

# Explorando o Conhecimento em Equipe na Engenharia de Software: Um Estudo de Caso Múltiplo

## Exploring Team Knowledge in Software Engineering: A Multiple Case Study

Danilo Monteiro Ribeiro<sup>1</sup>, Bruno Cartaxo<sup>2</sup> e Fabio Q. B. da Silva<sup>3</sup>.

<sup>1</sup>UNINABUCO – Olinda, Pernambuco.

<sup>13</sup>Universidade Federal de Pernambuco - Recife, Pernambuco.

<sup>2</sup>Instituto Federal de Pernambuco – Olinda, Pernambuco

{dmr,fabio}@cin.ufpe.br

brunocartaxo@gmail.com

**Abstract.** *Context:* Teamwork involves cognitive processes also called Team Knowledge (TK). However, the understanding of TK at software development context is still limited. **Objective:** Therefore, this work seeks to investigate the impacts of the different types of team knowledge in Software Engineering and which team knowledge is promoted by practices like ownership of shared code and daily meeting. **Methodology:** 30 semi-structured interviews and observations were carried out. Data were analyzed using grounded theory techniques. **Results:** In the context investigated, were found 13 factors affected by team knowledge and what are the knowledge in teams promoted by shared code ownership (4 TK) and daily meeting (5 TK).

**Keywords.** Teamwork, Team Knowledge, Software team.

**Resumo.** *Contexto:* O trabalho em equipe envolve processos cognitivos também chamados de conhecimento em equipe (CE). Contudo, o entendimento do CE em desenvolvimento de software ainda é limitado. **Objetivo:** Portanto, este trabalho busca investigar os impactos dos diversos tipos de conhecimento em equipe na Engenharia de Software e entender quais conhecimentos em equipe são promovidos por práticas como a posse de código compartilhada e a reunião diária. **Metodologia:** Foram realizadas 30 entrevistas semi-estruturadas e observações. Os dados foram analisados utilizando técnicas de teoria fundamentada. **Resultados:** Dentro do contexto investigado, foram encontrados 13 fatores que são afetados pelo o conhecimento em equipe e quais

*são os conhecimentos em equipes promovidos pela posse de código compartilhada (4 CE) e pela reunião diária (5 CE).*

*Palavras-Chave.* Trabalho em equipe, conhecimento em equipe e equipes de software.

## 1. Introdução

A atividade de desenvolvimento de software enfrenta crescentes desafios visando a diminuição de custo, esforço e um maior controle do cronograma. Isto faz com que os gestores e pesquisadores busquem alternativas para melhorar o processo de desenvolvimento. Uma dessas alternativas é o desenvolvimento ágil de software. O Manifesto Ágil afirma que devemos valorizar mais os indivíduos e interações do que o processo e as ferramentas (AGILE MANIFEST 2001). Ou seja, deve-se priorizar os aspectos humanos e sociais envolvidos no trabalho em equipe na Engenharia de Software.

Dentro do trabalho em equipe existem processos que combinam atividades cognitivas e comportamentais que surgem a partir das interações dos membros da equipe e que podem gerar melhores resultados (KOZLOWSKI e ILGEN 2006). Logo, uma das alternativas para conseguir um melhor desempenho das equipes no desenvolvimento de software passa também por um melhor entendimento das atividades cognitivas que acontecem no nível das equipes e dos indivíduos.

Na literatura, existe o constructo de cognição da equipe que pode ser definido como uma atividade cognitiva que ocorre no nível da equipe (COOKE *et al.* 2007). O principal subconjunto do estudo de cognição em equipe é o conhecimento em equipe (WILDMAN *et al.* 2012).

O conhecimento em equipe pode ser definido como estruturas de conhecimento emergentes que são desenvolvidas como resultado das interações das cognições de cada membro de uma equipe (KLIMOSKY e MOHAMMED, 1994 e COOKE *et al.* 2004). O conhecimento em equipe é largamente estudado com diversos resultados e enfoques em áreas como Psicologia, Administração e Militar (WILDMAN SALAS e SCOOT, 2014 e WILDMAN *et al.* 2012). Esses trabalhos visam conectar o conhecimento em equipe com outros constructos como: desempenho (COOKE *et al.* 2003, CHOI *et al.* 2010), criatividade (SUNG e CHOI 2012), motivação (AUBE *et al.* 2015), conflito (SANTOS *et al.* 2015) e aprendizado (HAAR *et al.* 2015).

A ausência ou a má utilização dos fatores relacionados ao conhecimento em equipe podem causar problemas nas equipes de desenvolvimento de software (FAEGRI *et al.* 2016). Por exemplo, Stray *et al.* (2016) observaram que quando as normas de codificação da equipe não estão claras entre os membros, os membros podem tomar decisões equivocadas, afetando assim o desempenho da equipe.

Embora haja estudos que busquem entender melhor os conhecimentos em equipe na Engenharia de Software, ainda há desafios em aberto. Por exemplo, o estudo de Faegri *et al.* (2016) propõe um framework teórico, a partir do framework de Wildman *et al.* (2012), para o conhecimento em equipe no contexto de desenvolvimento de software distribuído.

Além disso, Faegri *et al.* (2016) alegam que criar e manter o conhecimento em equipe tem um custo e que se deve priorizar quais conhecimentos em equipe serão mantidos em uma equipe de desenvolvimento de software. Moe *et al.* (2016) também observaram essa necessidade e apresentaram um conjunto de lições aprendidas para equipes distribuídas. Todavia, essas lições surgem dentro de um contexto específico e,

ainda assim, os autores afirmam que são necessários mais estudos para entender melhor o problema, inclusive em outros contextos.

Outro ponto importante é que existem práticas de desenvolvimento de software que são utilizadas no dia-a-dia das empresas como a posse compartilhada de código (RIBEIRO et al. 2016) e a reunião diária (STRAY et al. 2016), contudo, apesar de existirem estudos que encontram alguma relação dessas práticas com conhecimento em equipe, eles não buscam primariamente entender essas relações (SEDANO et al. 2016 e STRAY et al. 2016). Dessa forma, não se sabe realmente quais práticas devem ser utilizadas para promover determinados tipos de conhecimento.

Ou seja, apesar de existirem esforços que buscam entender melhor como o conhecimento em equipe pode ser promovido e qual é sua influência na área de desenvolvimento de software, ainda são necessários mais trabalhos que busquem compreender como isso ocorre na prática. Nesse contexto, este estudo visa responder às seguintes Perguntas de Pesquisa (PP):

*PP1: Quais conhecimentos em equipe são promovidos pelas práticas de desenvolvimento de software?*

*PP1.1 Quais conhecimentos em equipe são promovidos pela posse compartilhada de código?*

*PP1.2 Quais conhecimentos em equipe são promovidos pela reunião diária?*

*PP2 Como os diversos tipos de conhecimento em equipe impactam no desenvolvimento de software?*

Para responder às perguntas de pesquisa, conduzimos estudos de caso, coletando dados de observações e entrevistas semi-estruturadas. Conduzimos estudos em 5 empresas, totalizando 7 equipes de desenvolvimento de software e 30 pessoas. Como principais resultados, foram encontrados:

- ***Quais conhecimentos em equipe a posse compartilhada de código ajuda a promover*** (Modelo Mental, estratégia da tarefa, Conhecimento da tarefa e localização de expertise);
- ***Quais conhecimentos em equipe a reunião diária ajuda a promover*** (Visão compartilhada, localização de expertise, Modelo Situacional, estratégia da tarefa e conhecimento da tarefa);
- ***Como o conhecimento em equipe impacta no processo de desenvolvimento de software*** (quais são as relações positivas e negativas dos conhecimentos em equipe propostos por Faegri et al. (2016) na prática).
- ***A adição de um novo conhecimento em equipe, modelo situacional, ao framework de Faegri et al. (2016).***

O restante do trabalho está dividido em seção 2 – referencial teórico, que tem informações sobre o conhecimento em equipe e trabalhos relacionados; seção 3 – o detalhamento do método aplicado nesta pesquisa; seção 4 – resultados obtidos com este trabalho; seção 5 – discussão; e seção 6 – conclusão.

## 2. Referencial Teórico

Esta seção é dividida em explicações gerais sobre o conhecimento em equipe proveniente de outras áreas do conhecimento, resultados encontrados na área de computação e trabalhos relacionados.

### 2.1. Conhecimento em equipe

De acordo com Cooke *et al.* (2007), a cognição da equipe pode ser definida como os modelos mentais compartilhados por um grupo de indivíduos, permitindo-os assim, realizar as tarefas como uma unidade coordenada. Wildman et al. (2012) afirma que a cognição em equipe tem dois enfoques, o conhecimento em equipe e o enfoque de processo. No enfoque de conhecimento em equipe, a cognição da equipe não é meramente a soma do conhecimento dos membros individuais da equipe, mas sim uma estrutura de conhecimento emergente que resulta da interação (por exemplo, sobreposição, dispersão, complementaridade) das cognições individuais de cada membro da equipe. (COOKE et al., 2004)

Em uma outra definição, Wildman et al. (2014) comentam que o conhecimento em equipe pode ser representado por estruturas de conhecimento em nível de equipe relativamente estáveis e emergentes que existem nas cabeças dos membros da equipe e que juntas se combinam para representar a cognição da equipe.

Portanto, nesta pesquisa o constructo conhecimento em equipe é um dos enfoques da cognição em equipe. O conhecimento em equipe engloba outros constructos que envolvem o compartilhamento de modelos mentais por ao menos dois ou mais membros dentro de uma equipe, a partir de suas interações. Vale ressaltar que esses modelos podem ser inerentes aos indivíduos, no entanto, um indivíduo pode entender o modelo do outro indivíduo, e assim agir de acordo com as perspectivas que outro tem sobre como a tarefa deve ser feita, ou como a comunicação deve ser conduzida, por exemplo.

Em uma revisão sistemática da literatura, Wildman et al. (2012) dividem os modelos mentais compartilhados que formam o conhecimento em equipe em duas dimensões: estática e dinâmica.

Na divisão estática ficam os constructos que são estáveis ao longo de uma quantidade de tempo. Enquanto que na divisão dinâmica ficam os constructos que mudam ou evoluem rapidamente. Os autores não especificam a quantidade de tempo, mas exemplificam: o conhecimento da tarefa (o entendimento compartilhado sobre o conteúdo da tarefa) deve ficar estável durante toda a tarefa, no entanto, o modelo situacional (o entendimento coletivo de situações específicas) deveria mudar constantemente durante o trabalho. Mohammed, Hamilton e Lim (2009) comentam que distinguir entre estático e dinâmico é importante principalmente para realização de pesquisas, pois a mensuração das estruturas de conhecimento dinâmicas necessita ser realizada de forma mais longitudinal do que a mensuração das estruturas de conhecimento estáticas. A Figura 1 apresenta este framework.

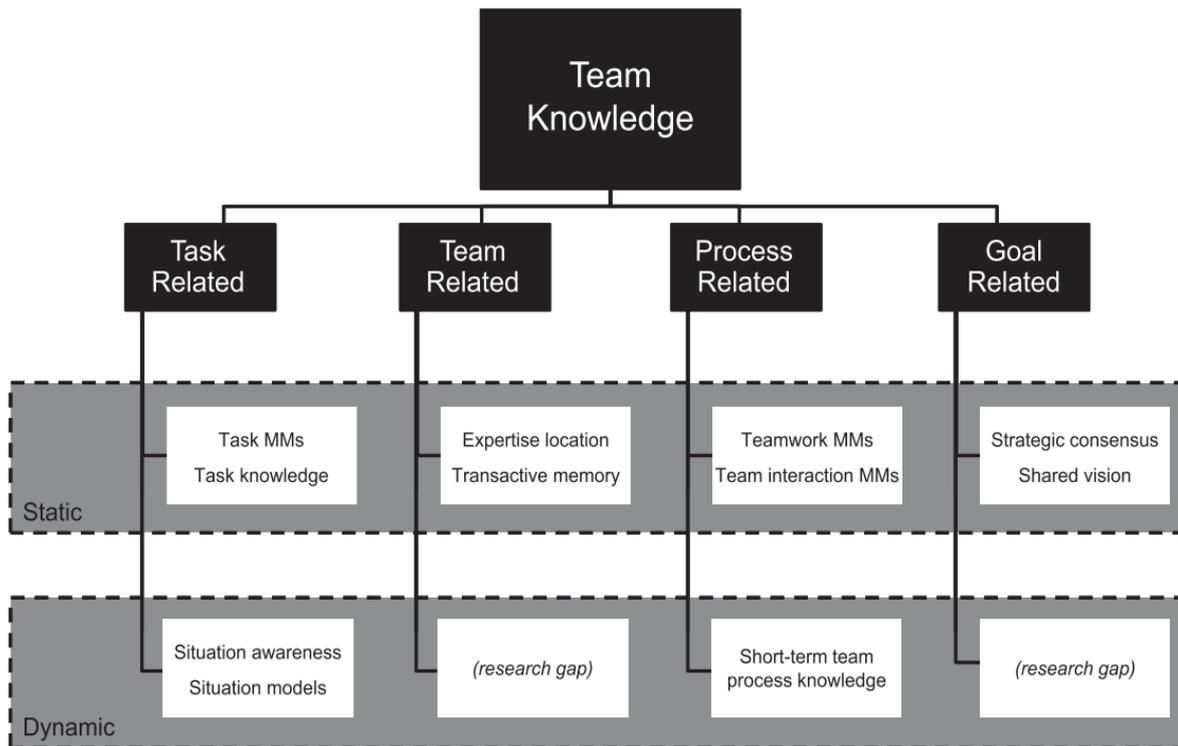


Figura 1. Framework proposto por Wildman et al. (2012)

Fonte: Wildman et al. (2012)

Wildman *et al.* (2012) ainda dividem o conhecimento em equipe em quatro categorias: (i) *conhecimento relacionado às tarefas*, (ii) *conhecimento relacionado ao time*, (iii) *conhecimento relacionado ao processo* e (iv) *conhecimento relacionado ao objetivo*.

O conhecimento da equipe *relacionado à tarefa* (i) é uma representação focada no conhecimento sobre como a tarefa está sendo executada pela equipe. A divisão estática é formada por dois constructos: Modelos Mentais da Tarefa e Conhecimento da Tarefa. De acordo com Mathieu *et al.* (2000), o constructo de **modelos mentais da tarefa** pode ser definido como o entendimento compartilhado sobre quais estratégias devem ser utilizadas para que realização da tarefa tenha sucesso, por exemplo, como o trabalho é alocado aos membros ou dividir a tarefa em subgrupos. Mathieu *et al.* (2000) também comentam que quanto mais imprevisíveis as tarefas forem, mais crucial será o efeito do constructo de modelos mentais da tarefa, pois ele permite aos membros discutirem sobre quais os procedimentos necessários para realizar a tarefa, possíveis cenários de contingência e estratégias para realizar a tarefa.

Já o **conhecimento da tarefa** para Mathieu *et al.* (2008) é o entendimento necessário para realizar a tarefa, por exemplo, o entendimento compartilhado entre os membros da equipe sobre como as partes de uma determinada tarefa interagem. Mathieu *et al.* (2008) asseguram que o conhecimento da tarefa é ainda mais importante para as equipes que realizam uma tarefa constantemente, pois além da equipe desempenhar melhor, também é capaz de definir as metas e planejar ações futuras mais adequadamente.

A divisão dinâmica do conhecimento em equipe relacionada à tarefa é dividida em consciência da situação e modelo situacional. A **consciência da situação** é a percepção dos elementos do ambiente, a compreensão do seu significado e uma projeção

de seus *status* em um futuro próximo (ENDSLEY,1995). Wildman, Salas e Scoot (2014) comentam que a consciência da situação representa uma compreensão dinâmica de determinada situação que pode mudar rapidamente a partir das alterações no ambiente. Na revisão feita por esses autores o constructo de consciência da situação foi o terceiro mais estudado, com 27 artigos dos 174 artigos totais. A consciência da situação é crucial nos sistemas táticos e estratégicos militares e na área da aviação onde o piloto tem que reconhecer parâmetros operacionais da aeronave, condições externas, informações de navegação, outras aeronaves e fatores hostis para executar suas ações (ENDSLEY,1995).

O **modelo situacional** pode ser definido como o entendimento coletivo do que está acontecendo em uma situação específica (COOKE, KIEKEL, e HELM, 2001). De acordo com Wildman *et al.* (2012), os modelos situacionais são representações mentais dinâmicas da equipe sobre todas as outras categorias propostas no *framework*. Ou seja, o entendimento da equipe em tempo real do que está acontecendo na equipe em todas as estruturas propostas pelo *framework* de Wildman *et al.* (2012).

O *conhecimento relacionado ao time ii*) pode ser definido como as estruturas mentais sobre as características e qualidades de seus companheiros de equipe ou da equipe como uma entidade social holística (WILDMAN *et al.* 2012). Esta categoria é apresentada com constructos apenas estáticos: Localização de expertise e memória transacional.

A **localização de expertise** é quando os membros têm o entendimento sobre quem é o especialista em determinado tópico de conhecimento em uma equipe ou quem fez determinada tarefa na equipe, ou seja, a compreensão de "QUEM sabe O QUÊ no time" (WILDMAN *et al.* 2012). Um dos estudos que abordam esse constructo é o de Kanawattanachai e Yoo (2007) que aponta para a importância da localização de expertise para o desempenho de equipe virtuais. Um exemplo desse constructo, é o fato de os membros saberem quem é melhor para realizar determinada tarefa, ou mesmo quem gosta de realizar a tarefa.

Ren e Agorte (2011) afirmam que **memória transacional** pode ser definida como um sistema mental compartilhado entre os membros da equipe para codificar, armazenar e recuperar informações sobre diferentes domínios. Para eles, o constructo tem dois componentes principais: (1) o conhecimento estruturado que pertence a um indivíduo de um grupo, e (2) os processos que ocorrem para codificar, armazenar e recuperar esse conhecimento. Por exemplo, pessoas que foram treinadas juntas conseguem memorizar mais palavras que pessoas que não foram treinadas juntas. De acordo com Hollingshead (2001) esses membros sabem como melhor lembrar uns aos outros do conhecimento que têm, também têm uma boa concepção do conhecimento do outro, evitam assim memorizar palavras dentro do domínio do parceiro.

Vale salientar também que de acordo com Wildman, Salas e Scoot (2014) a pesquisa que envolve o constructo de memória transacional concentra-se principalmente em avaliar dispersão e especialização do conhecimento dos indivíduos na equipe.

A categoria do *conhecimento de equipe relacionado ao processo (iii)* refere-se ao trabalho em equipe e processos interpessoais envolvidos em suas interações como a comunicação, coordenação e liderança. A divisão estática é dividida em dois constructos: Modelos mentais de interação e Modelos mentais do trabalho em equipe.

Os **Modelos mentais de interação** ocorrem quando a equipe compartilha modelos de interação, comunicação e liderança. Esses entendimentos podem ser sobre os papéis e responsabilidades dos membros da equipe, padrões de interação, fluxo de informação e canais de comunicação (MATHIEU *et al.* 2000). Por exemplo, a equipe pode ter o

entendimento compartilhado que a brevidade e a clareza são aspectos positivos da comunicação efetiva e que isso contribui para o sucesso da equipe (LIM e KLEIN, 2006).

O outro constructo é **modelos mentais do trabalho em equipe** que foi definido por Lim e Klein (2006) como um entendimento individual dos componentes do trabalho em equipe, como as tarefas da equipe, preferências dos membros, e os processos, bem como as relações entre esses componentes. Wildman, Salas e Scoot (2014) afirmam que o constructo de modelos mentais do trabalho em equipe é geralmente operacionalizado em termos de semelhança de modelos mentais entre os membros da equipe. Um exemplo desse constructo é quando os membros da equipe reconhecem como os outros membros conseguem aprender melhor, então os abordam utilizando de suas preferências.

Na divisão dinâmica, o **conhecimento dos processos de equipe a curto prazo** pode ser operacionalizado em como os membros da equipe são conscientes dos processos de curto prazo que estão ocorrendo dentro da equipe. Ou seja, o quanto os membros da equipe têm de percepção dos processos que estão ocorrendo dentro da equipe.

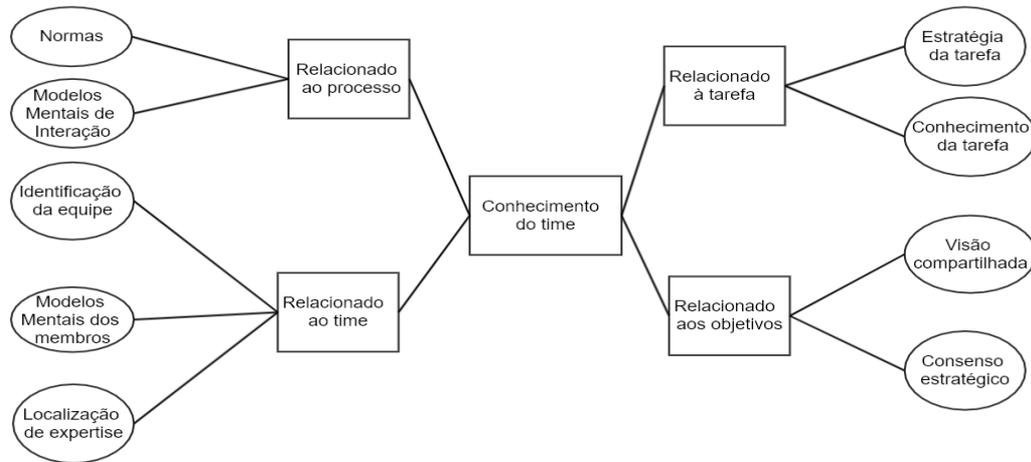
Finalmente, o *conhecimento em equipe relacionado aos objetivos* refere-se às construções que capturam a estrutura das representações mentais quanto às metas e objetivos da equipe, além de suas representações mentais quanto à realização desses objetivos. Ou seja, quais são os objetivos, visões e acordos gerais que devem ter seus entendimentos compartilhados em equipe.

Um dos constructos envolvidos nessa categoria é o **Consenso estratégico**. Wildman *et al.* (2012) definem o consenso estratégico como quando os membros da equipe compartilham uma compreensão da visão estratégica que eles, ou a organização visam atingir. Por exemplo, o fato do 12 s membros entenderem quais são os objetivos da organização e como ele pretende chegar neles.

Já a **visão compartilhada** é definida por Pearce e Ensley (2004), como um modelo mental compartilhado pela equipe de um futuro estado da equipe ou de suas tarefas que fornece uma base para as tomadas de decisões da equipe. Cohen (2008) e Gutiérrez Lloréns-Montes e Bustinza Sánchez (2009) observaram os benefícios da visão compartilhada entre os membros da equipe. Esses autores constataram o efeito positivo da visão compartilhada no comprometimento no trabalho e no desempenho organizacional.

## 2.2. Conhecimento em equipe na Engenharia de Software

O trabalho de Faegri et al. (2016) é baseado no Framework proposto por Wildman et al. (2012). O autor propõe uma adaptação do framework de Wildman para equipes de desenvolvimento de software distribuído. A adaptação proposta por Faegri et al. (2016) continua dividindo o framework em quatro categorias, no entanto, não contém as dimensões estática e dinâmica e seus constructos porque, de acordo com os autores, essas categorias são menos visíveis em equipes de desenvolvimento de software. Além disso, os autores sugerem subcategorias que são mais relevantes para serem pesquisadas dentro do contexto de equipes de desenvolvimento de software distribuídas. A Figura 2 apresenta o modelo e logo em seguida os constructos são apresentados.



**Figura 2. Framework proposto por Faegri et al. (2016)**

A categoria conhecimento em equipe relacionado à tarefa é dividida em dois conceitos (i) **Conhecimento das estratégias da tarefa** – é o modelo mental da tarefa do framework original. É o conhecimento de como a tarefa deve ser dividida e realizada pelos membros da equipe. Canfora et al. (2007) apresenta um exemplo de estudo sobre esse conceito na Engenharia de Software, onde os desenvolvedores que discutiam as estratégias encontraram uma solução de melhor qualidade do que os desenvolvedores que participaram individualmente.

(ii) **Conhecimento da tarefa**, utilizado no framework original de Wildman et al (2012). É o conhecimento que os membros da equipe têm compartilhado do que deve ser empregado na realização da tarefa. Espinosa et al. (2002) realizaram um trabalho onde eles encontraram que o conhecimento da tarefa pode afetar a coordenação de equipes de desenvolvimento de software.

O *conhecimento relacionado ao time* foi dividido em três constructos:

O primeiro constructo, (i) **Identificação da equipe**, onde os membros compartilham a sensação de pertencimento à equipe. Por exemplo: quando os membros se sentem bem ao trabalhar na equipe. Na literatura de Engenharia de Software, Marks e Lockyer (2005) afirmam que a identificação da equipe afeta positivamente a satisfação dos membros com o trabalho.

O segundo constructo, (ii) **localização de expertise** se assemelha ao framework original. É quando os membros sabem quem é o especialista naquela determinada linguagem ou método, por exemplo. He, Blutle e King (2007) afirmam que a localização de expertise ajuda no desempenho das equipes de desenvolvimento de software e Lin et al. (2015) afirmam que a localização de expertise pode ajudar as equipes de software a terem maior competência na resolução de problemas.

Existe também o constructo de (iii) **modelo mental dos membros** no qual os membros de uma equipe compartilham de estruturas de conhecimento que permitem formar explicações e expectativas precisas sobre a tarefa e, por sua vez, coordenar suas ações. Levesque et al. (2001) afirmam que em equipes de software temporárias onde os membros não esperam trabalhar juntos novamente, os membros da equipe têm dificuldades de manter modelos mentais dos membros.

O *conhecimento de equipe relacionado ao processo* é dividido em dois constructos:

(i) **Modelos mentais de interação**, semelhante ao constructo de nome igual do framework original de Wildman et al. (2012), onde os membros sabem como os outros preferem interagir. Bass (2014) afirma que algumas vezes em equipes distribuídas os membros adotam padrões que eles acham mais viáveis para o trabalho. No trabalho de Bass (2014) foi observado que os membros da equipe preferiam interagir com outros membros individualmente, do que realizar reuniões diárias coletivas.

(ii) **Normas**, que são códigos de conduta aceitos pelos membros da equipe e podem melhorar a eficácia da mesma. Essas normas não precisam ser necessariamente explícitas, podem ser criadas de maneira informal pelos os membros da equipe (FAEGRI et al 2016). Teh et al. (2012) conduziram um estudo em que eles observaram que as normas de grupos são essenciais para que as equipes desenvolvam as tarefas de maneira otimizada.

Por fim, o *conhecimento em equipe relacionado aos objetivos* é dividido em dois:

(i) **Visão compartilhada**, onde a equipe compartilha quais são seus objetivos, as expectativas, as métricas e como atingi-los. Crowston e Kammerer (1998) afirmam que quando a equipe tem objetivos e métricas compartilhadas, ela trabalha de maneira mais coordenada, o que pode levar a um maior desempenho. Preston e Karahanna (2009) sugerem que é importante que equipes de desenvolvimento de software tenham uma visão compartilhada para que a organização possa definir melhor suas estratégias.

(ii) **Consenso estratégico**, semelhante ao constructo do framework original de Wildman et al. (2012), onde a equipe chega à um acordo sobre metas estratégicas para a organização. Faegri et al. (2016) não achou nenhum exemplo de estudos dentro desta categoria. Na nossa revisão da literatura, encontramos o trabalho de Knight et al. (1999), que investigou em empresas de alta tecnologia, dentre elas empresas de desenvolvimento de software, e investigou como fatores demográficos e os processos do grupo influenciam o consenso estratégico da organização.

A Tabela 1. apresenta as diferenças e semelhanças entre os frameworks de Wildman et al. (2012) e Faegri et al. (2016).

**Tabela 1. Resumo das diferenças entre Faegri e Wildman.**

Constructo	Wildman et al.	Faegri et al.
Dimensões estática e dinâmica	Ambas	Sem distinção
Modelos mentais da tarefa (conhecimento das estratégias da tarefa)	X	X
Conhecimento da tarefa	X	X
Percepção da situação	X	
Modelo situacional	X	
Localização de expertise	X	X

Memória transitiva	X	
Modelos mentais do trabalho em equipe	X	X
Modelo mentais da interação da equipe	X	X
Conhecimento dos processos de equipe a curto prazo	X	
Consenso estratégico	X	X
Visão compartilhada	X	X
Normas de grupo		X
Identificação da equipe		X

### 2.3. Trabalhos relacionados

He, Butler e King (2007) realizaram uma pesquisa buscando entender como o conhecimento em equipe é desenvolvido e evoluído em equipes de desenvolvimento de software. No entanto, o conhecimento em equipe foi operacionalizado somente com os constructos de localização de expertise e o conhecimento da tarefa compartilhado. Eles encontraram que o efeito inicial da familiaridade da equipe sobre a localização de expertise e o conhecimento da tarefa decresce durante o tempo. Além disso, a localização de expertise e o conhecimento da tarefa são afetados positivamente pela familiaridade da equipe e pela comunicação frequente.

Este trabalho se diferencia do trabalho de He, Butler e King (2007), pois além de buscar observar todos os principais conhecimentos em equipes propostos Faegri et al. (2016) e por Wildman et al. (2012), busca relacionar as práticas de desenvolvimento de software (posse de código compartilhada e reunião diária) com o surgimento dos conhecimentos em equipe.

Moe *et al.* (2016) realizaram outro estudo baseado na adaptação do framework proposto por Faegri *et al.* (2016), em que apresentavam um conjunto de lições aprendidas sobre como uma equipe distribuída de desenvolvimento de software tentava promover o conhecimento em equipe. Durante o estudo, a equipe de desenvolvimento de software ficou reunida durante cinco dias, brevemente descritos abaixo.

Os primeiros dois dias foram destinados a um fórum de usuários, onde sete dos membros da equipe conheceram clientes e executaram demonstrações do produto. No terceiro dia, os membros que participaram do fórum apresentaram e discutiram o que aconteceu no fórum de usuários para toda equipe. Além disso, os membros das equipes de diferentes localidades realizaram entrevistas entre si para entender o que os outros membros fazem fora do trabalho.

No quarto dia, o *product owner* apresentou sua perspectiva sobre o futuro do produto seguido de uma sessão sobre perguntas e respostas dos membros da equipe. Ainda no quarto dia, os grupos identificaram tópicos que precisavam discutir entre si e implementaram um roteiro para isso. Por fim, os participantes em equipes fizeram caminhadas nas montanhas norueguesas sob condições adversas.

Foram ainda realizadas reuniões que falam do *roadmap* do produto para melhorar o conhecimento em equipe relacionado à tarefa. Para melhorar o conhecimento em equipe em relação aos processos, foram criados e discutidos tópicos (dia 4) que os membros achavam relevantes sobre o tema. Para melhorar o conhecimento em equipe relacionado ao time, os membros que participaram da reunião com os clientes apresentaram para toda a equipe o que foi comentado pelos clientes.

Além disso, os membros realizaram pequenas entrevistas entre diferentes equipes buscando entender quais eram os hobbies e interesses dos outros membros, o que é mais divertido no seu trabalho e em que a pessoa era boa no trabalho.

Quanto ao conhecimento em equipe relacionado aos objetivos, foi utilizada a estratégia de *roadmap* comentada anteriormente. Assim como a apresentação de pontos que chamaram a atenção dos membros que participaram das reuniões com os clientes.

Este trabalho vem complementar o estudo de Moe et al. (2016) que apresentam um conjunto de práticas que promovem o conhecimento em equipe de desenvolvimento de software distribuídas. O trabalho de Moe et al. (2016) apresenta um conjunto de possíveis ações que podem ser utilizadas pelas equipes para ter o conhecimento em equipe. Neste trabalho, duas práticas de desenvolvimento de software são investigadas e relacionadas especificamente aos conhecimentos promovidos.

Dentro do contexto dos trabalhos acima citados, a pesquisa que apresentamos neste artigo também vem complementar o trabalho de Faegri et al. (2016). Este trabalho apresenta como o *framework* teórico proposto por Faegri et al. (2016) pode impactar no desenvolvimento de software. Ademais, este trabalho busca verificar se os constructos que não foram levados em consideração por Faegri et al. (2016) do trabalho de Wildman et al. (2012) são observados na prática de Engenharia de Software.

### 3. Método

Nesta seção são apresentados os procedimentos metodológicos utilizados nesta pesquisa. Esta pesquisa segue o posicionamento construtivista/interpretativista. De acordo com Easterbrook et al. (2012), o paradigma construtivista rejeita a ideia de que o conhecimento científico pode ser separado do seu contexto.

#### 3.1. Planejamento do estudo de caso

Esta pesquisa procura entender o conhecimento em equipe na Engenharia de Software na prática. Visando atingir esse objetivo, foi escolhido como método de pesquisa o estudo de caso. De acordo com Merriam (2009), um estudo de caso pode ser definido como uma investigação empírica que busca entender um fenômeno contemporâneo dentro de seu real contexto.

Runeson e Host (2009) afirmam que o pesquisador deve deixar claro quais são seus objetivos e questões de pesquisa logo no planejamento do estudo. Os autores também afirmam que o objetivo do estudo de caso pode ser explorar, descrever ou explicar determinado fenômeno.

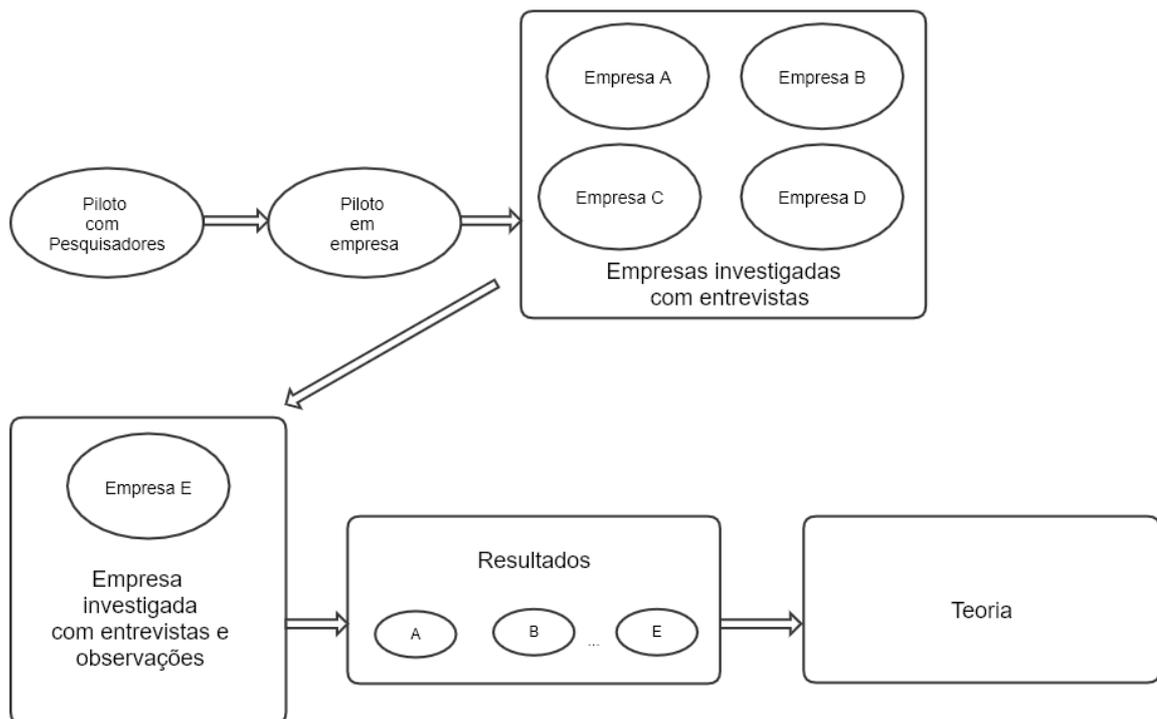
Nesta pesquisa, o **fenômeno** é o conhecimento em equipe e o **objetivo** é entender como o conhecimento em equipe impacta no desenvolvimento de software. Além disso, compreender quais conhecimentos em equipe as práticas de posse de código compartilhada e a reunião diária promovem.

Yin (2017) afirma que o caso pode ser praticamente qualquer coisa que seja um fenômeno contemporâneo em seu contexto real. Runeson e Host (2009) comentam também que na Engenharia de Software, o caso pode ser um projeto de desenvolvimento

de software, um indivíduo, um grupo de pessoas, um processo, um produto, etc. Neste trabalho, o **caso** é a equipe de desenvolvimento de software na prática industrial.

Na visão de Yin (2017), para desenvolver um estudo de caso também é necessário ter uma unidade de análise bem definida. Neste trabalho, busca-se entender o conhecimento em equipe, portanto, a **unidade de análise** são as equipes de desenvolvimento de software. De acordo com Yin (2017), o estudo de caso pode ser único ou múltiplo. Esta pesquisa realizou um **estudo de caso múltiplo holístico**, pois busca entender o mesmo caso em diferentes contextos. Yin (2017) comenta também que os estudos de caso múltiplos apresentam evidências mais convincentes e resultados gerais mais robustos, pois estudos de caso únicos podem ser limitados pelas condições de singularidade que envolvem o caso.

O desenho desta pesquisa é apresentado na Figura 3. Em um primeiro momento, foi realizado um piloto com pesquisadores (explicitado na seção 3.2) e posteriormente um piloto em uma empresa de desenvolvimento de software. Em seguida, foram investigadas entrevista em cinco equipes em quatro empresas. Foram realizadas observações e entrevistas em mais duas equipes em outra empresa. Os resultados foram analisados em conjunto e deram origem a uma teoria que surgiu a partir de técnicas da teoria fundamenta em dados.



**Figura 3** Processo utilizado na pesquisa.

### 3.2. Instrumentos de coleta de dados

Runeson e Host (2009) comentam que as principais decisões sobre os métodos de coleta de dados são definidas na fase de desenho do projeto, embora as decisões mais detalhadas sobre os procedimentos possam ser tomadas mais tarde. Esta pesquisa utiliza entrevistas, observações e análise de documentos como métodos de coletas de dados.

Para DeMarrais (2004), a entrevista é um processo no qual o pesquisador realiza uma conversa com os participantes da pesquisa para obter respostas para suas perguntas

de pesquisa. Nesta pesquisa as entrevistas utilizaram um roteiro semiestruturado. Os instrumentos de coleta de dados podem ser visualizados em <https://goo.gl/RT8PHw>.

Merriam (2009) comenta que ao usar um roteiro semiestruturado, o pesquisador desenvolve uma lista pré-definida de perguntas a serem exploradas, no entanto, a ordem das perguntas pode mudar e novas perguntas podem ser realizadas. Ela afirma também que este formato permite ao pesquisador responder à situação atual, entender a visão de mundo do entrevistado e ter novas ideias sobre o tema.

Os áudios das entrevistas foram gravados e armazenados de maneira confidencial, de forma que só o pesquisador responsável tem acesso às fontes originais. Todas as informações que possam identificar às empresas e os investigados são omitidas. Além disso, a qualquer momento durante a entrevista os entrevistados poderiam desistir de participar da pesquisa e não seriam penalizados por isso.

Adicionalmente, foram realizadas observações pelo pesquisador. Merriam (2009) afirma que existe duas vantagens em realizar observações: Primeiro, as observações ocorrem dentro do contexto em que o fenômeno de interesse acontece. Em segundo lugar, os dados provenientes das observações são dados colhidos durante o fenômeno, em vez de relatos dos entrevistados.

Guest *et al.* (2012) afirmam que observações permitem aos pesquisadores entenderem uma gama mais ampla de dados e que uma das principais razões para realizar observações é que muitos aspectos só são visíveis para quem está dentro do contexto. Os membros da equipe poderiam ficar receosos com uma pessoa desconhecida observando as realizações da tarefa da equipe (GUEST *et al.* 2012). Portanto, logo no início da observação, o pesquisador se apresentou e comentou um pouco sobre a pesquisa, por fim explicou que não estava avaliando o desempenho deles e que as informações eram sigilosas.

O observador acompanhou durante todo o expediente de trabalho durante um mês o time em suas reuniões e ficou no mesmo local de trabalho que os membros da equipe, no entanto, ele não desenvolveu software junto com a equipe nem opinou sobre nenhum aspecto técnico ou da equipe. Esse tipo de coleta de dados é chamado de *observação não participante* (MERRIAM 2009), onde a equipe sabe quais são as atividades do observador, o observador está junto da equipe, mas a sua participação é limitada a coletar os dados para a pesquisa. Com esse método, o pesquisador pode ter acesso uma grande quantidade de informações, mas o nível da informação revelada é controlado pelos membros do grupo que estão sendo investigados (MERRIAM et al. 2009).

Além das observações e das entrevistas, outros artefatos como quadros de tarefas e quadros das salas de reunião também foram analisados. O pesquisador identificou quais conhecimentos em equipes estão sendo transmitidos pelos artefatos. Vale salientar que o pesquisador não utilizou nem armazenou as informações coletadas sobre o projeto. Antes de cada entrevista o pesquisador informava para os entrevistados o teor da mesma e outras informações tais como: quem eram os pesquisadores responsáveis, pedia autorização formal para gravar e caso não fosse autorizado (ocorreu uma vez), o membro ficava durante um tempo na sala (para que os outros não percebessem) e era liberado.

### 3.3. Preparação para coleta

Nesta etapa, os roteiros de entrevista e de observação foram construídos. De acordo com Yin (2017), um estudo de caso piloto auxilia os pesquisadores na hora de aprimorar os planos para a coleta de dados. Este auxílio pode ser em relação ao conteúdo dos dados ou aos procedimentos que devem ser seguidos. De acordo com Merriam (2009),

os pilotos são cruciais para realizar testes no roteiro de entrevista, pois além de ajudar ao pesquisador a ter alguma prática na condução das entrevistas, também pode ajudar a descobrir questões que são confusas e que precisam ser reformuladas. Segundo Merriam (2009), é comum que os pesquisadores testem suas perguntas com alguém que entende do tópico em estudo. Yin (2017) afirma que os casos pilotos podem ser escolhidos por diversas razões como, por exemplo, participantes do caso piloto serem compatíveis com o que do caso que se deseja estudar.

Nesta pesquisa, o estudo piloto foi dividido em duas etapas. Na primeira etapa os três pesquisadores externos avaliaram os questionários. Todos os pesquisadores são da área de Engenharia de Software, com experiência tanto no desenvolvimento de software quanto no gerenciamento de equipes. Um dos pesquisadores é doutor e professor de uma instituição federal de ensino. Os outros dois são doutorandos e também, professores em instituições federais de ensino.

O objetivo nesta etapa era ter uma percepção inicial se o questionário explorava aspectos que levariam a responder as perguntas desta pesquisa, ter *feedbacks* sobre a estrutura do questionário e obter possíveis sugestões de melhoria. As perguntas eram realizadas e discutidas de forma individual com cada pesquisador. Com isso, ao final dessa etapa, o questionário sofreu alterações, foram retiradas e adicionadas novas perguntas, bem como também a ordem das perguntas foi alterada.

Com o novo questionário, foi realizado outro estudo piloto em uma empresa de desenvolvimento de software com uma equipe composta de três pessoas. Esta equipe adotava práticas de desenvolvimento ágeis, tais como reunião diária e posse compartilhada de código. Esse estudo piloto tem como objetivo testar o questionário na prática, bem como testar o roteiro de observação. Após esse piloto na empresa, tanto o roteiro de observação quanto as entrevistas foram alterados. Algumas perguntas do roteiro de entrevistas foram modificadas pois continham termos um pouco mais técnicos e alguns entrevistados tiveram dificuldade. Por exemplo, em alguns casos o termo “redundância de conhecimento em equipe” deixava os membros confusos. Além disso, o roteiro de observação era muito extenso e dificultava seu preenchimento, por exemplo: *Qual é o ambiente da ação? Como ela(reunião) está organizada? Como é o comportamento dos participantes? O líder utilizou alguma prática/ferramenta para que a equipe tenha redundância de conhecimento? Qual? Para qual tipo de conhecimento? Existe a necessidade de ter o conhecimento redundante? Qual?*

### 3.4. Análise de dados

De acordo com Merriam (2009), na pesquisa qualitativa, o pesquisador deve começar a analisar os dados logo no início de sua coleta. A partir da segunda entrevista o pesquisador busca entender as semelhanças e diferenças com a entrevista anterior, construindo assim sua teoria. Por isso, nesta pesquisa, ao final de cada entrevista o pesquisador responsável escrevia em seu caderno de anotações reflexões sobre ela.

Merriam (2009) também recomenda que pesquisadores escrevam comentários observacionais e criem *memos* sobre o que está sendo aprendido durante a coleta de dados. Isso nutre o pensamento crítico para que pesquisadores sejam mais que uma máquina de gravação. Por isso, ao final de cada dia o primeiro autor desta pesquisa se questionou sobre as observações feitas e escreveu *memos* em seu caderno de anotações.

Esta pesquisa utilizou na sua abordagem um processo indutivo e comparativo de codificação e agrupamento em categorias, seguindo as recomendações propostas por Merriam (2009) para análise de dados qualitativos. O método indutivo e comparativo é

baseado no método comparativo de análise de dados, proposto por Glaser e Strauss (1967) como meio para desenvolver a teoria fundamentada.

Merriam (2009) afirma que esse processo começa com a identificação de segmentos nas entrevistas que respondem às perguntas de pesquisa. Cada segmento é uma unidade de dados que potencialmente busca responder uma das perguntas de pesquisa. Merriam (2009) tem duas recomendações para construção desses segmentos. Primeiro, eles devem revelar informações relevantes para o estudo. Em segundo lugar, a unidade deve ser interpretável na ausência de informações adicionais. Segundo Merriam (2009), posteriormente, a tarefa é comparar uma unidade de informação com a próxima, em busca de regularidades recorrentes nos dados. O pesquisador deve criar categorias ou classes que reúnem esses segmentos novamente. Nesse processo, o pesquisador começa a discriminar mais claramente entre os critérios de alocação de dados para uma categoria ou outra. Como a comparação é constante, algumas categorias podem ser subdivididas e outras unidas em categorias mais abstratas.

Ademais, o roteiro de entrevista e observações foi construído a partir do *framework* apresentado por Faegri et al. (2016), por este ser uma proposta de adaptação para a área de Engenharia de Software. Mas vale salientar, que o pesquisador buscou também categorias presente no *framework* de Wildman et al (2012), e que uma delas surgiu nos dados (modelo situacional).

O processo de análise de dados foi conduzido pelo primeiro autor, sobre a supervisão dos outros autores. As entrevistas foram transcritas e inseridas no software MAXQDA para análise. Nesse processo, o primeiro autor procurava nas entrevistas trechos que respondessem às perguntas de pesquisas, e criava ou agrupava em sua respectiva categoria. A Figura 4 apresenta um exemplo dos resultados desse processo. Para construir o modelo apresentado na seção 4.2, o pesquisador utilizou das relações que apareciam em mais de um contexto. A Figura 4 apresenta um exemplo dos resultados desse processo (alguns dados precisaram ser omitidos).

Código: con.como.tarefa 24 codificações de 6 documentos e 4 de grupos de documentos

Tu consegue lembrar de situações que a equipe compartilhou um conteúdo de uma tarefa ?

Cara, isso rola muito...a galera sempre tá perguntado o que faz uma classe, como que chama tal método e isso é importante para que o cara não perca muito tempo, você vai ter que ler, entender e nem sempre isso é fácil, a gente sabe o que é o código, por mais que tenha padrão, isso não é fácil... saca? E as vezes você acaba fazendo

Comentário	Grupo de docu...	Nome do docu...	Código	Início	Fim	Resultado do p...	Pré-visualização
	empresa 3		con.como.tarefa	24	25	0	Tu consegue lem...
	Empresa 2		con.como.tarefa	9	10	0	Você consegue le...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 11	3 11	0	Programando em...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 13	3 13	0	está passa...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 15	3 15	0	A funcionária...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 16	3 16	0	necessito...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 17	3 17	0	precisou ...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 22	3 22	0	ve perg...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 23	3 23	0	Ligaram pela seg...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 26	3 26	0	passa con...
	observacoes emp...	Conhecimento na...	con.como.tarefa	3 27	3 27	0	Reuniao diaria, v...

**Figura 4 Exemplo de codificação**

### 3.5. Ameaças à validade

Merriam (2009) afirma que o pesquisador deve planejar formas para aumentar a credibilidade, confiabilidade e transferibilidade de uma pesquisa qualitativa. Além disso a autora afirma também que a credibilidade (também chamada de validade interna por Wholin et al. 2012) lida com a questão de como os resultados de pesquisa correspondem à realidade. Para isso, a autora alega que o pesquisador deve fazer constantemente as seguintes perguntas: Quão congruentes são os resultados com a realidade investigada? Os resultados captam o que realmente existe? Como forma de aumentar a credibilidade, o pesquisador retornou aos entrevistados para esclarecer eventuais dúvidas e apresentou os resultados finais para os participantes da última empresa investigada para perguntar se os resultados estão congruentes com a realidade que eles vivem. Ademais, os relatórios finais de cada empresa participante foi enviado para os responsáveis. Durante a observação, pesquisador também perguntava para os membros da equipe sobre alguma possível dúvida sobre o evento tinha acontecido.

Esta pesquisa seguiu as recomendações de Merriam (2009) quanto à confiabilidade (mesmo nome para Wohlin et al. 2012). O pesquisador deve explicar seus pressupostos, triangular dados, e fazer uma trilha de auditoria, descrevendo em detalhes como o estudo foi realizado e como os resultados foram derivados a partir dos dados (MERRIAM 2009). Nesta pesquisa os dados das entrevistas foram triangulados (entrevistas e observações) e sua origem é descrita nos resultados. Outro ponto importante é que os dados e resultados foram apresentados para uma banca de membros internos e externos do grupo de pesquisa ao qual o pesquisador principal faz parte.

Adicionalmente, também foi verificado a validade de constructo. De acordo com Wohlin et al. (2012), a validade de constructo visa buscar correspondência entre o que se pergunta e o que o participante responde. Para melhorar a validade de constructo foi realizado mais de um estudo piloto, um com acadêmicos e outro com uma equipe real, para tentar identificar perguntas que os membros tinham dificuldade de entender ou se as situações apresentadas nas respostas faziam jus ao constructo que se desejava verificar.

Merriam (2009) também menciona a importância da transferibilidade - chamada de validade externa por Wohlin et al. (2012) - que significa o quanto do resultado pode ser observado de maneira semelhante em outro contexto. Para aumentar transferibilidade uma descrição rica do contexto é apresentada na seção de resultados.

Glaser e Strauss (1967) comentam que existe um risco ao utilizar uma abordagem de análise de dados baseado em outra teoria, pois o pesquisador tende a se esforçar mais em selecionar os dados para cada categoria do que gerar categorias novas. No entanto, vale salientar que quando outros conhecimentos apareceram durante o processo de análise, eles foram integrados ao framework original.

## 4. Resultados

Nesta seção são descritos os resultados desta pesquisa.

### 4.1. Contexto

**Empresa A** – É uma empresa pública com foco em desenvolvimento de sistemas para auxiliar outros órgãos do governo. Nesta empresa foram consideradas duas equipes, escolhidas pela liberação dos gestores. Esta empresa tem mais de 100 sistemas em funcionamento e conta com cerca de 300 funcionários. Durante mais de 10 anos a empresa adotou o processo cascata, no entanto, nos últimos 5 anos tem adotado práticas de desenvolvimento ágeis, tais como a posse de código compartilhada e reunião diária.

Os membros da equipe possuem estabilidade pública. Em algumas situações, os membros são autorizados pelos outros membros a faltar, como em casos de doença de família, onde os outros membros da equipe são responsáveis por pegar a tarefa do membro faltoso e realizá-la. As equipes são responsáveis principalmente por prestar manutenção, além de eventualmente desenvolver novas funcionalidades no sistema. A empresa, na época da pesquisa, era certificada CMMI<sup>1</sup> nível 2, e buscava obter a certificação MPS.BR<sup>2</sup>.

A média de idade da equipe I é de cerca de 34 anos, onde o mais velho tem 39 e o mais novo 29. A média de tempo na empresa é de cerca de 2,4 anos, onde o mais recente tem um ano e o mais velho tem 4 anos. Em média eles têm cerca de 8,4 anos de experiência de mercado, com dois anos no mínimo e 13 anos no máximo. A equipe I tem quatro membros.

O sistema que essa equipe é responsável é o de arrecadação de outros órgãos do governo. Eles prestam manutenção no código e eventualmente, criam novas funcionalidades. Este é o software de maior importância para a empresa e que tem maior número de usuários.

A equipe II tem uma média aproximada de 31 anos, onde o mais novo tem 27 anos e o mais velho 38 anos. Os membros têm aproximadamente em média 8 anos de experiência, onde o membro com menos tempo de mercado tem 4 anos e no máximo 14 anos. Por fim, o tempo médio aproximado na empresa é de 4 anos, onde o membro que entrou mais recentemente tem apenas 1 ano na empresa e o máximo de 8 anos. O projeto em que essa equipe está envolvida tem 3 módulos. Essa equipe é responsável por um dos módulos de compra e vendas de imóveis. A equipe II tem cinco membros.

**Empresa B** – Esta é uma empresa multinacional com foco no desenvolvimento de novas tecnologias e sistemas com cerca de 90 empregados nesta unidade de pesquisa. O projeto desenvolvido pela equipe investigada está sobre sigilo e não pode ser detalhado, no entanto, é uma tecnologia que não existe no mercado. A equipe também utiliza práticas variadas de desenvolvimento de software, inclusive ágeis, como reunião diária e backlog. Todavia, as alocações das tarefas são feitas por um gerente. Algumas tarefas, como por exemplo o desenvolvimento de código complexo, tem um membro como responsável por garantir a qualidade do código feito pela equipe. Todas as equipes desta unidade ficam em um grande galpão (*open office*), onde eles podem conversar e circular com membros de outras equipes.

A equipe III tem seis membros. Os membros da equipe têm em média aproximadamente 7 anos de experiência com desenvolvimento de software. O menor tempo de experiência com desenvolvimento é de 2 anos e o maior é de 10 anos. O membro mais novo tem 22 anos, o mais velho tem 32 anos. A média aproximada de idade é de 26 anos. E a média de tempo na empresa é de 6,5 anos, com o membro mais antigo com 4 anos, e o mais novo com apenas 1 ano.

**Empresa C** – A empresa C é uma startup que possui uma plataforma de estudos para universitários com materiais compartilhados pela sua comunidade. Os membros pregam diversas práticas ágeis, como backlog, posse compartilhada de código, programação em par (em parte do tempo) e reuniões diárias. A equipe IV é responsável pelo sistema de conteúdo e conversa entre os assinantes. Por exemplo, o sistema deve reconhecer o conteúdo que foi enviado e categorizar ele de acordo com as regras do site. A equipe investigada tem quatro membros. O mais novo 28 anos, o mais velho 32, a

---

<sup>1</sup> <https://cmmiinstitute.com/>

<sup>2</sup> <https://softex.br/mpsbr/>

média da equipe é de aproximadamente 29 anos. O mais experiente tem 8 anos de desenvolvimento enquanto que o mais novo tem 2 anos. O membro com mais tempo na empresa desta equipe tem 2 anos, e o com menos tempo está prestes a completar um ano.

**Empresa D** – A empresa D é uma empresa recente no mercado, com apenas três anos de existência. Ela é uma empresa privada que cria soluções para empresas, sejam públicas ou privadas. No momento da entrevista a empresa e a equipe estavam focados em um sistema de estacionamento para prefeituras em que a população pode comprar o ticket para estacionar e a prefeitura recebe o dinheiro (Zona azul). Os membros utilizam práticas de desenvolvimento ágil como sprint, reunião diária, posse coletiva de código, mas não consideram a equipe totalmente ágil. A equipe V tem quatro pessoas com o mais velho com 28 anos, o mais novo 23 anos. O membro com mais tempo na empresa desta equipe tem 2 anos, o mais novo vai completar um ano.

**Empresa E** – Empresa localizada na cidade de Recife – Pernambuco com cerca de 15 funcionários em diversas equipes, para esta pesquisa foram utilizadas apenas duas equipes. A empresa desenvolve projetos em paralelo com outras empresas de software e dois produtos de sua criação. Alguns dos membros das equipes já participaram de mais de um projeto que estava ativo no momento. As equipes utilizam práticas ágeis como programação em par entre os membros menos experientes e os mais experientes, quadro de tarefas, reuniões diárias e posse de código compartilhada. A empresa tem um site explicando as normas da organização, quais roupas os membros não podem usar, a quem procurar quando tiverem um problema, o horário de funcionamento da empresa etc. Contudo, o site está desatualizado e os membros mais novos não tinham conhecimento sobre ele. Além disso, os membros da equipe ficam localizados na mesma sala, no entanto, os líderes das equipes ficam em uma sala separada. Na empresa E os próprios empregados definem os seus horários desde que cumpram a carga horária e participem das reuniões. Geralmente, a equipe define horários comuns para que eles consigam trabalhar juntos.

Dois equipes da empresa E foram observadas e entrevistadas, a descrição das equipes será apresentada a seguir:

Equipe VI: Formada por quatro integrantes que trabalham juntos desde outubro de 2013, e um integrante que entrou em outubro de 2016. O projeto é um sistema de gestão e controle de escolas de línguas, oferece uma versão *mobile* para que os alunos também possam controlar seu curso.

Equipe VII: Formada por três integrantes que trabalham juntos desde setembro de 2014, e outro integrante que entrou em outubro de 2016. O projeto é um sistema web que gerencia o administrativo de empresas. O sistema é evoluído pela empresa de forma constante.

#### 4.2. *Como os diversos tipos de conhecimento em equipe impactam no desenvolvimento de software?*

A seguir os trechos das entrevistas e observações que apresentam as evidências de como os diversos tipos de conhecimento em equipe impactam no desenvolvimento de software. Vale reforçar que os resultados aqui apresentados acontecem em pelo menos dois contextos. A partir desses trechos a Figura 5 foi construída onde as elipses são os conhecimentos em equipe e os retângulos são fatores afetados pelos conhecimentos em equipe. Os fatores podem ser afetados positivamente e negativamente.

O conhecimento em equipe relacionado às **estratégias da tarefa** está relacionado positivamente com o *engajamento da equipe* e está relacionado com um *maior entendimento comum de todo do software*.

*“[ter as estratégias da tarefa compartilhadas] é bastante importante...isso faz com que as pessoas se dediquem mais... e como toda a equipe tem o entendimento, o mesmo entendimento de como a tarefa deverá ser feita, tudo tá mais claro na cabeça de todo mundo ...fica mais fácil de ter um maior entendimento comum.” Empresa D*

Além disso, o conhecimento da **estratégia da tarefa** também está relacionado positivamente com a resolução de problemas, pois ele prover uma visão maior do todo. Além disso, ele também está positivamente relacionado com a localização de expertise do código.

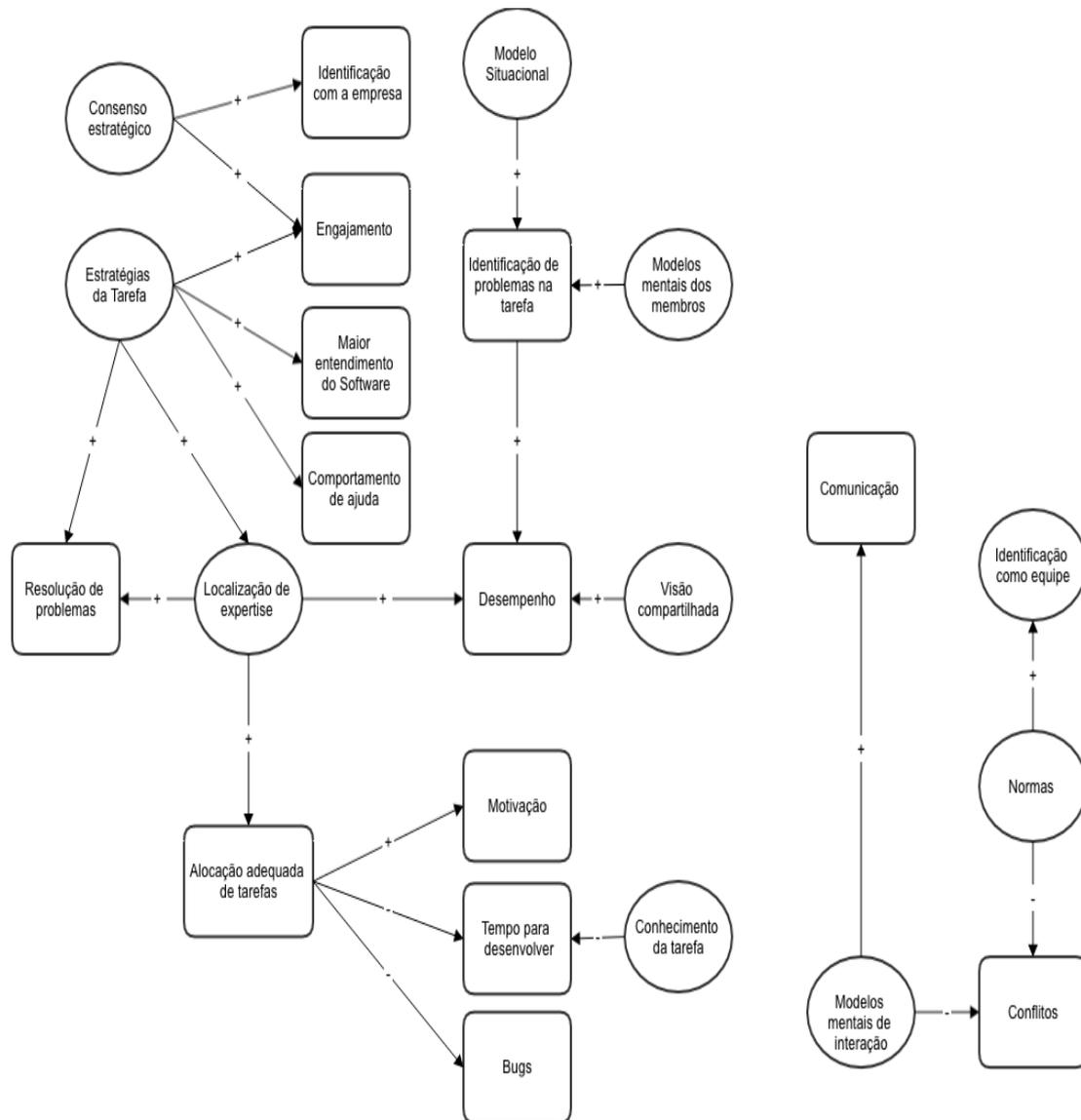
*“...todo mundo sabe como tá dividido as coisas, quem tá com determinada tarefa, e na hora que precisar, vai lá na pessoa certa... aqui a gente, tem tal problema, como que a gente vai resolver? isso também traz mais ideias de como resolver ... e aí o cara diz, resolve assim... ainda tem o fato de quando alguém tá fazendo algo... você já pensa assim: eita eu não posso fazer assim porque tem aquela outra parte de fulano...” Empresa E*

O conhecimento da **estratégia da tarefa** pode auxiliar a equipe a ter um melhor comportamento de ajuda, pois os membros sabem o quanto é importante determinadas tarefas e podem auxiliar quando observam que outros membros estão com dificuldades.

*“... a gente faz isso[compartilhar as estratégias da tarefa] toda semana quando a gente vai calcular a sprint, a gente vê as tarefas que vai precisar e a gente divide em conjunto ne? Esses dias eu estava em uma tarefa, e aí precisou fazer outra tarefa porque era mais crítico, então eu fiz front end para o beltrano e ele fez o backend, depois eu voltei para tarefa que estava fazendo...essa tarefa era mais importante para o projeto naquele momento, se eu não fizesse, o projeto podia atrasar...” Empresa C*

O **conhecimento da expertise dos membros** está relacionado positivamente com o fato dos membros reconhecerem a capacidade dos outros membros e isso faz com que os membros aloquem adequadamente as tarefas entre si, evitando assim a desmotivação.

*“Não adianta você passar uma tarefa para uma pessoa que você sabe que ela não vai conseguir fazer ou que ela vai ter muita dificuldade, isso pode gerar problemas do tipo ela atrasar ou ter muito erro... além disso, ela acaba desmotivando, né?” Empresa E*



**Figura 5 Modelo dos resultados**

Além disso, a alocação adequada pode levar as tarefas serem realizadas em menos tempo e com menos bugs.

*“Cada um faz o que mais gosta, a gente fica mais feliz com o trabalho, além disso, como a gente gosta disso[da tarefa], normalmente a gente sabe mais fazer[a tarefa]...então leva menos tempo e tem menos bugs também...” Empresa C*

A **localização de expertise** está relacionada positivamente com o desempenho da equipe porque os membros deixam as tarefas para quem eles acreditam que pode desempenhar melhor.

*“a gente se conhece... sabe mais ou menos em que cada membro é bom...isso ajudou muito no desempenho da equipe nesse projeto porque cada um ia, pegava uma tarefa e focava no que era melhor, a equipe já meio que sabia em que cada um se dava bem, e meio que deixava para o cara fazer aquela tarefa.” Empresa C*

A **localização de expertise** facilita a resolução de problemas porque os membros procuram quem fez determinada tarefa e quem sabe mais sobre o assunto:

*“Sim[é importante saber quem fez determinada tarefa?], porque as vezes eu preciso mexer e aí é bom saber quem fez para que eu não perca tempo entendendo o código. Se*

*tem algum problema ou função que eu não entenda, eu vou procurar a pessoa, ne?”  
Empresa D*

As **normas compartilhadas** estão relacionadas negativamente com o conflito porque quando os membros têm maiores conhecimentos das normas informais, mais eles podem evitar situações de conflito.

*“A gente sabe que quando fulando bota o fone, se você ficar chamando ele não gosta muito sabe? então quando eu vejo que ele tá com fone eu falo pelo aplicativo de celular sabe? por mais que ele sente ali na frente eu falo pelo aplicativo de celular... então a gente evita essas coisas pro bem da equipe...” Empresa E*

As **normas compartilhadas** estão relacionadas positivamente com a identificação dos membros como parte da equipe, os membros acabam criando normas que auxiliam na sensação de que os membros fazem parte da equipe.

*“Existem regras que só nossa equipe tem, tipo a gente tenta chegar no mesmo horário, a gente tenta sair junto para comer, todo mundo se sente mais parte da equipe fazendo essas coisas...seguido essas regras...” Empresa E*

O fato da equipe ter uma **visão compartilhada** está relacionado positivamente com o fato da equipe ter foco e buscar desempenhar melhor para atingir os objetivos.

*“Sim... eu acho que compartilhar as metas da equipe... onde a equipe quer chegar é fundamental. Todo mundo tem que saber aonde a equipe quer chegar...porque senão fica um bando de nego querendo ir para lados diferentes... quando isso tá claro para todo mundo, fica mais fácil de chegar no mesmo lugar, e conseqüentemente a equipe desempenha melhor.” Empresa C*

O conhecimento em equipe relacionado ao **consenso estratégico** está relacionado positivamente com o engajamento dos membros com a tarefa. Os membros ao compartilharem e terem a oportunidade de opinarem sobre os objetivos estratégicos ficam mais engajados na tarefa.

*“...é importante que toda a equipe tenha um conhecimento sobre a estratégia da organização... porque todo mundo sabe para onde ir, todo mundo opina, a gente quer chegar à tal resultado, todo mundo sabe onde quer chegar... quando equipe divide com a empresa os objetivos, você trabalha motivado, você produz melhor porque você quer chegar junto com a empresa naqueles objetivos.” Empresa A*

O conhecimento em equipe relacionado ao **consenso estratégico** afeta positivamente na identificação do membro com a empresa, fazendo com que os membros trabalhem juntos para chegar em um resultado comum.

*“Então...aqui a gente sempre faz isso [compartilha o conhecimento estratégico da organização] de um projeto para outro... a gente sempre para pra pensar o quanto o projeto foi importante para a empresa e qual a importância do novo projeto para empresa. tipo essa transparência ajuda muita a gente, a gente sabe que tá crescendo, se motiva mais ...meio que todo mundo trabalha junto para chegar em um resultado importante para empresa porque ela crescendo você cresce também” Empresa C*

O conhecimento em equipe do **modelo situacional** está relacionado positivamente com a identificação de problemas na tarefa. Isto faz com que membros possam saber como as tarefas da equipe estão naquele momento e possam ajudar os membros caso alguma atividade corra risco de atrapalhar o desempenho da equipe.

*“É tipo...um conhecimento de como tá cada um naquele momento, é como se fosse um print ... Se alguém tá em impedimento, certa forma outra pessoa pode ajudar ou não, é uma forma também de saber o que o outro tá fazendo e também solucionar alguns problemas que podem acontecer” Empresa E*

O entendimento compartilhado dos **modelos mentais de interação** auxilia na diminuição de conflito porque os membros sabem como se comunicar da maneira correta

com os outros membros da equipe.

*“...tem gente que a equipe já sabe como tem que falar porque é meio sentimental quando faz as coisas erradas, aí isso ajuda a evitar problemas” Empresa C*

O entendimento compartilhado dos **modelos mentais de interação** auxilia na melhora da comunicação porque os membros sabem onde deve postar e se comunicar de acordo com a necessidade.

*“...quando é algo mais urgente a gente liga ou vai diretamente na pessoa, quando é algo mais ou menos a gente manda um slack ou zap, e quando é algo sem mais criticidade a gente manda um email mesmo... isso é importante porque a gente já sabe o tamanho do problema pela maneira de comunicar...” Empresa B*

O entendimento compartilhado dos **modelos mentais dos membros** auxilia na identificação de problemas na tarefa dos outros membros

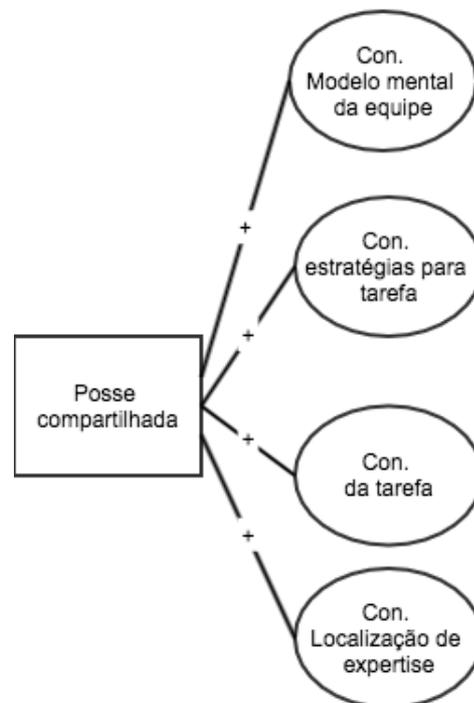
*“... é engraçado porque quando o fulando quando não consegue fazer as coisas ele vai logo fumar, tá ligado ? se levanta... aí a gente já sabe que tem coisa errada” Empresa A*

O **conhecimento da tarefa** pode diminuir o tempo para se realizar tarefa porque os membros já sabem como utilizar determinada classe ou funcionalidade.

*“Cara, isso[compartilhar o conhecimento da tarefa com a equipe] rola muito...a galera sempre tá perguntado o que faz uma classe, como que chama tal método e isso é importante para que o cara não perca muito tempo, você vai ter que ler, entender e nem sempre isso é fácil...E as vezes você acaba fazendo retrabalho, um código pior porque a equipe não tinha o entendimento comum do código, aí o projeto acaba atrasando” Empresa D*

#### 4.3. *Quais conhecimentos em equipe são promovidos pela posse compartilhada de código ?*

A Figura 6 apresenta quais conhecimentos em equipe relação com a posse compartilhada de código.



**Figura 6** Conhecimento em equipe promovidos pela posse compartilhada de código.

A seguir, são aprestadas quatro proposições sobre quais conhecimentos em equipe são promovidos pela posse compartilhada de código.

A posse compartilhada de código está relacionada positivamente conhecimento em equipe quanto a estratégias para realizar tarefas:

*“Olhe ... veja eu acho o que me motiva mais a essa pratica, é conhecer... aprender mais com os outros... outras formas... outros caminhos... outras estratégias para resolver determinados problemas ... para fazer o código ... que a gente acaba compartilhando ... isso acaba melhorando o desempenho da equipe, né?” Empresa B*

A posse compartilhada de código está relacionada com o conhecimento em equipe dos modelos mentais:

*“Quanto a prática de posse de código compartilhada... você acaba encontrando características dos outros desenvolvedores, em alguns momentos você olha uma parte do código e você sabe claramente por quem foi feito... isso faz você conhecer sua equipe ...saber...como eles pensam... ” Empresa B*

A posse compartilhada de código está relacionada positivamente com o conhecimento em equipe relacionado à localização de expertise da equipe:

*“Com essa prática eu tenho mais conhecimento de quem pode fazer determinadas coisas na equipe ... tipo... se eu vejo que fulano fez um código sobre determinada parte que preciso de ajuda, procuro logo ele... assim perco menos tempo...” Empresa E*

A posse compartilhada de código está relacionada positivamente com o conhecimento da tarefa:

*“É... como todo mundo detém o código, tem conhecimento do código, né, isso pode ser útil pra ajudar em... em outras coisas. Por exemplo, alguém entra de férias, a gente sempre faz um rodízio pra alguém pegar alguma coisa de... de cada um, sabe?” Empresa A*

#### **4.4 Quais conhecimentos em equipe são promovidos pela reunião diária?**

Outra prática também investigada foi a reunião diária. A Figura 7 apresenta um quais conhecimentos em equipe estão relacionados com a prática de reunião diária.

A reunião diária está relacionada positivamente com o conhecimento em equipe da visão compartilhada.

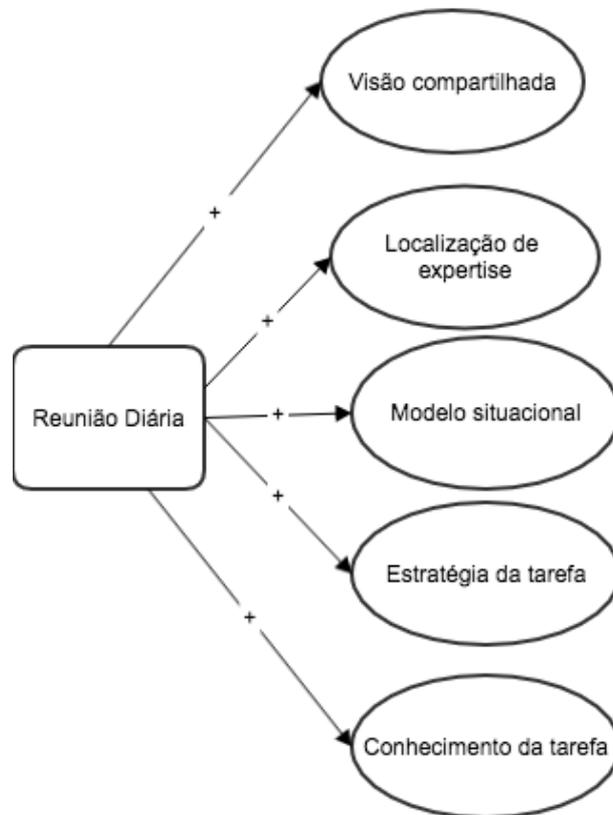
*“Sim, tipo eu acho que compartilhar as metas da equipe... onde a equipe quer chegar é fundamental. Todo mundo tem que saber onde a equipe quer chegar [...] quando a gente faz a reunião diária a gente tenta deixar isso bem claro para toda a equipe...meio que a gente assina um contrato...” Empresa A*

A reunião diária está relacionada positivamente com o conhecimento em equipe da localização de expertise da equipe.

*Durante o trabalho, o membro 1 tirou dúvidas sobre o funcionamento do framework com outro membro. O pesquisador perguntou como ele sabia que o outro engenheiro 2 saberia responder e o engenheiro 1 disse que já tinha visto o engenheiro 2 fazer atividades semelhantes pela reunião diária. (OBSERVAÇÃO Empresa E)*

A reunião diária está relacionada positivamente com o conhecimento em equipe da estratégia para realizar a tarefa da equipe

*A equipe começou a discutir sobre qual a melhor maneira de dividir as tarefas para que um não fique esperando pelo outro e qual seria a estratégia que eles usariam para não atrasar as entregas. (OBSERVAÇÃO Empresa E)*



**Figura 7** Conhecimento em equipe promovidos pela reunião diária

A reunião diária está relacionada positivamente com o conhecimento em equipe do conhecimento da tarefa.

*O Engenheiro apresenta como funciona a parte de integração que só ele estava fazendo com o banco. Ele explica como funciona a classe e aí surge algumas perguntas e dúvidas. Eles discutem sobre como poderia ser uma boa solução para problemas técnicos que poderiam surgir. (OBSERVAÇÃO Empresa E)*

## 5. Discussão

É interessante notar que os trabalhos de Moe *et al.* (2016) e Faegri *et al.* (2016) não apresentam o conhecimento em equipe do modelo situacional por não considerar a dimensão dinâmica do modelo de Wildman *et al.* (2012). No entanto, ele aparece durante as observações e entrevistas e tem um papel importante na identificação de problemas na equipe que auxilia na melhoria do desempenho. Embora não sejam encontrados estudos anteriores com relações diretas entre o modelo situacional e a identificação de problemas, o estudo de Stray *et al.* (2016) aponta que a reunião diária afeta a identificação de problemas. No mesmo trabalho de Stray *et al.* também foi observado que a reunião diária cria maiores níveis de modelo situacional na equipe.

Os resultados encontrados neste trabalho sugerem que o conhecimento compartilhado sobre as normas que foi adicionado por Faegri *et al.* (2016) está

relacionado com a diminuição de conflito dos membros. É interessante notar que existem outros trabalhos como o de Stray et al (2017) e o de Lenberg e Feld (2017) que apresentam relações entre normas e sua clareza com o desempenho das equipes. No entanto, no melhor de nosso conhecimento, não existem estudos prévios na Engenharia de Software que apresentem relações entre normas e conflitos.

Faegri et al. (2016) não encontrou trabalhos que tenham realizado algum estudo sobre consenso estratégico na Engenharia de Software. Este trabalho apresenta que o consenso estratégico tem influência no engajamento dos membros nas tarefas e na identificação dos membros com a empresa.

O conhecimento ligado às estratégias da tarefa apresenta relações positivas com o engajamento, com o maior entendimento do software, resolução de problemas, com o comportamento de ajuda e com outro conhecimento em equipe que é a localização de expertise. O maior engajamento já tinha sido previsto por Baliyepally et al. (2015), enquanto que a maior resolução de problemas já tinha sido prevista por Maruping et al (2009). A qualidade do software também foi prevista por Maruping et al (2009) mas não foi encontrado neste estudo, além da performance individual que pode ser consequência também das outras relações encontradas nesta pesquisa, como resolução de problemas e maior entendimento do software.

A localização de expertise apresenta ligações diretas com o desempenho, com a resolução de problemas e com a alocação adequada de tarefas, que por sua vez apresentou outras ligações com o aumento de motivação, com a diminuição de bugs e com a diminuição de tempo para desenvolver. Existem outros estudos sobre localização de expertise e a performance como o Sharman e Ghosh (2007) e o Hsu et al. (2012). Hsu et al. (2012) também comenta sobre a melhor alocação de tarefas. Além disso, McDonald e Ackerman (1998) sugerem sobre a relação de localização de expertise e a diminuição de bugs em seu estudo. No entanto, no melhor do nosso conhecimento não foi possível encontrar estudos que fale da relação entre localização de expertise e tempo para produzir o software.

Este trabalho identificou que os modelos mentais dos membros auxiliam na identificação de problemas. Algo que não foi observado na literatura de Engenharia de Software. Ao mesmo tempo, a literatura de Engenharia de Software apresenta resultados positivos com o sucesso da equipe (FARAJ e SPROULL, 2000).

Os modelos mentais de interação compartilhada, apresentam ligação positiva com a comunicação e negativa com o conflito. Bass (2015) afirma que os membros criam um padrão para se comunicar e isto foi observado neste estudo. No entanto, não foram encontrados resultados sobre a relação entre modelos mentais de interação compartilhados pelos membros da equipe e o conflito em equipes de desenvolvimento de software.

As normas apresentaram uma relação negativa com o conflito, embora essa relação não seja investigada na Engenharia de Software, ela existe em outras áreas (SANTOS *et al.* 2015). Além disso, existem outros resultados sobre a influência das normas na performance (TEH et al. 2012) e no aprendizado da equipe (CHANG et al. 2013) que não foram encontrados neste estudo.

Foi observado neste estudo que a visão compartilhada afeta positivamente o desempenho da equipe na Engenharia de Software. Cockburn e Highsmith (2001) apresentam resultados semelhantes onde eles afirmam que a visão compartilhada afeta positivamente o desempenho.

Os membros das equipes compartilham o conhecimento da tarefa quando encontram tarefas difíceis de serem realizadas e com alta importância para o funcionamento do sistema. Com isso, acreditam que quando for necessária a manutenção, outros membros irão poder realizar a tarefa em menos tempo.

Um outro ponto que vale a pena apresentar é que a localização de expertise e da estratégia da tarefa apareceram com destaque nos resultados pois eles eram os conhecimentos em equipe em maior evidência na observação e nas entrevistas. A localização de expertise é reconhecida em outras áreas como bastante importante, no entanto, não existem muitas pesquisas em outras áreas do conhecimento sobre as estratégias da tarefa compartilhada pelos membros (WILDMAN et al 2012 e WILDMAN, SALAS e SCOTT 2014).

A Tabela 2. apresenta uma sumarização dos resultados das relações encontradas neste artigo comparada aos resultados já encontrados na literatura da Engenharia de Software previamente.

**Tabela 2. Resumo dos resultados novos x literatura na área.**

Resultado	Novos resultados	Previamente encontrados
Conhecimento Estratégico das tarefas afeta positivamente o maior entendimento do software	X	
Conhecimento Estratégico das tarefas afeta positivamente Engajamento		Balijepally et al. (2015)
Conhecimento Estratégico das tarefas afeta positivamente resolução de problemas		Maruping et al (2009)
Conhecimento Estratégico das tarefas afeta positivamente comportamento de ajuda	X	
Conhecimento Estratégico das tarefas afeta positivamente localização de expertise	X	
Consenso Estratégico afeta positivamente o engajamento	X	
Consenso Estratégico afeta positivamente a identificação com a empresa	X	
Localização de expertise afeta positivamente a resolução de problemas	X	
Localização de expertise afeta positivamente o desempenho		Sharman e Ghosh (2007) e Hsu et al. (2012)
Localização de expertise afeta positivamente a alocação adequada de tarefa		Hsu et al. (2012)
Localização de expertise afeta indiretamente a motivação pela alocação adequada de tarefa	X	

Localização de expertise afeta indiretamente o tempo para desenvolvimento pela alocação adequada de tarefa	X	
Localização de expertise afeta indiretamente a quantidade de bugs pela alocação adequada de tarefa		McDonald e Ackerman (1998)
Modelo situacional afeta positivamente identificação de problemas	X	
Modelo situacional afeta positivamente identificação de problemas que afeta positivamente o desempenho	X	
Modelos mentais dos membros afeta positivamente a identificação de problemas	X	
Modelos mentais de interação afeta positivamente a comunicação		Bass (2015)
Modelos mentais de interação afeta negativamente o conflito	X	
Visão compartilhada afeta positivamente o desempenho		Cockburn e Highsmith (2001)
As normas compartilhadas afetam negativamente o conflito	X	
As normas compartilhadas afetam a identificação dos participantes como membros a equipe		Monaghan et al (2014)

No trabalho de Sedano *et al.* (2016), foi encontrado que a posse de código compartilhada afeta o entendimento do contexto do sistema da equipe. O contexto do sistema é definido pelo autor como o conhecimento do código, das decisões de design e do domínio do sistema. Os resultados deste trabalho confirmam os resultados de Sedano *et al.* (2016) e adicionam o conhecimento em equipe dos modelos mentais dos membros e da localização de expertise. Portanto, um líder de projeto que deseje melhorar os níveis desses constructos na equipe pode utilizar a posse de código compartilhada para isso.

Quanto aos resultados da reunião diária, o trabalho de Stray *et al.* (2016) que afirma que a reunião diária leva a maior conhecimento em equipe, principalmente ao modelo situacional compartilhado da equipe. Neste trabalho foi encontrado que a reunião diária promove os conhecimentos em equipe quanto à visão compartilhada, à localização de expertise, ao modelo situacional, à estratégia da tarefa e o conhecimento para realizar a tarefa.

A Tabela 3 apresenta uma sumarização dos resultados da posse de código e da reunião diária encontradas neste artigo e comparando com os resultados já encontrados na literatura de Engenharia de Software previamente.

**Tabela 3. Resumo dos resultados novos x literatura na área.**

Resultado	Novos resultados	Previamente encontrados
A posse compartilhada afeta positivamente o modelo mental dos membros	X	
A posse compartilhada afeta positivamente o Conhecimento da tarefa		Sedano <i>et al.</i> (2016),
A posse compartilhada afeta positivamente a estratégia da tarefa		Sedano <i>et al.</i> (2016),
A posse compartilhada afeta positivamente a localização de expertise	X	
A reunião diária afeta positivamente a visão compartilhada	X	
A reunião diária afeta positivamente o modelo situacional		Stray <i>et al.</i> (2016)
A reunião diária afeta positivamente o Conhecimento da tarefa	X	
A reunião diária afeta positivamente a estratégia da tarefa	X	
A reunião diária afeta positivamente a localização de expertise	X	

### 5.1 Limitações

Houve algumas limitações para conduzir este trabalho que serão discutidas nesta seção. Como os dados surgem de entrevistas e observações, os dados dependem das interpretações do pesquisador que realizou a análise, mesmo este pesquisador apresentando os dados e o resultados da análise, para outros pesquisadores e para a equipe que foi pesquisada, ela foi conduzida de maneira individual, sob uma ótica moldada pelas experiências do pesquisador principal. Isto é uma limitação inerente ao paradigma escolhido (MERRIAM, 2009).

Em pesquisas qualitativas também existem as interpretações dos entrevistados sobre a pergunta, isso é uma das ameaças da pesquisa em que a equipe tomou algumas precauções, mas que em algum momento pode ter passado despercebido para o pesquisador que conduziu as entrevistas e observações.

Vale salientar também que esta pesquisa não visa a generalização para a população, mas sim para teoria. Os pesquisadores não acreditam que todos os resultados podem ser repetidos em outro contexto para a mesma população de maneira igual. Ao invés disso, acreditamos que existem elementos do modelo apresentado que são semelhantes em outros contextos e que podem auxiliar outros pesquisadores, gerentes, e equipes no entendimento do seu funcionamento.

A observação foi de suma importância para os resultados deste trabalho porque permitiu ao pesquisador visualizar situações que nas entrevistas não apareceram, além de

permitir confirmar ou não situações que apareciam nas entrevistas, uma vez que o assunto discutido é o conhecimento e suas estruturas que estão dentro da mente dos indivíduos.

Dito isso, as primeiras limitações foram a quantidade de tempo realizando observações e a quantidade de equipes e empresas observadas. O pesquisador conseguiu realizar apenas um pouco mais de três semanas de observação em duas equipes de apenas uma das cinco empresas empresa. Mesmo observando durante 8 horas por dia.

Para mitigar essa limitação, uma sugestão seria que o pesquisador fosse membro de alguma equipe de desenvolvimento de alguma empresa para realizar um estudo mais completo com maior tempo na equipe. Apesar deste enfoque também ter suas limitações, traria ganhos valiosos para a área de pesquisa, visto que também são poucos os estudos que utilizam observação na área de Engenharia de Software (FAEGRI et al. 2016) e em outras áreas eles aparecem em maior quantidade (WILDMAN et al. 2012).

Outro ponto que gerou uma limitação é quanto ao tempo junto dos membros da equipe e da própria equipe. Uma vez que o conhecimento em equipe é construído pela interação dos membros, observar uma equipe que não se conhece previamente ou uma equipe recém-formada deve auxiliar no maior entendimento da área, assim como comparar os resultados com equipes que estão muito tempo juntas. Isso não foi possível neste trabalho.

As equipes investigadas utilizavam de várias técnicas ágeis e se consideravam ágeis, portanto, os resultados aqui apresentados devem ser vistos com um pouco de cautela para equipes que atuam com modelo cascata. Embora os resultados possam ser semelhantes, não existem evidências no nosso trabalho que isso seja verdadeiro.

Apesar de ter observado outras práticas de desenvolvimento como a reunião de *sprint* e a reunião de retrospectiva, a quantidade de observações e resultados das entrevistas foram insuficientes para concluir algo, portanto não foram colocadas outras práticas nos resultados desta pesquisa.

Outro ponto, é que não foram encontrados trabalhos que comentassem sobre características do ambiente que devessem ser levadas em consideração durante o processo de análise de dados. Isso é uma limitação, pois pode afetar os resultados da análise.

Por fim, a grande quantidade de conhecimentos em equipe investigados tornou a pesquisa difícil de ser conduzida e analisada de maneira mais profunda, visto que existem os conhecimentos em equipe se diferem, e existem conhecimento com poucas ligações com outros conhecimentos. Em uma outra oportunidade, selecionar apenas um subconjunto do conhecimento em equipe para analisar pode ser interessante para aumentar o foco da pesquisa. Outra opção é escolher apenas uma prática e ver suas relações com os conhecimentos em equipe a fundo.

## 6. Conclusão

Este trabalho investigou o impacto dos diversos tipos de conhecimento em equipe na Engenharia de Software e entendeu quais conhecimentos em equipe a posse de código compartilhada e a reunião diária promovem. Para isso, combinou os modelos teóricos propostos por Wildman et al. (2012) e Faegri et al. (2016) com entrevistas e observações em diferentes empresas e equipes.

Os modelos apresentados neste trabalho podem auxiliar na escolha de práticas de desenvolvimento de software de acordo com a necessidade de melhoria do conhecimento em equipe que as equipes precisam. Além disso, apresentam os impactos dos conhecimentos em equipe no desenvolvimento de software.

Os resultados mostram que os gerentes de uma equipe de desenvolvimento podem usar a posse compartilhada de código caso desejem promover o conhecimento da tarefa, a localização de expertise, o modelo mental dos membros e o conhecimento compartilhado das estratégias da tarefa de suas equipes. A reunião diária, por outro lado pode auxiliar os gerentes de projeto ou líderes de equipe na promoção da localização de expertise, no conhecimento da tarefa, no conhecimento das estratégias da tarefa, no modelo situacional e na visão compartilhada.

Outrossim, também foram apresentados os impactos dos constructos que envolvem o conhecimento em equipe no desenvolvimento de software. O consenso estratégico ajuda os membros a se identificar com a empresa e se engajar mais nas tarefas, o conhecimento das estratégias da tarefa também auxilia no engajamento, no entendimento do software, promove o comportamento de ajuda dos membros, auxilia na resolução de problemas e na localização de expertise. A localização de expertise também auxilia na resolução de problemas, no aumento do desempenho dos indivíduos, e na alocação adequada de tarefas, que aumenta a motivação, diminui o tempo para desenvolvimento, e é capaz de diminuir os bugs. Já o conhecimento da tarefa está ligado a diminuição do tempo para se desenvolver o software.

Tanto o modelo situacional, quanto os modelos mentais dos membros são capazes de auxiliar os membros da equipe na identificação de problemas na tarefa, que aumenta o desempenho da equipe. A visão compartilhada também tem impacto positivo no desempenho da equipe.

As normas auxiliam na identificação da equipe pelos membros, assim como são capazes de diminuir os conflitos em uma equipe de desenvolvimento de software. Os conflitos também são impactados pelos modelos mentais de interação dos membros da equipe. Esses modelos também são capazes de melhorar a comunicação.

Durante a pesquisa, também foi possível identificar que os constructos de localização de expertise e conhecimento das estratégias da tarefa são muito utilizados no dia-a-dia dentro de uma empresa de desenvolvimento de software. O modelo situacional descartado por Faegri et al. (2016) tem impacto no desenvolvimento de software e também deve ser levado em consideração tanto na construção do modelo quanto no entendimento do conhecimento em equipe na Engenharia de Software.

Como trabalhos futuros, foi identificado a necessidade de mensuração, tanto dos impactos dos conhecimentos em equipe quanto das relações propostas com a posse de código compartilhada de código e com a reunião diária. Um outro questionamento que pode ser investigado é se em equipes recém-criadas, a posse de código e a reunião diária também promovem os mesmos conhecimentos em equipe. Outra sugestão é investigar as diversas práticas de desenvolvimento de software que não apareceram nesta pesquisa por apresentarem dados insuficientes, mas que podem ajudar no conhecimento em equipe, como a programação em par, as reuniões de sprint e retrospectiva.

## Referências

- AGILE MANIFESTO. Manifesto for Agile Software Development, 17 February 2001. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso 25/07/2018.
- AUBÉ, Caroline; ROUSSEAU, Vincent; TREMBLAY, Sébastien. Perceived shared understanding in teams: The motivational effect of being 'on the same page'. *British Journal of Psychology*, v. 106, n. 3, p. 468-486, 2015. [[GS Search](#)]

- BALIJEPAALLY, VenuGopal; NERUR, Sridhar P.; MAHAPATRA, RadhaKanta. Task Mental Model and Software Developers' Performance: An Experimental Investigation. *CAIS*, v. 36, p. 53-76, 2015. [[GS Search](#)]
- BASS, Julian M. How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering*, v. 20, n. 6, p. 1525-1557, 2015. [[GS Search](#)]
- CANFORA, Gerardo et al. Evaluating performances of pair designing in industry. *Journal of Systems and Software*, v. 80, n. 8, p. 1317-1327, 2007. [[GS Search](#)]
- CHANG, Kuo-chung et al. Knowledge contribution in information system development teams: An empirical research from a social cognitive perspective. *International Journal of Project Management*, v. 31, n. 2, p. 252-263, 2013. [[GS Search](#)]
- CHEN, Xiaogang et al. Knowledge sharing in open source software project teams: A transactive memory system perspective. *International Journal of Information Management*, v. 33, n. 3, p. 553-563, 2013. [[GS Search](#)]
- CHOI, Sue Young; LEE, Heeseok; YOO, Youngjin. The impact of information technology and transactive memory systems on knowledge sharing, application, and team performance: a field study. *MIS quarterly*, p. 855-870, 2010. [[GS Search](#)]
- COCKBURN, Alistair; HIGHSMITH, Jim. Agile software development: The people factor. *Computer*, n. 11, p. 131-133, 2001. [[GS Search](#)]
- COHEN, Jason F. Contextual determinants and performance implications of information systems strategy planning within South African firms. *Information & Management*, v. 45, n. 8, p. 547-555, 2008. [[GS Search](#)]
- COOKE, Nancy J. et al. Team cognition. *Handbook of applied cognition*, v. 2, p. 239-268, 2007. [[GS Search](#)]
- COOKE, Nancy J. et al.. Advances in measuring team cognition. *Team cognition: Understanding the factors that drive process and performance*, p. 83-106, 2004. [[GS Search](#)]
- COOKE, Nancy J. et al.. Measuring team knowledge: A window to the cognitive underpinnings of team performance. *Group Dynamics: Theory, Research, and Practice*, v. 7, n. 3, p. 179-199, 2003. [[GS Search](#)]
- COOKE, Nancy J.; KIEKEL, Preston A.; HELM, Erin E. Measuring team knowledge during skill acquisition of a complex task. *International Journal of Cognitive Ergonomics*, v. 5, n. 3, p. 297-315, 2001. [[GS Search](#)]
- CROWSTON, Kevin; KAMMERER, Ericka Eve. Coordination and collective mind in software requirements development. *IBM Systems Journal*, v. 37, n. 2, p. 227-245, 1998. [[GS Search](#)]
- DEMARRAIS, Kathleen. Qualitative interview studies: Learning through experience. *Foundations for research: Methods of inquiry in education and the social sciences*, v. 1, n. 1, p. 51-68, 2004.. [[GS Search](#)]
- ENDSLEY, Mica R. Toward a theory of situation awareness in dynamic systems. *Human factors*, v. 37, n. 1, p. 32-64, 1995. [[GS Search](#)]

- ESPINOSA, J. Alberto et al. Team knowledge and coordination in geographically distributed software development. *Journal of management information systems*, v. 24, n. 1, p. 135-169, 2007. [[GS Search](#)]
- FÆGRI, Tor Erlend; STRAY, Viktoria; MOE, Nils Brede. Shared Knowledge in Virtual Software Teams: A Preliminary Framework. In: 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE). IEEE, 2016. p. 174-178. [[GS Search](#)]
- FARAJ, Samer; SPROULL, Lee. Coordinating expertise in software development teams. *Management science*, v. 46, n. 12, p. 1554-1568, 2000. [[GS Search](#)]
- GLASER, Barney G.; STRAUSS, Anselm L.; STRUTZEL, Elizabeth. The discovery of grounded theory; strategies for qualitative research. *Nursing research*, v. 17, n. 4, p. 364, 1968. [[GS Search](#)]
- GUEST, Greg; NAMEY, Emily E.; MITCHELL, Marilyn L. *Collecting qualitative data: A field manual for applied research*. Sage, 2012. [[GS Search](#)]
- HE, Jun; BUTLER, Brian S.; KING, William R. Team cognition: Development and evolution in software project teams. *Journal of Management Information Systems*, v. 24, n. 2, p. 261-292, 2007. [[GS Search](#)]
- HOLLINGSHEAD, Andrea B. Cognitive interdependence and convergent expectations in transactive memory. *Journal of personality and social psychology*, v. 81, n. 6, p. 1080, 2001. [[GS Search](#)]
- HSU, Jack Shih-Chieh et al. The impact of transactive memory systems on IS development teams' coordination, communication, and performance. *International Journal of Project Management*, v. 30, n. 3, p. 329-340, 2012. [[GS Search](#)]
- KANAWATTANACHAI, Prasert; YOO, Youngjin. The impact of knowledge coordination on virtual team performance over time. *MIS quarterly*, p. 783-808, 2007. [[GS Search](#)]
- KLIMOSKI, Richard; MOHAMMED, Susan. Team mental model: Construct or metaphor? *Journal of management*, v. 20, n. 2, p. 403-437, 1994. [[GS Search](#)]
- KNIGHT, Don et al. Top management team diversity, group process, and strategic consensus. *Strategic Management Journal*, p. 445-465, 1999. [[GS Search](#)]
- KOZLOWSKI, Steve WJ; ILGEN, Daniel R. Enhancing the effectiveness of work groups and teams. *Psychological science in the public interest*, v. 7, n. 3, p. 77-124, 2006. [[GS Search](#)]
- LENBERG, Per; FELDT, Robert. Psychological safety and norm clarity in software engineering teams. In: 2018 IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE). IEEE, 2018. p. 79-86. [[GS Search](#)]
- LEVESQUE, Laurie L.; WILSON, Jeanne M.; WHOLEY, Douglas R. Cognitive divergence and shared mental models in software development project teams. *Journal of Organizational Behavior*, v. 22, n. 2, p. 135-144, 2001. [[GS Search](#)]

- LIM, Beng-Chong; KLEIN, Katherine J. Team mental models and team performance: A field study of the effects of team mental model similarity and accuracy. *Journal of organizational behavior*, v. 27, n. 4, p. 403-418, 2006. [[GS Search](#)]
- LIN, Tung-Ching et al. The impact of team knowledge on problem solving competence in information systems development team. *International Journal of Project Management*, v. 33, n. 8, p. 1692-1703, 2015. [[GS Search](#)]
- MARKS, Abigail; LOCKYER, Cliff. Debugging the system: the impact of dispersion on the identity of software team members. *The International Journal of Human Resource Management*, v. 16, n. 2, p. 219-237, 2005. [[GS Search](#)]
- MARUPING, Likoebe M.; ZHANG, Xiaojun; VENKATESH, Viswanath. Role of collective ownership and coding standards in coordinating expertise in software project teams. *European Journal of Information Systems*, v. 18, n. 4, p. 355-371, 2009. [[GS Search](#)]
- MATHIEU, John E. et al. The influence of shared mental models on team process and performance. *Journal of applied psychology*, v. 85, n. 2, p. 273-283, 2000. [[GS Search](#)]
- MATHIEU, John et al. Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future. *Journal of management*, v. 34, n. 3, p. 410-476, 2008. [[GS Search](#)]
- MCDONALD, David W.; ACKERMAN, Mark S. Just talk to me: a field study of expertise location. In: *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. ACM, 1998. p. 315-324. [[GS Search](#)]
- Merriam, Sharan B. *Qualitative Research: a Guide to Design and Implementation*. 2. ed. San Francisco, CA: Jossey-Bass, 2009. [[GS Search](#)]
- MICHAELIS. *Moderno Dicionário da Língua Portuguesa*. Disponível em: <<http://michaelis.uol.com.br/moderno/portugues/index.php>>. Acesso em: 8 jun. 2017.
- MOE, Nils Brede et al. Enabling knowledge sharing in agile virtual teams. In: *Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on*. IEEE, 2016. p. 29-33. [[GS Search](#)]
- MOHAMMED, Susan; HAMILTON, Katherine; LIM, Audrey. The incorporation of time in team research: Past, current, and future. *Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches*, p. 321-348, 2009. [[GS Search](#)]
- MONAGHAN, Conal et al. Performance of student software development teams: the influence of personality and identifying as team members. *European Journal of Engineering Education*, v. 40, n. 1, p. 52-67, 2015. [[GS Search](#)]
- PEARCE, Craig L.; ENSLEY, Michael D. A reciprocal and longitudinal investigation of the innovation process: The central role of shared vision in product and process innovation teams (PPITs). *Journal of organizational Behavior*, v. 25, n. 2, p. 259-278, 2004. [[GS Search](#)]
- PRESTON, David; KARAHANNA, Elena. How to develop a shared vision: The key to IS strategic alignment. *MIS Quarterly Executive*, v. 8, n. 1-9, 2009. [[GS Search](#)]

- REN, Yuqing; ARGOTE, Linda. Transactive memory systems 1985–2010: An integrative framework of key dimensions, antecedents, and consequences *Academy of Management Annals*, v. 5, n. 1, p. 189-229, 2011. [[GS Search](#)]
- RUNESON, Per; HÖST, Martin. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, v. 14, n. 2, p. 131-164, 2009. [[GS Search](#)]
- SANTOS, Catarina Marques; UITDEWILLIGEN, Sjir; PASSOS, Ana Margarida. Why is Your Team More Creative Than Mine? The Influence of Shared Mental Models on Intra-group Conflict, Team Creativity and Effectiveness. *Creativity and Innovation Management*, v. 24, n. 4, p. 645-658, 2015. [[GS Search](#)]
- SEDANO, Todd; RALPH, Paul; PÉRAIRE, Cécile. Sustainable software development through overlapping pair rotation. In: *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 19-29, ACM, 2016.. [[GS Search](#)]
- SHARMA, Monika; GHOSH, Anjali. Does team size matter? A study of the impact of team size on the transactive memory system and performance of IT sector teams. *South Asian Journal of Management*, v. 14, n. 4, p. 96-115, 2007. [[GS Search](#)]
- STRAY, Viktoria; SJØBERG, Dag IK; DYBÅ, Tore. The daily stand-up meeting: A grounded theory study. *Journal of Systems and Software*, v. 114, p. 101-124, 2016. [[GS Search](#)]
- SUNG, Sun Young; CHOI, Jin Nam. Effects of team knowledge management on the creativity and financial performance of organizational teams. *Organizational Behavior and Human Decision Processes*, v. 118, n. 1, p. 4-13, 2012. [[GS Search](#)]
- TEH, Alvin et al. Social psychology and software teams: Establishing task-effective group norms. *IEEE software*, v. 29, n. 4, p. 53-58, 2012. [[GS Search](#)]
- WILDMAN, Jessica L. et al. Team knowledge research: Emerging trends and critical needs. *Human Factors*, v. 54, n. 1, p. 84-111, 2012. [[GS Search](#)]
- WILDMAN, Jessica L.; SALAS, Eduardo; SCOTT, Charles PR. Measuring cognition in teams: A cross-domain review. *Human factors*, v. 56, n. 5, p. 911-941, 2014.
- WOHLIN, Claes et al. *Experimentation in software engineering*. Springer Science & Business Media, 2012. [[GS Search](#)]
- YANG, Hee-Dong; KANG, Hye-Ryun; MASON, Robert M. An exploratory study on meta skills in software development teams: Antecedent cooperation skills and personality for shared mental models. *European Journal of Information Systems*, v. 17, n. 1, p. 47-61, 2008. [[GS Search](#)]
- YIN, Robert K. *Case study research and applications: Design and methods*. Sage publications, 2017.