



Universidade Federal do Ceará
Campus Quixadá
Curso de Sistemas de Informação

ANA LICE FREIRES DOS SANTOS

**UMA REVISÃO SISTEMÁTICA SOBRE FERRAMENTAS
COLABORATIVAS NO DESENVOLVIMENTO DE
SOFTWARE**

Quixadá
2018

ANA LICE FREIRES DOS SANTOS

UMA REVISÃO SISTEMÁTICA SOBRE FERRAMENTAS
COLABORATIVAS NO DESENVOLVIMENTO DE SOFTWARE

Monografia apresentada ao Curso de Sistemas de Informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação.

Orientador: Prof. Ma Jéssyka Flavyanne Ferreira Vilela

Quixadá
2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S233 Santos, Ana Lice Freires dos.

Uma revisão sistemática sobre ferramentas colaborativas no desenvolvimento de software / Ana Lice Freires dos Santos. – 2018.

61 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2018.

Orientação: Profa. Ma. Jéssyka Flavyanne Ferreira Vilela.

1. Ferramentas colaborativas. 2. Revisão sistemática. I. Título.

CDD 004.6

ANA LICE FREIRES DOS SANTOS

UMA REVISÃO SISTEMÁTICA SOBRE FERRAMENTAS
COLABORATIVAS NO DESENVOLVIMENTO DE SOFTWARE

Monografia apresentada ao Curso de Sistemas de Informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação.

Aprovada em: __/__/____

BANCA EXAMINADORA

Prof. Ma Jéssyka Flavyanne Ferreira Vilela
Universidade Federal do Ceará (UFC)

Prof. Dr. Jefferson de Carvalho Silva
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulyne Matthews Jucá
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

A Deus por ter me dado saúde, força e, sobretudo, coragem, para superar as dificuldades que se apresentaram ao longo dessa formação acadêmica.

A minha mãe, que com minha ajuda lutou pela vida e que hoje se encontra saudável físico e psicologicamente.

A minha amada tia Liduína, falecida há um ano, que foi minha segunda mãe e que sempre me incentivou nessa jornada de estudante.

Ao meu marido pelo amor e pela parceria na difícil rotina de estudar e, ao mesmo tempo, ser esposa e professora.

Aos gestores e professores da EEF Juscelino Kubitschek – Chorozinho –, na qual trabalho, pela compreensão, solicitude e colaboração.

A esta universidade, ao seu corpo docente, à direção à administração que oportunizaram a possibilidade que hoje vislumbro no meu futuro profissional, eivado pela acendrada confiança no mérito ética aqui presentes.

A minha orientadora Prof^a. Ms. Jéssyka Flavyanne Ferreira Vilela pela melhor orientação que um estudante pode ter, pelo suporte que me do início ao fim deste trabalho, pelo tempo dispensado em sua execução, maturação e concretização, pelos incentivos, por sua empatia, por seu profissionalismo e colaboração no seu paciente trabalho de revisão deste TCC.

Agradeço todos os professores por me proporcionarem conhecimento, não apenas o racional, mas também o que se manifesta no caráter na afetividade, elementos que considero fundamentais no processo de formação profissional, por toda a dedicação e por terem me ensinado a aprender. palavra mestre nunca fará justiça aos professores, aos quais, sem nominar, terão os meus eternos agradecimentos.

À Banca pela disponibilidade de tempo e justiça na avaliação deste trabalho.

RESUMO

Contexto: Ferramentas colaborativas tem possibilitado aproximação de equipes que estão geograficamente distantes. Elas possibilitam que *stakeholders* de um projeto desenvolvam suas atividades de maneira mais eficaz e interativa. Diversas ferramentas colaborativas existem para o desenvolvimento de software com diferentes características e preços, aplicáveis em diferentes contextos e em fases distintas do desenvolvimento. Sendo assim, é uma tarefa desafiadora para os *stakeholders* decidirem quais delas são adequadas ao seu projeto e atendem as suas necessidades. Objetivo: este trabalho visa identificar as ferramentas colaborativas usadas em todas as etapas de desenvolvimento de software. Método: para atingir esse objetivo, uma revisão sistemática da literatura foi conduzida que retornou 648 trabalhos de quatro bases de dados sendo 165 aceitos para extração de dados. Resultados: a partir dos dados extraídos na revisão, foi elaborado um catálogo de ferramentas colaborativas. Conclusão: espera-se contribuir para que os engenheiros de software e outros *stakeholders* envolvidos no desenvolvimento de software escolham quais ferramentas colaborativas podem ser utilizados em seus projetos. Assim, a partir da utilização de ferramentas adequadas, almeja-se que a cooperação, a colaboração e a comunicação entre os participantes da equipe sejam facilitadas.

Palavras-chaves: Ferramentas Colaborativas. Revisão Sistemática. Modelo 3C.

ABSTRACT

Context: Collaborative tools contribute to the approximation of teams that are geographically distant. They enable stakeholders to develop their activities more effectively and interactively. Several collaborative tools exist for the development of software with different characteristics and prices, applicable in different contexts and at different stages of development. Therefore, it is a challenging task for stakeholders to decide which ones are appropriate to their project and meet their needs. Objective: this work aims to identify the collaborative tools used in all stages of software development. Method: to achieve this goal, a systematic review of the literature was conducted which returned 648 works from four databases being 165 accepted for data extraction. Results: from the data extracted in the review, a catalog of collaborative tools was elaborated. Conclusion: it is expected to contribute to software engineers and other stakeholders involved in software development choosing which collaborative tools can be used in their projects. Thus, through the use of appropriate tools, it is hoped that cooperation, collaboration and communication among team members will be facilitated.

Key-words: Collaborative Tools. Systematic review. Model 3C.

LISTA DE ILUSTRAÇÕES

Figura 1 – Atividades do processo de desenvolvimento de software no modelo cascata	18
Figura 2 – Passos para realização de uma revisão sistemática.....	20
Figura 3 – Porcentagem de dados retornados por base de dados.....	27
Figura 4 – Porcentagem dos artigos selecionados em relação a quantidade de artigos retornados.....	28
Figura 5 – Distribuição dos artigos selecionados ao longo do tempo.....	28

LISTA DETABELAS

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto.....	14
Tabela 2 – Bases utilizadas na pesquisa.....	24
Tabela 3 – Resultado da seleção dos artigos.....	27
Tabela 4 – Ferramentas Colaborativas retornadas nos artigos da base ACM.....	29
Tabela 5 – Ferramentas Colaborativas retornadas nos artigos da Base IEEE (Parte 1). . .	30
Tabela 6 – Ferramentas Colaborativas retornadas nos artigos da base IEEE (Parte 2). . .	31
Tabela 7 – Ferramentas Colaborativas retornadas nos artigos da base Science Direct. . .	32
Tabela 8 – Ferramentas Colaborativas retornadas nos artigos da base Springer	33
Tabela 9 – Funcionalidades presentes nas Ferramentas Colaborativas.....	34
Tabela 10 – Funcionalidades presentes nas Ferramentas Colaborativas.....	35
Tabela 11 – Funcionalidades presentes nas Ferramentas Colaborativas.....	36
Tabela 12 – Funcionalidades presentes nas Ferramentas Colaborativas.....	37
Tabela 13 – Modelo do Processo	38
Tabela 14 – Licença das Ferramentas Colaborativas.....	39
Tabela 15 – Licença das Ferramentas Colaborativas.....	40
Tabela 16 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.	40
Tabela 17 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.	41
Tabela 18 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.	42
Tabela 19 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.	43
Tabela 20 – Fase do Processo de Desenvolvimento de Software presente nas Ferramentas Colaborativas.	44
Tabela 21 – Links das Ferramentas Colaborativas.....	44
Tabela 22 – Links das Ferramentas Colaborativas.....	45

LISTA DE ABREVIATURAS E SIGLAS

RSL	Revisão Sistemática da Literatura
IBIS	<i>Internet-Based Inspection System</i>
RTDWD	<i>Real-time Distributed Wideband-Delphi</i>
WiT	<i>Web inspection Tool</i>
CSI	<i>Collaborative Software Inspection</i>
CoP	<i>Communities of Practice</i>
AISA	<i>Asynchronous Inspector of Software Artifacts</i>
SCM	<i>Software Configuration Management</i>
RCS	<i>Revision Control System</i>
FADYRCOS	<i>FrAmework for Dynamic Reconfiguration of Networked</i>
DynC	<i>Dynamic Community framework COllaborative Systems</i>
DyCE	<i>Dynamic Cooperation Environment</i>
OSD	<i>Offshore Software Development</i>
IBERE	<i>Internet-Based Empirical RequirementsEvaluation</i>
RCS	<i>Revision Control System</i>
SVN	<i>Subversion</i>

SUMÁRIO

1. INTRODUÇÃO	11
2. TRABALHOS RELACIONADOS	13
3. OBJETIVOS	14
3.1 Objetivo Geral	14
3.2 Objetivos específicos	14
4. FUNDAMENTAÇÃO TEÓRICA	15
4.1 Sistemas Colaborativos	15
4.2 Processo de desenvolvimento de software	17
4.3 Revisão sistemática da literatura	19
5. PROCEDIMENTOS METODOLÓGICOS	20
6. PROTOCOLO DA REVISÃO SISTEMÁTICA	21
6.1 Identificação da necessidade da revisão	21
6.2 Comissionamento da revisão	21
6.3 Perguntas de Pesquisa	22
6.4 Palavras chaves utilizadas	22
6.5 String de busca	23
6.6 Bases de dados	23
6.7 Critérios de inclusão e exclusão	24
6.8 Procedimento para seleção de estudos	25
6.9 Ameaças à validade	25
7. RESULTADOS	26
7.1 P1. Quais são as ferramentas colaborativas existentes?	29
7.2 P2. Quais são as funcionalidades de um sistema colaborativo descritas no trabalho?	34
7.3 P3. As ferramentas são utilizadas em qual tipo de modelo de processo de desenvolvimento de software (ex: ágil, não-ágil, não-classificado)? . . .	38
7.4 P4. Qual a licença da ferramenta?	39
7.5 P5. Em que fase do processo de desenvolvimento do software as ferramentas são utilizadas (ex: requisitos, arquitetura, teste)?	40

8. CONCLUSÃO	46
REFERÊNCIAS	47

1 INTRODUÇÃO

Sistemas Colaborativos são ambientes virtuais que facilitam a colaboração e a comunicação entre um grupo de pessoas que almejam um objetivo em comum. Isto ocorre mesmo que elas não estejam em um mesmo espaço físico, ou ainda, que elas não estejam se comunicando de forma síncrona (COSTA; PIMENTEL, 2011).

A qualidade e o grau de colaboração são influenciados tanto pela distância geográfica quanto pela comunicação informal entre as pessoas (ALLEN *et al.*, 1984). Para Mendes *et al.* (2007), a interação é um processo de colaboração mútua que facilita a aquisição de conhecimento. Portanto, quanto mais distante as pessoas estão fisicamente, mais difícil se torna a comunicação e a colaboração, acarretando atrasos e desperdício de recursos.

A disseminação da internet e dos aparelhos computacionais proporcionou um maior aproveitamento de recursos tecnológicos que buscam maximizar a colaboração entre participantes de grupos co-localizados e distribuídos em ambientes virtuais ou em realidade aumentada (FILIPPO *et al.*, 2007). A comunicação, a colaboração e a cooperação entre os membros de uma equipe são de grande importância para o desenvolvimento de qualquer trabalho colaborativo.

A produção de um software envolve a colaboração das pessoas envolvidas nesse processo denominadas *stakeholders*. Estes realizam diferentes atividades, as quais necessitam ser integradas, desde a coleta e análise de requisitos até a implantação no ambiente do cliente (SOUZA; MARCZAK; PRIKLANDNICKI, 2012). Assim, caso seja necessário realizar uma modificação em algum aspecto do projeto, a equipe precisa se comunicar e artefatos precisam ser atualizados para manter a consistência e a integridade. Dessa maneira, os membros da equipe podem compreender as mudanças ocorridas, melhorando sua produtividade. Por conseguinte, o desenvolvimento de software envolve cooperação e comunicação entre pessoas em diferentes fases, tendo como principal objetivo a integração da equipe na construção de um software com qualidade.

Ferramentas como *Doors*¹ e *IRqA*² permitem que engenheiros de software compartilhem informações, descrevam os casos de usos remotamente, entre outras atividades. Na modelagem dos requisitos do software, estão disponíveis: *Gliffy*³ e *Creately*⁴, que suportam diferentes tipos de diagramas, além de oferecerem à equipe a possibilidade de criação de *blogs*.

¹ www.ibm.com/software/awdtools/doors

² www.visuresolutions.com

³ www.gliffy.com

⁴ <http://creately.com>

Os testes podem ser realizados utilizando-se, por exemplo, TestLink⁵ e Selenium⁶, que organizam e facilitam a interação ao sistema (LANUBILE *et al.*, 2010).

Existem várias ferramentas colaborativas no desenvolvimento de software disponíveis no mercado com diferentes características e preços, aplicáveis em diferentes contextos e em fases distintas do desenvolvimento. Sendo assim, é uma tarefa desafiadora para os *stakeholders* decidirem quais delas são adequadas ao seu projeto e atendem as suas necessidades.

A partir desse problema, constatou-se a premência em se fazer um levantamento das ferramentas colaborativas existentes. Como resultado desse levantamento, será elaborado um catálogo classificando as ferramentas de acordo com diferentes critérios como, por exemplo, a etapa do processo de desenvolvimento e o contexto mais adequado.

Alguns trabalhos sobre ferramentas colaborativas estão disponíveis na literatura (LANUBILE *et al.*, 2010; BOOCH; BROWN, 2003). (LANUBILE *et al.*, 2010) fornece uma visão geral de ferramentas e tecnologias para melhorar a colaboração no processo de desenvolvimento de software bem como resume as experiências e tendências observadas na *International Conference on Global Software Engineering*⁷. (BOOCH; BROWN, 2003) examina os pontos de atrito no processo de desenvolvimento de software e os mecanismos que reduzem esse atrito. Em seguida, apresenta uma variedade de sites, além do domínio da engenharia de software, que fornecem alguns desses mecanismos.

Contudo, os trabalhos apresentados acima apresentam uma visão geral com alguns exemplos de ferramentas organizadas de forma não sistemática. Sendo assim, este trabalho objetiva a criação de um catálogo sobre ferramentas colaborativas.

O catálogo foi elaborado por meio de uma revisão sistemática da literatura (RSL) (KITCHENHAM; CHARTERS, 2007), cujos passos são descritos na Seção 4.3, sobre os trabalhos que abordam ferramentas colaborativas no desenvolvimento de software. A partir do levantamento realizado, as ferramentas foram analisadas e classificadas considerando diferentes critérios a serem definidos. Dentre esses critérios, pode-se citar a etapa no processo de desenvolvimento do software, tais como: levantamento de requisitos, arquitetura, implementação, testes, implantação e manutenção no qual a ferramenta pode ser utilizada.

Espera-se que o presente trabalho contribua para que os engenheiros de software e outros *stakeholders* envolvidos no desenvolvimento de software escolham quais ferramentas

⁵ <http://testlink.sourceforge.net>

⁶ <http://seleniumhq.org>

⁷ <http://icgse.org/>

colaborativas podem ser utilizados em seus projetos. Assim, a partir da utilização de ferramentas adequadas, almeja-se que a cooperação, a colaboração e a comunicação entre os participantes da equipe sejam facilitadas.

O texto está estruturado da seguinte forma: no Segundo Capítulo, serão discutidos os principais trabalhos relacionados; no Terceiro, são apresentados os objetivos gerais e específicos; o Quarto sumariza a fundamentação teórica; o Quinto descreve os procedimentos metodológicos. No Sexto, é apresentado o protocolo da revisão sistemática. O Sétimo discute os resultados e, finalmente, o Oitavo apresenta conclusões e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Encontram-se diversos trabalhos relacionados as ferramentas colaborativas e a engenharia de software, porém são poucos aqueles que trazem informações que as sistematizem e as inter-relacionem. Na pesquisa realizada sobre ferramentas colaborativas no processo de desenvolvimento de software, foram encontrados três trabalhos relacionados: *Collaboration tools for global software engineering* (LANUBILE *et al.*, 2010), *Collaboration in Software Engineering: A Roadmap* (WHITEHEAD, 2007) e *Collaborative Development Environments* (BOOCH; BROWN, 2003) .

O trabalho de Lanubile *et al.* (2010) apresenta exemplos de ferramentas por etapa do desenvolvimento do software. Ele apresenta uma síntese das tecnologias pesquisadas em edições recentes do ICGSE (*International Conference on Global Software Engineering*) e em empresas que utilizam ferramentas distribuídas. Apesar de abordar a mesma problemática que a pesquisa realizada nesse trabalho, o artigo não apresenta um guia para escolha dessas ferramentas ou uma lista exaustiva de ferramentas disponíveis, obtida de forma sistemática.

A contribuição de (WHITEHEAD, 2007) assemelha-se bastante com esta pesquisa, pois analisa artigos que abordam ferramentas colaborativas no processo de desenvolvimento de software. Contudo, a lista de ferramentas não está atualizada, uma vez que o artigo foi publicado há onze anos. O autor aborda todas as etapas do desenvolvimento de um software, porém, enfatiza ferramentas de suporte a requisitos e ferramentas de colaboração na linguagem de modelagem unificada, do inglês *Unified Modeling Language* (UML). Esse foco o diferencia desta pesquisa, pois ela contemplará as ferramentas utilizadas em todo o processo de desenvolvimento de software, fazendo uma análise e avaliação por meio de comparações dos contextos, preços, funcionalidades, etc.

O artigo de (BOOCH; BROWN, 2003), assim como este trabalho, apresenta uma relação de ferramentas colaborativas de forma abrangente. Os autores definem as características necessárias para que um ambiente seja considerado colaborativo e classificam o espectro de ambiente de desenvolvimento colaborativo em diversos domínios, como gestão de ativos, serviços de informação, infraestrutura, comunidade e desenvolvimento de software, e apresentam exemplos. Sendo assim, o trabalho não analisa em detalhes as ferramentas usadas no processo de desenvolvimento de software, o que diferencia deste trabalho.

A tabela 1 apresenta as principais diferenças e semelhanças entre esta pesquisa e os trabalhos relacionados discutidos.

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto.

Critério/Trabalho	Lanubile et al. (2010)	Whitehead (2007)	Booch (2003)	Trabalho Proposto (2018)
Discute as características de sistemas colaborativos	Parcial	Parcial	Sim	Sim
Apresenta exemplos de ferramentas colaborativas	Não	Sim	Não	Sim
Aborda o processo de desenvolvimento de software	Sim	Sim	Parcial	Sim
Método de Pesquisa	Escolhidas a partir de publicações do IGSCCE	Não descrito	Não descrito	Revisão sistemática
Analisa outros domínios além da engenharia de software	Não	Não	Sim	Não

Fonte: Elaborada pela autora.

A partir dessa identificação das semelhanças e diferenças, os objetivos desse trabalho foram delineados conforme descrito no próximo capítulo.

3 OBJETIVOS

3.1 Objetivo Geral

Produzir uma revisão sistemática sobre ferramentas colaborativas utilizadas ao longo das diversas etapas do desenvolvimento de software.

3.2 Objetivos específicos

- a) Discutir as funcionalidades presentes em ferramentas colaborativas;
- b) Conduzir uma revisão sistemática sobre ferramentas colaborativas no desenvolvimento do software;

- c) Relacionar as ferramentas colaborativas de acordo com o cenário do projeto a ser desenvolvido.

Os conceitos envolvidos na realização desses objetivos são explicados no próximo capítulo.

4 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são conceituados os principais objetos que fundamentam o trabalho: sistemas colaborativos, processo de desenvolvimento de software e revisão sistemática da literatura.

4.1 Sistemas Colaborativos

A palavra “colaborativa”, segundo o dicionário Aurélio online⁸, é um adjetivo que significa aquilo que “envolve ajuda, auxílio produzido por outras pessoas”. Por sua vez, o vocábulo “sistemas” é definido como “reuniões” dos elementos abstratos e concretos que se interligam a fim de formar um todo organizado” ou como “Espaço para compartilhamento de ideias e aprendizado de experiências (KRAUT *et al.*, 1996).”

Para uma melhor compreensão do conceito de sistemas colaborativos, mais especificamente do modelo 3C (comunicação, coordenação, cooperação) (VIVACQUA; GARCIA *et al.*, 2011), é importante compreender a definição de ontologias. De acordo com os autores, uma ontologia tem como principal objetivo facilitar o diálogo entre os participantes de uma comunidade, possibilitando que pessoas externas possam compreender a sua visão sobre domínio.

Sistemas colaborativos são construídos a partir do conhecimento de quatro ontologias (VIVACQUA; GARCIA *et al.*, 2011): a formação de grupo; a comunicação; a coordenação; e a cooperação. Para que se possa compreender como se produz um ambiente colaborativo, nesta fundamentação teórica, explicar-se-á as quatro ontologias que formam a colaboração, um dos principais objetos desta pesquisa.

- **Formação de grupo:** trabalhar em equipe é uma atividade que envolve vários participantes. Cada participante é um indivíduo a quem a equipe consigna competência e confiança para desenvolver uma tarefa. Em contrapartida, o participante assume um compromisso com a equipe e com o contratante. Por meio de negociações, são definidos os objetivos

⁸ <https://www.dicio.com.br/>

necessários para gerar o produto. A equipe de profissionais torna-se responsável pela produção do produto, sendo assim, o grupo precisa manter seus membros motivados para desenvolverem as atividades que lhes forem designadas. O sucesso e fracasso não são resultados individuais, mas, sim, algo coletivo. Todos são responsáveis.

- **Comunicação:** essa ontologia é um requisito fundamental para se obter um trabalho em equipe. Se não houver comunicação, os participantes não entenderão as suas atividades e desenvolverão ações inúteis, sem importância para o projeto. No intuito de evitar que isso ocorra, primeiramente, deve ser verificado se o emissor e o receptor estão dispostos a trocar mensagens. Conseqüentemente, para que isso ocorra, é necessário um protocolo de comunicação, isto é, que o emissor inicie a conversa e que o receptor responda. Porém, para que a mensagem seja compreendida, o emissor deve escolher uma codificação que o receptor conheça. Em seguida, ele define qual é a melhor forma de transmitir a mensagem, ou seja, qual é o canal que utilizará (por telefone, e-mail, pessoalmente). Além disso, ele deve verificar qual a importância ou a urgência da mensagem, definido, assim, se será realizada de maneira síncrona (direta) ou assíncrona (indireta, sem que ambos estejam conectados no mesmo tempo ou espaço). O participante da equipe é emissor e, ao mesmo tempo, receptor, sendo que precisa ter compromisso em estabelecer comunicação.
- **Cooperação:** acontece quando a equipe possui ambiente compartilhado, mesmo que seja virtual. Dessa maneira, todos os participantes podem trabalhar juntos, podendo esclarecer suas dúvidas, aprender com o colega, ou até dividir recursos necessários para que as tarefas possam ser completadas rapidamente e assim a equipe concluir os objetivos necessários para finalização do trabalho.
- **Coordenação:** para que um objetivo em comum seja alcançado, é essencial que o trabalho seja dividido em atividades menores, assim a equipe pode alcançar o objetivo mais rapidamente, já que seus membros trabalharão ao mesmo tempo em atividades diferentes. Para que o trabalho seja coordenado, é elaborado um plano de trabalho, no qual encontra-se a definição e o controle de prazos, políticas (internas e externas), tarefas, papéis (cargos) dos participantes. Além disso, o plano apresenta tanto o requerimento, quanto a alocação de recursos.

As ontologias acima são úteis para quem utiliza, como forma de trabalho, a colaboração e o modelo 3C. Esse tipo de modelo de trabalho melhora a participação dos membros da

equipe, facilita o compartilhamento das informações e do conhecimento de forma coordenada e organizada (FUKS *et al.*, 2012).

São exemplos de ferramentas colaborativas conhecidas: o GitHub, o Trello, o Google Drive e o ShareLatex que permitem aos usuários realizarem a troca de informações e/ou trabalhar de forma simultânea em um mesmo arquivo.

Os sistemas colaborativos são o assunto principal desta pesquisa, pois serão analisados estudos que trazem como propostas as ferramentas colaborativas, as quais favorecem a comunicação, a colaboração e a coordenação das equipes que trabalham com a criação de software.

4.2 Processo de desenvolvimento de software

Atualmente, encontram-se milhares de softwares que são utilizados para todos os tipos de funções. Eles têm como principal objetivo facilitar a vida das pessoas, realizando atividades que proporcionem conforto e que minimizem o esforço humano. Contudo, muitos softwares que são produzidos não atendem as necessidades dos clientes ou apresentam erros devido a complexidade do seu desenvolvimento. Sendo assim, a fim de reduzir problemas, faz-se necessário a utilização de técnicas e etapas bem definidas para sua construção. (KOSCIANSKI; SOARES, 2006).

Para que os softwares sejam produzidos, a equipe envolvida em seu desenvolvimento precisa seguir algumas etapas, atividades e ações, sendo que essas etapas podem ser simultâneas ou não. O processo de desenvolvimento de software envolve, tradicionalmente, o levantamento de requisitos, o projeto da arquitetura, a implementação, o teste e a manutenção (SOMMERVILLE, 2011). Para muitos gerentes de projeto, que trabalham com o desenvolvimento ágil, essas atividades são realizadas repetidas vezes, sendo que, a cada interação, o software é incrementado até que esteja completo (PRESSMAN, 1995).

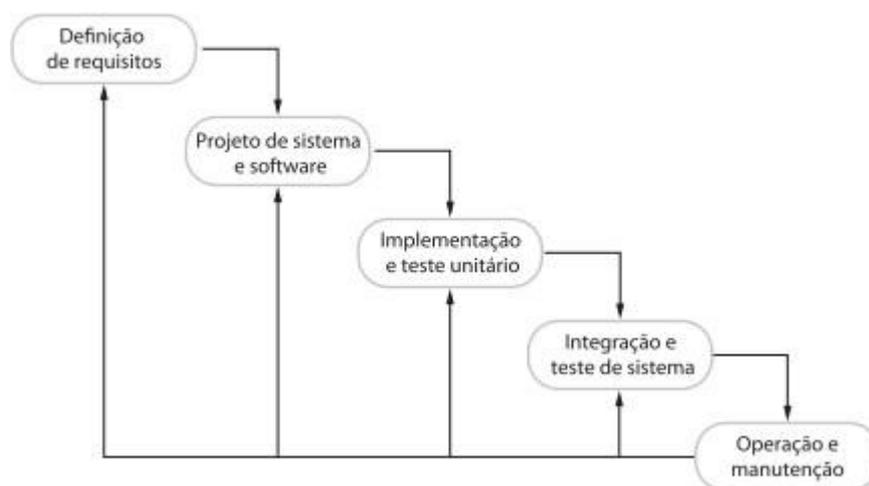
A Figura 1 apresenta o primeiro modelo do processo de desenvolvimento de software publicado, que contém atividades presentes em todos os modelos de processos de software. As principais etapas desse modelo são (SOMMERVILLE; ARAKAKI; MELNIKOFF, 2008):

- **Definição de requisitos:** nesta fase do desenvolvimento, também conhecida como engenharia de requisitos, são elicitados os requisitos a fim de que a equipe possa elaborar as funcionalidades, as restrições e os objetivos do sistema. As especificações do sistema são

definidas em detalhes, mostrando o funcionamento de cada tarefa. Ao final, os requisitos são validados com o cliente.

- **Projeto de sistema e software:** este processo, também denominado arquitetura de software, *design* ou projeto, envolve a distribuição dos recursos necessários tanto para o hardware como o software. Sendo assim, é definida uma arquitetura geral do sistema. Nesta fase, são descritas todas as abstrações essenciais do sistema de software e as relações entre elas.
- **Implementação e teste unitário:** na fase de implementação, é desenvolvido um conjunto de funcionalidades que compõem o software. Já na fase do teste unitário, é realizada a verificação de cada funcionalidade, para que se atenda os requisitos especificados.
- **Integração e teste de sistema.** O conjunto de funcionalidades é integrado ao sistema. Também são realizados testes para verificar se o sistema está de acordo com os requisitos especificados.
- **Operação e manutenção:** nesta fase, o sistema é entregue ao usuário final ou cliente e instalado. Nesse momento, podem ser realizadas correções no sistema, ampliação de serviços e melhoramento da implementação.

Figura 1 – Atividades do processo de desenvolvimento de software no modelo cascata .



Fonte: (SOMMERVILLE, 2011).

Nesta pesquisa serão analisadas ferramentas colaborativas usadas no processo de desenvolvimento do software apresentado acima seguindo a revisão sistemática da literatura como método de pesquisa.

4.3 Revisão sistemática da literatura

Uma revisão sistemática é um método de pesquisa empírico (KITCHENHAM; CHARTERS, 2007) que utiliza como fonte de dados a literatura sobre determinado tema (RF, 2007).

Existem vários motivos para realizar uma revisão sistemática. Os mais comuns são: resumir uma evidência existente sobre uma tecnologia ou um tratamento; identificar lacunas existentes com a finalidade de sugerir uma agenda de pesquisa; proporcionar uma estrutura para posicionar adequadamente uma pesquisa. Esse método também pode ser utilizado com o objetivo de investigação para contradizer uma hipótese ou criar novas teorias. Além disso, possui uma metodologia bem definida, que diminui a possibilidade de escrever pesquisas tendenciosas e que possibilita o acesso a uma grande quantidade de informações (KITCHENHAM; CHARTERS, 2007).

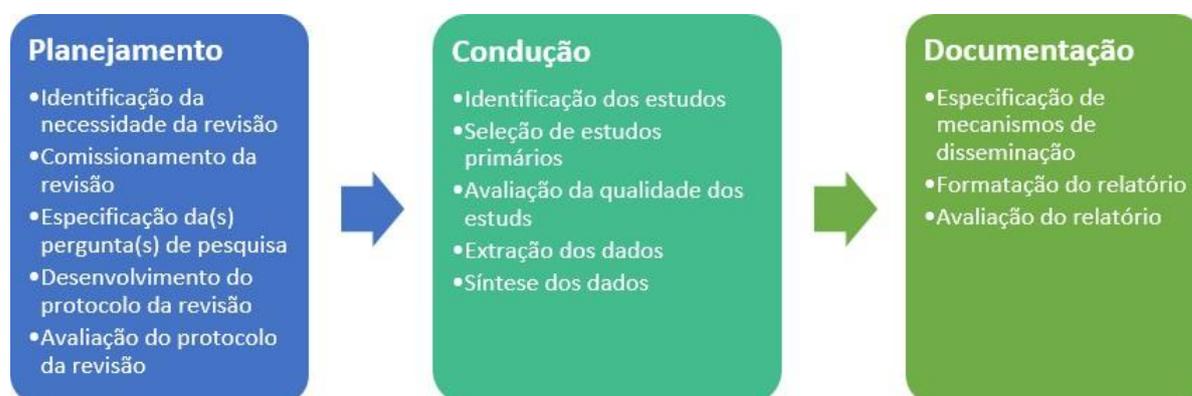
Kitchenham e Charters no artigo *Guidelines for performing Systematic Literature Reviews in Software Engineering* (KITCHENHAM; CHARTERS, 2007) definem as *guidelines* ou passos para realizar uma revisão sistemática. Essas *guidelines* são divididas em três fases principais: Planejando a revisão; Conduzindo a revisão; e Reportando a revisão, (KITCHENHAM; CHARTERS, 2007) conforme ilustrado na Figura 2.

Na Condução da revisão, as atividades planejadas anteriormente são executadas de acordo com o protocolo definido. Na etapa de Documentação são apresentados e divulgados os resultados obtidos na revisão.

A etapa de Planejamento, ilustrada na Figura 2, é responsável por definir as atividades necessárias para a realização da revisão resultando na definição do protocolo da revisão sistemática. O protocolo descreve as perguntas de pesquisa, palavras-chave, base de dados, critérios de inclusão e exclusão, entre outros aspectos.

A revisão sistemática da literatura é importante para esta pesquisa, pois ela será utilizada como principal método de pesquisa para a obtenção dos dados extração de dados e definição do catálogo.

Figura 2 – Passos para realização de uma revisão sistemática.



Fonte: Elaborada pela autora adaptada de (KITCHENHAM; CHARTERS, 2007).

No próximo capítulo é descrita a metodologia de pesquisa adotada nesse trabalho.

5 PROCEDIMENTOS METODOLÓGICOS

Nesta seção, apresenta-se as ações realizadas para o desenvolvimento desta pesquisa e, posteriormente, a criação de um catálogo sobre ferramentas colaborativas no desenvolvimento de software.

- **Estudo do Domínio:** realizou-se o estudo dos conceitos necessários para o desenvolvimento de um catálogo sobre ferramentas colaborativas no desenvolvimento de software. Portanto, a base de pesquisa foram livros e artigos obtidos no *Google* acadêmico, para encontrar artigos relacionados aos seguintes temas: sistemas colaborativos, processos de desenvolvimento de software e revisão sistemática da literatura. Além disso, a pesquisa utilizou-se de artigos publicados em bases de dados digitais como Elsevier, Springer, IEEE e ACM.
- **Elaboração do protocolo da revisão sistemática:** o protocolo, que define as questões de pesquisa, bases de dados, estratégias de busca, de seleção de estudos e de extração de dados, foi desenvolvido seguindo as *guidelines* de (KITCHENHAM; CHARTERS, 2007), explicadas na seção 4.3.
- **Condução da revisão sistemática:** o protocolo definido anteriormente foi seguido para obter os artigos a serem avaliados e analisados e, a partir deles, extrair

os dados. Como ferramenta de apoio, foi utilizada a ferramenta Parsifal⁹. Essa ferramenta suporta a realização de revisões sistemáticas, desde a elaboração do protocolo até a consolidação dos resultados.

- **Construção do catálogo:** a partir dos dados obtidos por meio da revisão sistemática, foi criada várias tabelas que trazem várias ferramentas de acordo com as perguntas do protocolo.

No próximo capítulo é descrito o protocolo da revisão sistemática da literatura utilizado nesse trabalho.

6 PROTOCOLO DA REVISÃO SISTEMÁTICA

A definição do protocolo a ser utilizado nessa pesquisa seguiu as *guidelines* necessárias de (KITCHENHAM; CHARTERS, 2007). O protocolo, que foi definido utilizando a ferramenta *Parsifal*, é detalhado nas próximas seções.

6.1 Identificação da necessidade da revisão

Antes de iniciar uma revisão sistemática, é imprescindível verificar se ela é necessária e, por conseguinte, precisa-se analisar revisões existentes na área de interesse.

Na seção 1, discutiu-se a inexistência de revisões sistemáticas nessa área de ferramentas colaborativas no desenvolvimento de software. Sendo assim, trabalhos relacionados foram comparados e constatou-se a necessidade de realizar a revisão proposta nesse trabalho.

6.2 Comissionamento da revisão

A fase de comissionamento de uma revisão sistemática não é necessária para esta pesquisa uma vez que será delegada para terceiros. A revisão será realizada pela própria aluna visando atender a necessidade da construção dos catálogos.

Como essa fase de comissionamento não será realizada, a estratégia de disseminação dos resultados foi incorporada ao protocolo de revisão. De acordo com (KITCHENHAM; CHARTERS, 2007), até agora, não há exemplos de revisões sistemáticas comissionadas no domínio de engenharia de software.

⁹ <https://parsif.al/>

6.3 Perguntas de Pesquisa

Segundo (KITCHENHAM; CHARTERS, 2007), a definição das perguntas de pesquisa é a etapa mais importante, pois são responsáveis por direcionar a metodologia a ser adotada.

O critério PICOC (População, Intervenção, Comparação, Resultado e Contexto), do inglês *Population, Intervention, Comparison, Outcome, Context*, foi utilizado para direcionar a definição das perguntas de pesquisa.

- **População:** Desenvolvedores de software.
- **Intervenção:** Coletar evidências empíricas em relação às ferramentas colaborativas no desenvolvimento do software.
- **Comparação:** Não se aplica. As ferramentas não serão comparadas.
- **Resultados:** Respostas para as perguntas de pesquisa.
- **Contexto:** Engenharia de Software desde que *stakeholders* utilizem ferramentas colaborativas para desenvolver software.

As questões a serem respondidas nessa revisão são:

- 1.P1. Quais são as ferramentas colaborativas existentes?
- 2.P2. Quais são as funcionalidades de um sistema colaborativo descritas no trabalho?
3. P3. As ferramentas são utilizadas em qual tipo de modelo de processo de desenvolvimento de software (ex: ágil, não-ágil, não-classificado)?
- 4.P4. Qual a licença da ferramenta?
5. P5. Em que fase do processo de desenvolvimento do software as ferramentas são utilizadas (ex: requisitos, arquitetura, teste)?

6.4 Palavras chaves utilizadas

As principais palavras chaves utilizadas no desenvolvimento da *string* de busca foram:

- Ferramentas colaborativas: *collaboration tools, collaborative development environment, collaborative system, computer-supported cooperative work (cscw)*
- Engenharia de Software: *distributed software development, global software development, global software engineering, software engineering*

6.5 String de busca

A *string* de busca foi especificada considerando as palavras chaves definidas. Foram realizadas várias buscas para refinar a *string* de maneira interativa. Foram excluídas palavras-chave cuja inclusão não retornou artigos adicionais nas pesquisas automáticas. Depois de várias iterações, a seguinte *string* de pesquisa foi definida para pesquisar nas palavras-chave, título, *abstract* e texto dos artigos:

(“collaboration tool” OR “collaborative tool” OR “communication tool” OR “cooperative tool” OR “coordination tool”) AND (“distributed software development” OR “global software development” OR “collaborative software development” OR “global software engineering” OR “collaborative software engineering”) AND (“problem” OR “issue” OR “challenge”)

Para aumentar a credibilidade da *string* foram selecionados 10 artigos que abordavam a temática pesquisada e os mesmos foram retornados pela *string*. São eles: *Tools used in Global Software Engineering: A Systematic Mapping Review* (PORTILLO-RODRÍGUEZ *et al.*, 2012); *Continuous Coordination Tools and their Evaluation* (SARMA *et al.*, 2010); *A Collaborative Virtual Workspace for Software Development* (SILVA *et al.*, 2015); *Collaboration in Software Engineering: A Roadmap* (WHITEHEAD, 2007); *Collaborative Software Engineering: Concepts and Techniques* (WHITEHEAD *et al.*, 2010); *Collaboration in Distributed software Development* (LANUBILE, 2006); *Supporting Collaborative Software Development Over GitHub* (ARORA; GOEL; MITTAL, 2017b); *Collaboration tools for global software engineering* (LANUBILE *et al.*, 2010); *Supporting Collaboration in the Geographically Distributed Work with Communication Tools in the Remote District SME's* (LIUKKUNEN *et al.*, 2010); *Collaborative software engineering: challenges and prospects* (MISTRÍK *et al.*, 2010)

6.6 Bases de dados

As bases de dados que foram utilizadas para a seleção dos artigos são descritas na Tabela 2.

Tabela 2 – Bases utilizadas na pesquisa.

Nome	URL
ACM Digital Library	http://portal.acm.org
IEEE Digital Library	http://ieeexplore.ieee.org
Science@Direct	http://www.sciencedirect.com
Springer Link	http://link.springer.com

Fonte: Elaborada pela autora.

6.7 Critérios de inclusão e exclusão

A definição de critérios de inclusão e exclusão é importante para conferir uma maior credibilidade à pesquisa, haja visto que não permite que a análise siga as expectativas do pesquisador. Os critérios adotados nesse trabalho são descritos a seguir tendo sempre como apoio a ferramenta online *Parsifal*.

Critérios de inclusão:

- Artigos que apresentem propostas de ferramentas colaborativas em algum processo de desenvolvimento de software;
- Artigos primários;
- Artigos publicados até julho de 2018;
- Artigos que citem ferramentas colaborativas usadas no processo de desenvolvimento de software;
- Artigos que respondam alguma pergunta de pesquisa.

Critérios de Exclusão:

- Artigos duplicados;
- Artigos menores que quatro páginas (short papers);
- Artigos que não são escritos em inglês;
- Artigos secundários;
- Literatura cinza (teses, dissertações, monografias, etc);
- Artigos cujo texto completo não sejam acessíveis;
- Artigos redundantes do mesmo autor;

- Estudos que não são relevantes à pesquisa.

6.8 Procedimento para seleção de estudos

O procedimento de seleção de estudos consistiu em cinco etapas principais. Na primeira, os estudos foram consultados e obtidos por meio de busca automática usando a *string* de pesquisa nas bases de dados apresentadas na Tabela 2. Os resultados das buscas foram armazenados na ferramenta Parsifal.

A segunda etapa consistiu na eliminação de artigos duplicados. Em seguida, no passo 3, selecionou-se os estudos primários obtidos na etapa anterior por meio da leitura de título e *abstract* e os critérios de inclusão e exclusão descritos na Seção 6.7. Caso houvesse dados insuficientes ou dúvidas, o artigo seguiu para o próximo passo.

O quarto passo consistiu na leitura completa dos artigos para responder as perguntas de pesquisa descritas na Seção 6.3.

6.9 Ameaças à validade

Nesta subsecção, apresentam-se as categorias de ameaça à validade, conceituadas por Wohlin *et al.* (2000), que inclui quatro tipos de validades: a de construção, a externa, a interna e a de conclusão.

Validade de constructo: esta ameaça está relacionada à generalização do resultado obtido no estudo para o conceito ou a teoria envolvida (WOHLIN *et al.*, 2000). Minimizou-se esta ameaça usando palavras e expressões sinônimas para construção da *String* de busca, por exemplo: “*collaboration tools*”, “*collaborative development environment*”, “*collaborative system, computer-supported cooperative work (cscw)*”.

Validade Interna: neste caso, as ameaças são causadas por possíveis erros de conclusão entre os tratamentos e os resultados (WOHLIN *et al.*, 2000). O principal objetivo de uma revisão sistemática é minimizar as ameaças à validade interna. Decisões subjetivas podem ocorrer durante a seleção de artigos e extração de dados uma vez que é comum estudos primários não fornecerem uma descrição clara ou objetivos e resultados apropriados, dificultando a aplicação objetiva dos critérios de inclusão/exclusão ou a imparcialidade. A fim de minimizar erros de seleção e extração, o processo de seleção foi realizado de forma iterativa de forma que quando ocorreu dúvida na aplicação de algum critério, o estudo não foi eliminado e passou para a próxima fase. Além disso, o processo de seleção foi realizado de forma colaborativa pela aluna

e por um colaborador de forma que conflitos fossem discutidos e solucionados pelos alunos em conjunto com a orientadora. Dessa forma, objetivou-se atenuar as ameaças devido ao viés pessoal na compreensão do estudo.

Validade externa: esta preocupa-se em estabelecer a generalização dos resultados da revisão, relacionando-a ao grau em que os estudos primários são representativos para o tema de revisão. No caso de uma revisão da literatura, a validade externa depende da literatura identificada; caso a literatura identificada não seja válida externamente, tampouco será a síntese do seu conteúdo (GASPARIC; JANES, 2016). Esta ameaça foi mitigada devido a utilização do critério de exclusão para eliminar, da pesquisa, os estudos provenientes de literatura cinza. Além disso, para mitigar as ameaças externas, o protocolo de pesquisa foi definido iterativamente e, validado, com o consenso da autora, do colaborador, da orientadora e os professores que participaram da banca do projeto de pesquisa.

Validade de conclusão: segundo Kitchenham e Charters (2007), nem todos os estudos primários que existem relacionados à pesquisa podem ser identificados, sendo assim, para minimizar esse ameaça, o processo da revisão foi cuidadosamente elaborado e discutido pelos autores para minimizar o risco de exclusão de estudos relevantes. Outro método empregado foi utilizar expressões e palavras sinônimas para os constructos dessa revisão sistemática, essa técnica objetiva uma maior cobertura de estudos possivelmente importantes a partir da pesquisa automática. Além disso, o processo de seleção do estudo foi conduzido em paralelo e de forma independente pela aluna e pelo colaborador. Posteriormente, os resultados foram harmonizados para mitigar o viés pessoal na seleção do estudo causado por revisores individuais. Finalmente, a orientadora supervisionou esse processo.

Na próxima seção, são apresentados os resultados da revisão sistemática.

7. RESULTADOS

As buscas realizadas nas bases de dados da Tabela 2 usando a *string* de busca retornou 648 artigos no total. Após a importação desses artigos na ferramenta *Parfisal* realizou-se a seleção deles com a participação de 3 pessoas, autora, orientadora e um colaborador.

A distribuição desses artigos e as bases correspondentes encontram-se na Tabela 3 juntamente com os artigos selecionados na segunda fase e com o percentual de concordância entre os participantes.

A ferramenta *Parsifal* utilizando-se dos resultados retornados pela *String* de busca,

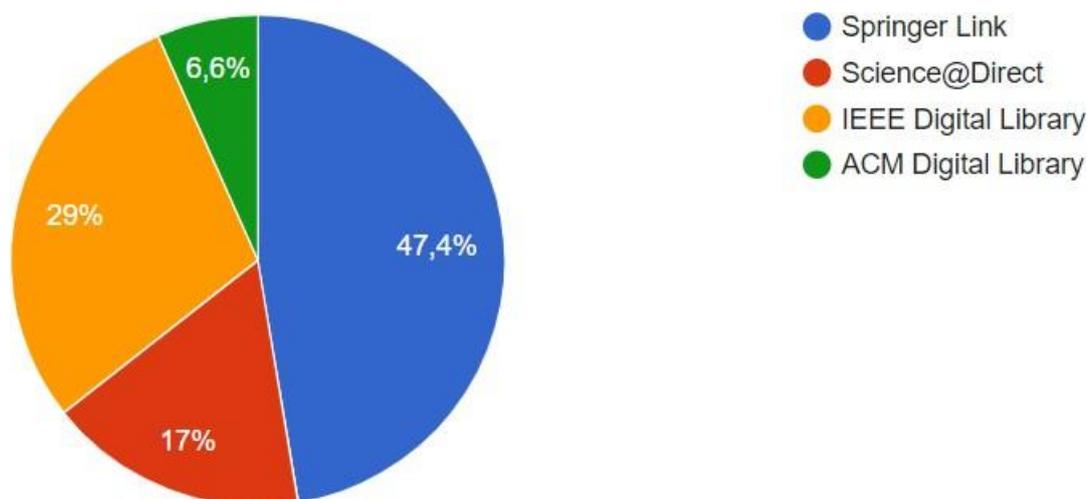
Tabela 3 – Resultado da seleção dos artigos.

Base de dados	Seleção automática	Seleção 2 fase	Concordância
ACM Digital Library	43	17	82,6%
IEEE Digital Library	188	133	90%
Science Direct	110	81	68%
Springer Link	307	170	71%

Fonte: Elaborada pela autora.

fez uma análise de percentual de artigos selecionados por base de pesquisa. A Figura 3 mostra que a base *Springer* teve um maior número de artigos selecionados e ACM o menor.

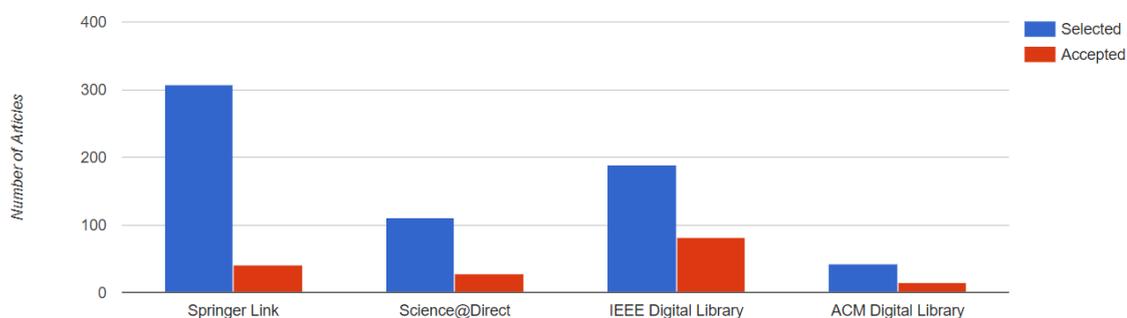
Figura 3 – Porcentagem de dados retornados por base de dados.



Fonte: ferramenta *Parsifal*

Essa mesma ferramenta também analisou o percentual sobre a quantidade de artigos selecionados automaticamente *versus* a quantidade de artigos aceitos após o processo de seleção. Dessa forma, percebe-se a base de dados *Springer Link* possui um maior número de artigos rejeitados.

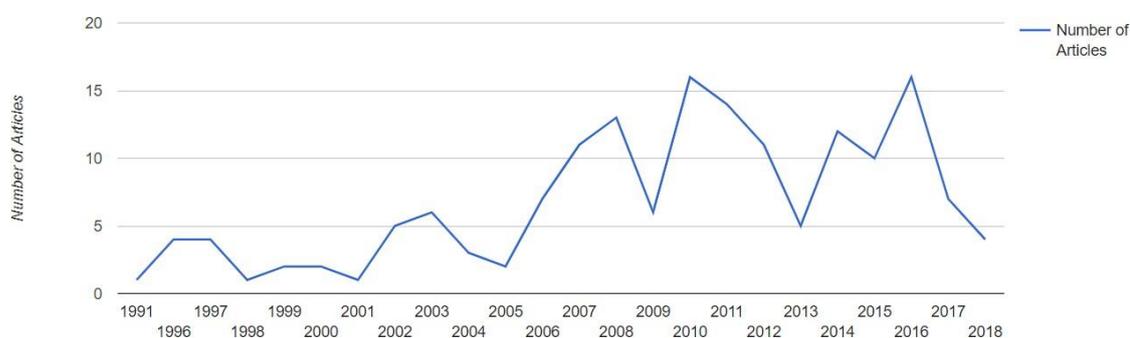
Figura 4 – Porcentagem dos artigos selecionados em relação a quantidade de artigos retornados.



Fonte: ferramenta *Parsifal*

O último gráfico apresentado pela ferramenta *Parsifal* traz o número de publicações de artigos relacionados as ferramentas colaborativas por ano. Observa-se, que os artigos selecionados possuem como ano marco 1991, sendo que os picos de maiores publicações foram os anos de 2011 e 2017. Provavelmente, deve-se esse pico de artigos relacionados a sistemas colaborativos, pois em 2011 aconteceram muitos avanços na tecnologia móvel, com criação de aplicativos que possibilitassem uma maior comunicação entre pessoas geograficamente distribuídas.

Figura 5 – Distribuição dos artigos selecionados ao longo do tempo.



Fonte: ferramenta *Parsifal*

Nas próximas seções, apresentam-se as respostas para as perguntas de pesquisa.

7.1 P1. Quais são as ferramentas colaborativas existentes?

A pesquisa realizada por meio da Revisão Sistemática de Literatura (KITCHENHAM; CHARTERS, 2007) nas bases citadas na Tabela 2 aponta diferentes Ferramentas, Sistemas e *Frameworks* colaborativos relacionados ao desenvolvimento de *software*. As ferramentas são apresentadas nas Tabelas: 4 (ACM), 5 (IEEE - Parte 1), 5 (IEEE - Parte 2), 7 (Science Direct) e 8 (Springer).

Tabela 4 – Ferramentas Colaborativas retornadas nos artigos da base ACM.

Autores	Ferramentas
(CABALLÉ <i>et al.</i> , 2014)	PFC Mobile
(SARMA; REDMILES; HOEK, 2010)	1. <i>Tesseract</i> 2. <i>SeeSoft</i> 3. <i>Jazz</i> 4. <i>Hackstat</i> 5. <i>Oasis</i> 6. <i>Stepin</i> 7. <i>EEL</i> 8. <i>Palantr</i> 9. <i>Chianti</i> 10. <i>Mylyn</i> 11. <i>CollabVS</i> 12. <i>CodeCity</i> 13. <i>TeamTracks</i> 14. <i>Creole</i>
(MEYER, 2008)	1. <i>Collaborator</i> 2. <i>X</i> – <i>Lite</i> 3. <i>EiffelStudio</i> 4. <i>WebEx</i>
(SINHA; SENGUPTA; CHANDRA, 2006)	1. <i>Egret</i> 2. <i>CVS</i> 3. <i>Collabnet</i> 4. <i>IBMRationalRequisitePro</i> 5. <i>Jazz</i> 6. <i>Sangam</i> 7. <i>Stellation</i> 8. <i>TelelogicDOORS</i>
(DEAN <i>et al.</i> , 1997)	1. <i>PSL/PSA</i> 2. <i>PLEXSYS</i> 3. <i>GroupSystems</i> 4. <i>GEMS</i> 5. <i>GroupDataModeler</i>
(TIWANA, 2008)	1. <i>REMAP</i>
(PAASIVAARA <i>et al.</i> , 2015)	1. <i>Flowdock</i> 2. <i>GitHub</i> 3. <i>AgileFant</i>
(SUZUKI; YAMAMOTO, 1999)	1. <i>SoftDock</i>
(FINHOLT <i>et al.</i> , 1999)	1. <i>NetMeeting</i>
(DAMIAN; LANUBILE; MALLARDO, 2008)	1. <i>IBIS</i>
(GARCIA; VIZCAINO; EBERT, 2011)	1. <i>EclipseProcessFrameworkComposer</i> 2. <i>OpenUnifiedProcess</i> 3. <i>RationalMethodComposer</i> 4. <i>AppianEnterprise</i> 6. <i>BizAgi</i> 7. <i>Intalio</i>
(HOSSAIN; BANNERMAN; JEFFERY, 2011)	1. <i>Team Foundation Server</i> 3. <i>Jira</i> 4. <i>ScrumWorks</i>
(GOTEL <i>et al.</i> , 2008)	1. <i>Smart Draw UML</i> 2. <i>java.net</i> 3. <i>Subversion</i>

Fonte: Elaborada pela autora.

Tabela 5 – Ferramentas Colaborativas retornadas nos artigos da Base IEEE (Parte 1).

Autores	Ferramentas
(SILVA <i>et al.</i> , 2015)	1. <i>TagSEA</i> 2. <i>Hackystat</i>
(GRINSVEN; VREEDE, 2003)	1. <i>GroupSystems</i>
(GOTEL <i>et al.</i> , 2009)	1. <i>CVS</i> 2. <i>java.net</i>
(PAPADAKI; POLEMI; DAMILOS, 2008)	1. <i>ISRM</i>
(CALEFATO; LANUBILE, 2016)	1. <i>Hipikat</i> 2. <i>Mylyn</i> 3. <i>Palantr</i> 4. <i>GitLab</i> 5. <i>HP ALM</i> 6. <i>Jenkins</i> 7. <i>TeamCity</i> 8. <i>Ansible</i> 9. <i>Maven</i> 10. <i>Jasmine</i> 11. <i>Karma</i> 12. <i>PhantomJS</i> 13. <i>Protractor</i> 14. <i>Selenium</i> 15. <i>Cucumber</i> 16. <i>Gatling</i> 17. <i>JU nit</i> 18. <i>JMeter</i> 19. <i>RestAssured</i> 20. <i>Capybara</i> 21. <i>Watir</i> 22. <i>Jira</i> 23. <i>Artifactory</i> 24. <i>Bower</i> 25. <i>Grunt</i>
(CAMPAGNOLO <i>et al.</i> , 2009)	1. <i>CAISE</i> 2. <i>CASDE</i> 3. <i>TagSEA</i> 4. <i>Palantr</i> 5. <i>CVS</i> 6. <i>COLLAB</i> 7. <i>LotusNotes</i>
(TREUDE; STOREY, 2012)	1. <i>Jazz</i> 2. <i>Bugzilla</i> 3. <i>TagSEA</i>
(BABAR <i>et al.</i> , 2004)	1. <i>Zwiki</i> 2. <i>InspectAnyWhere</i>
(COOK; CHURCHER, 2003)	1. <i>CVS</i> 2. <i>GroupCRC</i> 3. <i>Rosetta</i> 4. <i>SourceForge</i> 5. <i>CAISE</i>
(HILDENBRAND <i>et al.</i> , 2008)	1. <i>ARENA</i> 2. <i>SourceForge</i> 3. <i>ConversationBuilder</i> 4. <i>GENESIS</i> 5. <i>JAZZ</i> 6. <i>MILOS</i>
(BECKER, 1998)	1. <i>Cool Talk</i>
Autores	Ferramentas
(YILDIZ; TEKINERDOGAN, 2011)	1. <i>SoftFab</i> 2. <i>Jazz</i>
(ERRA; SCANNIELLO, 2010)	1. <i>CoFFEE</i> 1. <i>WinWin</i>
(GAMBLE; HALE, 2013)	1. <i>Jazz</i> 2. <i>SEREBRO</i> 3. <i>Subversion</i>
(ERRA; SCANNIELLO, 2011)	1. <i>CoFFEE</i>
(WU; GRAHAM; SMITH, 2003)	1. <i>Rose</i> 2. <i>LotusNotes</i>
(MONASOR; VIZCAINO; PIATTINI, 2010)	1. <i>CURE</i> 2. <i>iBistro</i> 3. <i>Jazz</i> 4. <i>OASIS</i> 5. <i>VECTOR</i>
(LEE <i>et al.</i> , 2000)	1. <i>ColD SPA</i> 2. <i>GroupSystems</i>
(KROPP; MEIER, 2016)	1. <i>Maven</i> 2. <i>Jenkins</i> 3. <i>GitHub</i> 4. <i>Trello</i> 5. <i>Jira</i>
(KUSUMASARI <i>et al.</i> , 2011)	1. <i>PBwork</i> 2. <i>mercurial</i> 3. <i>Mocking bird</i>
(SHROFF <i>et al.</i> , 2005)	1. <i>Subversion</i> 2. <i>Collabnet</i> 3. <i>IBM Quick place</i> 4. <i>Lotus Team Collaboration Suite</i> 5. <i>Microsoft Sharepoint Portal Server</i> 6. <i>SourceForge</i>
(BASHERI; BURD; BAGHAELI, 2012)	1. <i>DigiTile</i> 2. <i>WebSurface</i> 3. <i>CAMELA.MT –CollabUML</i>
(DEFRANCO-TOMMARELLO; DEEK, 2002)	1. <i>Groove</i> 2. <i>LotusNotes</i> 3. <i>GroupSystems</i> 4. <i>Rose</i> 5. <i>RequisitePro</i> 6. <i>WebMet</i> 7. <i>EvolvingArtifactory</i> 8. <i>CoNeX</i> 9. <i>Web –CCAT</i> 10. <i>SOLV EIT</i>
(DEFRANCO-TOMMARELLO <i>et al.</i> , 2003)	1. <i>Groove</i>
(WALTER; RILEY; GAMBLE, 2016)	1. <i>Drupal</i> 2. <i>Lucidchart</i> 3. <i>Taiga.io</i> 4. <i>Redmine</i> 5. <i>Trac</i> 6. <i>ApacheAllura</i> 7. <i>TACTIC</i> 8. <i>Endeavour</i> 9. <i>FusionForge</i> 10. <i>MantisBT</i> 11. <i>OpenProject</i> 12. <i>ProjectQtOr</i> 13. <i>Plandora</i> 14. <i>Collabtive</i> 15. <i>LibrePlan</i> 16. <i>Fossil</i> 17. <i>TheBugGenie</i> 18. <i>LucidChart</i> 19. <i>draw.io</i> 20. <i>GanttProject</i>
(LEI <i>et al.</i> , 2004)	1. <i>LotusSameTime</i> 2. <i>BuildTopia</i> 3. <i>OneSpace</i> 4. <i>MILOS</i> 5. <i>DevX</i> 6. <i>collab.net</i>
(ANTUNES; FERREIRA, 2011)	1. <i>ABTool</i> 2. <i>eyeView</i>
(XU <i>et al.</i> , 2009)	1. <i>Libra on Chat</i> 2. <i>D U M L</i>
(WATABE <i>et al.</i> , 1991)	1. <i>MERMAID</i>
(CASTRO-HERNÁNDEZ; SWIGGER; PONCE-FLORES, 2014)	1. <i>Redmine</i>
(MOE <i>et al.</i> , 2016)	1. <i>DNVGL</i>
(BARTHOLOMEW, 2008)	1. <i>MPK20</i>
(BABAR; WINKLER; BIFFL, 2007)	<i>EasyWinWin</i>
(CALEFATO; LANUBILE; SCALAS, 2007)	<i>eConference</i>
(CARLSON; XIAO, 2012)	1. <i>Git2.Assembla</i>
(NIINIMÄKI; LASSENIUS, 2008)	1. <i>LotusNotesSameTime</i>
(SUNINDYO <i>et al.</i> , 2010)	1. <i>MULE</i> 2. <i>Continuum</i> 3. <i>Hudson</i> 4. <i>Maven</i> 5. <i>Ant</i>
(MONASOR <i>et al.</i> , 2014)	1. <i>VENTURE</i>
(SIEVI-KORTE; SYSTÄ; HJELSVOLD, 2015)	1. <i>Atlassian Confluence</i> 2. <i>Atlassian Jira</i>

Fonte: Elaborada pela autora.

Tabela 6 – Ferramentas Colaborativas retornadas nos artigos da base IEEE (Parte 2).

Autores	Ferramentas
(LEPPÄNEN; LAHTINEN; IHANTOLA, 2016)	1. <i>Cockburn</i> 2. <i>Agile fant</i> 3. <i>IRC</i> 4. <i>Trello</i> 5. <i>GitLab</i>
(TREUDE; STOREY, 2009)	1. <i>Jazz</i> 2. <i>TagSEA</i>
(TAKAHASHI <i>et al.</i> , 1996)	1. <i>EColabor</i> 2. <i>Cogent</i> 3. <i>MIME</i> 4. <i>PEARnet</i> 5. <i>SIBYL</i>
(GHORASHI; JENSEN, 2017)	1. <i>Jimbo</i> 2. <i>Collabode</i> 3. <i>FASTDash</i> 4. <i>Palantr</i> 5. <i>IRC</i> 6. <i>Git</i> 7. <i>SVN</i>
(PESOLA <i>et al.</i> , 2011)	1. <i>Jazz</i> 2. <i>Jenkins</i> 3. <i>Subversion</i> 4. <i>OpenMeetings</i> 5. <i>T RAC</i> 6. <i>Git</i> 7. <i>Mercurial</i>
(BAN <i>et al.</i> , 2017)	1. <i>GitHub</i>
(RAS, 2009)	1. <i>Trac</i> 2. <i>SnipSnap</i> 3. <i>WikiDoc</i> 4. <i>TeamWeaverWiki</i> 5. <i>Moodle Wiki</i> 6. <i>Jotspot Wiki</i> 7. <i>SVN</i> 8. <i>TikiWiki</i>
(PALACIO <i>et al.</i> , 2011a)	1. <i>CWSM</i> 2. <i>Palantir</i>
(GERVIGNY; NAGOWAH, 2017)	1. <i>MASE</i>
(SU; SCHARFF, 2010)	1. <i>Rational Team Concert</i>
(CANFORA; CIMITILE; VISAGGIO, 2003)	1. <i>NetMeeting</i>
(KÖBLER <i>et al.</i> , 2010)	1. <i>LocaTag</i> 2. <i>Effusia</i> 3. <i>LotusSametime</i>
(CAJANDER <i>et al.</i> , 2012)	1. <i>SVN Trello</i>
(CASTRO-HERNÁNDEZ <i>et al.</i> , 2016)	1. <i>Redmine</i>
(MAK; KRUCHTEN, 2007)	1. <i>NextMove</i>
(BANI-SALAMEH; JEFFERY, 2014)	1. <i>GateKeeper</i> 2. <i>Jazz</i> 3. <i>BSCW</i> 4. <i>SCI</i>
(WEERD <i>et al.</i> , 2006)	1. <i>Software Product Management Workbench</i>
(KALLIAMVAKOU <i>et al.</i> , 2015)	1. <i>GitHub</i> 2. <i>SVN</i> 3. <i>Jira</i> 4. <i>Asana</i> 5. <i>PivotalTracker</i>
(SILVA <i>et al.</i> , 2013)	1. <i>OPERAM</i> 2. <i>TagSEA</i>
(DULLEMOND; GAMEREN; SOLINGEN, 2011)	1. <i>Communico</i> 2. <i>IRC</i> 3. <i>GroupBanter</i> 4. <i>Babble</i> 5. <i>Loops</i> 6. <i>ReachOut</i> 7. <i>T hreadedChat</i> 8. <i>OpenMessenger</i>
(WU, 2012)	1. <i>RearViewMirror</i> 2. <i>SharePoint</i> 3. <i>Groove</i> 4. <i>GroupSystems</i> 5. <i>SourceForge</i> 6. <i>SourceCast</i> 7. <i>RationalSuiteEnter prise</i>
(CASEY; RICHARDSON, 2006)	1. <i>NetMeeting</i>
(NIEMINEN, 2012)	1. <i>CoRED</i> 2. <i>Git</i> 3. <i>Doodle</i>
(CHUNG <i>et al.</i> , 2010)	1. <i>ConversationBuilder</i> 2. <i>GroupKit</i>
(GRUNDY; HOSKING; MUGRIDGE, 1996)	1. <i>Dimdim</i>
(ARORA; GOEL; MITTAL, 2017a)	1. <i>GitHub</i> 2. <i>Subversion</i> 3. <i>Perforce</i> 4. <i>Saros</i> 5. <i>GCodeEdit</i> 6. <i>Collabode</i> 7. <i>COLLECE</i> 8. <i>Palantr</i> 9. <i>Syde</i> 10. <i>WeCode</i> 11. <i>T MED-COG</i>
(BANDINELLI; NITTO; FUGGETTA, 1996)	1. <i>Oval</i> 2. <i>gIBIS</i> 3. <i>Regatta</i> 4. <i>GroupKit</i> 5. <i>ClearBoard</i> 6. <i>SPADE</i> – 1 7. <i>ImagineDesk toolkit</i>
(THUM; SCHWIND, 2010)	1. <i>SLIM</i> 2. <i>WebEx</i> 3. <i>Adobe Connect</i>
(KROPP; MEIER; BIDDLE, 2016)	1. <i>Jenkins</i> 2. <i>GitHub</i> 3. <i>Trello</i> 4. <i>OpenOLAT</i>
(SARMA; HOEK; REDMILES, 2018)	1. <i>LotusNotes</i> 2. <i>IRC</i> 3. <i>ICQ</i> 4. <i>AOL</i> 5. <i>Ariadne</i> 6. <i>Tesseract</i> 7. <i>SmallBlue</i> 8. <i>MyVine</i> 9. <i>OASiS</i> 10. <i>Palantir</i> 11. <i>Chianti</i> 12. <i>LightHouse</i> 13. <i>CollabV S</i> 14. <i>Git</i> 15. <i>GROVE</i> 16. <i>VNC</i> 17. <i>SynchronEyes</i> 18. <i>Sangam</i> 19. <i>ShrEdit</i> 20. <i>Jazz</i> 21. <i>Travis</i> 22. <i>SPADE</i> 23. <i>EPOS</i> 24. <i>Bugzilla</i> 25. <i>Trac</i> 26. <i>Milos</i> 27. <i>Autoplan</i> 28. <i>MSPro ject</i>
(DAMIAN; ZOWGHI, 2002)	1. <i>RequisitePro</i> 2. <i>Netmeeting</i>
(SATELI; ANGIUS; WITTE, 2013)	1. <i>ReqWiki</i>
(PALACIO <i>et al.</i> , 2011b)	1. <i>CWSM</i> 2. <i>Palantr</i> 3. <i>ProjectWatcher</i> 4. <i>FAST Dash</i> 5. <i>YooHoo</i>
(DUBEY; HUDEPOHL, 2013)	1. <i>ANT</i> 2. <i>SVN</i> 3. <i>ClearCase</i>
(BARBOSA; MALDONADO, 2006)	1. <i>CoWeb2</i> 2. <i>LaTeX</i> 3. <i>WebCT</i>
(SCHARFF; GOTEL; KULKARNI, 2010)	1. <i>Jazz</i> 2. <i>Rally</i>
(CHENG <i>et al.</i> , 2016)	1. <i>WeChat</i>
(EVANS <i>et al.</i> , 2014)	1. <i>BAESystems</i>
(GRUNDY; MUGRIDGE; HOSKING, 1997)	1. <i>Dora</i> 2. <i>SPE</i> 3. <i>Regatta</i> 4. <i>TeamFLOW</i> 5. <i>SPADE</i> 6. <i>ProcessWEAVER</i> 7. <i>EPOS</i> 8. <i>ADELE</i> – <i>T EMPO</i> 9. <i>Oz</i> 10. <i>GroupKit</i> 11. <i>Rendezvous</i> 12. <i>SerendipityPCE</i>
(KWAN; DAMIAN; STOREY, 2006)	1. <i>Babble</i> 2. <i>EGRET</i> 3. <i>Jazz</i>
(TREUDE; STOREY, 2012)	1. <i>Jazz</i> 2. <i>Bugzilla</i> 3. <i>TagSEA</i>

Fonte: Elaborada pela autora.

Tabela 7 – Ferramentas Colaborativas retornadas nos artigos da base Science Direct.

Autores	Ferramenta
(KALIDINDI, 2015)	1.Dropbox 2.Github 3.SourceForge 4.HU Bzero 5.nanoHU B 6.GoogleDocs 7.Authorea 8.ShareLaTex 9.Mendeley 10.ResearchGate 11.Plot.ly
(BUONO; COSTABILE; LANZI- LOTTI, 2014)	1.CBP(<i>ClockBoxPlot</i>)
(GIUFFRIDA; DITTRICH, 2015)	<i>Spira</i>
(BRAVO; DUQUE; GALLARDO, 2013)	1.COLLECE (<i>COLLaborative Edition, Compilation and Execution of programs</i>)
(ALI; LAI, 2016)	1.OSS (<i>Online Shopping System</i>)
(SHRIVASTAVA; RATHOD, 2017)	1.DAD (<i>Distributed Agile Development</i>)
(VLIETLAND; SOLINGEN; VLIET, 2016)	1.SV F (<i>Scrum Value Chain Framework</i>)
(YAGüE <i>et al.</i> , 2016)	1.Redmine 2.SonarQuBe 3.SVN(<i>Subversion</i>)
(CRUZ; MORIYA-HUZITA; FEL- TRIM, 2018)	1.GitHub
(GAUBATZ; LYTRA; ZDUN, 2015)	1.CoCoADvISE 2.Cloud9 3.Koding 4.Creatly 5.Lucidchart
(TRAINER; REDMILES, 2018)	1.GitHub 2.JIRA 3.Theseus
(MOULIN <i>et al.</i> , 2016)	1.JIRA 2.GIT 3.SVN(<i>Subversion</i>)
(SATZGER <i>et al.</i> , 2014)	1.JIRA 2.GIT 3.SVN(<i>Subversion</i>)
(RAIBULET; FONTANA, 2018)	1.GitHub 2.SonarQube 3.Microsoft Project in Education
(KURNIAWAN <i>et al.</i> , 2015)	1.Collabode 2.CodeR
(KOSHIMA; ENGLEBERT, 2015)	1.DiCoMEF
(PAASIVAARA; LASSENIUS, 2014)	1.CoP (<i>Communities of Practice</i>)
(SCANNIELLO; ERRA, 2014)	1.COFFEE
(SWIGGER <i>et al.</i> , 2004)	1.COFFEE
(LUCIA <i>et al.</i> , 2007)	1.STEVE (<i>Synchronous Collaborative Modelling Tool Enhanced with VErsioning Management</i>) 2.ADAMS (<i>ADvanced Arte fact Management System</i>) 3.SPE(<i>SmartProgrammingEnvironment</i>)
(KAMOUN; TAZI; DRIRA, 2012)	1.FADYRCOS(<i>Framework for Dynamic Reconfiguration ofnetworkedCollaborativeSystems</i>)
(FERNANDES <i>et al.</i> , 2012)	1.iThink
(LIN <i>et al.</i> , 2003)	1.VRCASE
(CELIK <i>et al.</i> , 2011)	1.Kuali
(LIU; WANG; ZHAO, 2012)	1.Red Hat 2.Sakai OAE 3.Kuali Rice
(ALVERTIS <i>et al.</i> , 2016)	1.GitHub 2.BitBucket 3.Kiln 4.SonarQube 5.Squale 6.Kickstarter 7.Indiegogo
(PORTILLO-RODRÍGUEZ <i>et al.</i> , 2014)	1.Ariadne 2.TraVis 3.WorldView 4.Tesseract 5.FonsecaTool 6.CodeSaw.
(EAST; KIRBY; LIU, 2008)	1.DrChecksSM (<i>Design Review and Checking System</i>)

Tabela 8 – Ferramentas Colaborativas retornadas nos artigos da base Springer.

Autores	Ferramenta
(SCHÜMMER; HAAKE, 2001)	1.TUKAN
(CHAN; CHUNG, 2002)	1.IPPM (<i>Integrated Project and Process Management</i>)
(ARANDA; VIZCAÍNO; PIATTINI, 2010)	1.WinWin Spiral Model 2. RE-GSD Framework
(GEISSER; HILDENBRAND, 2006)	1.ARENA 2. IBERE (<i>Internet – Based Empirical Requirements Evaluation</i>)
(LARRUCEA <i>et al.</i> , 2008)	1. SourceForge 2.Ezforge
(MISHRA; MISHRA, 2010)	1. AISA (<i>Asynchronous Inspector of Software Artifacts</i>); 2. <i>Collaborative Software Inspection (CSI)</i> ; 3. <i>Internet Based Inspection System (IBIS)</i> ; 4. <i>Web inspection Tool (WiT)</i>
(BABAR, 2010)	<i>Process – based Architecture Knowledge Management Environment (PAKME)</i>
(LEWANDOWSKI; BOURGUIN, 2008)	1. SourceForge; 2. <i>Freshmeat</i> 3. <i>Cooperative Layer for Software Development (CoolDev)</i>
(BAGHERI; GHORBANI, 2010)	1. DOME; 2. DFX; 3. <i>Groove Networks</i> 4. TOMSCOP; 5. EML 6. Jdiff 7. UMLDiff 8. <i>Bisimilarity Based Merge</i> 9. <i>Distributed consistency checking</i> 10. <i>Unified Traceability Schema</i> 11. KS-DMME; 12. <i>CoCreate OneSpace</i> 13. CollabCAD;
(KALLIAMVAKOU <i>et al.</i> , 2016)	1. GitHub
(SOHAN; RICHTER; MAURER, 2010)	1. VersionOne 2. ScrumPad 3. Xplanner
(LANUBILE, 2006)	1. Subversion 2. <i>CruiseControl</i> 4. <i>SourceForge</i>
(IMTIAZ; HAUGE; CHEN, 2007)	1.Opentaps
(SQUIRE, 2018)	1. RubyGems 2. <i>RubyForge</i> 3. Github
(BANIJAMALI <i>et al.</i> , 2017)	1. Github 2.JIRA
(VALETTO; KAISER, 1996)	1.Oz
(CEPÉDA <i>et al.</i> , 2010)	1.EvolTrack 2. JigaDrew
(SCHOLTES; MAVRODIEV; SCHWEITZER, 2016)	1.MYLIN 2. JAZZ 3. GITHUB 4. Open Source Software (OSS)
(JABANGWE; BÖRSTLER; PETERSEN, 2015)	1. <i>Source Code Monitor Tool</i>
(GUZMÁN <i>et al.</i> , 2010)	1. VTManager 2. <i>Concurrent Versioning System</i> 3. ISABEL CSCW 4. <i>Project Coordinator</i>
(FUKUI, 2002)	<i>Software Configuration Management (SCM)</i>
(GUPTA; CRK; BONDADE, 2011)	1.Knowfact
(KURTZ <i>et al.</i> , 2008)	1. <i>Mobile Scenario Presenter</i> 2. Arena
(TELLIOĞLU; WAGNER, 1997)	1.COLLECE (<i>COLLaborative Edition, Compilation and Execution of programs</i>)
(BERKLING <i>et al.</i> , 2007)	1. TraVis (<i>Trace Visualization</i>); 2. CodeBeamer; 3. Ibere
(BELSIS; KOUTOUMANOS; SGOUROPOULOU, 2014)	<i>Patterns – Based, Unsupervised Requirements Clustering (PBURC)</i>
(DUSTDAR; NASTIĆ; ŠČEKIĆ, 2017)	1. Smart 2. Gitlab
(YE; NAKAKOJI; YAMAMOTO, 2007)	<i>Dynamic Community Framework (DynC)</i>
(HATTORI; LANZA; ROBBES, 2012)	<i>Software Configuration Management (SCM)</i>
(AIELLO <i>et al.</i> , 2006)	RTDWD (<i>Real time Distributed Wideband Def phi</i>)
(DEWAN; HEGDE, 2007)	CollabVS
(WANG; HAAKE; WESSNER, 2002)	<i>Dynamic Cooperation Environment (DyCE)</i>
(HAAKE, 2000)	1. CoVER 2. Verso 3. GroupDesk
(GUTWIN <i>et al.</i> , 2005)	1.TUKAN 2. Augur 3. NetBSD 4.Apache httpd 5. Subversion
(BANI-SALAMEH; JEFFERY, 2011)	1.(SCDE), 2. SVN (Subversion) 3. CollabVS 4. <i>SourceForge</i> 5. <i>SourceCast</i>
(SOUZA; HILDENBRAND; REDMILES, 2007)	1. Offshore Software Development (OSD)2. TRaVis 3. Ariadne
(SARJOUGHIAN; NUTARO; JOSHI, 2011)	1. <i>Collaborative DEV S Modelling (CDM)</i>
(YU <i>et al.</i> , 2006)	1. Framework EkSarva

Fonte: Elaborada pela autora.

7.2 P2. Quais são as funcionalidades de um sistema colaborativo descritas no trabalho?

As funcionalidades de cada ferramenta encontrada na Revisão Sistemática da Literatura são apresentadas na Tabela 9, na Tabela 10, na Tabela 11 e na Tabela 12. Ressalta-se que algumas ferramentas que foram citadas não encontram-se nas tabelas, devido a inexistência dessa informação nos artigos.

Tabela 9 – Funcionalidades presentes nas Ferramentas Colaborativas.

índice	Ferramenta	Funcionalidade- Gerenciamento de artefatos
1.	PFC Mobile	Fórum
2.	Tesseract	Rastreamento de <i>bugs</i> , registros de comunicação
3.	SeeSoft	Gerenciamento de artefatos
4.	Creole	Gerenciamento de artefatos
5.	CodeCity	Gerenciamento de artefatos
6.	Team Tracks	Gerenciamento de artefatos
7.	Jazz	Gerenciamento de tarefas
8.	Hackystat	Gerenciamento de tarefas
9.	Oasis	Comunicação
10.	Step <i>n</i>	Comunicação
11.	EEL	Comunicação
12.	Palantír	Comunicação
13.	Chianti	Gerenciamento de artefatos
14.	Mylyn	Gerenciamento de artefatos
15.	CollabVS	Gerenciamento de artefatos
16.	Collaborator	Gerência de configuração
17.	X-Lite	Audio-conferência
18.	Eiffel Studio	wiki
19.	WebEx	Compartilhamento de tela
20.	Egret	Mensagens síncronas, email
21.	CVS	Controle de versão
22.	IBM Rational RequisitePro	Gerenciamento de configuração, rastreamento de bugs, gerenciamento de tarefas, discussão
23.	Sangam	Editor de código compartilhado
24.	Taiga.io	Rastreamento de bugs
25.	Redmine	Wiki, uploads de arquivos
26.	PSL/PSA	Gerenciamento de artefatos
27.	PLEXSYS	Gerenciamento de artefatos
28.	GroupSystems	Suporte a reunião
29.	GEMS	Suporte a reunião
30.	Group Data Modeler	Captura colaborativa de dados
31.	REMAP	Rastreamento de requisitos
32.	Flowdock	Comunicação assíncrona
33.	GitHub	Gerenciamento de configuração
34.	AgileFant	Compartilhamento de informações
35.	SoftDock	Gerenciar informação
36.	NetMeeting	Compartilhamento de informações, vídeo-conferência
37.	IBIS	Gerenciar informação
38.	<i>Eclipse Process Framework Composer</i>	Compartilhamento de informações, compartilhamento de conhecimento
39.	Open Unified Process	Orientação para RUP
40.	Rational Method Composer	Compartilhamento de informações, compartilhamento de conhecimento
41.	Appian Enterprise	Compartilhamento de informações, compartilhamento de conhecimento
42.	ARIS Platform	Compartilhamento de conhecimento
43.	BizAgi	Compartilhamento de informações, compartilhamento de conhecimento
44.	Intalio	Compartilhamento de informações, compartilhamento de conhecimento

Tabela 10 – Funcionalidades presentes nas Ferramentas Colaborativas.

índice	Ferramentas	Funcionalidades
45.	WEB-PerformCharts	Notificação por e-mail, gerenciamento de projeto, compartilhamento de arquivo
46.	Team Foundation Server	Compartilhamento de código
47.	Creatly	Gerenciamento de projeto
48.	Jira	Compartilhamento de arquivos e rastreamento
49.	ScrumWorks	Gerenciamento de projeto
50.	Smart Draw UML	Compartilhamento de arquivos
51.	java.net	Compartilhamento de código
52.	Subversion	Compartilhamento de código, gerenciamento de versão
53.	TagSEA	Desenvolvimento assíncrono
54.	<i>Microsoft Sharepoint Portal Server</i>	Ambiente de dev
55.	ISRM	Fórum, notificação por <i>e-mail</i>
56.	IBM Quickplace	Suporte as tarefas
57.	FusionForge	<i>Upload</i> de documentos, <i>wiki</i>
58.	Ansible	Entrega contínua
59.	Mocking bird	Design de interface
60.	DigiTile	Aprendizagem colaborativa
61.	Karma	Teste
62.	PBwork	Wiki
63.	Jenkins	Integração contínua
64.	CoNeX	Editor de negociação, gerenciamento de arquivos, mensagem informal
65.	Maven	Gerenciamento de construção
66.	Jasmine	Teste
67.	SOLVEIT	Teste de componentes
68.	Protractor	Teste
69.	Selenium	Teste
70.	Cucumber	Teste
71.	Gatling	Teste
72.	JUnit	Teste
73.	JMeter	Teste
74.	RestAssured	Teste
75.	Capbara	Teste
76.	Watir	Teste
77.	Artifactory	Gerenciamento de pacotes
78.	Bower	Gerenciamento de pacotes
79.	Grunt	Automação de tarefas
80.	CAISE	Ambiente de desenvolvimento colaborativo
81.	CASDE	Ambiente de desenvolvimento colaborativo
82.	eConference	Suporte a reunião
83.	Bugzilla	Rastreamento de <i>bugs</i>
84.	Evolving Artifact	Construção e refinamento de representações de design
85.	Zwiki	Detecção de erros
86.	InspectAnyWhere	Detecção de erros
87.	ConversationBuilder	Versionamento de artefatos
88.	MILOS	Modelagem de processo, planejamento de projeto e implementação
89.	Cool Talk	Compartilhamento de informação
90.	SoftFab	Automatização de construção e teste
91.	CoFFEE	Ambiente de desenvolvimento colaborativo
92.	WinWin	Sistema de arquivos multimídia
93.	SEREBRO	Ambiente de desenvolvimento colaborativo
94.	CURE	Salas virtuais
95.	iBistro	Comunicação informal
96.	Lucidchart	Modelagem e diagramação
97.	VECTOR	Ambiente de treinamento virtual
98.	CoID SPA	Modelagem colaborativa
99.	Trello	Quadro de tarefas

Fonte: Elaborada pela autora.

Tabela 11 – Funcionalidades presentes nas Ferramentas Colaborativas.

Índice	Ferramentas	Funcionalidades
100	Rose	Compartilhamento assíncrono
101	Mercurial	Gerenciamento de configuração
102.	CAMEL	Design de software
103.	MT-CollabUML	Design de software
104.	Groove	Compartilhamento de arquivo, mensagem instantânea, audio-conferencia
105.	OSD	Permite integrar as atividades concluídas ao Software;
106.	NetBSD	Coordenação de projetos
107.	Augur	Coordenação de projetos
108.	Apache httpd	Coordenação de projetos
109.	DyCE	Compartilhamento de conhecimento e de artefatos
110.	DynC	Fornece dados compartilhados dinamicamente replicados, bem como suporte transacional para acesso e modificação desses dados compartilhados
111.	CodeBeamer;	Permite a visualização e análise dos dependências diferentes entre vários artefatos dentro da CSDP.
112.	RCS	Rastreamento de bugs o sistema.
113.	Mobile Scenario Presenter	Gerenciamento de requisitos utilizando cenários .
114.	Requisite Pro	Gerenciamento de requisitos
115.	Trac	Controle de versão, wiki, notificação por e-mail
116.	Apache Allura	<i>Wiki, uploads</i> de arquivos
117.	TACTIC	Compartilhamento de arquivo
118.	Endeavour	Gerenciamento de projeto, <i>Wiki</i> , notificações <i>pore-mail</i>
119.	ProjectQtOr	Gerenciamento de documentos, reunião
120.	Plandora	Lista tarefas
121.	LibrePlan	Calendário
122.	Fossil	Rastreamento de bugs, wiki
123.	The Bug Genie	Rastreamento de bugs, wiki
124.	GanttProject	Controle de versão
125.	BuildTopia	Design colaborativo
126.	OneSpace	Salas de chat
127.	ABTool	<i>Brainstorming</i> síncrono
128.	eyeView	Sistema de reunião
129.	Libra-on-Chat	Modelagem colaborativa
130.	D-UML	Modelagem colaborativa
131.	MERMAID	Suporte a tomada de decisão
132.	GroupMeter	<i>Feedback</i>
133.	DNV GL	Compartilhamento de conhecimento
134.	MPK20	ambiente de desenvolvimento colaborativo tri-dimensional
135.	EasyWinWin	Suporte a tarefas de engenharia de requisitos
136.	MULE	Plataforma de integração distribuída
137.	Continuum	Integração contínua
138.	Hudson	Integração contínua
139.	Atlassian Confluence	<i>Wiki</i> , compartilhamento de material
140.	Atlassian Jira	Gerencia tarefas
141.	Cockburn	Gerencia tarefas
142.	Jimbo	Programação em par
143.	FASTDash	Gerenciamento de informação
144.	IRC	Comunicação remota
145.	SnipSnap	<i>Wiki</i> , integração de documentação
146.	WikiDoc	<i>Wiki</i>

Fonte: Elaborada pela autora.

Tabela 12 – Funcionalidades presentes nas Ferramentas Colaborativas.

índice	Ferramentas	Funcionalidades
147.	TeamWeaver Wiki	Wiki
148.	Moodle Wiki	Wiki
149.	Jotspot Wiki	Wiki
150.	TikiWiki	Wiki
151.	LocaTag	Mensagem instantânea
152.	Software Product Management Workbench	Gerenciamento de produto
153.	Pivotal Tracker	Rastreadores de bugs
154.	OPERAM	Gerenciamento de projeto
155.	Communico	Espaço de conversa aberto
156.	GroupBanter	Espaço de conversa aberto
157.	Loops	Espaço de conversa aberto
158.	ReachOut	Espaço de conversa aberto
159.	Threaded Chat	Espaço de conversa aberto
160.	Rear View Mirror	Comunicação síncrona
161.	SharePoint	Comunicação assíncrona
162.	SourceCast	Comunicação assíncrona
163.	CoRED	Editor em tempo real
164.	Doodle	Suporte a reuniões
165.	Dimdim	Chat
166.	Saros	Editor compartilhado
167.	GCodeEdit	Editor compartilhado
168.	Syde	Compartilhamento de mudanças
169.	gIBIS	Colaboração assíncrona
170.	GroupKit	Comunicação síncrona
171.	ClearBoard	Comunicação síncrona
172.	SPADE	Ambiente de desenvolvimento com suporte a cooperação
173.	ImagineDesk toolkit	Kit de ferramentas com suporte a coordenação
174.	SLIM	Editor síncrono
175.	OpenOLAT	Aprendizagem eletrônica
176.	Lotus Notes	Aprendizagem eletrônica
177.	Babble	Espaço de conversa aberto
178.	ICQ	Mensagem instantânea
179.	AOL	Mensagem instantânea
180.	SmallBlue	Rastreamento de bugs, registros de comunicação
181.	LightHouse	Gerência de configuração
182.	SynchronEyes	Editor compartilhado
183.	ShrEdit	Editor compartilhado
184.	EPOS	Ambiente de dev. com suporte a cooperação
185.	YooHoo	Compartilhamento de mudanças online
186.	ClearCase	Gerência de configuração
187.	WebCT	Comunicação de dados
188.	Rally	Ambiente integrado
189.	BAE Systems	Suporte a comunicação, suporte a reuniões
190.	Dora	Ambiente de desenvolvimento integrado
191.	SPE	Ambiente de desenvolvimento integrado
192.	TeamFLOW	Gerenciamento de fluxo de trabalho
193.	ProcessWEAVER	Ambiente de desenvolvimento com suporte a cooperação
194.	CoP(Communities of Practice)	Desenvolvimento de Software
195.	ADELE-TEMPO	Ambiente de desenvolvimento com suporte a cooperação
196.	GroupKit	Comunicação síncrona
197.	Rendezvous	Comunicação síncrona
198.	Serendipity PCE	Suporte a coordenação
199.	GitLab	Controle de versão
200.	nanoHUB	Permite integrar as atividades concluídas ao Software;

Fonte: Elaborada pela autora.

7.1 P3. As ferramentas são utilizadas em qual tipo de modelo de processo de desenvolvimento de software (ex: ágil, não-ágil, não-classificado)?

A classificação das ferramentas, sistemas e *frameworks* colaborativos encontra-se na Tabela 13.

Tabela 13 – Modelo do Processo

Metodologia	Ferramentas
Ágil	7. Jazz 32. Flowdock 33. GitHub 34. AgileFant 38. Eclipse Process Framework Composer 39. Open Unified Process 40. Rational Method Composer 41. Appian Enterprise 42. ARIS Platf orm 43. BizAgi 44. Intalio 46. Team Foundation Server 47. Lotus Notes 48. Jira 49. ScrumWorks 52. Subversion 55. GitLab 62. PBwork 63. Jenkins 65. Maven 93. SEREBRO 96. Lucidchart 99. Trello 111. CodeBeamer 112. RCS 116. Apache Allura 120. Plandora 133. DNV GL 141. Cockburn 154. OPERAM 162. SourceCast 163. CoRED 176. Lotus Notes 180. SmallBlue 194. CoP 196. FADYRCOS 198. Serendipity PCE 200. nanoHU B
Não-Ágil	
Não-Classificada	1. PFCMobile 2. Tesseract 3. SeeSoft 4. Creole 5. CodeCity 6. TeamTracks 8. Hackystat 9. Oasis 10. Step1 n 11. EEL 12. Palantr 13. Chianti 14. Mylyn 15. CollabV S 16. Collaborator 17. Xtite 18. EiffelStudio 19. WebEx 20. Egret 21. CVS 22. IBM Rational Requisite Pro 23. Sangam 24. Taiga.io 25. Redmine 26. PSL/PSA 27. PLEX SY S 28. GroupSystems 29. GEMS 30. GroupDataModeler 31. REMAP 35. SoftDock 36. NetMeeting 37. IBIS 45. WEB PerformanceCharts 50. SmartDrawU ML 51. java.net 53. TagSEA 54. MicrosoftSharepointPortalServer 56. IBMQuickplace 57. FusionForge 58. Ansible 59. Mockingbird 60. DigiTile 61. Karma 64. CoNeX 66. Jasmine 67. SOLV EIT 68. Protractor 69. Selenium 70. Cucumber 71. Gatling 72. JUnit 73. JMeter 74. RestAssured 75. Capybara 76. Watir 77. Artifactory 78. Bower 79. Grunt 80. CAISE 81. CASDE 82. 83. Bugzilla 84. EvolvingArtif act 85. Zwiki 86. InspectAnyWhere 87. ConversationBuilder 88. MILOS 89. CoolTalk 90. SoftFab 91. CoFFEE 92. WinWin 94. CURE 95. iBistro 97. VECTOR 98. ColDSPA 100. Rose 101. Mercurial 102. CAMEL 103. MTCollabU ML 104. Groove 105. OSD 106. NetBSD 107. Augur 108. Apachehttpd 109. DyCE 110. DynC 113. MobileScenarioPresenter 114. RequisitePro 115. Trac 117. TACTIC 118. Endeavour 119. ProjectQtOr 121. LibrePlan 122. Fossil 123. TheBugGenie 124. GanttProject 125. BuildTopia 126. OneSpace 127. ABTool 128. eyeView 129. LibrationChat 130. D U-ML 131. MERMAID 132. GroupMeter 134. MPK20 135. EasyWinWin 136. eConference 136. MULE 137. Continuum 138. Hudson 139. AtlassianConfluence 140. AtlassianJira 142. Jimbo 143. FAST Dash 144. IRC 145. SnipSnap 146. WikiDoc 147. TeamWeaverWiki 148. MoodleWiki 149. JotspotWiki 150. TikiWiki 151. LocaTag 152. SoftwareProductManagementWorkbench 153. PivotalTracker 155. Communico 156. GroupBanter 157. Loops 158. ReachOut 159. ThreadedChat 160. RearViewMirror 161. SharePoint 164. Doodle 165. Dimdim 166. Saros 167. GCodeEdit 168. Syde 169. gIBIS 170. GroupKit 171. ClearBoard 172. SPADE 173. ImagineDesktoolkit 174. SLIM 175. OpenLAT 177. Babble 178. ICQ 179. AOL 181. LightHouse 182. SynchronEyes 183. ShrEdit 184. EPOS 185. YooHoo 186. ClearCase 187. WebCT 188. Rally 189. BAESystems 190. Dora 191. SPE 192. TeamFLOW 193. 195. ADELE 197. Rendezvous

Fonte: Elaborada pela autora.

7.2 P4. Qual a licença da ferramenta?

A maioria dos sistemas, ferramentas e *frameworks* encontrados nesta pesquisa possuem licenças gratuitas (código aberto ou livre). Portanto, abaixo a Tabela 14 e Tabela 15 apresenta apenas as ferramentas proprietárias ou as ferramentas em que os artigos não revelaram a licença.

Tabela 14 – Licença das Ferramentas Colaborativas.

Pagas	Pagas e gratuitas	Não informa
Oasis	PSL/PSA	PFC Mobile
Collaborator	ARIS Platform	Tesseract
WebEx	TeamFLOW	SeeSoft
PSL/PSA	GitLab	Creole
Flowdock		Team Tracks
Rational Method Composer		StepIn
Appian Enterprise		EEL
ARIS Platform		Palantír
BizAgi		Chianti
Intalio		CollabVS
Team Foundation Serve		Egret
Lotus Notes		Stellation
ScrumWorks		Telelogic DOORS
Smart Draw UML		PLEXSYS
GroupSystems		Group Data Modeler
HP ALM		REMAP
Ansible		IBIS
Artifactory		java.net
COLLAB		Palantír
Mocking bird		CAISE
Collabnet		CASDE
Lotus Team Collaboration Suite		InspectAnyWhere
Microsoft Sharepoint Portal Server		GroupCRC
Groove		GENESIS
OpenProject		Cool Talk
The Bug Genie		SoftFab
Lotus SameTime		CoFFEE
Assembla		CoVER
MULE		Verso
Ant		GroupDesk
Atlassian Confluence		SCM
Atlassian Jira		EvoTrack
Agilefant		AISA
Cogent		CSI
Effusia		IBIS
GateKeeper		WiT
Jira		DOMÉ
Asana		DFX
Pivotal Tracker		Groove Networks
Doodle		TOMSCOP
Perforce		EML
Adobe Connect		diff
VNC		UMLDiff
MS Project		Bisimilarity based merge

Fonte: Elaborada pela autora.

Tabela 15 – Licença das Ferramentas Colaborativas.

Pagas	Pagas e gratuitas	Não informa
Rally		Distributed consistency checking;
CoCoADvISE		Unified Traceability Schema
Cloud9		KS-DMME
Koding		CoCreate OneSpace
Creatly		CollabCAD
Lucidchart		Software Configuration Management (SCM)
VRCASE		Rose
Knowfact		iBistro
Mobile Scenario Presenter		CURE
Arena		CoID SPA
RTDWD (Real-time Wideband-Delphi)	Distributed	IBM Quickplace
		Requisite Pro
		MT-CollabUML

Fonte: Elaborada pela autora.

7.3 P5. Em que fase do processo de desenvolvimento do software as ferramentas são utilizadas (ex: requisitos, arquitetura, teste)?

As ferramentas, os sistemas e os *frameworks* coletados na pesquisa correspondem a um processo de desenvolvimento de software mostrado na Figura 1. A Tabela 16, Tabela 17, Tabela 18, Tabela 19 e Tabela 20 apresentam uma relação entre as fases do processo de desenvolvimento do software e as ferramentas existentes. Esse relação foi determinada mediante as características e as funcionalidades encontradas no decorrer da leitura dos artigos. Também encontram-se os *links* de algumas dessas ferramentas na Tabela 21 e na Tabela 22.

Tabela 16 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.

índice	Ferramenta	Fase do Processo de Desenvolvimento
1.	PFC Mobile	Definição de requisitos
2.	Tesseract	Rastreamento de <i>bugs</i> , registros de comunicação
3.	SeeSoft	Implementação, integração e operação e manutenção
4.	Creole	Implementação, integração e operação e manutenção
5.	CodeCity	Implementação, integração e operação e manutenção
6.	Team Tracks	Implementação, integração e operação e manutenção
7.	Jazz	Implementação, integração e operação e manutenção
8.	Hackstat	Implementação, integração e operação e manutenção
9.	Oasis	Em todas as fases do processo
10.	Step _i n	Em todas as fases do processo
11.	EEL	Em todas as fases do processo
12.	Palantír	Em todas as fases do processo
13.	Chianti	Implementação, integração e operação e manutenção
14.	Mylyn	Implementação, integração e operação e manutenção
15.	CollabVS	Implementação, integração e operação e manutenção
16.	Collaborator	Operação e manutenção
17.	X-Lite	Definição de requisitos, projeto de sistema e software
18.	Eiffel Studio	Projeto de sistemas e software
19.	WebEx	Implementação
20.	Egret	Definição de requisitos

Fonte: Elaborada pela autora.

Tabela 17 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.

21.	CVS	Controle de versão
22.	IBM Rational RequisitePro	Integração, manutenção e teste
23.	Sangam	Implementação e teste
24.	Taiga.io	Integração e teste
25.	Redmine	Projeto de sistemas e software
26.	PSL/PSA	Implementação, integração e operação e manutenção
27.	PLEXSYS	Implementação, integração e operação e manutenção
28.	GroupSystems	Operação e manutenção
29.	GEMS	Definição de requisitos, projeto de sistema e software
30.	Group Data Modeler	Integração, teste e operação e manutenção
31.	REMAP	Definição de requisitos
32.	Flowdock	Definição de requisitos,
33.	GitHub	Integração e teste
34.	AgileFant	implementação e teste
35.	SoftDock	Definição de requisitos, projeto de sistema, implementação, na integração e na operação e teste
36.	NetMeeting	implementação e na definição de requisitos
37.	IBIS	Gerenciar informação
38.	<i>Eclipse Process Framework Composer</i>	projeto de sistema e software e implementação
39.	Open Unified Process	projeto de sistema, implementação e teste
40.	Rational Method Composer	Definição de requisitos, projeto de sistema
41.	Appian Enterprise	Definição de requisitos, projeto de sistema
42.	ARIS Platform	Definição de requisitos, projeto de sistema
43.	BizAgi	Definição de requisitos, projeto de sistema
44.	Intalio	Definição de requisitos, projeto de sistema
45.	WEB-PerformCharts	Definição de requisitos, projeto de sistema e implementação
46.	Team Foundation Server	Implementação e teste
47.	Creatly	projeto de pesquisa
48.	Jira	Integração e teste
49.	ScrumWorks	Projeto de pesquisa e software
50.	Smart Draw UML	projeto de pesquisa
51.	java.net	Implementação e Teste
52.	Subversion	Implementação e Teste, integração e operação e manutenção
53.	TagSEA	Implementação
54.		
55.	ISRM	Definição de requisitos e projeto de pesquisa
54.	<i>Microsoft Sharepoint Portal Server</i>	implementação
55.		
55.	GitLab	Integração e testes, operação e manutenção
56.	IBM Quickplace	Projeto de sistemas e software
57.	FusionForge	Projeto de sistemas e software
58.	Ansible	Implementação e integração
59.	Mocking bird	Implementação
60.	DigiTile	Em todas as fases do processo de desenvolvimento
61.	Karma	Integração e testes do sistema
62.	PBwork	Projeto de sistemas e software
63.	Jenkins	Integração contínua
64.	CoNeX	Editor de negociação, gerenciamento de arquivos, mensagem informal
65.	Maven	implementação e teste, projeto de sistemas
66.	Jasmine	Integração e testes do sistema
67.	SOLVEIT	Integração, testes do sistema e operação e manutenção
68.	Protractor	Integração, testes do sistema e operação e manutenção
69.	Selenium	Integração, testes do sistema e operação e manutenção
70.	Cucumber	Integração, testes do sistema e operação e manutenção
71.	Gatling	Integração, testes do sistema e operação e manutenção
72.	JUnit	Integração, testes do sistema e operação e manutenção
73.	JMeter	Integração, testes do sistema e operação e manutenção
74.	RestAssured	Integração, testes do sistema e operação e manutenção
75.	Capybara	Integração, testes do sistema e operação e manutenção
76.	Watir	Integração, testes do sistema e operação e manutenção
77.	Artifactory	Integração e teste de sistema

Fonte: Elaborada pela autora.

Tabela 18 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.

índice	Ferramentas	Fase do Processo de Desenvolvimento
78.	CASDE	Implementação e teste
79.	eConference	Definição de requisitos
80.	Bugzilla	implementação, integração e Operação e manutenção
81.	Evolving Artifact	projeto de sistemas e software
82.	Zwiki	Operação e manutenção
83.	InspectAnyWhere	Operação e manutenção
84.	ConversationBuilder	Implementação e teste
85.	MILOS	Definição de requisitos, projeto de sistema e software e implementação e teste
86.	Cool Talk	Definição de requisitos
87.	SoftFab	Implementação, integração e operação
88.	CoFFEE	Implementação e teste
89.	WinWin	Definição de requisitos
90.	SEREBRO	Implementação e teste
91.	CURE	Definição de requisitos
92.	iBistro	Definição de requisitos
93.	Lucidchart	Projeto de sistema e software
94.	VECTOR	Operação e manutenção
95.	CoID SPA	Projeto de sistema e software
96.	Trello	Projeto de sistema e software
97.	Rose	Implementação e teste
98.	Mercurial	Operação e manutenção
99.	CAMEL	Projeto de sistema e software
100.	MT-CollabUML	Projeto de sistema e software
101.	Groove	Projeto de sistema e software
102.	Offshore Software Development (OSD)	Implementação e teste
103.	NetBSD	Projeto de sistema e software
104.	Augur	Projeto de sistema e software
105.	Apache httpd	Projeto de sistema e software
106.	Dynamic Cooperation Environment (DyCE)	Implementação e teste
107.	Dynamic Community framework (DynC)	Implementação e teste
108.	CodeBeamer;	Projeto de sistemas e software.
109.	Revision Control System (RCS)	Integração do sistema
110.	Mobile Scenario Presenter	Definição dos requisitos .
111.	Requisite Pro	Definição dos requisitos
112.	Trac	Projeto de sistemas e softwares e integração
113.	Apache Allura	Definição dos requisitos
114.	TACTIC	Projeto de sistemas e softwares
115.	Endeavour	Projeto de sistemas e softwares, definição dos requisitos
116.	ProjectQtOr	Projeto de sistemas e softwares
117.	Plandora	Projeto de sistemas e softwares
118.	LibrePlan	Projeto de sistemas e softwares
119.	Fossil	Integração de sistema
120.	The Bug Genie	Integração de sistema
121.	GanttProject	Integração de sistema
122.	BuildTopia	Projeto de sistemas e softwares
123.	OneSpace	Definição de requisitos
124.	ABTool	Definição de requisitos
125.	eyeView	Definição de requisitos, projeto de sistema e softwares
126.	Libra-on-Chat	Projeto de sistemas e softwares

Fonte: Elaborada pela autora.

Tabela 19 – Fase do Processo de Desenvolvimento presentes nas Ferramentas Colaborativas.

Índice	Ferramentas	Fase do Processo de Desenvolvimento
130.	D-UML	Projeto de sistemas e softwares
131.	MERMAID	Projeto de sistemas e softwares
132.	GroupMeter	Projeto de sistemas e softwares
133.	DNV GL	Projeto de sistemas e softwares
134.	MPK20	Implementação e teste
135.	EasyWinWin	Definição de requisitos
136.	MULE	Integração do sistema
137.	Continuum	Integração do sistema
138.	Hudson	Integração do sistema
139.	Atlassian Confluence	Projeto de sistemas e softwares, Implementação e Teste
140.	Atlassian Jira	Projeto de sistemas e softwares
141.	Cockburn	Projeto de sistemas e softwares
142.	Jimbo	Implementação e teste
143.	FASTDash	Projeto de sistemas e softwares
144.	IRC	Definição de requisitos
145.	SnipSnap	Projeto de sistemas e softwares
146.	WikiDoc	Projeto de sistemas e softwares
147.	TeamWeaver	Projeto de sistemas e softwares
148.	Moodle Wiki	Projeto de sistemas e softwares
149.	Jotspot Wiki	Projeto de sistemas e softwares
150.	TikiWiki	Projeto de sistemas e softwares
151.	LocaTag	Definição de requisitos
152.	Software Product Management Workbench	Projeto de sistemas e softwares
153.	Pivotal Tracker	Integração e Teste
154.	OPERAM	Projeto de sistemas e softwares
155.	Communico	Definição de requisitos e projeto de sistemas e softwares
156.	GroupBanter	Definição de requisitos e projeto de sistemas e softwares
157.	Loops	Definição de requisitos e projeto de sistemas e softwares
158.	ReachOut	Definição de requisitos e projeto de sistemas e softwares
159.	Threaded Chat	Definição de requisitos e projeto de sistemas e softwares
160.	Rear View Mirror	Definição de requisitos e projeto de sistemas e softwares
161.	SharePoint	Definição de requisitos e projeto de sistemas e softwares
162.	SourceCast	Definição de requisitos e projeto de sistemas e softwares
163.	CoRED	Implementação e teste
164.	Doodle	Definição de requisitos e projeto de sistemas e softwares
165.	Dimdim	Definição de requisitos e projeto de sistemas e softwares
166.	Saros	Implementação e teste
167.	GCodeEdit	Implementação e teste
168.	Syde	Integração do sistemas
169.	gIBIS	Definição de requisitos e projeto de sistemas e softwares
170.	GroupKit	Definição de requisitos e projeto de sistemas e softwares
171.	ClearBoard	Definição de requisitos e projeto de sistemas e softwares
172.	SPADE	Implementação e teste
173.	ImagineDesk toolkit	Projeto de sistemas e software
174.	SLIM	Implementação e teste
175.	OpenOLAT	Implementação e teste
176.	Lotus Notes	Implementação e teste
177.	Babble	Definição de requisitos e projeto de sistemas e softwares

Fonte: Elaborada pela autora.

Tabela 20 – Fase do Processo de Desenvolvimento de Software presente nas Ferramentas Colaborativas.

índice	Ferramentas	Fase do Processo de Desenvolvimento
178.	ICQ	Definição de requisitos e projeto de sistemas e softwares
179.	AOL	Definição de requisitos e projeto de sistemas e softwares
180.	SmallBlue	Integração e teste de sistema
181.	LightHouse	Integração e teste de sistema
182.	SynchronEyes	Implementação e teste
183.	ShrEdit	Implementação e teste
184.	EPOS	Implementação e teste
185.	YooHoo	Integração e teste de sistema
186.	ClearCase	Integração e teste de sistema
187.	WebCT	Integração e teste de sistema
188.	Rally	Implementação e teste
189.	BAE Systems	Implementação e teste
190.	Dora	Implementação e teste
191.	SPE	Implementação e teste
192.	TeamFLOW	Projeto de sistemas e softwares
193.	ProcessWEAVER	Implementação e teste
194.	CoP	Implementação e teste
195.	ADELE-TEMPO	Implementação e teste
196.	GroupKit	Definição de requisito
197.	Rendezvous	Definição de requisito
198.	Serendipity PCE	Projeto de sistemas e softwares
199.	GitLab	Implementação e Teste
200.	nanoHub	Integração e teste de sistema

Fonte: Elaborada pela autora.

Tabela 21 – Links das Ferramentas Colaborativas.

índice	Links das ferramentas
5.	https://wettel.github.io/codecity-download.html
7.	http://www.jazz.net
8.	https://hackystat.github.io
9.	http://www.oasiscms.com/
12.	https://www.palantir.net/
14.	https://www.eclipse.org/mylyn/
15.	http://research.microsoft.com/enus/projects/collabvs/default.aspx
16.	www.smartbear.com
18.	dev.eiffel.com
19.	https://www.webex.com.br/
21.	http://www.nongnu.org/cvs/
24.	taiga.io
25.	redmine.org
28.	https://www.groupsupport.com/index.html
33.	https://github.com/mozilla
34.	https://www.agilefant.com/
43.	www.bizagi.com
44.	http://subversion.tigris.org
46.	https://go.microsoft.com/fwlink/?linkid=861638
48.	https://br.atlassian.com/software/jira
49.	https://www.collab.net/products/scrumworks
50.	https://www.smartdraw.com/uml-diagram/

Fonte: Elaborada pela autora.

Tabela 22 – Links das Ferramentas Colaborativas.

índice	Links das ferramentas
51.	http://java.net
52.	www.tigris.org/subversion
53.	http://tagsea.sourceforge.net
55.	http://seari.mit.edu/isrm.php
57.	fusionforge.org
102.	http://camel.apache.org/
58	https://www.ansible.com
59.	https://gomockingbird.com
60.	http://digitile.gulbenkian.pt/
61.	https://karma-runner.github.io/latest/index.html
106.	www.netbsd.org
62.	http://www.pbworks.com/
66.	https://jasmine.github.io/
68.	https://www.protractortest.org/
69.	https://www.seleniumhq.org/
70.	https://cucumber.io/
71.	https://gatling.io
72.	https://junit.org/junit5
73.	https://jmeter.apache.org/
74.	http://rest-assured.io/
75.	http://github.com/teamcapybara/capybara
76.	http://watir.com/
77.	https://jfrog.com/artifactory/
78.	https://bower.io/
79.	https://gruntjs.com
83.	https://www.bugzilla.org/
96.	lucidchart.com
99.	https://trello.com
101.	https://www.mercurial-scm.org/
104.	https://products.office.com/pt-br/sharepoint/collaboration?ms.officeurl=sharepointrtc=1
108.	www.apache.org
115.	trac.edgewall.org
116.	allura.apache.org
117.	community.southpawtech.com
118.	endeavor-mgmt.sourceforge.net
119.	projectqtor.org
120.	plandora.org
121.	libreplan.com
122.	fossil-scm.org
123.	thebuggenie.com
124.	ganttproject.biz
131.	https://mermaidjs.github.io/
133.	https://www.dnvgl.com/
136.	http://www.mulesoft.org
137.	http://continuum.apache.org/
138.	http://hudson-ci.org
150.	https://tiki.org/HomePage
153.	https://www.pivotaltracker.com/
162.	http://sourcecast.org/
163.	https://jenkins.io/
164.	https://doodle.com/
165.	https://en.softonic.com/download/dimdim/web-apps/post-download
166.	http://www.saros-project.org
175.	https://www.openolat.com/?lang=en

8 CONCLUSÃO

Existem muitas ferramentas colaborativas relacionadas ao desenvolvimento de *software*. Contudo, até o presente momento elas não estavam relacionadas as fases de desenvolvimento de *software*: Definição de requisitos, Projeto, Implementação e teste, Integração e operação e manutenção.

Este trabalho teve como objetivo identificar as ferramentas colaborativas usadas em todas as etapas de desenvolvimento de software. Para alcançar tal meta, uma revisão sistemática da literatura foi realizada. A revisão retornou 648 trabalhos dos quais 165 foram aceitos para extração. A partir das informações obtidas, foram criadas diversas tabelas que relacionam as ferramentas colaborativas ao que foi proposto.

Esta pesquisa utilizando-se de tabelas relacionou as ferramentas com: as fases de desenvolvimentos, as funcionalidades, a licença, os métodos citados nos artigos e fez também referência aos autores. Com esta pesquisa pretende-se contribuir com os *Stakeholders* envolvidos em trabalhos colaborativos e que precisem de orientações sobre qual ferramenta utilizar.

Pode-se observar por meio da análise de dados que a maioria dos artigos apresentam ferramentas gratuitas. Além disso, apesar de encontra-se as ferramentas em todas as fases do desenvolvimento, observou-se uma maior porcentagem de ferramentas colaborativas voltadas para as fases de: projeto de software e para definição de requisitos, sendo que algumas delas podem ser usadas em todos os processos do desenvolvimento.

Observa-se a necessidade de validação destes dados e a criação de um sistema que venha facilitar à busca por essas ferramentas tanto para desenvolvedor e gerente de projeto como para qualquer *Stakeholders* envolvido.

REFERÊNCIAS

- AIELLO, G.; ALESSI, M.; COSSENTINO, M.; URSO, A.; VELLA, G. Rtdwd: Real-time distributed wideband-delphi for user stories estimation. In: SPRINGER. **International Workshop on Rapid Integration of Software Engineering Techniques**. [S.l.], 2006. p. 35–50.
- ALI, N.; LAI, R. A method of requirements change management for global software development. **Information and Software Technology**, v. 70, p. 49 – 67, 2016. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584915001640>. Acesso em : 20 de julho de 2018
- ALLEN, T. J. *et al.* Managing the flow of technology: Technology transfer and the dissemination of technological information within the r&d organization. **MIT Press Books**, The MIT Press, v. 1, 1984.
- ALVERTIS, I.; KOUSSOURIS, S.; PAPASPYROS, D.; ARVANITAKIS, E.; MOUZAKITIS, S.; FRANKEN, S.; KOLVENBACH, S.; PRINZ, W. User involvement in software development processes. **Procedia Computer Science**, v. 97, p. 73 – 83, 2016. ISSN 1877-0509. 2nd International Conference on Cloud Forward: From Distributed to Complete Computing. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050916320981>. Acesso em : 20 de julho de 2018
- ANTUNES, P.; FERREIRA, A. Developing collaboration awareness support from a cognitive perspective. In: **2011 44th Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2011. p. 1–10. ISSN 1530-1605.
- ARANDA, G. N.; VIZCAÍNO, A.; PIATTINI, M. A framework to improve communication during the requirements elicitation process in gsd projects. **Requirements engineering**, Springer, v. 15, n. 4, p. 397–417, 2010.
- ARORA, R.; GOEL, S.; MITTAL, R. K. Supporting collaborative software development in academic learning environment: A collaborative pair and quadruple programming based approach. In: **2017 Tenth International Conference on Contemporary Computing (IC3)**. [S.l.: s.n.], 2017. p. 1–7. ISSN 2572-6129.
- ARORA, R.; GOEL, S.; MITTAL, R. K. Supporting collaborative software development over github. **Software: practice and experience**, Wiley Online Library, v. 47, n. 10, p. 1393–1416, 2017.
- BABAR, M. A. A web-based system for managing software architectural knowledge. In: **Web-based Support Systems**. [S.l.]: Springer, 2010. p. 305–332.
- BABAR, M. A.; KITCHENHAM, B.; ZHU, L.; JEFFERY, R. An exploratory study of groupware support for distributed software architecture evaluation process. In: **11th Asia-Pacific Software Engineering Conference**. [S.l.: s.n.], 2004. p. 222–229. ISSN 1530-1362.
- BABAR, M. A.; WINKLER, D.; BIFFL, S. Evaluating the usefulness and ease of use of a groupware tool for the software architecture evaluation process. In: **First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)**. [S.l.: s.n.], 2007. p. 430–439. ISSN 1949-3770.
- BAGHERI, E.; GHORBANI, A. A. An exploratory classification of applications in the realm of collaborative modeling and design. **Information Systems and e-Business Management**, Springer, v. 8, n. 3, p. 257–286, 2010.

- BAN, A.; PA, N. C.; DIN, J.; YAA'COB, N. A. M. Integrating social collaborative features in learning management system: A case study. In: **2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)**. [S.l.: s.n.], 2017. p. 67–72.
- BANDINELLI, S.; NITTO, E. D.; FUGGETTA, A. Supporting cooperation in the spade-1 environment. **IEEE Transactions on Software Engineering**, v. 22, n. 12, p. 841–865, Dec 1996. ISSN 0098-5589.
- BANI-SALAMEH, H.; JEFFERY, C. Teaching and learning in a social software development tool. In: **Social media tools and platforms in learning environments**. [S.l.]: Springer, 2011. p. 17–35.
- BANI-SALAMEH, H.; JEFFERY, C. Notifications management in distributed development environments: A case study. In: **2014 International Conference on Collaboration Technologies and Systems (CTS)**. [S.l.: s.n.], 2014. p. 49–55.
- BANIJAMALI, A.; DAWADI, R.; AHMAD, M. O.; SIMILÄ, J.; OIVO, M.; LIUKKUNEN, K. Empirical investigation of scrumban in global software development. In: HAMMOUDI, S.; PIRES, L. F.; SELIC, B.; DESFRAY, P. (Ed.). **Model-Driven Engineering and Software Development**. Cham: Springer International Publishing, 2017. p. 229–248. ISBN 978-3-319-66302-9.
- BARBOSA, E. F.; MALDONADO, J. C. Towards the establishment of a standard process for developing educational modules. In: **Proceedings. Frontiers in Education. 36th Annual Conference**. [S.l.: s.n.], 2006. p. 5–10. ISSN 0190-5848.
- BARTHOLOMEW, R. Evaluating a networked virtual environment for globally distributed avionics software development. In: **2008 IEEE International Conference on Global Software Engineering**. [S.l.: s.n.], 2008. p. 227–231. ISSN 2329-6305.
- BASHERI, M.; BURD, L.; BAGHAEI, N. Collaborative software design using multi-touch tables. In: **2012 4th International Congress on Engineering Education**. [S.l.: s.n.], 2012. p. 1–5.
- BECKER, S. A. A proposed learning environment for goal-specific improvements. In: **Proceedings of the Thirty-First Hawaii International Conference on System Sciences**. [S.l.: s.n.], 1998. v. 3, p. 389–398 vol.3.
- BELIS, P.; KOUTOUMANOS, A.; SGOUROPOULOU, C. Pburc: a patterns-based, unsupervised requirements clustering framework for distributed agile software development. **Requirements Engineering**, v. 19, n. 2, p. 213–225, Jun 2014. ISSN 1432-010X. Disponível em: <https://doi.org/10.1007/s00766-013-0172-9>. Acesso em : 20 de julho de 2018
- BERKLING, K.; GEISSER, M.; HILDENBRAND, T.; ROTHLAUF, F. Offshore software development: transferring research findings into the classroom. In: SPRINGER. **International Conference on Software Engineering Approaches for Offshore and Outsourced Development**. [S.l.], 2007. p. 1–18.
- BOOCH, G.; BROWN, A. W. Collaborative development environments. **Advances in computers**, Citeseer, v. 59, n. 1, p. 1–27, 2003.
- BRAVO, C.; DUQUE, R.; GALLARDO, J. A groupware system to support collaborative programming: Design and experiences. **Journal of Systems and Software**, v. 86, n. 7, p. 1759 – 1771, 2013. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121212002439>. Acesso em : 20 de julho de 2018
- BUONO, P.; COSTABILE, M. F.; LANZILOTTI, R. A circular visualization of people's

activities in distributed teams. **Journal of Visual Languages & Computing**, v. 25, n. 6, p. 903 – 911, 2014. ISSN 1045-926X. Distributed Multimedia Systems DMS2014 Part I. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1045926X14001189>. Acesso em : 20 de julho de 2018

CABALLÉ, S.; XHAFÁ, F.; RAYA, J.; BAROLLI, L.; UCHIDA, K. Building a software service for mobile devices to enhance awareness in web collaboration. In: **2014 International Conference on Intelligent Networking and Collaborative Systems**. [S.l.: s.n.], 2014. p. 369–376.

CAJANDER, A.; DANIELS, M.; KULTUR, C.; DAG, L. R.; LAXER, C. Managing international student collaborations: An experience report. In: **2012 Frontiers in Education Conference Proceedings**. [S.l.: s.n.], 2012. p. 1–6.

CALEFATO, F.; LANUBILE, F. A hub-and-spoke model for tool integration in distributed development. In: **2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)**. [S.l.: s.n.], 2016. p. 129–133. ISSN 2329-6313.

CALEFATO, F.; LANUBILE, F.; SCALAS, M. Evolving a text-based conferencing system: An experience report. In: **2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)**. [S.l.: s.n.], 2007. p. 427–431.

CAMPAGNOLO, B.; TACLA, C. A.; PARAISO, E. C.; SATO, G. Y.; RAMOS, M. P. An architecture for supporting small collocated teams in cooperative software development. In: **2009 13th International Conference on Computer Supported Cooperative Work in Design**. [S.l.: s.n.], 2009. p. 264–269.

CANFORA, G.; CIMITILE, A.; VISAGGIO, C. A. Lessons learned about distributed pair programming: what are the knowledge needs to address? In: **WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003**. [S.l.: s.n.], 2003. p. 314–319. ISSN 1080-1383.

CARLSON, P.; XIAO, N. Experience and recommendations for distributed software development. In: **2012 Second International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)**. [S.l.: s.n.], 2012. p. 21–24.

CASEY, V.; RICHARDSON, I. Project management within virtual software teams. In: **2006 IEEE International Conference on Global Software Engineering (ICGSE'06)**. [S.l.: s.n.], 2006. p. 33–42. ISSN 2329-6305.

CASTRO-HERNÁNDEZ, A.; SWIGGER, K.; PONCE-FLORES, M. P. Effects of cohesion-based feedback on the collaborations in global software development teams. In: **10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing**. [S.l.: s.n.], 2014. p. 74–83.

CASTRO-HERNÁNDEZ, A.; SWIGGER, K.; PONCE-FLORES, M. P.; TERÁN-VILLANUEVA, J. D. Measures for predicting task cohesion in a global collaborative learning environment. In: **2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)**. [S.l.: s.n.], 2016. p. 31–36. ISSN 2329-6313.

CELIK, N.; LEE, S.; MAZHARI, E.; SON, Y.-J.; LEMAIRE, R.; PROVAN, K. G. Simulation-based workforce assignment in a multi-organizational social network for alliance-based software development. **Simulation Modelling Practice and Theory**, v. 19, n. 10, p. 2169 – 2188, 2011. ISSN 1569-190X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1569190X11001250>. Acesso em : 20 de julho de 2018

CEPÊDA, R. d. S. V. de; MAGDALENO, A. M.; MURTA, L. G. P.; WERNER, C. M. L. Evoltrack: improving design evolution awareness in software development. **Journal of the Brazilian Computer Society**, Springer, v. 16, n. 2, p. 117–131, 2010.

CHAN, K. C.; CHUNG, L. M. Integrating process and project management for multi-site software development. **Annals of Software Engineering**, v. 14, n. 1, p. 115–143, Dec 2002. ISSN 1573-7489. Disponível em: <https://doi.org/10.1023/A:1020553624256>. Acesso em : 20 de julho de 2018

CHENG, X.; LIU, J.; DRUCKENMILLER, D.; FU, S. Trust development in globally distributed collaboration: A case study in china. In: **2016 49th Hawaii International Conference on System Sciences (HICSS)**. [S.l.: s.n.], 2016. p. 480–489. ISSN 1530-1605.

CHUNG, E.; JENSEN, C.; YATANI, K.; KUECHLER, V.; TRUONG, K. N. Sketching and drawing in the design of open source software. In: **2010 IEEE Symposium on Visual Languages and Human-Centric Computing**. [S.l.: s.n.], 2010. p. 195–202. ISSN 1943-6106.

COOK, C.; CHURCHER, N. An extensible framework for collaborative software engineering. In: **Tenth Asia-Pacific Software Engineering Conference, 2003**. [S.l.: s.n.], 2003. p. 290–299.

COSTA, A. M. Nicolaci-da; PIMENTEL, M. Sistemas colaborativos para uma nova sociedade e um novo ser humano. **Sistemas colaborativos**. PIMENTEL, M.; FUKS, H.(Orgs.). Rio de Janeiro: Elsevier, 2011.

CRUZ, G. A. M. da; MORIYA-HUZITA, E. H.; FELTRIM, V. D. Arsenal-gsd: A framework for trust estimation in virtual teams based on sentiment analysis. **Information and Software Technology**, v. 95, p. 46 – 61, 2018. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584916304517>. Acesso em : 20 de julho de 2018

DAMIAN, D.; LANUBILE, F.; MALLARDO, T. On the need for mixed media in distributed requirements negotiations. **IEEE Transactions on Software Engineering**, v. 34, n. 1, p. 116–132, Jan 2008. ISSN 0098-5589.

DAMIAN, D. E.; ZOWGHI, D. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In: **Proceedings IEEE Joint International Conference on Requirements Engineering**. [S.l.: s.n.], 2002. p. 319–328. ISSN 1090-705X.

DEAN, D. L.; LEE, J. D.; PENDERGAST, M. O.; HICKEY, A. M.; JR., J. F. N. Enabling the effective involvement of multiple users: Methods and tools for collaborative software engineering. **Journal of Management Information Systems**, Routledge, v. 14, n. 3, p. 179–222, 1997. Disponível em: <https://doi.org/10.1080/07421222.1997.11518180>. Acesso em : 20 de julho de 2018

DEFRANCO-TOMMARELLO, J.; DEEK, F. P. Collaborative software development: a discussion of problem solving models and groupware technologies. In: **Proceedings of the 35th Annual Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2002. p. 568–577.

DEFRANCO-TOMMARELLO, J.; HILTZ, S. R.; DEEK, F. P.; PEREZ, C.; KEENAN, J. P. Collaborative software development: experimental results. In: **36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the**. [S.l.: s.n.], 2003. p. 10 pp.–.

DEWAN, P.; HEGDE, R. Semi-synchronous conflict detection and resolution in asynchronous software development. In: **ECSCW 2007**. [S.l.]: Springer, 2007. p. 159–178.

- DUBEY, A.; HUDEPOHL, J. Towards global deployment of software engineering tools. In: **2013 IEEE 8th International Conference on Global Software Engineering**. [S.l.: s.n.], 2013. p. 129–133. ISSN 2329-6305.
- DULLEMOND, K.; GAMEREN, B. van; SOLINGEN, R. van. Overhearing conversations in global software engineering - requirements and an implementation. In: **7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)**. [S.l.: s.n.], 2011. p. 1–8.
- DUSTDAR, S.; NASTIĆ, S.; ŠĆEKIĆ, O. Programmatic management of human coordination and collaboration activities. In: **Smart Cities**. [S.l.]: Springer, 2017. p. 113–151.
- EAST, E. W.; KIRBY, J. G.; LIU, L. Y. Verification and validation of a project collaboration tool. **Automation in Construction**, v. 17, n. 2, p. 201 – 214, 2008. ISSN 0926-5805. 22nd Symposium on Automation and Robotics in Construction, ISARC 2005. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0926580507000611>. Acesso em : 20 de julho de 2018
- ERRA, U.; SCANNIELLO, G. Assessing communication media richness in requirements negotiation. **IET Software**, v. 4, n. 2, p. 134–148, April 2010. ISSN 1751-8806.
- ERRA, U.; SCANNIELLO, G. Assessing think-pair-square in distributed modeling of use case diagrams. In: **Workshop on Empirical Requirements Engineering (EmpiRE 2011)**. [S.l.: s.n.], 2011. p. 77–84. ISSN 2329-6348.
- EVANS, R. D.; GAO, J. X.; MARTIN, N.; SIMMONDS, C. Using web 2.0-based groupware to facilitate collaborative design in engineering education scheme projects. In: **2014 International Conference on Interactive Collaborative Learning (ICL)**. [S.l.: s.n.], 2014. p. 397–402.
- FERNANDES, J.; DUARTE, D.; RIBEIRO, C.; FARINHA, C.; PEREIRA, J. M.; SILVA, M. M. da. itthink: A game-based approach towards improving collaboration and participation in requirement elicitation. **Procedia Computer Science**, v. 15, p. 66 – 77, 2012. ISSN 1877-0509. 4th International Conference on Games and Virtual Worlds for Serious Applications(VS-GAMES'12). Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050912008216>. Acesso em : 20 de julho de 2018
- FILIPPO, D.; RAPOSO, A.; ENDLER, M.; FUKS, H. Ambientes colaborativos de realidade virtual e aumentada. **C. Kirner, R. Siscoutto, EDS**, p. 168–191, 2007.
- FINHOLT, T. A.; ROCCO, E.; BREE, D.; JAIN, N.; HERBSLEB, J. D. Netmeeting: A field trial of netmeeting in a geographically distributed organization. **SIGGROUP Bull.**, ACM, New York, NY, USA, v. 20, n. 1, p. 66–69, abr. 1999. ISSN 2372-7403. Disponível em: <http://doi.acm.org/10.1145/327556.327726>. Acesso em : 20 de julho de 2018
- FUKS, H.; RAPOSO, A.; GEROSA, M. A.; PIMENTEL, M.; FILIPPO, D.; LUCENA, C. d. Teorias e modelos de colaboração. **Sistemas colaborativos**, p. 16–33, 2012.
- FUKUI, S. Introduction of the software configuration management team and defect tracking system for global distributed development. In: **Software Quality—ECSQ 2002**. [S.l.]: Springer, 2002. p. 217–225.
- GAMBLE, R. F.; HALE, M. L. Assessing individual performance in agile undergraduate software engineering teams. In: **2013 IEEE Frontiers in Education Conference (FIE)**. [S.l.: s.n.], 2013. p. 1678–1684. ISSN 0190-5848.

- GARCIA, F.; VIZCAINO, A.; EBERT, C. Process management tools. **IEEE Software**, v. 28, n. 2, p. 15–18, March 2011. ISSN 0740-7459.
- GASPARIC, M.; JANES, A. What recommendation systems for software engineering recommend: A systematic literature review. **Journal of Systems and Software**, Elsevier, v. 113, p. 101–113, 2016.
- GAUBATZ, P.; LYTRA, I.; ZDUN, U. Automatic enforcement of constraints in real-time collaborative architectural decision making. **Journal of Systems and Software**, v. 103, p. 128 – 149, 2015. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121215000345>. Acesso em : 20 de julho de 2018
- GEISSER, M.; HILDENBRAND, T. A method for collaborative requirements elicitation and decision-supported requirements analysis. In: **Advanced software engineering: expanding the frontiers of software technology**. [S.l.]: Springer, 2006. p. 108–122.
- GERVIGNY, M. L. I.; NAGOWAH, S. D. Knowledge sharing for agile distributed teams: A case study of mauritius. In: **2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)**. [S.l.: s.n.], 2017. p. 413–419.
- GHORASHI, S.; JENSEN, C. Integrating collaborative and live coding for distance education. **Computer**, v. 50, n. 5, p. 27–35, May 2017. ISSN 0018-9162.
- GIUFFRIDA, R.; DITTRICH, Y. A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams. **Information and Software Technology**, v. 63, p. 11 – 30, 2015. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S095058491500049X>. Acesso em : 20 de julho de 2018
- GOTEL, O.; KULKARNI, V.; SAY, M.; SCHARFF, C.; SUNETNANTA, T. A global and competition-based model for fostering technical and soft skills in software engineering education. In: **2009 22nd Conference on Software Engineering Education and Training**. [S.l.: s.n.], 2009. p. 271–278. ISSN 1093-0175.
- GOTEL, O.; KULKARNI, V.; SCHARFF, C.; NEAK, L. Working across borders: Overcoming culturally-based technology challenges in student global software development. In: **2008 21st Conference on Software Engineering Education and Training**. [S.l.: s.n.], 2008. p. 33–40. ISSN 1093-0175.
- GRINSVEN, J. van; VREEDE, G. J. de. Addressing productivity concerns in risk management through repeatable distributed collaboration processes. In: **36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the**. [S.l.: s.n.], 2003. p. 10 pp.–.
- GRUNDY, J. C.; HOSKING, J. G.; MUGRIDGE, W. B. Serving up a banquet: towards an environment supporting all aspects of software development. In: **Proceedings 1996 International Conference Software Engineering: Education and Practice**. [S.l.: s.n.], 1996. p. 465–472.
- GRUNDY, J. C.; MUGRIDGE, W. B.; HOSKING, J. G. Utilising past event histories in a process-centred software engineering environment. In: **Proceedings of Australian Software Engineering Conference ASWEC 97**. [S.l.: s.n.], 1997. p. 127–136.
- GUPTA, A.; CRK, I.; BONDADE, R. Leveraging temporal and spatial separations with the 24-hour knowledge factory paradigm. **Information Systems Frontiers**, Springer, v. 13, n. 3, p. 397–405, 2011.

GUTWIN, C.; SCHNEIDER, K.; PAQUETTE, D.; PENNER, R. Supporting group awareness in distributed software development. In: BASTIDE, R.; PALANQUE, P.; ROTH, J. (Ed.).

Engineering Human Computer Interaction and Interactive Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 383–397. ISBN 978-3-540-31961-0.

GUZMÁN, J. G.; RAMOS, J. S.; SECO, A. A.; ESTEBAN, A. S. How to get mature global virtual teams: a framework to improve team process management in distributed software teams.

Software Quality Journal, v. 18, n. 4, p. 409–435, Dec 2010. Disponível em:

<https://doi.org/10.1007/s11219-010-9096-5>. Acesso em : 20 de julho de 2018

HAAKE, J. M. Structural computing in the collaborative work domain? In: SPRINGER.

International Workshop on Structural Computing. [S.l.], 2000. p. 108–119.

HATTORI, L. P.; LANZA, M.; ROBBES, R. Refining code ownership with synchronous

changes. **Empirical Software Engineering**, v. 17, n. 4, p. 467–499, Aug 2012. Disponível em:

<https://doi.org/10.1007/s10664-010-9145-5>. Acesso em : 20 de julho de 2018

HILDENBRAND, T.; ROTHLAUF, F.; GEISSER, M.; HEINZL, A.; KUDE, T. Approaches to collaborative software development. In: **2008 International Conference on Complex,**

Intelligent and Software Intensive Systems. [S.l.: s.n.], 2008. p. 523–528.

HOSSAIN, E.; BANNERMAN, P. L.; JEFFERY, R. Towards an understanding of tailoring scrum in global software development: A multi-case study. In: **Proceedings of the 2011**

International Conference on Software and Systems Process. New York, NY, USA: ACM, 2011. (ICSSP '11), p. 110–119. ISBN 978-1-4503-0730-7. Disponível em:

<http://doi.acm.org/10.1145/1987875.1987894>. Acesso em : 20 de julho de 2018

IMTIAZ, A.; HAUGE, J. B.; CHEN, S. Collaboration within the tool-and-die manufacturing industry through open-source modular erp/crm systems. In: CAMARINHA-MATOS, L. M.;

AFSARMANESH, H.; NOVAIS, P.; ANALIDE, C. (Ed.). **Establishing the Foundation of Collaborative Networks**. Boston, MA: Springer US, 2007. p. 469–476. ISBN

978-0-387-73798-0.

JABANGWE, R.; BÖRSTLER, J.; PETERSEN, K. Handover of managerial responsibilities in global software development: a case study of source code evolution and quality. **Software**

Quality Journal, v. 23, n. 4, p. 539–566, Dec 2015. ISSN 1573-1367. Disponível em:

<https://doi.org/10.1007/s11219-014-9247-1>. Acesso em : 20 de julho de 2018

KALIDINDI, S. R. 8 - materials innovation cyberinfrastructure. In: KALIDINDI, S. R. (Ed.).

Hierarchical Materials Informatics. Boston: Butterworth-Heinemann, 2015. p. 207 – 212.

ISBN 978-0-12-410394-8. Disponível em:

<http://www.sciencedirect.com/science/article/pii/B9780124103948000084>. Acesso em : 20 de julho de 2018

KALLIAMVAKOU, E.; DAMIAN, D.; BLINCOE, K.; SINGER, L.; GERMAN, D. M. Open source-style collaborative development practices in commercial projects using github. In: **2015**

IEEE/ACM 37th IEEE International Conference on Software Engineering. [S.l.: s.n.],

2015. v. 1, p. 574–585. ISSN 0270-5257.

KALLIAMVAKOU, E.; GOUSIOS, G.; BLINCOE, K.; SINGER, L.; GERMAN, D. M.; DAMIAN, D. An in-depth study of the promises and perils of mining github. **Empirical**

Software Engineering, Springer, v. 21, n. 5, p. 2035–2071, 2016.

KAMOON, A.; TAZI, S.; DRIRA, K. Fadyrcos, a semantic interoperability framework for collaborative model-based dynamic reconfiguration of networked services. **Computers in Industry**, v. 63, n. 8, p. 756 – 765, 2012. ISSN 0166-3615. Special Issue on Sustainable Interoperability: The Future of Internet Based Industrial Enterprises. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0166361512001261>. Acesso em : 20 de julho de 2018

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. 2007.

KOSCIANSKI, A.; SOARES, M. d. S. Qualidade de software. **São Paulo: Novatec**, v. 3, 2006.

KOSHIMA, A. A.; ENGLEBERT, V. Collaborative editing of emf/core meta-models and models: Conflict detection, reconciliation, and merging in dicomef. **Science of Computer Programming**, v. 113, p. 3 – 28, 2015. ISSN 0167-6423. Model Driven Development (Selected & extended papers from MODELSWARD 2014). Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167642315001380>. Acesso em : 20 de julho de 2018

KRAUT, R.; SCHERLIS, W.; MUKHOPADHYAY, T.; MANNING, J.; KIESLER, S. Homenet: A field trial of residential internet services. In: ACM. **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. [S.l.], 1996. p. 284–291.

KROPP, M.; MEIER, A. Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight. In: **2016 IEEE Global Engineering Education Conference (EDUCON)**. [S.l.: s.n.], 2016. p. 1003–1011. ISSN 2165-9567.

KROPP, M.; MEIER, A.; BIDDLE, R. Teaching agile collaboration skills in the classroom. In: **2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)**. [S.l.: s.n.], 2016. p. 118–127. ISSN 2377-570X.

KURNIAWAN, A.; KURNIAWAN, A.; SOESANTO, C.; WIJAYA, J. E. C. Coder: Real-time code editor application for collaborative programming. **Procedia Computer Science**, v. 59, p. 510 – 519, 2015. ISSN 1877-0509. International Conference on Computer Science and Computational Intelligence (ICCCSI 2015). Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050915020608>.

KURTZ, G.; GEISSER, M.; HILDENBRAND, T.; KUDE, T. Mobile technologies in requirements engineering. In: **Advances in Computer and Information Sciences and Engineering**. [S.l.]: Springer, 2008. p. 317–322. Acesso em : 20 de julho de 2018

KUSUMASARI, T. F.; SUPRIANA, I.; SURENDRO, K.; SASTRAMIHARDJA, H. Collaboration model of software development. In: **Proceedings of the 2011 International Conference on Electrical Engineering and Informatics**. [S.l.: s.n.], 2011. p. 1–6. ISSN 2155-6830.

KWAN, I.; DAMIAN, D.; STOREY, M. Visualizing a requirements-centred social network to maintain awareness within development teams. In: **2006 First International Workshop on Requirements Engineering Visualization (REV'06 - RE'06 Workshop)**. [S.l.: s.n.], 2006. p. 7–7.

KöBLER, F.; KOENE, P.; KRUMHOLTZ, H.; ALTMANN, M.; LEIMEISTER, J. M. Locatag - an nfc-based system enhancing instant messaging tools with real-time user location. In: **2010 Second International Workshop on Near Field Communication**. [S.l.: s.n.], 2010. p. 57–61.

- LANUBILE, F. Collaboration in distributed software development. In: **Software Engineering**. [S.l.]: Springer, 2006. p. 174–193.
- LANUBILE, F.; EBERT, C.; PRIKLADNICKI, R.; VIZCAÍNO, A. Collaboration tools for global software engineering. **IEEE software**, IEEE, v. 27, n. 2, 2010.
- LARRUCEA, X.; FERNANDEZ, R.; SORIANO, J.; MARTÍNEZ, A. L.; GONZALEZ-BARAHONA, J. M. A service based development environment on web 2.0 platforms. In: SPRINGER. **European Conference on a Service-Based Internet**. [S.l.], 2008. p. 38–48.
- LEE, J. D.; HICKEY, A. M.; ZHANG, D.; SANTANEN, E.; ZHOU, L. Cold spa: a tool for collaborative process model development. In: **Proceedings of the 33rd Annual Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2000. p. 10 pp. vol.1–.
- LEI, H.; CHAKRABORTY, D.; CHANG, H.; DIKUN, M. J.; HEATH, T.; LI, J. S.; NAYAK, N.; PATNAIK, Y. Contextual collaboration: platform and applications. In: **IEEE International Conference on Services Computing, 2004. (SCC 2004). Proceedings. 2004**. [S.l.: s.n.], 2004. p. 197–206.
- LEPPÄNEN, M.; LAHTINEN, S.; IHANTOLA, P. Hammer and nails - crucial practices and tools in ad hoc student teams. In: **2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)**. [S.l.: s.n.], 2016. p. 142–146. ISSN 2377-570X.
- LEWANDOWSKI, A.; BOURGUIN, G. An eclipse-based framework for supporting software development cooperative activities. In: MANOLOPOULOS, Y.; FILIPE, J.; CONSTANTOPOULOS, P.; CORDEIRO, J. (Ed.). **Enterprise Information Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 254–268. ISBN 978-3-540-77581-2.
- LIN, Q.; LOW, C. P.; NG, J. M.; BU, J.; LIU, X. Multiuser collaborative work in virtual environment based case tool. **Information and Software Technology**, v. 45, n. 5, p. 253 – 267, 2003. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S095058490300017X>. Acesso em : 20 de julho de 2018
- LIU, M.; WANG, H. J.; ZHAO, J. L. Technology flexibility as enabler of robust application development in community source: The case of kuali and sakai. **Journal of Systems and Software**, v. 85, n. 12, p. 2921 – 2928, 2012. ISSN 0164-1212. Self-Adaptive Systems. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121212001744>. Acesso em : 20 de julho de 2018
- LIUKKUNEN, K.; LINDBERG, K.; HYYSALO, J.; MARKKULA, J. Supporting collaboration in the geographically distributed work with communication tools in the remote district sme's. In: **IEEE. Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on**. [S.l.], 2010. p. 155–164.
- LUCIA, A. D.; FASANO, F.; SCANNIELLO, G.; TORTORA, G. Enhancing collaborative synchronous uml modelling with fine-grained versioning of software artefacts. **Journal of Visual Languages & Computing**, v. 18, n. 5, p. 492 – 503, 2007. ISSN 1045-926X. In honour of Stefano Levialdi. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1045926X07000481>. Acesso em : 20 de julho de 2018

- MAK, D. K. M.; KRUCHTEN, P. B. Nextmove: A framework for distributed task coordination. In: **2007 Australian Software Engineering Conference (ASWEC'07)**. [S.l.: s.n.], 2007. p. 399–408. ISSN 1530-0803.
- MENDES, C. C.; MILLA, G. L.; MIRANDA, R. P.; MORAES, R. L.; ALBERTI, T. F.; BEHAR, P. A. Texto coletivo: possibilidades e limites no processo ensino-aprendizagem a distância. **RENOTE: revista novas tecnologias na educação [recurso eletrônico]. Porto Alegre, RS**, 2007.
- MEYER, B. Design and code reviews in the age of the internet. **Commun. ACM**, ACM, New York, NY, USA, v. 51, n. 9, p. 66–71, set. 2008. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/1378727.1378744>. Acesso em : 20 de julho de 2018
- MISHRA, D.; MISHRA, A. A software inspection process for globally distributed teams. In: MEERSMAN, R.; DILLON, T.; HERRERO, P. (Ed.). **On the Move to Meaningful Internet Systems: OTM 2010 Workshops**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 289–296. ISBN 978-3-642-16961-8.
- MISTRÍK, I.; GRUNDY, J.; HOEK, A. Van der; WHITEHEAD, J. Collaborative software engineering: challenges and prospects. In: **Collaborative Software Engineering**. [S.l.]: Springer, 2010. p. 389–403.
- MOE, N. B.; FÆGRI, T. E.; CRUZES, D. S.; FAUGSTAD, J. E. Enabling knowledge sharing in agile virtual teams. In: **2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)**. [S.l.: s.n.], 2016. p. 29–33. ISSN 2329-6313.
- MONASOR, M. J.; PARKES, J.; NOLL, J.; VIZCAÍNO, A.; PIATTINI, M.; BEECHAM, S. Global software development education: A commercial perspective from a case study. In: **2014 IEEE 9th International Conference on Global Software Engineering**. [S.l.: s.n.], 2014. p. 173–182. ISSN 2329-6305.
- MONASOR, M. J.; VIZCAÍNO, A.; PIATTINI, M. A training tool for global software development. In: **2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET)**. [S.l.: s.n.], 2010. p. 9–16.
- MOULIN, C.; KAERI, Y.; SUGAWARA, K.; ABEL, M.-H. Capitalization of remote collaborative brainstorming activities. **Computer Standards & Interfaces**, v. 48, p. 217 – 224, 2016. ISSN 0920-5489. Special Issue on Information System in Distributed Environment. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0920548915001397>. Acesso em : 20 de julho de 2018
- NIEMINEN, A. Real-time collaborative resolving of merge conflicts. In: **8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)**. [S.l.: s.n.], 2012. p. 540–543.
- NIINIMÄKI, T.; LASSENIUS, C. Experiences of instant messaging in global software development projects: A multiple case study. In: **2008 IEEE International Conference on Global Software Engineering**. [S.l.: s.n.], 2008. p. 55–64. ISSN 2329-6305.
- PAASIVAARA, M.; BLINCOE, K.; LASSENIUS, C.; DAMIAN, D.; SHEORAN, J.; HARRISON, F.; CHHABRA, P.; YUSSUF, A.; ISOTALO, V. Learning global agile software engineering using same-site and cross-site teams. In: **Proceedings of the 37th International Conference on Software Engineering - Volume 2**. Piscataway, NJ, USA: IEEE Press, 2015. (ICSE '15), p. 285–294. Disponível em: <http://dl.acm.org/citation.cfm?id=2819009.2819053>. Acesso em : 20 de julho de 2018

- PAASIVAARA, M.; LASSENIUS, C. Communities of practice in a large distributed agile software development organization – case ericsson. **Information and Software Technology**, v. 56, n. 12, p. 1556 – 1577, 2014. ISSN 0950-5849. Special issue: Human Factors in Software Development. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584914001475>. Acesso em : 20 de julho de 2018
- PALACIO, R. R.; MORAN, A. L.; VIZCAINO, A.; GONZALEZ, V. M. Knowledge flow as facilitator for getting into collaboration in distributed software development. In: **2011 44th Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2011. p. 1–10. ISSN 1530-1605.
- PALACIO, R. R.; VIZCAINO, A.; MORAN, A. L.; GONZALEZ, V. M. Tool to facilitate appropriate interaction in global software development. **IET Software**, v. 5, n. 2, p. 157–171, April 2011. ISSN 1751-8806.
- PAPADAKI, E.; POLEMI, D.; DAMILOS, D. K. A holistic, collaborative, knowledge-sharing approach for information security risk management. In: **2008 The Third International Conference on Internet Monitoring and Protection**. [S.l.: s.n.], 2008. p. 125–130.
- PESOLA, J.; TANNER, H.; ESKELI, J.; PARVIAINEN, P.; BENDAS, D. Integrating early v amp;v support to a gse tool integration platform. In: **2011 IEEE Sixth International Conference on Global Software Engineering Workshop**. [S.l.: s.n.], 2011. p. 95–101. ISSN 2329-6305.
- PORTILLO-RODRÍGUEZ, J.; VIZCAÍNO, A.; PIATTINI, M.; BEECHAM, S. Tools used in global software engineering: A systematic mapping review. **Information and Software Technology**, Elsevier, v. 54, n. 7, p. 663–685, 2012.
- PORTILLO-RODRÍGUEZ, J.; VIZCAÍNO, A.; PIATTINI, M.; BEECHAM, S. Using agents to manage socio-technical congruence in a global software engineering project. **Information Sciences**, v. 264, p. 230 – 259, 2014. ISSN 0020-0255. Serious Games. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025514000152>. Acesso em : 20 de julho de 2018
- PRESSMAN, R. S. **Engenharia de software**. [S.l.]: Makron books São Paulo, 1995. v. 6.
- RAIBULET, C.; FONTANA, F. A. Collaborative and teamwork software development in an undergraduate software engineering course. **Journal of Systems and Software**, v. 144, p. 409 – 422, 2018. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121218301389>. Acesso em : 20 de julho de 2018
- RAS, E. Investigating wikis for software engineering - results of two case studies. In: **2009 ICSE Workshop on Wikis for Software Engineering**. [S.l.: s.n.], 2009. p. 47–55.
- RF, S. Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. **Revista brasileira de fisioterapia**, SciELO Brasil, v. 11, n. 1, p. 83–89, 2007.
- SARJOUGHIAN, H. S.; NUTARO, J. J.; JOSHI, G. Towards collaborative component-based modelling. **Journal of Simulation**, Taylor & Francis, v. 5, n. 2, p. 77–88, 2011.
- SARMA, A.; AL-ANI, B.; TRAINER, E.; FILHO, R. S. S.; SILVA, I. A. da; REDMILES, D.; HOEK, A. van der. Continuous coordination tools and their evaluation. In: **Collaborative Software Engineering**. [S.l.]: Springer, 2010. p. 153–178.

SARMA, A.; HOEK, A. V. der; REDMILES, D. The coordination pyramid: A perspective on the state of the art in coordination technology. **Computer**, p. 1–1, 2018. ISSN 0018-9162.

SARMA, A.; REDMILES, D.; HOEK, A. v. d. Categorizing the spectrum of coordination technology. **Computer**, v. 43, n. 6, p. 61–67, June 2010. ISSN 0018-9162.

SATELI, B.; ANGIUS, E.; WITTE, R. The reqwiki approach for collaborative software requirements engineering with integrated text analysis support. In: **2013 IEEE 37th Annual Computer Software and Applications Conference**. [S.l.: s.n.], 2013. p. 405–414. ISSN 0730-3157.

SATZGER, B.; ZABOLOTNYI, R.; DUSTDAR, S.; WILD, S.; GAEDKE, M.; GÖBEL, S.; NESTLER, T. Chapter 8 - toward collaborative software engineering leveraging the crowd. In: MISTRİK, I.; BAHSOON, R.; KAZMAN, R.; ZHANG, Y. (Ed.). **Economics-Driven Software Architecture**. Boston: Morgan Kaufmann, 2014. p. 159 – 182. ISBN978-0-12-410464-8. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780124104648000088>. Acesso em : 20 de julho de 2018

SCANNIELLO, G.; ERRA, U. Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments. **Journal of Visual Languages & Computing**, v. 25, n. 4, p. 494 – 517, 2014. ISSN 1045-926X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1045926X14000329>. Acesso em : 20 de julho de 2018

SCHARFF, C.; GOTEL, O.; KULKARNI, V. Transitioning to distributed development in students' global software development projects: The role of agile methodologies and end-to-end tooling. In: **2010 Fifth International Conference on Software Engineering Advances**. [S.l.: s.n.], 2010. p. 388–394.

SCHOLTES, I.; MAVRODIEV, P.; SCHWEITZER, F. From aristotle to ringelmann: a large-scale analysis of team productivity and coordination in open source software projects. **Empirical Software Engineering**, Springer, v. 21, n. 2, p. 642–683, 2016.

SCHÜMMER, T.; HAAKE, J. M. Supporting distributed software development by modes of collaboration. In: SPRINGER. **ECSCW 2001**. [S.l.], 2001. p. 79–98.

SHRIVASTAVA, S. V.; RATHOD, U. A risk management framework for distributed agile projects. **Information and Software Technology**, v. 85, p. 1 – 15, 2017. ISSN 0950-5849. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584916304815>. Acesso em : 20 de julho de 2018

SHROFF, G.; MEHTA, A.; AGARWAL, P.; SINHA, R. Collaborative development of business applications. In: **2005 International Conference on Collaborative Computing: Networking, Applications and Worksharing**. [S.l.: s.n.], 2005. p. 6 pp.–.

SIEVI-KORTE, O.; SYSTÄ, K.; HJELSVOLD, R. Global vs. local — experiences from a distributed software project course using agile methodologies. In: **2015 IEEE Frontiers in Education Conference (FIE)**. [S.l.: s.n.], 2015. p. 1–8.

SILVA, E. J. D.; TACLA, C. A.; BARTHÈS, J.-P. A.; RAMOS, M. P.; PARAISO, E. C. A collaborative virtual workspace for software development. In: IEEE. **Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on**. [S.l.], 2015. p. 24–29.

SILVA, E. J. d.; TORQUATO, E.; RAMOS, M. P.; PARAISO, E. C. Operam: A collaborative semantic workspace for software verification. In: **2013 IEEE International Conference on Systems, Man, and Cybernetics**. [S.l.: s.n.], 2013. p. 1026–1031. ISSN 1062-922X.

SILVA, E. J. da; TACLA, C. A.; BARTHÈS, J. P. A.; RAMOS, M. P.; PARAISO, E. C. A collaborative virtual workspace for software development. In: **2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)**. [S.l.: s.n.], 2015. p. 24–29.

SINHA, V.; SENGUPTA, B.; CHANDRA, S. Enabling collaboration in distributed requirements management. **IEEE Software**, v. 23, n. 5, p. 52–61, Sept 2006. ISSN 0740-7459.

SOHAN, S.; RICHTER, M. M.; MAURER, F. Auto-tagging emails with user stories using project context. In: SPRINGER. **International Conference on Agile Software Development**. [S.l.], 2010. p. 103–116.

SOMMERVILLE, I. **Software engineering**. [S.l.]: New York: Addison-Wesley, 2011.

SOMMERVILLE, I.; ARAKAKI, R.; MELNIKOFF, S. S. S. **Engenharia de software**. [S.l.]: Pearson Prentice Hall, 2008.

SOUZA, C. R. B. de; HILDENBRAND, T.; REDMILES, D. Toward visualization and analysis of traceability relationships in distributed and offshore software development projects. In: MEYER, B.; JOSEPH, M. (Ed.). **Software Engineering Approaches for Offshore and Outsourced Development**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 182–199. ISBN 978-3-540-75542-5.

SOUZA, R. B. d.; MARCZAK, S.; PRIKLANDNICKI, R. Desenvolvimento colaborativo de software. 2012.

SQUIRE, M. Data sets describing the circle of life in ruby hosting, 2003–2016. **Empirical Software Engineering**, Springer, p. 1–30, 2018.

SU, S. H.; SCHARFF, C. Know yourself and beyond: A students' global software development project experience with agile methodology. In: **2010 5th International Conference on Computer Science Education**. [S.l.: s.n.], 2010. p. 412–417.

SUNINDYO, W. D.; MOSER, T.; WINKLER, D.; BIFFL, S. Foundations for event-based process analysis in heterogeneous software engineering environments. In: **2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications**. [S.l.: s.n.], 2010. p. 313–322. ISSN 2376-9505.

SUZUKI, J.; YAMAMOTO, Y. Leveraging distributed software development. **Computer**, v. 32, n. 9, p. 59–65, Sept 1999. ISSN 0018-9162.

SWIGGER, K.; ALPASLAN, F.; BRAZILE, R.; MONTICINO, M. Effects of culture on computer-supported international collaborations. **International Journal of Human-Computer Studies**, v. 60, n. 3, p. 365 – 380, 2004. ISSN 1071-5819. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1071581903001800>. Acesso em : 20 de julho de 2018

TAKAHASHI, K.; POTTS, C.; KUMAR, V.; OTA, K.; SMITH, J. D. Hypermedia support for collaboration in requirements analysis. In: **Proceedings of the Second International Conference on Requirements Engineering**. [S.l.: s.n.], 1996. p. 31–40.

TELLIOĞLU, H.; WAGNER, I. Negotiating boundaries. configuration management in software development teams. **Computer Supported Cooperative Work (CSCW)**, Springer, v. 6, n. 4, p. 251–274, 1997.

THUM, C.; SCHWIND, M. Synchronite - a service for real-time lightweight collaboration. In: **2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing**. [S.l.: s.n.], 2010. p. 215–221.

TIWANA, A. Impact of classes of development coordination tools on software development performance: A multinational empirical study. **ACM Trans. Softw. Eng. Methodol.**, ACM, New York, NY, USA, v. 17, n. 2, p. 11:1–11:47, maio 2008.

TRAINER, E. H.; REDMILES, D. F. Bridging the gap between awareness and trust in globally distributed software teams. **Journal of Systems and Software**, v. 144, p. 328 – 341, 2018. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121218301250>. Acesso em : 20 de julho de 2018

TREUDE, C.; STOREY, M. How tagging helps bridge the gap between social and technical aspects in software development. In: **2009 IEEE 31st International Conference on Software Engineering**. [S.l.: s.n.], 2009. p. 12–22. ISSN 0270-5257.

TREUDE, C.; STOREY, M. Work item tagging: Communicating concerns in collaborative software development. **IEEE Transactions on Software Engineering**, v. 38, n. 1, p. 19–34, Jan 2012. ISSN 0098-5589.

VALETTO, G.; KAISER, G. E. Enveloping sophisticated tools into process-centered environments. In: **Computer Aided Software Engineering**. [S.l.]: Springer, 1996. p. 125–161.

VIVACQUA, A. S.; GARCIA, A. C. B. *et al.* **Ontologia de colaboração**. [S.l.]: Campus, 2011. 34–49 p.

VLIETLAND, J.; SOLINGEN, R. van; VLIET, H. van. Aligning codependent scrum teams to enable fast business value delivery: A governance framework and set of intervention actions. **Journal of Systems and Software**, v. 113, p. 418 – 429, 2016. ISSN 0164-1212. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0164121215002435>. Acesso em : 20 de julho de 2018

WALTER, C.; RILEY, I.; GAMBLE, R. Configuring an appropriate team environment to satisfy relevant criteria. In: **2016 IEEE Frontiers in Education Conference (FIE)**. [S.l.: s.n.], 2016. p. 1–9.

WANG, W.; HAAKE, J. M.; WESSNER, M. Supporting cooperative learning in distributed project teams. In: SPRINGER. **OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"**. [S.l.], 2002. p. 266–285.

WATABE, K.; SAKATA, S.; MAENO, K.; FUKUOKA, H.; OHMORI, T. Distributed desktop conferencing system with multiuser multimedia interface. **IEEE Journal on Selected Areas in Communications**, v. 9, n. 4, p. 531–539, May 1991. ISSN 0733-8716.

WEERD, I. V. D.; BRINKKEMPER, S.; NIEUWENHUIS, R.; VERSENDAAL, J.; BIJLSMA, L. On the creation of a reference framework for software product management: Validation and tool support. In: **2006 International Workshop on Software Product Management (IWSPM'06 - RE'06 Workshop)**. [S.l.: s.n.], 2006. p. 3–12.

- WHITEHEAD, J. Collaboration in software engineering: A roadmap. In: IEEE COMPUTER SOCIETY. **2007 Future of Software Engineering**. [S.l.], 2007. p. 214–225.
- WHITEHEAD, J.; MISTRÍK, I.; GRUNDY, J.; HOEK, A. Van der. Collaborative software engineering: concepts and techniques. In: **Collaborative Software Engineering**. [S.l.]: Springer, 2010. p. 1–30.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering: an introduction**. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN 0-7923-8682-5.
- WU, J.; GRAHAM, T. C. N.; SMITH, P. W. A study of collaboration in software design. In: **2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings**. [S.l.: s.n.], 2003. p. 304–313.
- WU, S. Overview of communication in global software development process. In: **Proceedings of 2012 IEEE International Conference on Service Operations and Logistics, and Informatics**. [S.l.: s.n.], 2012. p. 474–478.
- XU, D.; KUROGI, J.; OHGAME, Y.; HAZEYAMA, A. Distributed collaborative modeling support system associating uml diagrams with chat messages. In: **2009 33rd Annual IEEE International Computer Software and Applications Conference**. [S.l.: s.n.], 2009. v. 1, p. 367–372. ISSN 0730-3157.
- YAGÜE, A.; GARBAJOSA, J.; DÍAZ, J.; GONZÁLEZ, E. An exploratory study in communication in agile global software development. **Computer Standards & Interfaces**, v. 48, p. 184 – 197, 2016. ISSN 0920-5489. Special Issue on Information System in Distributed Environment. Disponível em:
<http://www.sciencedirect.com/science/article/pii/S0920548916300381>. Acesso em : 20 de julho de 2018
- YE, Y.; NAKAKOJI, K.; YAMAMOTO, Y. Reducing the cost of communication and coordination in distributed software development. In: MEYER, B.; JOSEPH, M. (Ed.). **Software Engineering Approaches for Offshore and Outsourced Development**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 152–169. ISBN 978-3-540-75542-5.
- YILDIZ, B. M.; TEKINERDOGAN, B. Architectural viewpoints for global software development. In: **2011 IEEE Sixth International Conference on Global Software Engineering Workshop**. [S.l.: s.n.], 2011. p. 9–16. ISSN 2329-6305.
- YU, J.; REDDY, Y. V. R.; BHARADWAJ, V.; REDDY, S.; KANKANAHALLI, S. Workflow-centric distributed collaboration in heterogeneous computing environments. In: SHEN, W.-m.; CHAO, K.-M.; LIN, Z.; BARTHÈS, J.-P. A.; JAMES, A. (Ed.). **Computer Supported Cooperative Work in Design II**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 504–515. ISBN 978-3-540-32970-1.