todos os ambientes aplicados. Cinco anos depois, lançou a segunda edição do livro, incluindo no XP o valor “respeito” aos valores que haviam sido definidos previamente (comunicação, simplicidade, coragem e feedback), as doze práticas originais foram quebradas em treze com onze corolários.

Para fins de simplificar a conceituação, e visando maior congruência com o trabalho base de Hummel, Rosenkranz e Holten(2013), são explicados apenas os valores e práticas advindos da primeira edição. A Tabela 1 apresenta os quatro valores XP e suas descrições, e a Tabela 2 apresenta as 12 práticas XP bem como suas finalidades.

O XP é uma das abordagens do desenvolvimento ágil que coloca o ideal em prática através da consideração da estrutura dos valores XP, aproximando as pessoas envolvidas com o processo de desenvolvimento.

Tabela 1 – Valores XP e suas descrições

| Valor XP | Descrição |
|---------------------|--|
| Comunicação | Uma comunicação precisa, objetiva e oportuna ajuda a todos os envolvidos compreenderem o que o projeto é, quais são seus resultados esperados e como proceder para alcançar tais objetivos. |
| Simplicidade | Com o intuito de minimizar o impacto em mudanças e o esforço gasto com elas, a solução mais simples é sempre a almejada. |
| Respeito | Necessário para ouvir e compreender melhor, ou seja, para melhorar a comunicação entre todos os envolvidos, acatando ideias novas que podem ser mais simples e promover resultados mais eficazes ou discuti-las, caso contrário. |
| Coragem | O princípio da coragem no XP está associado à capacidade de aceitar mudanças, acomoda-las e obter bons resultados apesar delas. |

Fonte: adaptado de Beck (1999).

Tabela 2 – Práticas XP e suas motivações

| Prática XP | Motivação |
|--|---|
| Propriedade Coletiva de Código | Cada um dos membros da equipe deve enxergar o código produzido como se fosse próprio. |
| Padrões de codificação | A aderência de padrões coletivos de codificação suporta a propriedade coletiva de código e promove compreensibilidade e manutenção. |
| Integração Contínua | A integração de entregas funcionais em produção assim que concluídos ao invés de após um período, evita conflitos. Além disso, a integração realizada de forma contínua ajuda a saber o status real do projeto. |
| Cliente sempre presente | O envolvimento das áreas de negócio ajuda a equipe a responder questões, resolver discordâncias e determinar prioridades. |
| Programação em pares | O desenvolvimento do código deve ser realizado em duplas de programadores: enquanto um digita o código, outro promove feedbacks. |
| Jogo de Planejamento | O XP possui dois níveis de planejamento: o de release e o da iteração. Neles, a área de negócio aproxima-se da área técnica para decidirem sobre escopo, prioridades, composição das releases e prazos, com a intenção de entregar valor ao cliente mantendo o projeto gerenciável. |
| Refatoração | Constantemente o código produtivo é reestruturado sem alteração das funcionalidades observadas externamente para facilitar o entendimento e a manutenção num período futuro do ciclo de vida do produto. |
| Projeto simples | Deve-se facilitar o processo geral do desenvolvimento evitando grandes desenhos logo de início, de maneira que o time de desenvolvimento possa entregar funcionalidades com maior antecedência. |
| Pequenas releases | Releases pequenas e constantes tornam mais fácil lidar com as diversas mudanças de requisitos e de prioridades enquanto é entregue software de valor incremental ao cliente. |
| Ritmo sustentável (semana de 40 horas) | É necessário um ritmo sustentável, nem demasiado corrido nem latente demais, para que os membros individuais da equipe possam desempenhar uma melhor performance a longo prazo. |

| | |
|---------------------------------------|---|
| Metáfora | A metáfora refere-se à uma visão comum desenvolvida e evoluída por toda a equipe em relação à aplicação e à forma como ela opera. |
| Programação orientada a testes | O desenvolvimento orientado por teste exige que os testes sejam escritos antes da escrita do código que pretende-se testar. |

Fonte: adaptado de Beck (1999).

A intenção do *framework* XP, segundo Newkirk (2002), era endereçar as necessidades de pequenas equipes confrontadas com requisitos vagos ou de alta fluidez (alterados constantemente). Pela adoção das práticas e integração dos valores mencionados acima, o XP busca aumentar a eficiência dos projetos de desenvolvimento de software, mitigando o risco inerente às diversas e constantes alterações no mesmo.

2.2 COMUNICAÇÃO

A fundamentação epistemológica da comunicação é consolidada pelas diversas publicações acadêmicas, que explicitam os principais conflitos e problemáticas conceituais concernentes ao tema, em suas diversas abordagens. O objetivo deste trabalho demanda a compreensão mínima do conceito genérico da comunicação.

A palavra é originária do latim *communicare*, literalmente “tornar comum” (HARPER, 2017), que é, por sua vez, derivada de *communis*, traduzido como comum ou compartilhado. A compreensão etimológica apresenta a principal função do termo, que se refere à expansão dos limites de conhecimento referente a determinado objeto-alvo.

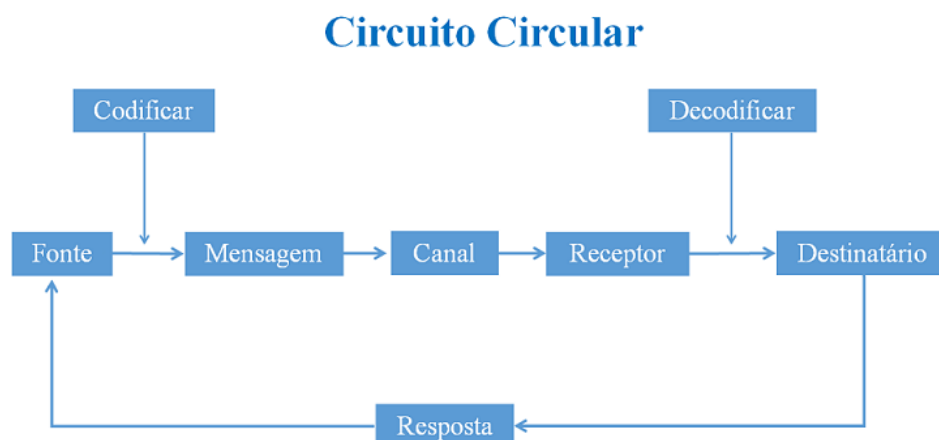
Sabe-se que o objeto da comunicação, ou seja, aquilo que deseja-se tornar comum, é sempre alguma mensagem ou informação. Posto isso, é possível estabelecer que a comunicação empreende, ao menos, duas partes (neste caso, duas pessoas): uma emissora da informação e outra receptora dela. Em sentido amplo, a definição para o termo se põe como o compartilhamento de informações entre transmissor e seu(s) respectivo(s) receptor(es) através de um meio que viabilize o processo e provoque significados comuns entre ambos através de ferramentas simbólicas – em semiótica

denominados signos – sendo as posições de emissor e receptor intercambiáveis (HOHLFELDT, 2010).

Há diversos atributos que complementam a definição acima mas destacam-se, pela contundência em relação ao tema, seu caráter imanente à possibilidade da vida em sociedade (devido à busca humana por consensos), sua irreversibilidade e seu caráter não repetível. Isso denota a sensibilidade conferida ao tratamento da comunicação humana, pois toda a vida em comunidades exige a existência desse processo e, uma vez que ocorre, não pode ser nem refeito, nem desfeito.

Bessa (2006) propôs um viés relacional entre emissor, receptor e mensagem transmitida pelo qual se considera que a efetivação da comunicação se dá na observância de seus aspectos cognitivos. No primeiro caso, denominado linear, tem-se que apenas a mensagem recebida livre de ruídos ou interrupções garante a efetividade da comunicação, sendo que o mesmo só é obtido quando o receptor compreende a informação e se relaciona com ela, no caso da chamada comunicação linear, ilustrada pela Figura 6. Ainda um terceiro caso define a interação aberta considerando o contexto da troca de informações e seus relacionamentos entre si como determinantes da efetividade do processo de comunicação.

Figura 6 – processo circular da interação comunicativa.



Fonte: adaptado de Bessa (2006).

Isso quer dizer que os emissores envolvidos devem analisar qual processo comunicativo almejam e o contexto em que ocorre para mitigarem o risco de falhas de comunicação inerentes a cada um. De acordo com Guarnieri (2010), uma das

interpretações da literatura a respeito do assunto critica o caráter puramente formal de linguagem e comunicação nos processos descritos acima, argumentando que se atêm mais às definições nominalistas que ao processo humano de comunicar. Com isso acabam colocando humanos nas posições de meros sistemas técnicos, ao invés de levar em consideração a complexidade existente no processo social de colocar em contato dois (ou mais) mundos internos, próprios e distintos a cada ser humano.

Não cabe aqui demonstrar amplamente os aspectos em discussão acadêmica a respeito da comunicação, mas vale ressaltar que o tempo e o espaço não mais limitam o processo comunicativo e, graças à internet, diversas abordagens em relação à visão de comunicação foram desenvolvidas, no sentido de caracterizá-la como um processo contínuo de duas vias que é interativo e participativo em todos os níveis de forma que os significados são criados e compartilhados entre os participantes mutuamente – dicotomizando a comunicação em discussão e diálogo (VAN RULER, 2015). Tal característica é relevante ao presente trabalho pois é a que justamente ocorrerá em um projeto de desenvolvimento de software ágil: um processo contínuo de construção de significado compartilhado entre diversas áreas de estruturas organizacionais.

2.3 REVISÃO SISTEMÁTICA DE LITERATURA

Devido ao amplo crescimento da área e os aumentos nos estudos, uma quantidade considerável de publicações é produzida anualmente, geralmente com descobertas conflituosas e discussões controversas. Esse número elevado de resultados dificulta a percepção da visão global de um tema específico ou ainda, de quais resultados são os mais confiáveis para respaldar decisões práticas. Uma revisão sistemática de literatura – SLR, *Systematic Literature Review* – é um trabalho que visa o apontamento desses conflitos, bem como sua sintetização e possíveis comentários analíticos, providenciando implicações para práticas e políticas ou, ainda, a descrição de uma direção para futuras pesquisas.

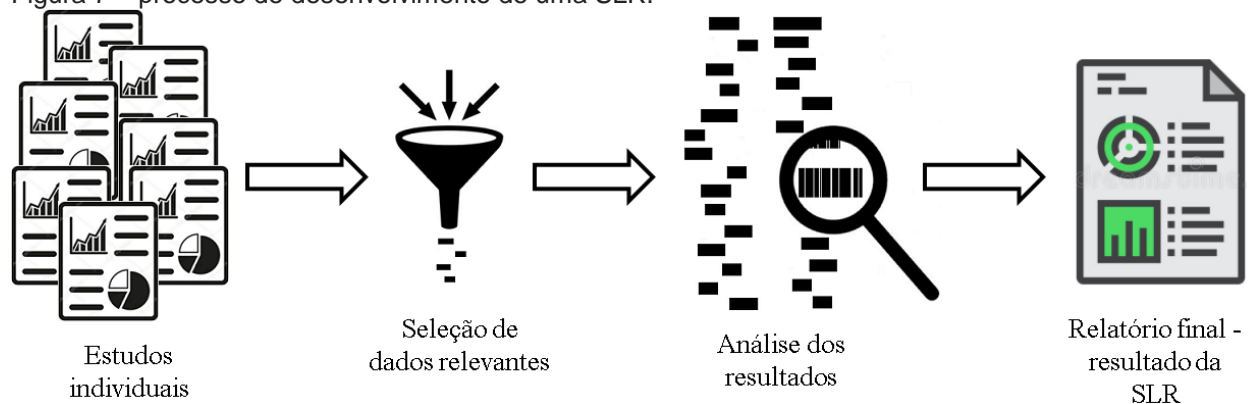
Além de examinar até que ponto uma evidência empírica suporta ou contradiz hipóteses teóricas, outras razões comuns para justificar a necessidade da realização de SLRs na área de tecnologia da informação são: sumarização das evidências ou estudos concernentes a uma tecnologia, identificação de lacunas nas pesquisas atuais para sugestão de novas investigações, providência de uma estrutura ou um *background* para posicionar novas atividades de pesquisa.

Como o próprio nome sugere, é um trabalho que consiste em revisar, ou analisar, as publicações referentes a um tema utilizando um sistema próprio e bem definido para o estudo, ou seja, uma metodologia pré-definida. Trata-se de um estudo realizado para identificar, avaliar criticamente e integrar as descobertas e/ou discussões de todas as publicações relevantes que abordam uma ou mais questões de pesquisa definidas no protocolo da SLR, de forma imparcial e, até certo ponto, repetível (KITCHENHAM; CHARTERS, 2007).

Há diversas maneiras de estruturar uma SLR, sobretudo as utilizadas no meio acadêmico da medicina, mas costuma-se haver concordância sobre as maiores etapas do processo: Planejamento, Condução e Relatório da revisão.

O Planejamento geralmente inclui a identificação da necessidade da revisão, levantamento de uma ou mais questões de pesquisa, o desenvolvimento de um protocolo de revisão e a avaliação de cada item precedente. A etapa de Condução da revisão inclui a efetiva identificação das pesquisas relevantes ao tema ou às questões de pesquisa, a seleção através dos critérios de inclusão ou exclusão, a avaliação da qualidade, o monitoramento da extração dos dados e sua síntese analítica. A última etapa é a escrita do Relatório informativo da revisão, contendo também a análise dos resultados obtidos. A Figura 7 representa a etapa de Condução da pesquisa e subsequente análise de uma SLR descrita neste capítulo.

Figura 7 – processo de desenvolvimento de uma SLR.



Fonte própria.

Apesar das três etapas do processo descrito acima parecerem sequenciais, muitas das atividades envolvem iteração, em particular algumas que são iniciadas durante o desenvolvimento do protocolo e refinadas posteriormente na revisão (KITCHENHAM; CHARTERS, 2007).

3. A PESQUISA DE HUMMEL, ROSENKRANZ E HOLTEN

Este capítulo tem o objetivo de apresentar quais foram os principais itens e discussões referentes à pesquisa para a qual se deseja atualizar neste presente trabalho: seus objetivos, metodologias, resultados e conclusões.

3.1 INTRODUÇÃO

O trabalho de Hummel, Rosenkranz e Holten (2013) mostra uma visão integrada do papel da comunicação em projetos de desenvolvimento de software ágeis através da condução de uma SLR publicada sobre o tema. Argumentando elevada fragmentação do conhecimento sobre a implementação da mudança de paradigma comunicativo promovida pelas abordagens ágeis, bem como sua influência no desenvolvimento de software, propõem a seguinte questão de pesquisa: *qual o papel da comunicação em projetos de desenvolvimento de software que empregam práticas ágeis?*

Os autores definem como objetivos principais do trabalho reportar os resultados passados, identificar e estruturar as lacunas no conhecimento atual sobre o papel da comunicação em desenvolvimento de software ágil e providenciar orientação para pesquisas futuras. O resultado final foi a seleção de um total de 333 artigos relevantes publicados entre 2001 e 2013, e a classificação analítica de cada um conforme o impacto no sucesso do desenvolvimento de software de seu tema apresentado.

3.2 METODOLOGIA DE PESQUISA

O trabalho dos autores não é muito claro a respeito do protocolo de revisão utilizado, mas é possível compreender razoavelmente a estrutura metodológica da

pesquisa através da leitura de seu segundo Capítulo, item primeiro, que, complementado pelo Apêndice A, descreve dois estágios no processo de pesquisa.

O primeiro estágio focou nos jornais e conferências mais populares sobre desenvolvimento de software ágil tanto no domínio da engenharia de software como no dos sistemas de informação. Foi utilizada uma “*string*” de pesquisa genérica e cada resultado foi examinado para encontrar assuntos relacionados à comunicação, definindo assim dois critérios de inclusão. Todas as publicações cujo foco não era desenvolvimento de software ágil foram excluídas dos resultados, além daquelas não baseadas em pesquisa científica rigorosa (o que levou à exclusão de relatórios, editoriais e *short papers*). Foram excluídos também as publicações dos canais relacionados à educação e à prática, e revistas não revisadas.

Na segunda etapa foi realizada uma busca em oito bases de pesquisa (Springer Link, IEEEExplore, ScienceDirect, Wiley Online Library, Taylor and Francis Online, Informa, ACM Digital Library, e uma não encontrada) utilizando a mesma “*string* de pesquisa” mas com o operador AND agregando o termo “*communication*” para a pesquisa do título, resumo e palavras-chave. Nesta etapa foram incluídos os resultados de todos os canais omitidos previamente e também os não baseados em pesquisa científica rigorosa.

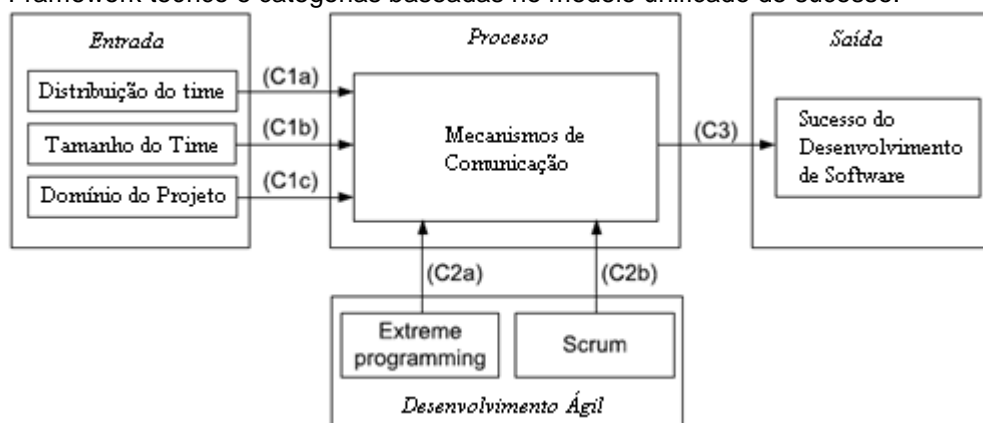
De acordo com o Anexo A do trabalho de Hummel, Rosenkranz e Holten(2013), a *string* de pesquisa teve de ser adaptada de acordo com as respectivas bases de pesquisa, já que nem todas oferecem as mesmas possibilidades de utilização. Conforme a menção dos autores no trabalho, a *string* genérica considerada foi a seguinte:

PublicationTitle: “Name” AND ((Title: “Agile” NOT Title: “Manufacturing”) OR (Title: “Agility” NOT Title: “Manufacturing”) OR Title: “Extreme Programming” OR Title: “Scrum” OR Title: “Pair Programming” OR Abstract: “Extreme Programming” OR Abstract: “Scrum” OR Abstract: “Pair Programming” OR (Abstract: “Agility” AND Abstract: “Development”) OR (Abstract: “Agile” AND Abstract: “Development”) OR Keywords: “Flexible Information Systems Development” OR Keywords: “Agile Software Development” OR Keywords: “Agile Information Systems Development”).

O processo de pesquisa e coleta de dados seguiu as orientações dos autores Webster e Watson (2002), especialmente as que dizem respeito ao caráter focado em

conceito da estrutura da revisão. Por isso, usaram uma matriz conceitual baseada em três categorias que foram concebidas de acordo com a proposta dos autores Siau, Long e Ling (2010) para um modelo unificado para determinação dos fatores de sucesso em desenvolvimento de software. A matriz foi elaborada buscando considerar uma representação sumária do processo de desenvolvimento de software, inclusive os de entrada e saída, conforme apresentado pela Figura 8.

Figura 8 – Framework teórico e categorias baseadas no modelo unificado de sucesso.



Fonte: Hummel, Rosenkranz e Holten (2013).

A primeira categoria trata as variáveis de entrada que presumivelmente influenciam os mecanismos de comunicação em equipes de desenvolvimento de software – a distribuição física dos membros da equipe (em termos de distância), número de membros de cada time no projeto, e o domínio do projeto. A segunda diz respeito à influência das técnicas e práticas utilizadas pela abordagem ágil que é adotada – Scrum ou XP (*Extreme Programming*) – sobre os mecanismos de comunicação das equipes. A última aborda a maneira como os mecanismos de comunicação das equipes influenciam o sucesso do desenvolvimento do software (*Software Development* – SD). A Tabela 3 representa a matriz conceitual com as três categorias utilizadas para a classificação dos trabalhos.

Tabela 3 – Matriz conceitual

| Categoria | Subcategorias | Descrição |
|---|----------------------|---|
| Categoria 1 - Influência dos fatores de entrada | C1a | Distribuição física dos membros da equipe |
| | C1b | Número de membros de cada time no projeto |
| | C1c | Domínio/Características do projeto |
| Categoria 2 - Influência da abordagem ágil adotada | C2a | Extreme Programming (XP) |
| | C2b | Scrum |
| Categoria 3 - Influência dos fatores de saída | C3 | Impacto dos mecanismos de comunicação sobre o sucesso do desenvolvimento de software ágil |

Fonte: adaptado de Hummel, Rosenkranz e Holten (2013).

3.3 RESULTADOS DA PESQUISA DE HUMMEL, ROSENKRANZ E HOLTEN

Todos os 333 trabalhos relevantes encontrados foram analisados e classificados de acordo com as categorias apresentadas anteriormente, sendo realizado um levantamento de quantos papéis abordavam cada uma delas, e se havia quaisquer subcategorias dentre elas.

Para a primeira categoria, no subitem que aborda a distribuição da equipe, foram encontrados 155 estudos e entre as principais descobertas provenientes deles destacam-se: a maior facilidade de comunicação nas equipes localizadas mais próximas, ao passo que times distantes enfrentam inúmeros desafios, por exemplo fusos horários, idiomas ou culturas diferentes; o *layout* se mostra um fator importante para melhorar a frequência e eficiência da comunicação; a frequência da comunicação com equipes distribuídas é menor em comparação em equipes próximas, o que pode resultar em falhas de comunicação, mal entendidos e confusões entre membros da equipe; o framework ágil ajuda a superar desafios de equipes distribuídas ao forçar comunicação através de ferramentas que possibilitam tanto a comunicação síncrona (face-a-face) quanto assíncrona (e-mails, wikis, etc.).

No subitem referente à análise das implicações do tamanho da equipe nos mecanismos de comunicação foram encontrados 33 trabalhos, apontando principalmente o fato de que empresas maiores tendem a ter equipes mais distribuídas, dificultando assim sua comunicação, conforme mencionado no parágrafo anterior. Além

disso observou-se que equipes menores tendem a adotar uma comunicação mais informal e, portanto facilitada, ao passo que tal comunicação pode ser mais problemática quanto maior o número de pessoas envolvidas, devendo ser adotada a comunicação formal, nesse caso.

O subitem que diz respeito à influência do domínio do projeto para a comunicação de equipes ágeis, destacam-se dois pontos principais: para aplicações cujas finalidades são críticas, a comunicação face-a-face com o cliente é essencial devido a mudanças nos requisitos; em termos de projetos de manutenção, faz-se necessária a documentação formal para retenção de conhecimento.

Para a segunda categoria, as descobertas sobre o impacto na comunicação dos times pela escolha de *Extreme Programming* envolveram 95 trabalhos apontando, em geral, que o maior benefício do XP é a maior frequência de comunicação entre as equipes em qualquer nível/fase do projeto. O maior foco de investigação relacionada à comunicação é a programação em pares, sobre a qual foram encontrados 47 trabalhos constatando que tal prática promove habilidades de comunicação e melhora as interações entre os programadores através de compartilhamento de experiências anteriores, revisões contínuas do objetivo esperado, explicações, discussões, e assuntos não relacionados.

As descobertas sobre o impacto do Scrum na comunicação entre equipes apresentaram menos resultados em relação ao XP, num total de 64 trabalhos que destacam o fato de que o Scrum incentiva suas equipes a conversarem a respeito do projeto e incentiva-os a trabalharem fisicamente aproximados, o que implica maiores frequência e qualidade de comunicação. O maior foco de investigação está nas práticas de reunião diária, com 42 artigos relacionados a este tópico e apontando-o como a prática mais importante para desenvolver a comunicação entre equipes Scrum.

Os autores encontraram 100 trabalhos que discutem o relacionamento entre a comunicação e o sucesso dos desenvolvimentos de software utilizando metodologias ágeis, sendo aquela repetidamente referenciada como um dos principais fatores deste, especialmente em ambientes distribuídos, trazendo benefícios como alta produtividade, maior qualidade de software, menor necessidade de tempo e menor custo. Os efeitos da

comunicação no desempenho dos times são mediados pela confiança, compreensão mútua e identificação da equipe. O principal impacto negativo encontrado pelos autores é que a comunicação informal também é relatada como desafiadora e opressiva em caso de algumas partes interessadas (no caso em que os *stakeholders* esperam documentações mais formalizadas, ou quando membros da equipe possuem fracas habilidades comunicativas, por exemplo) ou na retenção do conhecimento.

O resumo dos resultados encontrados pelos autores Hummel, Rosenkranz e Holten por categorias pode ser observado na Tabela 4.

Tabela 4 – Resultados classificados por conceitos

| Categoria | Número de Trabalhos | Principais descobertas através da SLR |
|--|----------------------------|--|
| C1a (Distribuição das equipes) | 155 | <ul style="list-style-type: none"> • Equipes fisicamente mais próximas: comunicação é mais fácil • Equipes distribuídas: muitos desafios associados à comunicação (ex: fusos horários diferentes) • O ágil ajuda a superar esses desafios forçando equipes a se comunicarem |
| C1b (Tamanho das equipes) | 33 | <ul style="list-style-type: none"> • Comunicação informal é ideal para pequenas equipes • Equipes maiores têm que usar comunicação mais formal |
| C1c (Domínio do projeto) | 4 | <ul style="list-style-type: none"> • Projetos de sistemas críticos: requisitos comunicados presencialmente evitam falha de comunicação. • Projetos de manutenção: comunicação formal ajuda a manter o conhecimento. |
| C2a (XP - Extreme Programming) | 95 | <ul style="list-style-type: none"> • Práticas facilitam comunicação informal (ex: programação em pares) |
| C2b (Scrum) | 64 | <ul style="list-style-type: none"> • Práticas facilitam comunicação informal (ex: daily scrum) |
| C3 (Sucesso do desenvolvimento de software ágil) | 100 | <ul style="list-style-type: none"> • Impacto positivos: a comunicação informal é reportada como um dos principais fatores de sucesso especialmente em um ambiente distribuído • Impacto negativo: a comunicação informal também é relatada como desafiadora e opressiva em caso de algumas partes interessadas, ou para manter o conhecimento • Mediadores entre a comunicação e o DS ágil identificados: compreensão mútua, confiança, identificação do time. • Principais benefícios observados: maiores produtividade e qualidade de software; menores orçamento e necessidade de tempo |

Fonte: adaptado de Hummel, Rosenkranz e Holten (2013).

Os resultados também foram classificados de acordo com uma estrutura baseada no trabalho de Urbach, Smolnik e Riempp (2009), conforme a 1.) abordagem da pesquisa, 2.) metodologias usadas e 3.) técnica para coleta dos dados que foram empregadas nos trabalhos encontrados.

A abordagem da pesquisa classifica o trabalho revisado em empírico ou não empírico, de forma que os empíricos são baseados em observações e aplicam algum tipo de método empírico: pesquisas quantitativas (baseadas em questionários ou experimento de laboratório) ou qualitativas (estudos de casos ou entrevistas), já os não empíricos baseiam-se primordialmente em conceitualizações, ideias, *frameworks* e especulações ao invés de observação sistemática.

As metodologias de pesquisa distinguidas por Urbach, Smolnik e Riempp (2009) são: descrição de estrutura ou de modelo conceitual, especulação ou comentários realizados sobre experiência ou teoria de outros autores e a revisão de literatura.

A técnica de coleta de dados se refere à geração de evidências primárias, ou seja, o processo genérico que guia o projeto de pesquisa no que diz respeito às coletas de dados empíricos (pesquisa, entrevista, estudo de caso ou experiência em laboratório).

Tabela 5 – Resultados classificados por características de pesquisa

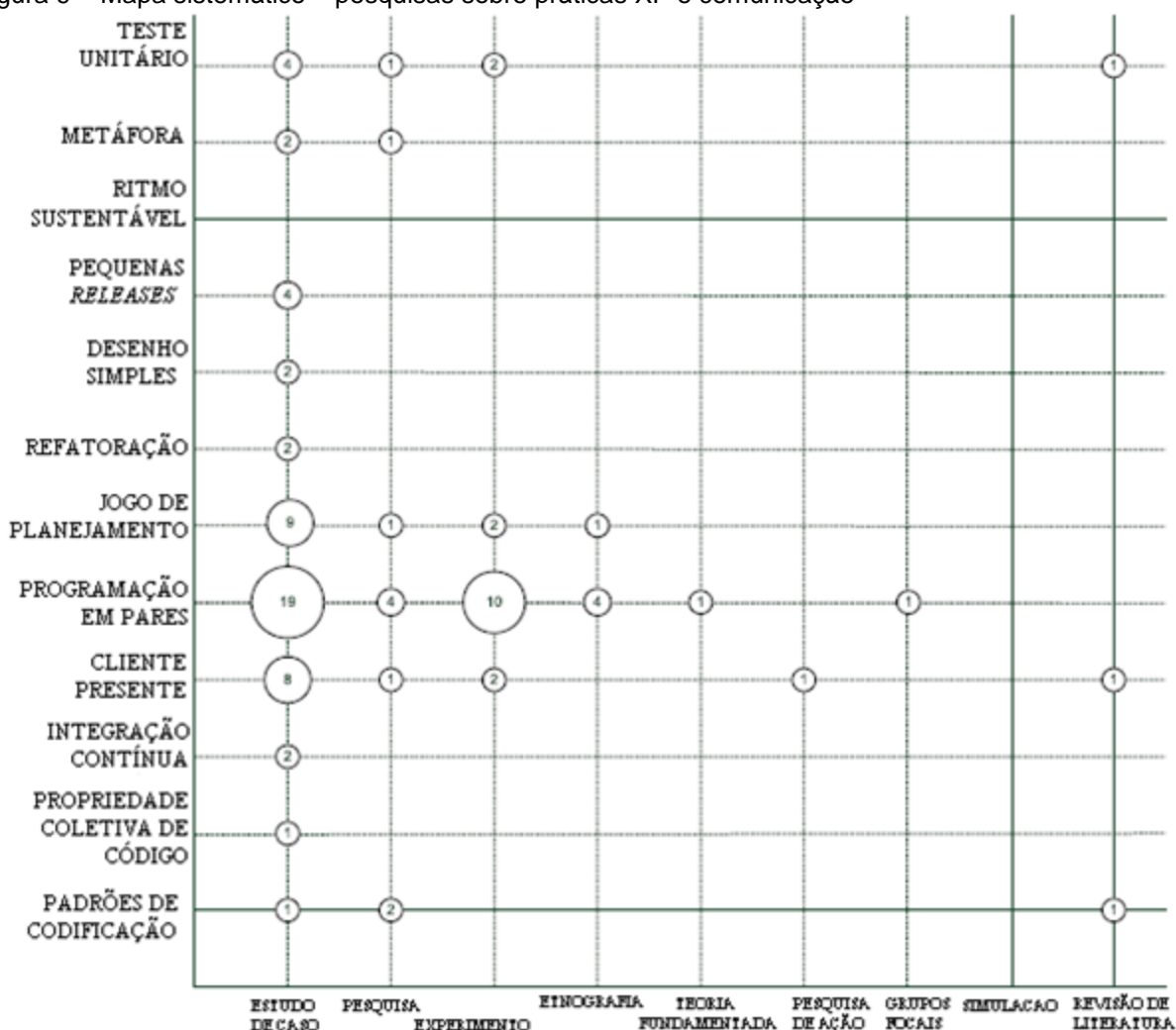
| Categoria | Classificação | Número de Trabalhos | Porcentagem |
|----------------------------|----------------------------|----------------------------|--------------------|
| Abordagem de Pesquisa | Qualitativo | 173 | 51.95% |
| | Quantitativo | 70 | 21.02% |
| | Conceitualização | 63 | 18.92% |
| | Relatórios de experimentos | 39 | 11.71% |
| Método de Pesquisa | Estudo de caso | 128 | 38.44% |
| | Pesquisa | 26 | 7.81% |
| | Experimento | 21 | 6.31% |
| | Revisão de literatura | 20 | 6.01% |
| | Etnografia | 14 | 4.20% |
| | Teoria fundamentada | 12 | 3.60% |
| | Grupo de foco | 5 | 1.50% |
| | Pesquisa de ação | 5 | 1.50% |
| | Simulação | 3 | 0.90% |
| | Nenhuma | 103 | 30.93% |
| Técnica de Coleta de Dados | Entrevistas | 103 | 30.93% |
| | Questionários | 42 | 12.61% |
| | Arquivos de log | 36 | 10.81% |
| | Notas de campo | 33 | 9.91% |
| | Gravação de vídeo | 6 | 1.80% |
| | Gravação de áudio | 3 | 0.90% |
| | Desconhecido | 19 | 5.71% |
| | Nenhuma | 109 | 32.73% |

Fonte: adaptado de Hummel, Rosenkranz e Holten(2013).

O resultado dessa classificação pode ser observado na Tabela 5, que apresenta os dados das características das pesquisas não totalizando 100% pois as mesmas podem contribuir para mais de uma categoria.

A partir dos resultados obtidos pelas duas matrizes estruturadas – a conceitual e a de características – foram criados dois mapas sistemáticos cruzando os resultados de cada uma delas nos casos de abordagem XP e Scrum, e para outras categorias, conforme as Figuras 9, 10 e 11. Dessa forma, puderam detalhar a extensão em que as pesquisas investigaram as categorias propostas e encontrar quais lacunas os estudos não preencheram.

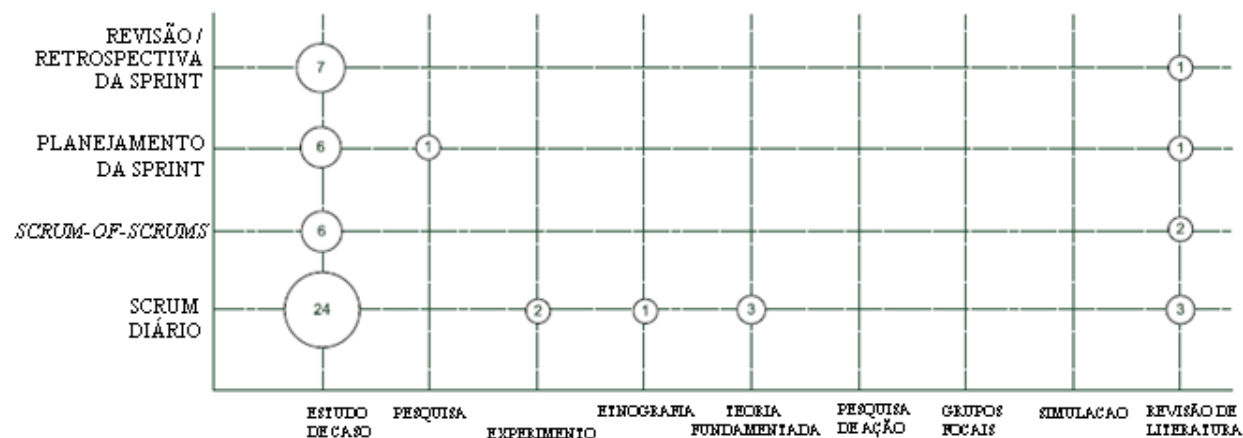
Figura 9 – Mapa sistemático – pesquisas sobre práticas XP e comunicação



Fonte: Traduzido de Hummel, Rosenkranz e Holten (2013).

Observaram que a pesquisa associada à comunicação na abordagem XP está focada na programação em pares enquanto as outras práticas são pouco estudadas.

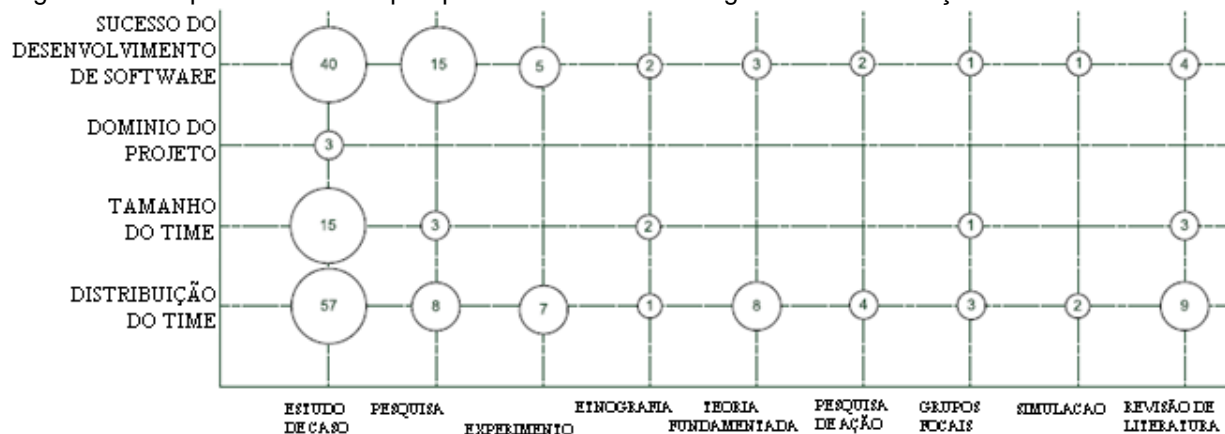
Figura 10 – Mapa sistemático – pesquisas sobre práticas Scrum e comunicação.



Fonte: Traduzido de Hummel, Rosenkranz e Holten (2013).

Além disso, os autores observaram que a pesquisa associada à comunicação na abordagem Scrum está focada no *daily Scrum* (reuniões diárias) enquanto as outras práticas são pouco frequentes, e a maior parte dos estudos encontrados possui abordagem de natureza qualitativa.

Figura 11 – Mapa sistemático – pesquisas sobre outras categorias e comunicação.



Fonte: Traduzido de Hummel, Rosenkranz e Holten (2013).

Referente às outras categorias, encontraram estudos explanatórios e qualitativos sobre o impacto da distribuição da equipe e do tamanho da equipe, bem como sobre o

efeito da comunicação sobre o sucesso do desenvolvimento do software, mas pouco foi encontrado sobre outros fatores de entrada, como o domínio do projeto.

Com os mapas criados, puderam realizar um levantamento de *gaps* (lacunas) de pesquisa para cada categoria da matriz conceitual apresentando oportunidades para trabalhos futuros.

Para a categoria de entrada, encontraram duas questões que consideraram não respondidas pelas publicações do período: quais condições envolvem e quais fatores no nível ambiental, organizacional, de grupo e individual afetam a comunicação bem-sucedida em desenvolvimento de software ágil, e qual o impacto do domínio do projeto e do contexto do desenvolvimento (fatores no nível ambiental, organizacional, de grupo e individual) sobre o uso e o efeito dos mecanismos de comunicação dentro das equipes ágeis?

Para a categoria dos processos, os autores observaram que as questões a seguir não foram respondidas: quais os impactos das práticas do Scrum e/ou do XP na informalidade, frequência, e qualidade da comunicação em comparação com o desenvolvimento tradicional e não ágil? Em outras palavras, o uso das práticas Scrum e/ou XP levam a uma comunicação mais informal e melhor?

Para a categoria de saída, encontraram as seguintes lacunas: quais são as implicações do paradigma de comunicação em mudança do desenvolvimento de software ágil para o resultado do desenvolvimento? As equipes ágeis de desenvolvimento de software são mais bem-sucedidas do que as tradicionais e não ágeis, devido à melhoria da comunicação? Em outras palavras, maior frequência e melhor comunicação leva a melhores resultados?

4. EXTENSÃO DO TRABALHO DE HUMMEL ROSENKRANZ E HOLTEN

Este capítulo apresenta o desenvolvimento do trabalho de atualização da SLR originalmente proposta e realizada por Hummel, Rosenkranz e Holten (2013) para o período entre o fim da pesquisa dos autores, 2012/2013 e final de 2017.

4.1 PROTOCOLO DA SLR

Posto que o trabalho de Hummel, Rosenkranz e Holten não é claro a respeito do protocolo utilizado para revisão, este trabalho baseia-se em um previamente desenvolvido e fornecido pela orientadora (BOMBONATTI; GOULÃO; MOREIRA, 2016), sendo este adaptado considerando-se a metodologia do trabalho original e as condições de pesquisa disponíveis pelo acesso aos acervos de pesquisa fornecido pela USP, em conjunto com a Capes. Este artefato tem como objetivo auxiliar e orientar a pesquisa tanto em termos conceituais – para que não se perca o foco temático – quanto em organizacionais – definindo indiretamente o que deve-se fazer e em que ordem.

Conforme menção adiante, no protocolo constam o planejamento cronológico da SLR, alguns dados básicos (título, período de cobertura, equipe envolvida, objetivos, motivações, necessidade e estudos similares), os principais conceitos envolvidos no processo, as questões de pesquisa (RQ1, RQ2 e RQ3), as fontes de pesquisa (acervos, jornais e conferências), as palavras-chave e *strings* de consulta (bem como os testes realizados para cada *string*), os critérios de exclusão e inclusão, a estratégia de extração e consolidação de dados, a estratégia de revisão e uma planilha contendo as características das seleções dos artigos.

Observou-se que no artigo original de Hummel, Rosenkranz e Holten (2013), alguns resultados não foram encontrados até o ano de publicação da mesma (2013) e por isso a pesquisa deve considerar um período de cobertura desde a última data encontrada pelos autores até as publicações de 2018. A intenção que justifica a atualização é avaliar se a comunicação continua sendo um tema relevante nas pesquisas

mais recentes, além de construir um mapa atualizado das práticas ágeis vigentes que possuem maior foco na melhoria da comunicação.

No protocolo também foram elencados e brevemente descritos os principais conceitos abordados no trabalho, que são os seguintes: comunicação, Ágil, XP, Scrum e SLR, todos já detalhados anteriormente no texto.

As questões de pesquisa (RQ1, RQ2 e RQ3), mencionadas anteriormente, foram descritas no protocolo junto à estrutura PICOC – *Population (Problem), Interest, Comparison, Outcomes e Context* – uma ferramenta de estratégia para definir componentes chave antes do início da pesquisa (UMAN, 2011). A população foi definida como: Project Owner, Scrum Master, time de desenvolvimento e time estendido, clientes/usuários. A intervenção foi definida como: pesquisa focada nos frameworks XP e Scrum. A comparação foi definida como: uma comparação entre os resultados obtidos para Scrum ou XP. Os resultados foram estipulados como: práticas ágeis relacionadas a algum impacto na comunicação e vice-versa. O contexto ficou definido como: Artigos voltados à práticas acadêmicas e industriais.

Foi realizado um estudo para levantar as fontes utilizadas no trabalho original, suas últimas datas de publicação encontradas no trabalho original e disponibilidade de consulta através do acesso para consulta de publicações acadêmicas da biblioteca da USP, conforme a Tabela 6. Foi verificado que apenas três das fontes utilizadas não estavam disponíveis e algumas possuíam resultados apenas até 2017.

Tabela 6 – Fontes de pesquisa utilizadas no trabalho a ser atualizado

| Source Name | Outlet name | Outlet type | Latest included issue | Observation |
|---------------|--|-------------|-----------------------|---------------------------------|
| Springer Link | Empirical Software Engineering | Journal | Apr-13 | Disponibilizado todo o material |
| Springer Link | EJIS - European Journal of Information Systems | Journal | Jan-13 | Disponibilizado todo o material |
| IEEEExplore | IEEE Transactions on Software Engineering | Journal | Vol. 2 2013 | Disponibilizado todo o material |
| IEEEExplore | IET Software | Journal | Dec-12 | Disponibilizado todo o material |
| ScienceDirect | Information and Software Technology | Journal | Mar-13 | Disponibilizado todo o material |
| ScienceDirect | Journal of Systems and Software | Journal | Mar-13 | Disponibilizado todo o material |

| | | | | |
|---------------------------|---|------------|-----------|---|
| ScienceDirect | Journal of Systems Architecture | Journal | Jan-13 | Disponibilizado todo o material |
| IEEEExplore | Agile Conference | Conference | 2012 | Encontrado através de pesquisa refinada em acervo de publicações (material agrupado por conferência anual) |
| IEEEExplore | ESEM - Empirical Software Engineering And Measurement | Conference | 2012 | Encontrado através de pesquisa refinada em acervo de publicações |
| Springer Link | Journal of Information Technology | Journal | Dec-12 | Disponibilizado todo o material |
| Wiley Online Library | Information Systems Journal | Journal | Mar-13 | Disponibilizado todo o material |
| Taylor and Francis Online | Journal of Management Information Systems | Journal | Fall 2012 | Disponibilizado todo o material |
| Informa | Information Systems Research | Journal | Dec-12 | Disponibilizado todo o material |
| ScienceDirect | Journal of Strategic Information Systems | Journal | Dec-12 | Disponibilizado todo o material |
| IEEEExplore | EuroMicro International Conference | Conference | 2012 | Encontrado através de pesquisa refinada em acervo de publicações |
| Springer Link | EuroSPI | Conference | 2012 | Disponíveis apenas as publicações até 2016 |
| ACM Digital Library | ICSE | Conference | 2012 | Disponibilizado todo o material |
| IEEEExplore | ICGSE | Conference | 2012 | Disponibilizado todo o material |
| Springer Link | IFIP Conferences | Conference | 2012 | Disponibiliza busca na série de livros IFIP Advances in Information and Communication Technology |
| Springer Link | PROFES | Conference | 2012 | Disponíveis apenas as publicações até 2016 |
| Springer Link | XP - International Conference on Agile Software Development | Conference | 2012 | Disponibilizado todo o material |
| Springer Link | ICIS | Conference | 2012 | Disponíveis apenas as publicações até 2010 pela consulta no site |
| Springer Link | LESS (Lean Enterprise Software and Systems) | Conference | 2010 | Apenas as conferências de 2010 e 2013 disponíveis |
| Not Found | Journal of the AIS | Journal | Jan-13 | Não foram encontradas bases acadêmicas cujos acessos são garantidos pela SIBInet, apenas o site do editor (PPV - pay per view) e a AIS eLibrary |
| Not Found | MIS Quarterly | Journal | Mar-13 | Não foram encontradas bases acadêmicas cujos acessos são garantidos pela SIBInet, apenas o site do editor (PPV - pay per view) e a AIS eLibrary |

Fonte: própria.

A partir deste levantamento, foi discutida a necessidade dos autores Hummel, Rosenkranz e Holten terem dividido a pesquisa original em duas partes, julgando-se insubstancial e até contraproducente a realização de buscas em espaços amostrais contidos entre si. Ou seja, uma vez que a primeira parte considerava um universo amostral menor (jornais e congressos, além de algumas outras limitações) e inserido no que foi considerado na segunda parte da pesquisa (todo o acervo de dados), não há a necessidade de realizar ambas separadamente, apenas a amostra mais ampla. Outrossim, se houve algum motivo específico para tanto, o mesmo não foi descrito no artigo, originando tal discussão. A lógica foi demonstrada de forma empírica para o aleatoriamente escolhido artigo de Datta (2017), que fora encontrado tanto pela consulta no jornal “*Empirical Software Engineering*” quanto em todo o banco de dados da *SpringerLink*; não foi realizada uma busca mais abrangente pois, uma vez que não há mecanismo de busca específica por jornais, a demonstração se tornaria insustentável. Portanto, este trabalho considera apenas a busca em todo o acervo.

Dessa forma, foram realizadas pesquisas nas sete bases de pesquisa utilizadas no trabalho anterior com variações da *string* de pesquisa genérica usada pelos autores, que teve de ser alterada para conformar-se às especificações de pesquisa (sintaxe) dos portais web que dão acesso às bases sem, no entanto, alterar seu sentido original. Para tanto, foram feitos diversos testes confrontando os resultados com e sem a junção do termo “communication” à *string*, e utilizando ou não os filtros para datas entre o período específico (2012 a 2018), observando que os resultados foram alterados coerentemente pelas modificações na semântica e adquirindo uma base comparativa para a grandeza do número de resultados.

Para análise dos artigos encontrados, foram definidos nove critérios de inclusão ou exclusão, conforme a Tabela 7.

Tabela 7 – Critérios para inclusão ou exclusão

| Id | Tipo | Critério |
|----|----------|---|
| C1 | Exclusão | <i>Resultados anteriores a última data encontrada</i> |
| C2 | Exclusão | <i>Trabalhos duplicados</i> |
| C3 | Exclusão | <i>Resultados que não atendem às questões de pesquisa</i> |
| C4 | Fusão | <i>Mesmos conteúdos em diferentes artigos</i> |
| C5 | Exclusão | <i>Artigos não escrito nos idiomas: Inglês, Espanhol, Português</i> |
| C6 | Inclusão | <i>Artigos que respondem às questões de pesquisa</i> |
| C7 | Exclusão | <i>Artigos que já se encontram no trabalho original</i> |
| C8 | Exclusão | <i>Estudos que não sejam primários</i> |
| C9 | Exclusão | <i>Conteúdo não disponível</i> |

Fonte: própria.

Foi definido previamente que os critérios seguiriam a mesma sequência de aplicação de seus *Ids*, mas observou-se outra ordem de aplicação durante a execução da seleção, conforme segue: C1 (filtro realizado diretamente nos portais de cada base), C2//C4, C3//C5//C8, C7, C9 e C6. Foram definidos os papéis dos pesquisadores durante a seleção (nesse caso apenas dois pois, o trabalho foi realizado com um aluno e uma orientadora, mas poderiam ter sido criados diversas responsabilidades, caso houvessem mais envolvidos) como selecionador e validador, sendo que os artigos que representassem dúvidas de incorporação aos critérios seriam marcados pelo selecionador e avaliados em reunião com o validador. A execução do filtro foi realizada em quatro *rounds*, conforme mostra a Tabela 8.

Tabela 8 – Rounds de seleção de artigos

| Round | Descrição |
|-------|---|
| 1 | Leitura dos títulos e temas dos artigos para realizar seleção preliminar |
| 2 | Leitura do abstract dos artigos para seleção intermediária |
| 3 | Leitura completa dos artigos para seleção final |
| 4 | Seleção definitiva após validação com orientador |

Fonte: própria.

A execução dos filtros definidas pelos critérios através dos rounds resultou em uma seleção de artigos que foram dispostos em uma tabela seguindo um layout conforme apresentado pela Tabela 9. Os critérios para criação deste layout levaram em

consideração as classificações implementadas pelos autores conforme apresentadas anteriormente nas Tabelas 3 e 5.

Tabela 9 – Layout pra auxílio na criação de matrizes de apresentação dos resultados

| | |
|--|--|
| 1. Paper Basic Information: | |
| 1.1 Paper id: | [Definido pela ordem da pesquisa] |
| 1.2 Paper title: | |
| 1.3 Paper conference/journal: | |
| 1.4 Paper year: | |
| 1.5 Author(s): | |
| 1.6 Digital library: | |
| 1.7 DOI | |
| 2. Abstract: | |
| 2.1 Abstract (full): | |
| 3. Research | |
| 3.1 What is the research approach used. | [e.g: Conceptualization, Qualitative, Quantitative, Experience Report] |
| 3.2 What is the research method used. | [e.g: Grounded Theory, Literature Review, Ethnography, Experiment, Action Research, Field Study, Case Study, Simulation, Focus Group (may be a combination)] |
| 3.3 What is the primary data collection methods used. | [e.g: Survey, Interviews, Field Notes, Audio Recordings, Video Recordings, Literature, Log Files, Questionnaire (may be a combination)] |
| 4. Input Dimension | |
| 4.1 Team distribution | [mark an "x" for yes, blank for no] |
| 4.2 Team size | [mark an "x" for yes, blank for no] |
| 4.3 Project location | [mark an "x" for yes, blank for no] |
| 5. Agile | |
| 5.1 Extreme Programming | Coding standards, Collective code ownership, Continuous integration, On-site customer, Pair programming, Planning game, Refactoring, Simple design, Small releases, Sustainable pace, System metaphor, Unit testing (incl. automated acceptance tests), User Stories |
| 5.2 Scrum | Daily Scrum, Sprint Planning, Sprint Review/ Retrospective, Scrum-of-Scrums/LeSS/SAFe, Product/ Sprint Backlog, Product Increment |
| 6. Output dimension | |
| 6.1 SD Success | |
| 7. Another criteria? | |

Fonte: própria.

Finalmente, foram definidos a estratégia para consolidar e apresentar os dados resultantes da pesquisa como similares aos do trabalho anterior – tabulado e sumarizado por narrativa – inclusive usando as mesmas classificações que os autores utilizaram na matriz conceitual, previamente apresentadas na Tabela 3.

4.2 RESULTADOS DA SLR

Os artigos encontrados, bem como sua seleção e demais resultados da SLR podem ser observados pela descrição do protocolo, disponibilizado através do link: https://drive.google.com/open?id=11I_Ga3Z996H70MTzTY-a6Uubyl61AUAj

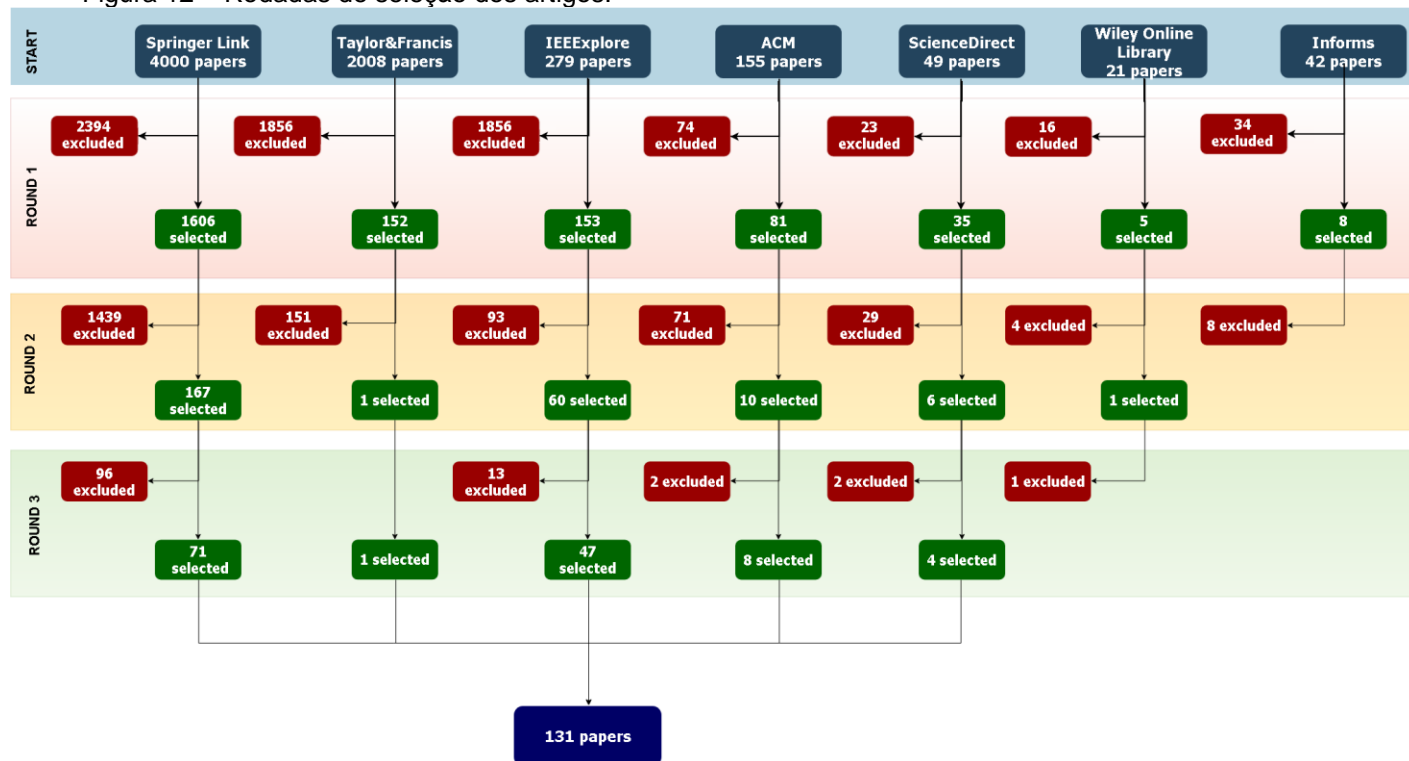
Os rounds de seleção tiveram finalidade de filtrar os artigos gradualmente para que restem apenas os que são efetivamente relevantes ao tema. Ao fim da pesquisa nas sete bases foram obtidos 6563 artigos, dos quais apenas 131 foram selecionados para análise e apresentação.

No primeiro round foram lidos todos os títulos e selecionados preliminarmente 2040 para o próximo round, excluindo artigos com temas muito distantes das questões de pesquisa, o que ressalta as possibilidades de que ou a *string* de busca não foi tão restritiva como se propunha ou que o mecanismo de busca não é muito aderente às restrições da *string* de consulta.

No segundo round foram lidos os abstracts dos 2040 *papers* para realizar uma seleção intermediária, dispensando os *papers* que abordavam temas não relacionados ao desenvolvimento de software ágil (por exemplo agilidade em *supply chain* ou agilidade organizacional) e os resultados que não descrevem estudos primários (capítulos de livros e SLRs, por exemplo), além dos artigos que já se encontravam no trabalho original.

No terceiro round foram lidos os textos dos 245 artigos para selecionar os que respondem às questões de pesquisa e foram escolhidos 131 a partir dessa seleção. A validação com a orientadora foi realizada continuamente através de e-mails e conversas via Skype. A Figura 12 apresenta as seleções dos estudos conforme a evolução das rodadas.

Figura 12 – Rodadas de seleção dos artigos.



Fonte: própria.

Conforme menção prévia, foi realizada a análise partindo-se das mesmas premissas que os autores, ou seja, as mesmas classificações e estruturas conceituais foram utilizadas para a apresentação dos resultados e principais descobertas. A Tabela 10 apresenta os principais resultados do impacto dos fatores de entrada (distribuição e tamanho da equipe e domínio do projeto), de saída (sucesso do projeto) e da abordagem ágil adotada sobre a comunicação (o resultado da soma não é necessariamente o número total de *papers* encontrados porque os mesmos podem ser enquadrados em mais de uma categoria).

Tabela 10 – Síntese dos resultados classificados por conceitos

| Categoria | Número de trabalhos | Principais descobertas através da SLR |
|--|---------------------|--|
| C1a (distribuição das equipes) | 48 | <ul style="list-style-type: none"> Equipes distribuídas geograficamente apresentam maiores obstáculos para comunicarem-se ao passo que as co-localizadas possuem maiores facilidades (não representa novidades) Uso de técnicas ágeis de escalamento podem facilitar a comunicação em equipes distribuídas (Scrum-of-Scrums) Melhorias simples nas ferramentas e infraestrutura de comunicação podem "reduzir as distâncias". |
| C1b (tamanho das equipes) | 23 | <ul style="list-style-type: none"> Grandes equipes podem dificultar o direcionamento da gestão Equipes pequenas em empresas pequenas tendem a adotar comunicação menos formal Diversos <i>frameworks</i> podem ser utilizados para escalar o desenvolvimento ágil em grandes empresas |
| C1c (domínio do projeto) | 19 | <ul style="list-style-type: none"> Aplicações críticas – garantias de segurança e certificações Aplicação em projetos de migração para Cloud: superação de barreiras técnicas e desafios organizacionais Aplicação na indústria de jogos: melhoria de qualidade |
| C2a (XP) | 51 | <ul style="list-style-type: none"> <i>Pair programming</i> promovem comunicação informal enquanto <i>user stories</i> exigem comunicação bem-sucedida <i>Sprint/Product Backlog</i> exigem eficiência na comunicação e <i>daily meetings</i> promovem compartilhamento de conhecimento |
| C2b (Scrum) | 100 | |
| C3 (sucesso e desafios do desenv. ágil) | 32 | <ul style="list-style-type: none"> A estrutura da comunicação organizacional influi na arquitetura final do sistema Práticas ágeis facilitam a comunicação e contribuem para a qualidade final dos projetos A maioria dos impedimentos estão associados a problemas de gerenciamento, coordenação e comunicação |

Fonte: adaptado (atualizado) de Hummel, Rosenkranz e Holten (2013).

Garbajosa, Yagüe e Gonzalez (2014) mencionam que nenhum dos obstáculos para comunicação entre equipes em empresas *multi-site* são realmente novas, mas publicações e revisões recentes ainda apresentam serias preocupações. Além disso, informa que o desenvolvimento de software ágil global (AGSD – *Agile Global Software Development*) tem ganhado relevância e importância no cenário atual e que a comunicação é ainda mais crítica nesses casos. Por isso as equipes distribuídas dependem fortemente das ferramentas para comunicação face a face (*Skype*, *VSee*, por exemplo), da infraestrutura (*Smartboards*, câmeras, telas, repositórios centralizados como *SVN*, *Redmine* ou até *Github*, e internet de banda-larga, por exemplo), e das equipes ágeis usarem técnicas de escalamento ágil como *Scrum-of-Scrums* (VALLON et al., 2013). No caso de equipes co-localizadas, um ambiente informativo encoraja a

comunicação e auxilia na partilha de conhecimento, desempenhando um papel central para equipes ágeis (SCHWARZER et al., 2016).

Com o empoderamento individual permitindo flexibilidade no desenvolvimento ágil de larga escala, o controle gerencial é ligeiramente sacrificado, inviabilizando o direcionamento das intenções estratégicas. Isso pode ser resolvido com a implementação de incentivos voltados ao direcionamento que se deseja para a equipe – recompensas associadas à criação de valor para o time, por exemplo (BLAU et al., 2013).

Apesar de possuírem acesso a menos recursos que grandes companhias, pequenas empresas de software têm menor necessidade de comunicação documentada, aderindo às práticas menos formais de comunicação, facilitando assim a aplicação de abordagens ágeis (BABB; HODA; NØRBJERG, 2014).

Conforme seus objetivos, modelo operacional e necessidades, grandes empresas podem escolher uma série de *frameworks* para escalar o desenvolvimento ágil, dentre eles o Scaled Agile Framework (SAFe), Scrum-of-Scrums (SoS), Large Scale Scrum (LeSS), Disciplined Agile Delivery (DAD) e Spotify Model. Tais frameworks podem ser observados numa matriz comparativa providenciada e mantida por Dolman e Spearman (2017). Contudo, segundo Razzak et al. (2017), pouco se publica a respeito dessas abordagens escaladas para a adoção do ágil em pequenas e médias empresas (PMEs). De acordo com os autores, em organizações menores, as diversas cerimônias e os variados papéis e responsabilidades não são possíveis de completar, de modo que são necessárias avaliações através de métricas ou a realização de auto avaliações para entender quais práticas devem ser adotadas ou adequadas para a melhor implementação em PMEs.

Ao desenvolver softwares críticos (por exemplo para a área da saúde) as definições dos requisitos devem levar em consideração algumas garantias de *compliance* impostas pelo domínio, especialmente as de padrões de segurança e os organismos de certificação. Para mitigar os riscos associados a esse problema, sugere-se a utilização de práticas como: introdução de especialista no projeto, testes extensos (por exemplo TDD), introdução de padrões de segurança, melhorar a garantia de

qualidade de artefatos anteriores ao código (por exemplo *User Stories*), manter altos padrões de código (GÓRSKI; ŁUKASIEWICZ, 2013).

Ainda no que diz respeito ao impacto do domínio do projeto sobre a comunicação, foi encontrado um relatório de pesquisa em progresso referente a desenvolvimento ágil de software médico embutido, que enfatiza a necessidade do design de hardware simultâneo ao processo de desenvolvimento e a dependência da comunicação entre as diversas áreas envolvidas e o sucesso do projeto (DEMISSIE; KEENAN; MCCAFFERY, 2016).

No domínio da indústria de *games*, os principais impactos do uso de abordagens ágeis foram a melhoria de comunicação entre profissionais, a qualidade dos jogos, e a descoberta de funcionalidades divertidas e implementáveis de forma mais rápida.

A aplicação de abordagens ágeis em projetos de migração de aplicações para nuvem pode ajudar a vencer desafios organizacionais e técnicos. Conforme Stavru, Krasteva e Ilieva (2013), a principal sugestão para o uso do ágil dada por especialistas é realizar pequenas releases ou *Sprints* possibilitando o rápido recebimento de feedback e aumentando a capacidade de resposta a mudanças e a problemas.

A Tabela 11 apresenta a quantia de trabalhos que reportam a relação entre práticas ágeis XP ou Scrum e comunicação (o resultado da soma não é necessariamente 100% porque várias práticas podem ser abordadas em cada *paper*). No período da atualização, as principais práticas estudadas do XP foram *pair programming* e *user stories*, enquanto que no Scrum foram o *sprint* ou *product backlog* e as reuniões diárias da *daily Scrum*.

Nota-se que tanto *user stories* no XP quanto os *backlogs* no Scrum são práticas ágeis para levantar e/ou rastrear requisitos e exigem, efetivamente, comunicação bem-sucedida, engajamento e participação ativa entre os envolvidos para a eficiente transferência de conhecimentos e informações. Tais práticas podem ser facilitadas pela escolha dos artefatos de comunicação – que é influenciada pela cultura organizacional, como explica Plachkinova, Peffers e Moody (2015) – ou por ferramentas ou infraestruturas de comunicação, como mencionado previamente.

As práticas de *pair programming* e *daily meetings* influenciam fortemente a linguagem informal e fortalecem o compartilhamento de conhecimento entre membros da equipe (KUUSINEN et al., 2017). De acordo com os autores, colegas de equipe preferem compartilhar conhecimento entre si ao invés de com clientes pois a motivação para fazê-lo entre times é a satisfação enquanto que com clientes é o estreitamento de laços. Foi identificado que a escolha de parceiros na programação em pares pode afetar o resultado no projeto, pois pares com personalidades e temperamentos distintos apresentam melhores performances (SFETSOS et al., 2012).

Tabela 11 – Frameworks abordados nas pesquisas sobre desenvolvimento de software ágil e comunicação

| Framework Ágil | Prática Ágil | # Artigos | Percentual |
|---------------------------------|---|-----------|------------|
| XP (eXtreme Programming) | Coding standards | 5 | 3.82% |
| | Collective code ownership | 7 | 5.34% |
| | Continuous integration | 8 | 6.11% |
| | On-site customer | 4 | 3.05% |
| | Pair programming | 28 | 21.37% |
| | Planning game | 10 | 7.63% |
| | Refactoring | 9 | 6.87% |
| | Simple design | 5 | 3.82% |
| | Small releases | 10 | 7.63% |
| | Sustainable pace | 4 | 3.05% |
| | System metaphor | 6 | 4.58% |
| | Unit testing (+ automated acceptance tests) | 8 | 6.11% |
| | User Stories | 18 | 13.74% |
| Scrum | Daily Scrum | 55 | 41.98% |
| | Sprint Planning | 54 | 41.22% |
| | Sprint Review/ Retrospective | 45 | 34.35% |
| | Scrum-of-Scrums/LeSS/SAFe | 14 | 10.69% |
| | Product/ Sprint Backlog | 73 | 55.73% |
| | Product Increment | 30 | 22.90% |

Fonte: própria.

Conforme mencionado anteriormente, a comunicação tem grande influência no sucesso dos projetos de desenvolvimento, e pode ser facilitada pela adoção de práticas ágeis. Wojciechowski, Wesolowski e Complak (2010) concluem experimentalmente que

a prática XP de cliente *on-site* influencia positivamente a qualidade da comunicação e a velocidade da produção de software. Os autores indicaram a prática como chave para sucesso devido à comunicação face a face, à possibilidade de trocar conhecimento usando recursos simples (notas, desenhos, etc.), além do controle da produção do software pelo cliente poder ser feita instantaneamente.

O impacto da comunicação sobre projetos em geral para desenvolvimento de software global também foi identificado diversas vezes como essencial. Yagüe et al. (2016), por exemplo, afirma que a comunicação é crítica devido ao desenvolvimento distribuído e as reuniões frequentes previstas pelo ágil e realçadas pelo contato face a face.

O impacto da estrutura organizacional de comunicação sobre a estrutura do *design* do sistema foi verificado por Santana et al. (2013), confirmando a hipótese de que a arquitetura final do sistema seria mais simples e potencialmente menos redundante se os membros da equipe estivessem em constante interação, assim comprovando parcialmente a “Lei de Conway” (CONWAY, 1968).

Além disso, foram observados alguns impedimentos relacionados aos resultados do projeto e, portanto, foi incluída na categoria C3 da matriz conceitual o critério “desafios” relacionados ao desenvolvimento ágil. A maioria dos impedimentos podem ser associados ao gerenciamento do projeto, coordenação, comunicação, de acordo com Wiklund et al. (2013). Os autores também verificaram que parte dos impedimentos nos resultados de projetos ágeis correspondem à fraca infraestrutura que define a divisão de responsabilidades para testes e, portanto, dificultando a comunicação entre as equipes de desenvolvimento e as equipes de testes.

A Tabela 12 apresenta as características gerais das pesquisas referentes a desenvolvimento de software ágil e comunicação (o resultado da soma não é necessariamente 100% pois cada *paper* pode contribuir diversas vezes nas categorias apresentadas).

Tabela 12 – Características gerais das pesquisas analisadas

| Categoria | Item | # Artigos | Porcentagem |
|-----------------------------------|--|------------------|--------------------|
| Abordagem de Pesquisa | Qualitativo | 83 | 63.36% |
| | Quantitativo | 27 | 20.61% |
| | Conceitualização | 24 | 18.32% |
| | Relatórios de experimentos | 20 | 15.27% |
| Método de Pesquisa | Estudo de caso | 78 | 59.54% |
| | Experimento | 8 | 6.11% |
| | Revisão de literatura | 2 | 1.53% |
| | Etnografia | 2 | 1.53% |
| | Teoria fundamentada | 25 | 19.08% |
| | Grupo de foco | 7 | 5.34% |
| | Pesquisa de ação | 7 | 5.34% |
| | Simulação | 11 | 8.40% |
| | Estatístico | 2 | 1.53% |
| | Observacional | 2 | 1.53% |
| | Estudo de Campo | 2 | 1.53% |
| | Avaliação de Especialista | 2 | 1.53% |
| | Design Science | 1 | 0.76% |
| | Mapas conceituais | 1 | 0.76% |
| | Desconhecido | 3 | 2.29% |
| Técnica de Coleta de Dados | Entrevistas | 59 | 45.04% |
| | Questionários | 42 | 32.06% |
| | Arquivos de log | 3 | 2.29% |
| | Notas de campo | 9 | 6.87% |
| | Gravação de vídeo | 7 | 5.34% |
| | Gravação de áudio | 14 | 10.69% |
| | Captura de Tela | 1 | 0.76% |
| | Fotos | 6 | 4.58% |
| | Observações | 24 | 18.32% |
| | Documentos do Projeto (incl. Código Fonte) | 22 | 16.79% |
| | Grupos Focais | 1 | 0.76% |
| | Workshops | 3 | 2.29% |
| | Literatura | 9 | 6.87% |
| | (PROM) PRO Metrics | 2 | 1.53% |
| | Sensor de Resposta Galvânica da Pele | 1 | 0.76% |
| | Polígrafo | 1 | 0.76% |
| | Desconhecido | 18 | 13.74% |

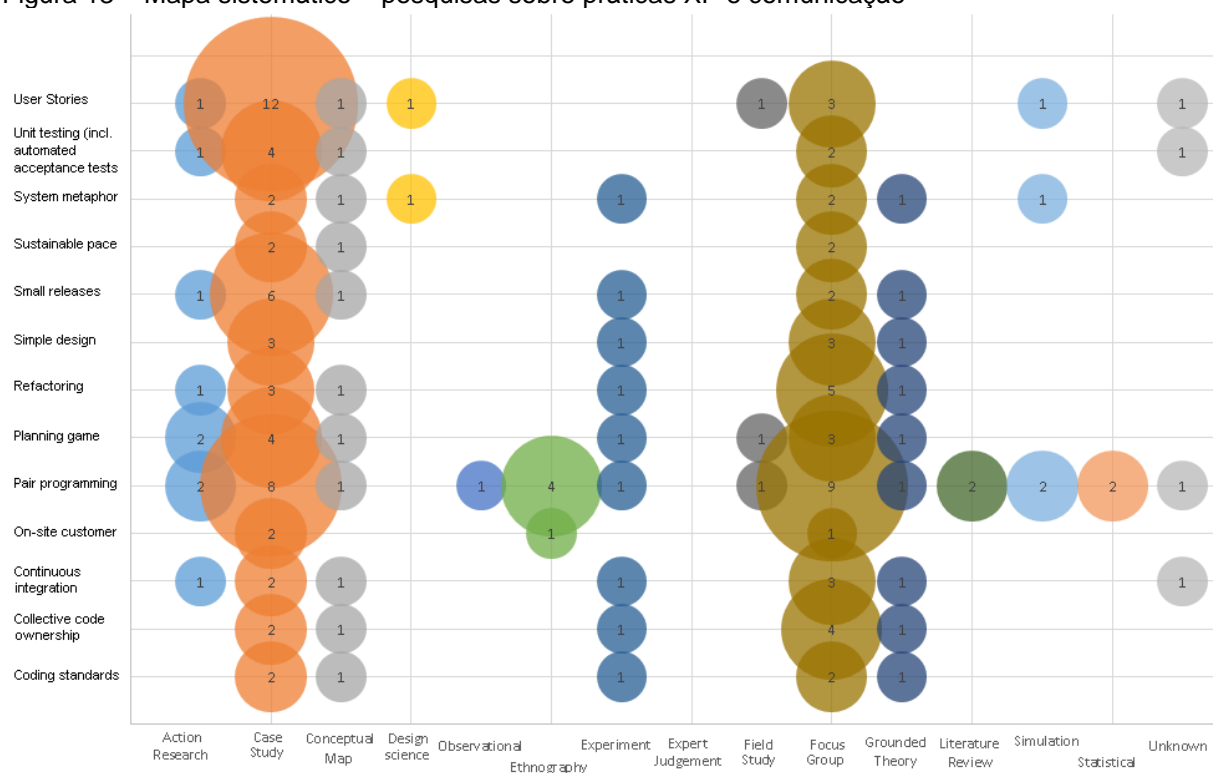
Fonte: adaptado (atualizado) de Hummel, Rosenkranz e Holten (2013).

Vale ressaltar que esta pesquisa encontrou técnicas de coleta primária de dados diferenciados em relação à original. Um dos destaques foi o uso da ferramenta PROM (PRO Metrics), uma automação desenvolvida para coleta e análise de métricas de desenvolvimento de software (como tempo de desenvolvimento, *bugs* encontrados e corrigidos, tamanho do software, entre outros) que é explicada em maiores detalhes por Sillitti, Janes e Succi (2003). Também destacam-se o uso de outros sensores para aferir

respostas físicas dos comportamentos humanos em reuniões, como os sensores de resposta galvânica da pele (mede o grau da condutância da pele, que varia conforme a ação de glândulas sudoríparas, indicando o nível do estado fisiológico de excitação) ou polígrafos para registrar medidas de respiração, pulso, pressão, entre outras (DAFOULAS et al., 2017).

A Figura 13 apresenta o mapa conceitual da pesquisa sobre práticas do XP e comunicação. Observa-se uma distribuição relativamente proporcional sobre todo o mapa (tanto através dos métodos de pesquisa quanto pelas práticas do *framework*), com ênfase em pesquisas do tipo estudo de caso e grupos focais para *pair programming* e *user stories*.

Figura 13 – Mapa sistemático – pesquisas sobre práticas XP e comunicação

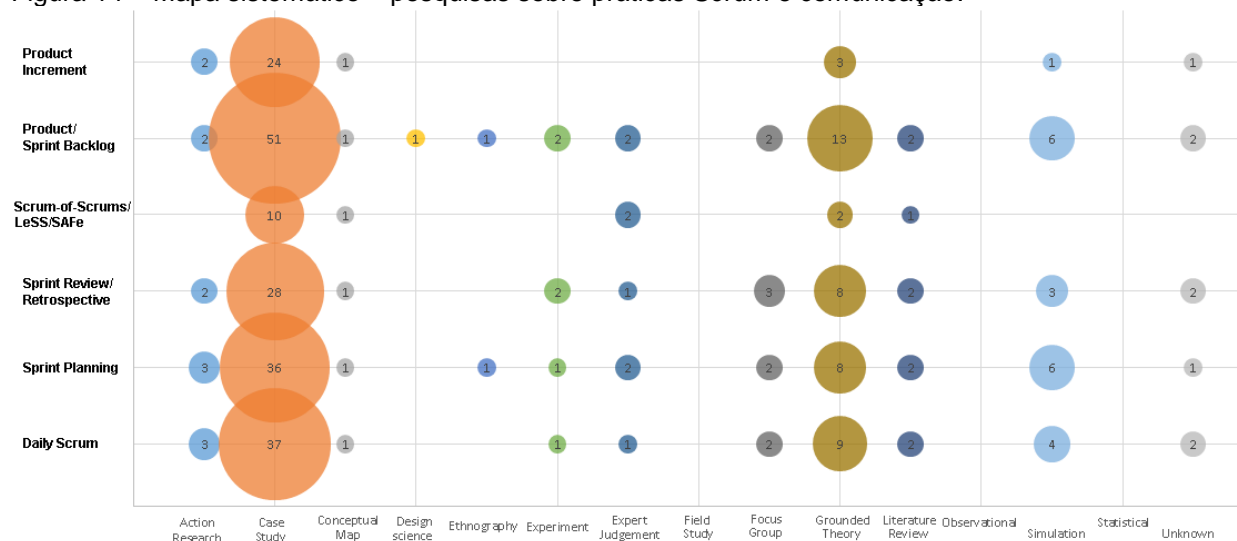


Fonte: própria.

A Figura 14 apresenta o mapa conceitual da pesquisa sobre práticas do Scrum e comunicação. Observa-se uma distribuição menos proporcional em relação aos métodos de pesquisa, focando em pesquisas do tipo estudo de caso e teoria fundamentada. A

extensão das práticas Scrum no mapa conceitual apresenta relativa proporção em sua distribuição, apesar de mostrar que o maior foco de pesquisa é o *backlog* do produto ou da *sprint*.

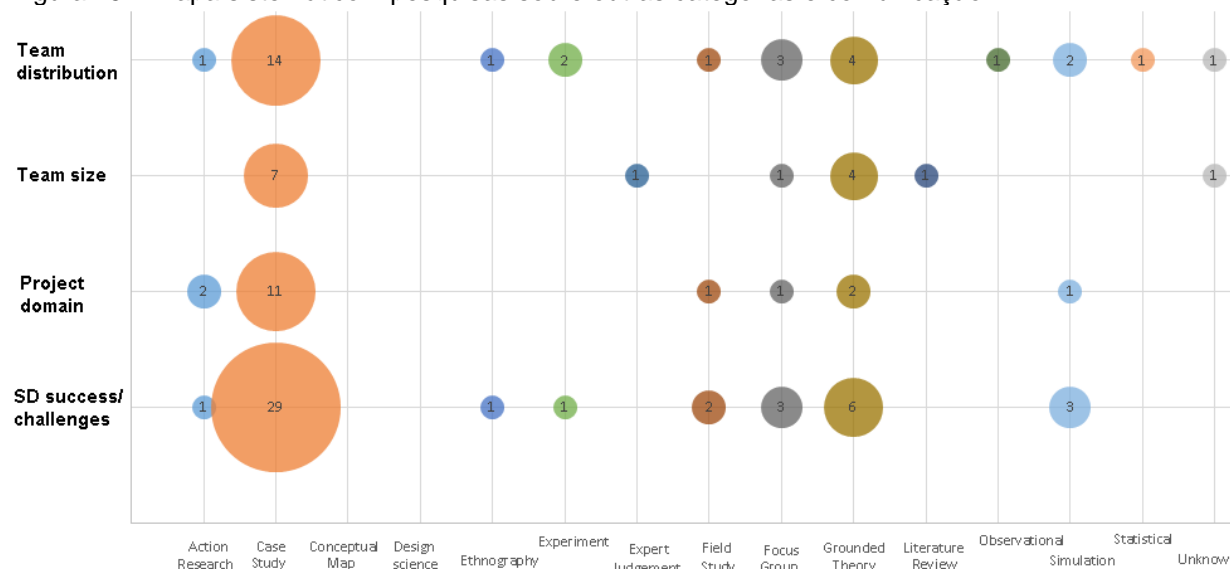
Figura 14 – Mapa sistemático – pesquisas sobre práticas Scrum e comunicação.



Fonte: própria.

A Figura 15 apresenta o mapa conceitual da pesquisa sobre outras categorias e comunicação.

Figura 15 – Mapa sistemático – pesquisas sobre outras categorias e comunicação.



Fonte: própria.

Não foram encontrados estudos que apresentam evidências empíricas ou estatísticas comparando o uso das abordagens ágeis (Scrum ou XP) e indicando qual das duas corresponde a um melhor resultado, de forma que a lacuna de pesquisa anterior ainda persiste. Além disso os resultados apenas dão a entender que melhores técnicas de comunicação influenciam num melhor resultado de projeto ou que baixa qualidade na comunicação classifica um desafio ou barreira ao sucesso do projeto. Em outras palavras, a maioria dos estudos em todos os domínios da matriz conceitual apresenta uma combinação de estudos de casos, pesquisas teóricas/bibliográficas e pesquisas de opinião (estudos de campo e grupos focais envolvendo entrevistas, questionários e avaliações de especialistas). Faltam experiências, simulações, dados concretos em geral que respaldem as sugestões apresentadas pelas demais pesquisas.

5. CONCLUSÕES

Foi observado através do desenvolvimento da revisão sistemática aqui apresentada que o papel da comunicação permanece representando um valor de fundamental importância ao desenvolvimento de software ágil e, portanto um tema de interesse no meio acadêmico. As práticas que foram apresentadas como facilitadoras de comunicação no trabalho de Hummel, Rosenkranz e Holten apresentaram nesta atualização um maior índice de resultados em pesquisas, o que reforça a validade do estudo original e responde às questões de pesquisa RQ1 e RQ2 aqui propostas.

No tocante à questão de pesquisa RQ3, depreende-se dos resultados deste trabalho que as mesmas lacunas de pesquisa apontadas por Hummel, Rosenkranz e Holten foram encontradas para as categorias de saída e para a de processos em todas as subcategorias. Já para a categoria de entrada persistem as mesmas lacunas de pesquisa apenas para a subcategoria do domínio do projeto, enquanto que diversas pesquisas foram encontradas associando o tamanho e/ou distribuição das equipes à barreiras comunicativas em relação ao sucesso do projeto, propondo soluções como adoção de ágil escalonado, práticas de projeto, ou o uso de artefatos tecnológicos para infraestrutura comunicacional.

Apesar de não ser completamente abrangente, uma das lacunas de pesquisa referente à categoria de entrada apontadas trabalho de Hummel, Rosenkranz e Holten foi encontrada neste trabalho, explicando quais condições envolvem e quais fatores no nível organizacional, de grupo e individual afetam a comunicação bem-sucedida em desenvolvimento de software ágil confirmando, inclusive, a chamada “Lei de Conway”, conforme apresentado no Capítulo 4.2.

Grande parte dos resultados tratam sobre o escalonamento da abordagem ágil para grandes empresas ou em organizações internacionais, o que não havia na SLR original. Além disso, com o avanço dos chamados *design patterns*, foram introduzidos nos projetos as diversas arquiteturas (MVC, TDD, DDD, entre outros) e, por isso algumas delas foram encontradas nos resultados das pesquisas como influenciadoras no

desenvolvimento, o que também diferiu da pesquisa original. Este aspecto pode mostrar um possível amadurecimento no uso de técnicas mais avançadas de projeto, análise e desenvolvimento.

A comparação das lacunas de pesquisa encontradas por Hummel, Rosenkranz e Holten e as encontradas neste presente trabalho pode responder à terceira questão de de pesquisa aqui proposta, além de orientar futuros trabalhos acadêmicos referente ao tema da comunicação no desenvolvimento de software ágil.

O trabalho exige muita organização, disciplina e empenho, portanto o contato com o orientador é fundamental para guia teórico e manutenção do foco no planejamento. As *strings* de consulta das bases de pesquisa podem trazer resultados maiores que os previstos, de modo que a quantidade de artigos para leitura completa pode se tornar insustentável. Isso indica que a área das engenharias não possui uma catalogação organizada de palavras-chave para as publicações dos artigos.

REFERÊNCIAS

BABB, J. S.; HODA, R.; NØRBJERG, .. XP in a Small Software Development Business: Adapting to Local Constraints. **Lecture Notes In Business Information Processing**, [s.l.], p.14-29, 2014. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-09546-2_2.

BECK, K.. **Extreme Programming Explained: Embrace Change**. [S.l.]: Addison Wesley Professional, 1999. 207 p. ISBN-13: 9780201616415.

BECK, K.. Embracing change with extreme programming. **Computer**, [s.l.], v. 32, n. 10, p.70-77, 1999. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/2.796139>.

BESSA, D. D.. **Teorias da Comunicação**. Brasília: Laboratório de Geoiconografia e Multimídias - Lagim, 2006. 108 p. ISBN 85-86290-65-3.

BLAU, B. S. et al. Steering through Incentives in Large-Scale Lean Software Development. **Communications In Computer And Information Science**, [s.l.], p.32-48, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-32341-6_3.

BOMBONATTI, D.; GOULÃO, M.; MOREIRA, A.. Synergies and tradeoffs in software reuse - a systematic mapping study. **Software: Practice and Experience**, [s.l.], v. 47, n. 7, p.943-957, 1 ago. 2016. Wiley. <http://dx.doi.org/10.1002/spe.2416>.

CHAOS Summary for 2010. N.i: The Standish Group International, Inc, 2010. Disponível em: <<https://zenexmachina.wordpress.com/2012/07/19/waterfall-vs-agile-a-knowledge-problem-not-a-requirements-problem/>>. Acesso em: 25 dez. 2017.

CHAOS Manifesto. N.i: The Standish Group International, Inc, 2011. Disponível em: <<https://thoughtensemble.com/why-do-technology-projects-fail/>>. Acesso em: 25 dez. 2017.

CONWAY, M. E.. How do Committees Invent? **Datamation**, [s.l.], v. 14, n. 5, p.25-31, abr. 1968. Disponível em: <<http://www.melconway.com/Home/pdf/committees.pdf>>. Acesso em: 21 abr. 2018.

DAFOULAS, G. et al. Understanding Collaboration in Global Software Engineering (GSE) Teams with the Use of Sensors: Introducing a Multi-sensor Setting for Observing Social and Human Aspects in Project Management. **2017 International Conference On Intelligent Environments (ie)**, [s.l.], p.114-121, ago. 2017. IEEE. <http://dx.doi.org/10.1109/ie.2017.40>.

DATTA, S.. How does developer interaction relate to software quality? an examination of product development data. **Empirical Software Engineering**, [s.l.], p.1-35, 3 ago. 2017. Springer Nature. <http://dx.doi.org/10.1007/s10664-017-9534-0>.

DEMISSIE, S.; KEENAN, F.; MCCAFFERY, F.. Investigating the Suitability of Using Agile for Medical Embedded Software Development. **Communications In Computer And Information Science**, [s.l.], p.409-416, 2016. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-38980-6_29.

DOLMAN, R.; SPEARMAN, S.. **AGILE SCALING KNOWLEDGE: THE MATRIX**. 2017. Disponível em: <<http://www.agilescaling.org/ask-matrix.html>>. Acesso em: 22 jun. 2018. FONSECA FILHO, Cleuzio. **História da Computação: O caminho do pensamento e da tecnologia**. Porto Alegre: Edipucrs, 2007. 205 p. (ISBN 978-85-7430-691-9). Disponível em: <<http://www.pucrs.br/edipucrs/online/historiadacomputacao.pdf>>. Acesso em: 17 maio 2017.

GALEALE, G. P. et al. Internet das Coisas aplicada a negócios – Um estudo bibliométrico. **Proceedings Of The 13th Contecsi International Conference On Information Systems And Technology Management**, [s.l.], p.2247-2265, 3 jun. 2016. TECSI. <http://dx.doi.org/10.5748/9788599693124-13contecsi/ps-3969>.

GARBAJOSA, J.; YAGÜE, A.; GONZALEZ, E.. **Communication in Agile Global Software Development: An Exploratory Study**. On The Move To Meaningful Internet Systems: OTM 2014 Workshops, [s.l.], p.408-417, 2014. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-662-45550-0_41.

GÓRSKI, J.; ŁUKASIEWICZ, K.. Towards Agile Development of Critical Software. **Lecture Notes In Computer Science**, [s.l.], p.48-55, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-40894-6_4.

GREENFIELD, J.. **O caso das fábricas de software**. 2004. Microsoft Corporation. Disponível em: <<https://msdn.microsoft.com/pt-br/library/aa480032.aspx>>. Acesso em: 25 jul. 2017.

GUARNIERI, A. C. M.. **Marcondes Filho, Maturana, Bateson e Watzlawick: Novas Tecnologias e Ausência de Comunicação**. Fasci-tech, São Caetano do Sul, v. 1, n. 3, p.22-35, dez. 2010. Disponível em: <<http://www.fatecsaocaetano.edu.br/fascitech/index.php/fascitech/article/view/24>>. Acesso em: 28 out. 2017.

HARPER, D.. **Online Etymology Dictionary**. Disponível em: <<https://www.etymonline.com/word/communication>>. Acesso em: 26 out. 2017.

HOHLFELDT, A. (Ed.). **Comunicação**. In: Enciclopédia INTERCOM de comunicação: Volume 1 – Conceitos. Volume 1 – Conceitos. São Paulo: Intercom, 2010. p. 234-235. ISBN: 978-85-88537-66-8.

HUMMEL, M.; ROSENKRANZ, C.; HOLTEN, R.. The Role of Communication in Agile Systems Development. **Business & Information Systems Engineering**, [s.l.], v. 5, n. 5, p.343-355, 31 jul. 2013. Springer Nature. <http://dx.doi.org/10.1007/s12599-013-0282-4>.

KUUSINEN, K. et al. Knowledge Sharing in a Large Agile Organisation: A Survey Study. **Lecture Notes In Business Information Processing**, [s.l.], p.135-150, 2017. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-57633-6_9.

JIANG, L.; EBERLEIN, A.. **An analysis of the history of classical software development and agile development**. 2009 IEEE International Conference On Systems, Man And Cybernetics, [s.l.], p.3733-3738, out. 2009. IEEE. <http://dx.doi.org/10.1109/icsmc.2009.5346888>.

KAPITSAKI, G. M.; CHRISTOU, M.. Learning from the Current Status of Agile Adoption. **Communications In Computer And Information Science**, [s.l.], p.18-32, 2015. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-27218-4_2.

KITCHENHAM, B.; CHARTERS, S.. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. Durham, Uk: N.i., 2007. 57 p. EBSE Technical

Report. Disponível em:
 <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.471&rep=rep1&type=pdf>>. Acesso em: 04 dez. 2017.

LIMA, M. D. C.; ABBUD, M. E. de O. P.. **Comunicação Organizacional: Histórico, Conceitos e Dimensões**. In: CONGRESSO DE CIÊNCIAS DA COMUNICAÇÃO NA REGIÃO NORTE, 14., 2015, Manaus. Anais... . Manaus: Intercom, 2015. p. 1 - 15. Disponível em: <<http://www.portalintercom.org.br/anais/norte2015/resumos/R44-0415-1.pdf>>. Acesso em: 27 out. 2017.

NEWKIRK, J.. **Introduction to agile processes and extreme programming**. Proceedings Of The 24th International Conference On Software Engineering - Icse '02, Florida, p.695-696, 2002. ACM Press. <http://dx.doi.org/10.1145/581339.581450>.

PLACHKINOVA, M.; PEFFERS, K.; MOODY, G.. Communication Artifacts for Requirements Engineering. **New Horizons In Design Science: Broadening the Research Agenda**, [s.l.], p.104-118, 2015. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-18714-3_7.

RAZZAK, M. A. et al. Transition from Plan Driven to SAFe®: Periodic Team Self-Assessment. **Product-focused Software Process Improvement**, [s.l.], p.573-585, 2017. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-69926-4_47.

REGAN, G. O. A Brief History of Computing. London: Springer-verlag, 2008. 252 p.

SANTANA, A. M. de et al. Relationships Between Communication Structure and Software Architecture: An Empirical Investigation of the Conway's Law at the Federal University of Pernambuco. **2013 3rd International Workshop On Replication In Empirical Software Engineering Research**, [s.l.], p.34-42, out. 2013. IEEE. <http://dx.doi.org/10.1109/reser.2013.18>.

SCHWABER, K.. **SCRUM Development Process. Business Object Design And Implementation:** OOPSLA'95 Workshop Proceedings, Austin, p.117-134, 1997. Springer London. http://dx.doi.org/10.1007/978-1-4471-0947-1_11.

SCHWABER, K.; SUTHERLAND, Jeff. **Resources.** Elaborado por Scrum.org. Disponível em: <<https://www.scrum.org/resources/scrum-guide>>. Acesso em: 17 out. 2017.

SCHWARZER, J. et al. Ambient Surfaces: Interactive Displays in the Informative Workspace of Co-located Scrum Teams. In: NORDICHI '16 PROCEEDINGS OF THE 9TH NORDIC CONFERENCE ON HUMAN-COMPUTER INTERACTION, 9., 2016, Gothenburg, Sweden. **Proceedings...** . Gothenburg, Sweden: Acm, 2016. p. 1 - 4.

SFETSOS, P. et al. Investigating the Impact of Personality and Temperament Traits on Pair Programming: A Controlled Experiment Replication. **2012 Eighth International Conference On The Quality Of Information And Communications Technology**, [s.l.], p.57-65, set. 2012. IEEE. <http://dx.doi.org/10.1109/quatic.2012.36>.

SIAU, K.; LONG, Y.; LING, M.. **Toward a Unified Model of Information Systems Development Success.** Journal Of Database Management, Hershey, Pa, Usa, v. 21, n. 1, p.80-101, 2010. IGI Global. <http://dx.doi.org/10.4018/jdm.2010112304>.

SILLITTI; JANES; SUCCI. Collecting, integrating and analyzing software metrics and personal software process data. **Proceedings Of The 20th IEEE Instrumentation Technology Conference (cat no 03ch37412) Eurmic-03**, [s.l.], p.336-342, 2003. IEEE. <http://dx.doi.org/10.1109/eurmic.2003.1231611>.

STAVRU, S.; KRASTEVA, I.; ILIEVA, S.. Challenges for Migrating to the Service Cloud Paradigm: An Agile Perspective. **Web Information Systems Engineering – Wise 2011 And 2012 Workshops**, [s.l.], p.77-91, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-38333-5_10.

SUTHERLAND, J.; SCHWABER, K.. **The Scrum Guide:** The Definitive Guide to Scrum: The Rules of the Game. 2016. Disponível em: <<http://www.scrumguides.org/download.html>>. Acesso em: 17 out. 2017.

SUTHERLAND, J.. **Agile Can Scale:** Inventing and Reinventing SCRUM in Five Companies. Cutter Business Technology Journal. [s.l.], p. 5-11. 30 nov. 2001.

TAKEUCHI, H.; NONAKA, I.. **The New New Product Development Game.** 1986. Harvard Business Review. Disponível em: <<https://hbr.org/1986/01/the-new-new-product-development-game>>. Acesso em: 10 out. 2017.

UMAN, L. S.. **Systematic Reviews and Meta-Analyses.** Journal Of The Canadian Academy Of Child And Adolescent Psychiatry, [s.l.], v. 20, n. 1, p.57-59, fev. 2011. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3024725/>>. Acesso em: 18 jun. 2018.

URBACH, N.; SMOLNIK, S.; RIEMPP, G.. **The State of Research on Information Systems Success.** Business & Information Systems Engineering, [s.l.], v. 1, n. 4, p.315-325, 18 jul. 2009. Springer Nature. <http://dx.doi.org/10.1007/s12599-009-0059-y>.

VALLON, R. et al. Inter-organizational Co-development with Scrum: Experiences and Lessons Learned from a Distributed Corporate Development Environment. **Lecture Notes In Business Information Processing**, [s.l.], p.150-164, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-38314-4_11.

VAN RULER, B.. **Agile public relations planning: The Reflective Communication Scrum.** Public Relations Review, [s.l.], v. 41, n. 2, p.187-194, jun. 2015. Elsevier BV. <http://dx.doi.org/10.1016/j.pubrev.2014.11.008>.

WEBSTER, J.; WATSON, R. T.. **Analyzing the past to prepare for the future: writing a literature review.** Mis Quarterly. Minneapolis, Mn, Usa, p. 13-22. jun. 2002.

WELLS, D.. **Extreme Programming:** A gentle introduction. 1999. Disponível em: <<http://www.extremeprogramming.org/>>. Acesso em: 20 out. 2017.

WEST, D.. **Updates to the Scrum Guide:** The 5 Scrum values take center stage. 2016. Elaborado por: Scrum.org. Disponível em: <<https://www.scrum.org/resources/blog/updates-scrum-guide-5-scrum-values-take-center-stage>>. Acesso em: 17 out. 2017.

WIKLUND, K. et al. Impediments in Agile Software Development: An Empirical Investigation. **Product-focused Software Process Improvement**, [s.l.], p.35-49, 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-39259-7_6.

WIRTH, N.. A Brief History of Software Engineering. **IEEE Annals Of The History Of Computing**, [s.l.], v. 30, n. 3, p.32-39, jul. 2008. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mahc.2008.33>.

WOJCIECHOWSKI, A.; WESOLOWSKI, M.; COMPLAK, W.. Experimental Evaluation of 'On-Site Customer' XP Practice on Quality of Software and Team Effectiveness. **On The Move To Meaningful Internet Systems: OTM 2010 Workshops**, [s.l.], p.269-278, 2010. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-16961-8_45.

YAGÜE, A. et al. An exploratory study in communication in Agile Global Software Development. **Computer Standards & Interfaces**, [s.l.], v. 48, p.184-197, nov. 2016. Elsevier BV. <http://dx.doi.org/10.1016/j.csi.2016.06.002>.