

## UMA ANÁLISE DO USO DE GROUNDED THEORY EM ENGENHARIA DE SOFTWARE

### AN ANALYSIS OF THE USE OF GROUNDED THEORY IN SOFTWARE ENGINEERING

Marcelo Werneck Barbosa\* E-mail: [mwerneck@pucminas.br](mailto:mwerneck@pucminas.br)

\*Pontifícia Universidade Católica de Minas Gerais (PUC Minas), Belo Horizonte, MG

**Resumo:** A Engenharia de Software (ES) é composta de fenômenos constituídos sócio-culturalmente e não tecnicamente. As pesquisas conduzidas nesta área devem compreender tais fenômenos. Entretanto, métodos qualitativos, adequados para este tipo de pesquisa, ainda não são comumente utilizados em ES. Apesar disso, tem-se observado um crescimento recente de publicações que adotam métodos qualitativos na ES, mais particularmente, da *Grounded Theory* (GT). A *Grounded Theory* utiliza um conjunto de procedimentos sistemáticos de coleta e análise dos dados para gerar, elaborar e validar teorias substantivas sobre fenômenos essencialmente sociais. Diante deste contexto, este trabalho teve como objetivo realizar uma revisão sistemática para caracterizar estudos na área de ES que utilizaram o método *Grounded Theory*. Os resultados da revisão sistemática mostraram um aumento no número de trabalhos usando GT em Engenharia de Software, particularmente em pesquisas que envolvem comportamentos humanos em projetos que usam métodos ágeis ou são desenvolvidos de maneira distribuída. Foi possível observar ainda que a maioria dos trabalhos não chega a enunciar uma teoria de fato ao final, mas, usam este método de pesquisa como meio para identificar categorias e conceitos no contexto pesquisado.

**Palavras-chave:** Grounded Theory. Métodos qualitativos. Engenharia de Software. Revisão Sistemática.

**Abstract:** Software Engineering (SE) is constituted by phenomena that are built socially and culturally, not technically. Research conducted in this area should comprehend such phenomena. However, qualitative methods, suited for this kind of research, are not commonly used in SE. Despite this fact, a recent growth in the number of qualitative studies in the area has been observed, particularly using Grounded Theory (GT). Grounded Theory is based on a set of systematic procedures of collecting and analyzing data in order to generate, elaborate and validate substantive theories on socially phenomena. In this context, this work aimed at performing a systematic literature review for characterizing studies on SE that used Grounded Theory. Results show an increase in the number of SE studies using GT, particularly in research involving human behavior in projects that use agile methods or are developed in a distributed way. It was observed that most studies do not phrase a specific theory actually, but use this research method as a way of identifying categories and concepts of the context studied.

**Keywords:** Grounded Theory. Qualitative Methods. Software Engineering. Systematic Review.

## 1 INTRODUÇÃO

O estudo da Engenharia de Software (ES) sempre foi complexo e difícil. Esta complexidade surge de questões técnicas, mas também do papel central do comportamento humano no processo de desenvolvimento de software (Seaman,

1999). Assim, a Engenharia de Software é um fenômeno constituído sócio-culturalmente e não tecnicamente. Logo, as pesquisas conduzidas nesta área não podem ser baseadas exclusivamente em abordagens da ciência natural, mas devem incluir um modo de se compreender fenômenos psicológicos, sociais e culturais (COLEMAN & O'CONNOR, 2007). Entretanto, métodos qualitativos ainda não são comumente utilizados em pesquisas em ES. Por isso, faz-se necessária uma maior discussão de como métodos de análise qualitativa podem ser utilizados em pesquisas em ES (CONTE, CABRAL, & TRAVASSOS, 2012).

A análise qualitativa é geralmente mais trabalhosa do que a análise quantitativa. Resultados qualitativos normalmente são considerados mais “leves” ou “nebulosos” do que resultados quantitativos, especialmente em comunidades técnicas como a de software. Eles são mais difíceis de sumarizar ou simplificar, mas assim são os problemas estudados na ES (SEAMAN, 1999). Pesquisadores em ES estão constantemente buscando melhorar a quantidade e a qualidade de seus resultados de pesquisa pelo uso de uma metodologia de pesquisa apropriada. Nos últimos tempos, tem havido um crescimento no número de pesquisadores explorando aspectos humanos e sociais da ES, muitos destes tem usado a *Grounded Theory* (GT) (HODA, NOBLE, & MARSHALL, 2012) ou Teoria Fundamentada nos Dados.

Apesar de a GT ter sido criada no final dos anos 60 e de ter sido extensivamente aplicada nas áreas de ciências sociais, o uso da GT em estudos de desenvolvimento de software não é muito comum. Pode-se dizer que na ES, estudos de GT são relativamente raros (MONTONI & ROCHA, 2010). Assim, o domínio de pesquisa em ES está para trás das ciências sociais no uso de metodologias de pesquisa qualitativas empíricas como a GT (BADREDDIN & LETHBRIDGE, 2012).

A GT é uma ferramenta disponível para gerar teorias substantivas que explicam fenômenos. Ao se trabalhar com dados qualitativos, e ao se comparar com outros métodos de pesquisa qualitativa, tais como narrativa ou etnografia, a GT gera uma teoria substantiva que explica o comportamento dos participantes como um conjunto de hipóteses integradas (ADOLPH & KRUCHTEN, 2013). A GT permite ao pesquisador começar a partir de questões de pesquisa gerais e refinar tais questões (DAGENAIS, et al., 2009). É um método bastante efetivo para o estudo do comportamento humano e, a ida ao campo com um referencial teórico

em formação permite olhar além das teorias existentes trazendo novas perspectivas e contribuindo para o desenvolvimento da sensibilidade do pesquisador (BIANCHI & IKEDA, 2008).

Dentro da ES, tem-se observado estudos envolvendo GT relativos ao processo de adoção de métodos ágeis, uma vez que a transformação para os métodos ágeis é tida como um processo sócio-técnico que envolve uma grande mudança em práticas técnicas e culturais em empresas de software (GANDOMANI, ZULZALIL, & NAFCHI, 2014). Outra área de uso potencial da GT em ES é a área de testes de software uma vez que a prática de testes é um fenômeno organizacional complexo com nenhuma teoria abrangente estabelecida que pudesse ser testada com observações empíricas (KASURINEN, TAIPALE, & SMOLANDER, 2009). Há ainda trabalhos que envolvem melhoria de processos de software (MONTONI & ROCHA, 2010) e desenvolvimento distribuído de software (DORAIRAJ, NOBLE, & MALIK, 2012) (SIVA DORAIRAJ, NOBLE, & ALLAN, 2013).

Diante deste contexto, este artigo tem como objetivo identificar e analisar trabalhos com aplicação do método de pesquisa *Grounded Theory* na Engenharia de Software. Tal análise foi realizada por meio de uma revisão sistemática em uma importante base de dados científica da área.

O restante deste artigo está organizado da seguinte maneira. A Seção 2 apresenta o referencial teórico, destacando a GT como método qualitativo de pesquisa. A Seção 3 descreve os procedimentos metodológicos utilizados neste trabalho. A Seção 4 apresenta os trabalhos encontrados sobre a aplicação da *Grounded Theory* em pesquisas de ES e uma análise destes trabalhos. A Seção 5 conclui o trabalho e apresenta direções para novas pesquisas.

## **2 PESQUISAS QUALITATIVAS E GROUNDED THEORY**

A pesquisa qualitativa está direcionada primariamente a coletar e analisar dados não numéricos com o objetivo de alcançar profundidade da informação ao invés de largura. Enquanto a pesquisa quantitativa está preocupada com perguntas como “Quanto?”, “Quantos?” ou “Com que frequência?”, a pesquisa qualitativa está relacionada com questões como “Por que?”, “Como?” e “De que maneira?” (COLEMAN & O’CONNOR, 2007). O objetivo da análise qualitativa é

sistematicamente consolidar, reduzir e interpretar dados obtidos de várias fontes e dar algum significado a eles (FRANÇA, FELIX, & SILVA, 2012).

Segundo (VERGARA, 2005), a *Grounded Theory*, ou Teoria Fundamentada em Dados, procura desenvolver uma teoria a partir da análise de dados coletados e categorizados pelo pesquisador. Trata-se de uma teoria que emerge dos dados, pois é representativa de um grupo ou situação específicos. Esta teoria desenvolve-se durante o processo de pesquisa e é um produto de contínuas interações entre análise e coleta sistemática de dados pertencentes a um fenômeno (GOULDING, 2001). É considerada uma teoria substantiva, uma teoria específica para determinado grupo ou situação e que não visa à generalização além da sua área substantiva. As teorias substantivas se contrapõem às teorias formais, que são conceituais e abrangentes.

O objetivo da GT é gerar uma teoria como um conjunto inter-relacionado de hipóteses geradas através da comparação constante de dados em níveis crescentes de abstração. A característica que distingue a GT é a ausência de um problema claro de pesquisa ou hipótese; ao contrário, o pesquisador procura revelar o problema de pesquisa ao longo do processo (PARRY, 1998). O processo da GT é um processo adaptativo de pesquisa para encontrar uma teoria emergente que não pode ser definida previamente à pesquisa. Não somente a teoria é emergente, mas também o pesquisador aprende mais e mais sobre o fenômeno envolvido e, conseqüentemente, quais dados devem ser buscados (BERRY et al., 2013).

A metodologia foi primeiramente apresentada por Glaser e Strauss no livro "*The discovery of Grounded Theory*" em 1967. Há duas frentes de uso da GT, a glasseriana e a straussiana, nomes dados em função de seus autores. A vertente defendida por Glaser dá ênfase à característica emergente do método e aos processos indutivos (CONTE et al., 2012). Glaser tem uma posição radical de que o pesquisador deve ir ao campo sem uma teoria pré-determinada para não enviesar sua interpretação (BIANCHI & IKEDA, 2008). A vertente straussiana buscou sistematizar e explicitar técnicas para operacionalização da metodologia, sendo por isso, mais disseminada (VERGARA, 2005). Strauss parte do pressuposto de que o conhecimento prévio é um meio indispensável para que os dados empíricos tenham sentido.

Segundo Strauss e Corbin (1998), o processo de análise de dados envolve etapas de codificação. Durante a codificação, são identificados códigos e categorias. Um código, também chamado de conceito, dá nome a um fenômeno de interesse para o pesquisador; abstrai um evento, objeto, ação, ou interação que tem significado para o pesquisador. Os códigos gerados podem ser classificados como: códigos de primeira ordem, diretamente associados às citações (chamados código in vivo); e códigos abstratos ou teóricos, associados a outros códigos, sem necessariamente estarem ligados a alguma citação (CONTE et al., 2012).

A aplicação da GT envolve a definição de um tema, problema e objeto de estudo; a seleção de sujeitos entrevistados e realização das entrevistas com registro de notas de campo (VERGARA, 2005). Para a definição da questão da pesquisa, deve-se formular questões abertas que induzam a análise do comportamento com toda a profundidade que se faz necessário no uso deste método. A questão também deve induzir a flexibilidade de opções de busca e análise de dados, já que a proposta do método é desenvolver teoria (BIANCHI & IKEDA, 2008).

Em seguida, dá-se início à fase que recebe o nome de codificação aberta. Nesta fase, todo o material coletado é transcrito, as frases são analisadas e as palavras-chave são selecionadas, sendo que pode haver centenas delas (BIANCHI & IKEDA, 2008). Os conceitos são agrupados em categorias e suas propriedades e dimensões identificadas. Os autores Strauss & Corbin (1998) o definem como o processo de desmembramento, exame, comparações, conceituações e categorização dos dados. Durante este processo, o pesquisador deve ir formulando várias questões para si mesmo, no que se refere ao fenômeno que está sendo estudado. Fazer comparações é o que dá aos conceitos da GT, sua precisão e especificidade, e permite que, pelo processo de comparação, os códigos possam ser agrupados um a um por suas similaridades e diferenças conceituais para formar as categorias (YUNES & SZYMANSKI, 2005).

A amostragem teórica é o processo de coleta de dados para gerar a teoria pela qual o pesquisador coleta, codifica e analisa os dados e decide quais dados coletar na próxima iteração e onde encontrá-los, de forma a desenvolver sua teoria na medida em que ela emerge (GLASER & STRAUSS, 2006). Esta amostragem é composta de indivíduos, situações e eventos idealizados para o processo de análise. Intencionalmente forma-se um grupo alvo para o estudo e, ao longo dos

Revista Produção Online, Florianópolis, SC, v. 17, n. 1, p. 26-48, jan./mar. 2017.

trabalhos, o grupo se torna teórico à medida que suporta a criação de hipóteses e desenvolve teorias (BIANCHI & IKEDA, 2008). Entrevistas, tanto formais quanto informais, são centrais no processo de coleta de dados. Pelo fato de o pesquisador usando GT não saber de antemão aonde a teoria o vai levar, apenas a amostragem inicial pode ser planejada. Baseados na teoria emergente, pesquisadores podem mudar suas listas de perguntas feitas para refletir mais de perto as categorias emergentes. Baseados no desenvolvimento de categorias, os pesquisadores podem então escolher entrevistar certos tipos de indivíduos ou buscar outras fontes de dados (COLEMAN & O'CONNOR, 2007).

O pesquisador é responsável por escolher os nomes das categorias, que devem ser nomeadas de forma abstrata e de maneira a apresentar uma “força conceitual”, ou seja, ser representativa de grupos de conceitos ou subcategorias, e não apenas representar um assunto (YUNES & SZYMANSKI, 2005). Strauss & Corbin (1998) dizem que o nome da categoria deve ser dado pensando que a categoria deve ser fácil de ser lembrada, pensada, e acima de tudo, começada a ser desenvolvida analiticamente. Nomear as categorias é considerado uma das etapas mais difíceis deste trabalho. Durante todo o processo de codificação e nomeação de categorias e subcategorias, o pesquisador deve fazer anotações e partir de ideias que surgem a respeito dos códigos, categorias e relações entre as categorias (YUNES & SZYMANSKI, 2005).

Posteriormente, os relacionamentos entre as categorias são identificados, fase chamada de codificação axial (VERGARA, 2005). Esta etapa envolve um conjunto de procedimentos na qual os dados são agrupados de novas formas, após a codificação aberta, por meio das conexões entre as categorias (YUNES & SZYMANSKI, 2005). Esta fase se faz necessária em função do grande volume de conceitos originados na fase anterior. Trata-se de analisar os conceitos selecionados, fazer uma reorganização e, destes extrair uma ideia central e suas subordinações (BIANCHI & IKEDA, 2008). A codificação axial também pode ser usada para se obter uma descrição mais dinâmica do conjunto de relações entre categorias e subcategorias. A codificação axial deve ser realizada até que o pesquisador esteja confiante que todas as categorias identificadas durante a codificação aberta foram incluídas em alguma relação categoria – subcategoria (CHAKRABORTY & DEHLINGER, 2009).

A codificação seletiva, fase seguinte, consiste no refinamento e integração dos resultados. A codificação seletiva refina todo o processo identificando a categoria central da teoria, com a qual todas as outras estão relacionadas. A categoria central deve ser capaz de integrar todas as outras categorias e expressar a essência do processo social que ocorre entre os envolvidos. Esta categoria central pode ser uma categoria existente, ou uma nova categoria pode ser criada (CONTE et al., 2012). A parte final da análise consiste em buscar o fenômeno central, ou seja, aquele que estabelecerá o elo entre as categorias. A tarefa, nesta etapa, é integrar todas as categorias para formar a teoria fundamentada nos próprios dados (YUNES & SZYMANSKI, 2005). A regra geral em GT é continuar o processo de coletar e analisar sistematicamente os dados até a saturação teórica ser atingida (BANDEIRA DE MELLO & CUNHA, 2006). Para a narrativa da teoria gerada, Strauss e Corbin (STRAUSS & CORBIN, 1998) propõem um formato específico: (A) condições levam ao (B) fenômeno, que surge num (C) contexto que leva à (D) ações e depois a (E) consequências.

Por fim, a teoria deve ser validada pela checagem dos resultados com os entrevistados e comparação da teoria gerada com as existentes. O trabalho termina com a elaboração de um relatório de pesquisa. (VERGARA, 2005) enfatiza ainda que os passos não são pré-determinados. Estudos com GT nunca são avaliados como “certos” ou “errados” e não demandam testes adicionais, pois a teoria nasce diretamente dos dados (DORAIRAJ ET AL., 2012).

### **3 PROCEDIMENTOS METODOLÓGICOS**

O trabalho, caracterizado como pesquisa exploratória, se baseou na realização de uma revisão sistemática do uso de *Grounded Theory* em pesquisas da área de Engenharia de Software. A revisão sistemática é uma ferramenta utilizada para identificar, avaliar e interpretar toda pesquisa disponível e relevante sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse, ou seja, uma forma de executar revisões abrangentes da literatura de forma não tendenciosa (KITCHENHAM, 2004).

Esse tipo de investigação disponibiliza um resumo das evidências relacionadas a uma estratégia de intervenção específica mediante a aplicação de métodos explícitos e sistematizados de busca, apreciação crítica e síntese da

informação selecionada. A revisão sistemática é útil para integrar as informações do conjunto de estudos realizados separadamente sobre determinada visão, que podem apresentar resultados conflitantes e/ou coincidentes, bem como evidenciar temas e auxiliar na orientação das intervenções (SAMPAIO & MANCINI, 2007).

O procedimento de revisão sistemática usado foi o mesmo definido por Kitchenham (KITCHENHAM, 2004). Buscou-se por artigos em inglês somente, disponíveis na base de dados eletrônica IEEEExplore. Foi utilizada somente uma base científica, pois contrariando outros trabalhos que afirmaram haver poucos trabalhos usando *Grounded Theory* na Engenharia de Software, a pesquisa em somente uma base científica já pode identificar vários trabalhos deste tipo. Para obtenção das publicações foi realizada nesta base pesquisa pelas palavras-chave “*grounded theory*” AND “*software engineering*”.

Foram retornados 204 artigos. Os artigos foram todos analisados. Os critérios de inclusão adotados foram: os artigos precisavam tratar do uso do método *Grounded Theory* em pesquisas da área de ES especificamente; os artigos precisavam ter sido publicados nos últimos 10 anos, ou seja, entre 2005 e 2015. Foram identificados 25 artigos (12,25%) que atendiam aos critérios solicitados.

A análise destes trabalhos foi realizada com o objetivo de responder às seguintes perguntas:

- A *Grounded Theory* tem sido aplicada mais frequentemente em quais disciplinas da Engenharia de Software?
- Tem havido aumento das publicações em Engenharia de Software usando o método *Grounded Theory* nos últimos anos?
- Como tem sido a aplicação das etapas previstas no processo da *Grounded Theory* nos trabalhos de Engenharia de Software? Quão completa tem sido a utilização de GT? Há o uso de software de apoio?

A próxima seção apresenta os resultados e discussões sobre o procedimento da revisão sistemática.



## 4 RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados e analisados os resultados da revisão sistemática com o objetivo de responder às questões de pesquisa colocadas ao final da seção anterior.

Para apresentar o resultado da revisão sistemática, foi elaborado o Quadro 1, que apresenta as obras identificadas que atenderam aos critérios de inclusão da revisão sistemática. Para cada obra, são apresentados: autores, ano de publicação (como referência), título do artigo, disciplina ou tema trabalhado dentro da Engenharia de Software, aspectos importantes do uso da GT como método de coleta de dados ou uso de software de apoio e ainda algumas observações sobre o trabalho. Neste campo Observações, procurou-se tentar resumir os principais achados de cada trabalho. Este resumo tem o objetivo apenas de guiar e estimular o leitor para investigar melhor estes trabalhos. Por serem trabalhos qualitativos, existe a tendência de gerar análises muito ricas e certamente qualquer síntese corre o sério risco de ignorar conclusões importantes. Sendo assim, recomenda-se a leitura de cada trabalho para se ter um maior conhecimento acerca dos achados de cada artigo. Em seguida, é realizada uma análise dos dados dos artigos mostrados no Quadro 1 com o objetivo de responder às questões de pesquisa deste trabalho.

**Quadro 1** – Compilação de trabalhos obtidos pelo processo de revisão sistemática

(continua)

REFERÊNCIA	TÍTULO	DISCIPLINA DAS	OBSERVAÇÕES SOBRE O USO DA GT	RESULTADOS E CONCLUSÕES DO TRABALHO
(SIVA DORAIRAJ et al., 2013)	Agile Software Development with Distributed Teams: Senior Management Support	Métodos ágeis; Desenvolvimento distribuído de software	Método: entrevistas. Houve referência a identificação de uma teoria (teoria enunciada).	O trabalho alega ter descrito a “Teoria de Uma Equipe” que explica como uma equipe distribuída em um desenvolvimento ágil de software adota estratégias explícitas para reduzir distâncias temporais, espaciais e socioculturais, ao mesmo tempo em que lida com fatores críticos de impacto para trabalhar coesa como uma equipe.
(GANDOMANI et al., 2014)	Agile transformation: What is it about?	Métodos ágeis	Método: entrevistas. Houve identificação de uma categoria central (codificação seletiva).	A teoria final que emergiu neste estudo foi a de transição para adoção de processos ágeis; entretanto, o trabalho não discutiu a teoria final. Vários aspectos e dimensões da transição para os métodos ágeis incluem conceitos (identificados pela GT) como pré-requisitos, facilitadores, um modelo de transição, gerenciamento da mudança e treinamento.
(KASURINEN et al., 2009)	Analysis of Problems in Testing Practices	Testes de Software	Método: entrevistas. Houve identificação de categorias e seus atributos (codificação aberta).	O trabalho procurou identificar problemas em equipes de testes e relacionar a existência de tais problemas com o tamanho da organização. Os autores identificaram que questões relativas a pessoal, processo de testes e estratégia de teste são independentes da orientação ou tamanho da empresa. A análise sugere que organizações de testes não têm benefícios em pertencer a uma organização maior.
(DAGENAIS et al., 2009)	A qualitative study on project landscapes	Gerência de Projetos; Desenvolvimento distribuído de software	Método: entrevistas. Houve identificação inicial de categorias (codificação aberta).	O trabalho procurou identificar percepções e sentimentos de profissionais quando alocados como membros de um projeto. Resultados preliminares identificaram Documentação, Experiência de colegas e Aprender por experimentação como categorias a serem estudadas.
(MONTONI & ROCHA, 2010)	Applying Grounded Theory to Understand Software Process Improvement Implementation	Melhoria de Processos de Software	Método: questionário e entrevistas. Software: Atlas.ti Foram elaboradas proposições e validadas junto a comunidade de especialistas (teoria enunciada).	Foram identificadas duas categorias centrais chamadas de “contexto institucional para implementar melhoria de processo de software” e “condução estratégia para implementar melhoria de processo de software”, representando respectivamente, os fatores contextuais que tem influência no sucesso de iniciativas de melhoria de processo e ações estratégias tomadas pelos atores da melhoria. Foram explicadas as influências de tais fatores no sucesso de iniciativas de melhoria de processo.

**Quadro 1** – Compilação de trabalhos obtidos pelo processo de revisão sistemática

(continuação)

REFERÊNCIA	TÍTULO	DISCIPLINA DAS	OBSERVAÇÕES SOBRE O USO DA GT	RESULTADOS E CONCLUSÕES DO TRABALHO
(CHAKRABORTY & DEHLINGER, 2009)	Applying the Grounded Theory Method to Derive Enterprise System Requirements	Engenharia de Requisitos de Software	Foi realizada apenas uma demonstração da GT como ferramenta para derivar requisitos.	Este trabalho apresentou o uso de um processo para utilizar GT para construir os requisitos de uma arquitetura corporativa a partir de uma grande base de dados de documentos. O trabalho deve apoiar alinhamento com requisitos e a rastreabilidade para os documentos.
(MARTINI, BOSCH, & CHAUDRON, 2014)	Architecture Technical Debt: Understanding Causes and a Qualitative Model	Arquitetura de Software	Método: workshops. Houve identificação de categorias e relacionamento entre elas (codificação axial).	Os resultados do artigo mostram os fatores e tendências atuais que influenciam o fenômeno da acumulação e recuperação de débito técnico arquitetural. Os fatores e tendências obtidos pelo estudo qualitativo revelam implicações importantes à luz das práticas gerenciais de débito técnico arquitetural.
(BADREDDIN & LETHBRIDGE, 2012) e (BADREDDIN, 2013)	Combining experiments and grounded theory to evaluate a research prototype Lessons from the umple model oriented programming technology	Engenharia de Requisitos de Software; Implementação	Método: questionário e entrevistas. Foram apresentadas conclusões da aplicação do método, mas sem apresentação de categorias ou conceitos identificados.	O objetivo principal do artigo foi guiar o desenvolvimento da linguagem Umple por meio do <i>feedback</i> de usuários e usar a percepção e experiências destes usuários para construir teorias sobre como a modelagem textual é percebida e usada na prática. A linguagem Umple foi comparada com UML e Java. Lições aprendidas do uso da <i>grounded theory</i> são apresentadas. O artigo “Empirical evaluation of research prototypes at variable stages of maturity”, também encontrado pela revisão, cita o mesmo procedimento de uso da <i>Grounded Theory</i> , não trazendo nada de novo neste sentido.
(HODA et al., 2012)	Developing a grounded theory to explain the practices of self-organizing Agile teams	Métodos ágeis	Método: entrevista e observação. Houve a identificação de uma categoria central e uma teoria enunciada como “Atos de equilíbrio” mostrando que as práticas equilibram conceitos diferentes e contrastantes	O trabalho procurou identificar práticas de equipes auto-organizadas. Por meio de um estudo qualitativo, as principais práticas adotadas por equipes auto organizadas que usam métodos ágeis foram identificadas como: liberdade concedida pela gerência sênior e expectativa com sua responsabilidade em troca; especialização entre os diferentes papéis funcionais e áreas de experiência técnica e aprendizagem contínua com o esforço de manter a equipe auto organizada.

Quadro 1 – Compilação de trabalhos obtidos pelo processo de revisão sistemática

(continuação)

REFERÊNCIA	TÍTULO	DISCIPLINA DA ES	OBSERVAÇÕES SOBRE O USO DA GT	RESULTADOS E CONCLUSÕES DO TRABALHO
(ADOLPH & KRUCHTEN, 2013)	Generating a useful theory of software engineering	Processo de desenvolvimento de software	Método utilizado não está explícito. É enunciada uma teoria chamada de “Perspectivas conciliatórias”, sem apresentação de detalhes das etapas previstas na <i>Grounded Theory</i> .	Este é um artigo de posicionamento que discute as possibilidades de haver teorias úteis na ES. A pesquisa realizada mostra que isso é possível ao se coletar dados de campo com o uso de métodos como <i>Grounded Theory</i> . Os autores afirmam ter sido possível gerar uma teoria que explica como os praticantes gerenciam o processo de desenvolvimento de software. A teoria mostra que o desenvolvimento de software é um processo social e que a criação de produtos de trabalho aceitos depende das habilidades dos indivíduos em realizar negociações e se proteger de interrupções.
(PARIZI, GANDOMANI, & NAFCHI, 2014)	Hidden facilitators of agile transition: Agile coaches and agile champions	Métodos ágeis	Método: entrevista. Autores afirmam não mostrar detalhes do processo de pesquisa por limitação de espaço, mas analisam 2 categorias centrais (codificação seletiva)	Este trabalho procura explicar dois papéis importantes na adoção de métodos ágeis – os <i>coaches</i> e <i>champions</i> . Eles foram considerados facilitadores na adoção de métodos ágeis que diretamente influenciam as pessoas envolvidas no processo de mudança.
(STOJANOV, DOBRILOVIC, & JEVTIC, 2011)	Identifying properties of software change request process: Qualitative investigation in very small software companies	Engenharia de Requisitos de Software	Método: entrevista e grupo focal. Foi identificada uma categoria central: processo de solicitação de mudança de software (codificação seletiva)	O artigo procura identificar propriedades do processo de solicitações de mudança em empresas de software muito pequenas. O processo de solicitação de mudanças é identificado como categoria central no modelo desenvolvido. Propriedades desta categoria central são: implementação ad hoc, nível de automação, <i>feedback</i> , iteratividade e duração do processo.
(VALENTIM & CONTE, 2014)	Improving a Usability Inspection Technique Based on Quantitative and Qualitative Analysis	Usabilidade	Método: análise de documento (questionário dos inspetores). Foi identificada uma categoria central (codificação seletiva)	O estudo apresentou a avaliação e evolução de uma técnica de inspeção de usabilidade (MIT 2 (v2)), através dos resultados quantitativos e qualitativos de um estudo experimental. Os estudos qualitativos focaram nos comentários de inspetores ao executar a técnica com o objetivo de identificar melhorias na mesma.

**Quadro 1** – Compilação de trabalhos obtidos pelo processo de revisão sistemática

(continuação)

REFERÊNCIA	TÍTULO	DISCIPLINA DAS	OBSERVAÇÕES SOBRE O USO DA GT	RESULTADOS E CONCLUSÕES DO TRABALHO
(SALINGER, ZIERIS, & PRECHELT, 2013)	Liberating pair programming research from the oppressive driver/observer regime	Métodos ágeis	Método: Análise de gravação de vídeo. Foram identificadas categorias e sub-categorias relacionadas (codificação axial).	Foi estudada a prática de programação em pares usando <i>Grounded Theory</i> com o objetivo de encontrar um modelo mais realista que o modelo atual de dois papéis: observador e programador. O estudo identificou a existência de outros papéis além destes dois. Estes papéis são assumidos e não assumidos gradualmente. Múltiplos papéis podem ser exercidos por uma pessoa ao mesmo tempo e algumas de suas facetas são sutis.
(STEINMACHER, CHAVES, CONTE, & GEROSA, 2014)	Preliminary Empirical Identification of Barriers Faced by Newcomers to Open Source Software Projects	Gerência de Projetos	Método: entrevista e questionário. Software: Atlas.ti Foram identificadas categorias e sub-categorias relacionadas às barreiras (codificação axial).	Este estudo buscou identificar barreiras que impedem novatos de trabalhar em projetos de software de código aberto. A análise identificou 58 barreiras agrupadas em seis diferentes categorias: diferenças culturais, características dos novatos, questões sobre receptividade, orientação, barreiras técnicas e problemas de documentação.
(KHAN, MAHRIN, & CHUPRAT, 2013)	Situational factors affecting Requirement Engineering process in Global Software Development	Engenharia de Requisitos de Software; Desenvolvimento distribuído de software	Método: revisão sistemática. Foram identificadas categorias e sub-categorias relacionadas aos fatores situacionais (codificação axial).	O trabalho procurou identificar fatores que afetam a Engenharia de Requisitos em projetos de desenvolvimento distribuído de software. Foi obtida uma lista de fatores situacionais que influenciam a Engenharia de Requisitos em ambiente de desenvolvimento distribuído de software.
(GREILER, VAN DEURSEN, & STOREY, 2012)	Test confessions: A study of testing practices for plug-in systems	Testes de software	Método: entrevista. Foram identificadas, mas não apresentadas, categorias e conceitos (codificação axial), que foram validados por especialistas.	O trabalho procurou compreender o que pensam desenvolvedores e testados ao testar sistemas que dependem de plug-ins. 25 praticantes foram entrevistados e um conjunto de práticas de testes adotadas atualmente foi identificado, assim como barreiras para sua adoção. Foi realizada uma etapa quantitativa para validar os resultados da fase qualitativa.

**Quadro 1** – Compilação de trabalhos obtidos pelo processo de revisão sistemática

(continuação)

REFERÊNCIA	TÍTULO	DISCIPLINA DAS	OBSERVAÇÕES SOBRE O USO DA GT	RESULTADOS E CONCLUSÕES DO TRABALHO
(CLEAR, HUSSAIN, & MACDONELL, 2012)	The Many Facets of Distance and Space: The Mobility of Actors in Globally Distributed Project Teams	Desenvolvimento distribuído de software	Método: análise documental (e-mails, notas de pesquisa, documentos de cursos, etc). Foram identificadas categorias de recomendações (codificação axial).	O trabalho procurou identificar características relacionadas a tempo e espaço em projetos de desenvolvimento distribuído de software. Algumas recomendações foram identificadas como: necessidade de se aumentar o espaço físico, otimizar o espaço virtual e reduzir o tamanho dos ciclos de entrega.
(WHITWORTH & BIDDLE, 2007)	The Social Nature of Agile Teams	Métodos ágeis	Método: entrevista. Foram identificadas categorias de práticas ágeis e seus atributos (codificação axial).	O estudo identificou que há forças sociais fortes em equipes ágeis que reforçam o valor de métodos ágeis. Práticas ágeis como as reuniões diárias e radiadores de informação fornecem sentimentos de segurança, controle e responsabilidade social. Foram identificados problemas como dificuldade de integração de membros, estresse e mudança de cultura.
(FRANÇA et al., 2012)	Towards an Explanatory Theory of Motivation in Software Engineering: A Qualitative Case Study of a Government Organization	Processo de desenvolvimento de software	Método: entrevista e análise documental. Software: QSR NVivo. Foi identificada uma categoria central (codificação seletiva). Não chega a ser enunciada uma teoria, mas um conjunto de proposições sobre a motivação.	Este trabalho procurou desenvolver teorias exploratórias para identificar fatores que influenciam a motivação em organizações de software. O estudo foi realizado por meio de entrevistas, estudos diários e análise documental. A teoria que explica a motivação na organização é baseada no equilíbrio entre as características motivacionais centrais, positivas ou negativas, e como a coesão e sinergia da equipe e o ambiente político externo afetam este equilíbrio.
(FRANCA, CARNEIRO, & SILVA, 2012)	Towards an Explanatory Theory of Motivation in Software Engineering: A Qualitative Case Study of a Small Software Company	Processo de desenvolvimento de software	Método: entrevista. Software: QSR NVivo. O texto alega que foi identificada uma categoria central (codificação seletiva). Não chega a ser enunciada uma teoria, mas um conjunto de proposições.	Este trabalho procurou desenvolver teorias exploratórias para identificar fatores que influenciam a motivação no contexto de uma pequena empresa privada de software. Aprendizado e crescimento foram identificados como os principais impulsionadores da motivação, que por sua vez, aumenta o compromisso e cria condições para melhor desempenho da equipe.

**Quadro 1** – Compilação de trabalhos obtidos pelo processo de revisão sistemática

(conclusão)

REFERÊNCIA	TÍTULO	DISCIPLINA DAS	OBSERVAÇÕES SOBRE O USO DA GT	RESULTADOS E CONCLUSÕES DO TRABALHO
(DORAIRAJ et al., 2012)	Understanding lack of trust in distributed agile teams: a grounded theory study	Métodos ágeis; Desenvolvimento distribuído de software	Método: entrevista. Foram identificadas categorias de causas e consequências para a falta de confiança, identificada como categoria central (codificação seletiva).	O trabalho identificou que a confiança é uma preocupação importante para equipes ágeis, particularmente as distribuídas geograficamente. A natureza auto-organizada destas equipes aumenta a importância da confiança no desenvolvimento ágil de software.
(COLEMAN & O'CONNOR, 2007)	Using grounded theory to understand software process improvement: A study of Irish software product companies	Melhoria de processos de software	Método: entrevista Software: Atlas.ti Foi encontrada uma categoria central (codificação seletiva). Um framework de categorias foi apresentado como teoria enunciada.	Por meio de entrevistas com 21 empresas, o estudo identificou fatores que influenciam a forma como a melhoria de processo de software é estabelecida. Foram identificados dois temas – Formação do processo e Evolução do processo – e uma categoria central – Custo dos processos. Categorias relacionadas a cada tema ou à categoria central foram identificadas e relacionadas.
(2012)	Using grounded theory to understand testing engineers' soft skills of third-party software testing centers	Testes de software	Método: entrevista e grupo focal. Software: Atlas.ti Não foi possível identificar resultados do uso da <i>Grounded Theory</i> .	O trabalho procurou compreender as habilidades “soft” de engenheiros de testes e sua influência em projetos. O trabalho não apresenta nenhuma evidência do uso da <i>Grounded Theory</i> como conceitos ou categorias identificados. A conclusão indica direções para trabalhos futuros.

Fonte: Dados da pesquisa

Para responder a pergunta sobre em quais áreas ou disciplinas da Engenharia de Software tem sido aplicada mais frequentemente a *Grounded Theory*, foi elaborada a Tabela 1. Como pode ser observado, os temas mais frequentes são o desenvolvimento de software com métodos ágeis, com 7 artigos, e o desenvolvimento distribuído de software, com 5 artigos. Os resultados corroboram o que afirma Hoda, Noble e Marshall (HODA et al., 2012) que dizem ser a *Grounded Theory* uma metodologia de pesquisa útil para explorar os aspectos humanos e sociais da Engenharia de Software, pois permite capturar efetivamente as experiências de como as pessoas constroem software. Pode-se observar que os dois temas que mais utilizam GT possuem questões muito presentes quanto aos aspectos humanos. Os métodos ágeis são centrados nas pessoas e nas equipes. Comunicação, colaboração e coordenação entre desenvolvedores, clientes e gerentes são aspectos importantes para o sucesso de um projeto ágil. Por sua vez, projetos desenvolvidos de maneira distribuída encaram desafios como manter a equipe unida apesar das separações no tempo e espaço e diferenças culturais (Siva Dorairaj et al., 2013).

**Tabela 1** – Disciplinas da Engenharia de Software mais frequentes nos trabalhos encontrados.

<b>Disciplinas / Temas da Engenharia de Software</b>	<b>Número de artigos relacionados</b>
Métodos ágeis	7
Desenvolvimento distribuído de software	5
Requisitos de Software	4
Testes de Software	3
Processo de desenvolvimento de software	3
Gerência de Projetos	2
Melhoria de Processos de Software	2
Usabilidade	1
Arquitetura de Software	1
Implementação	1

**Fonte:** Dados da pesquisa

Para responder a pergunta se tem havido aumento das publicações em Engenharia de Software usando o método *Grounded Theory* nos últimos anos, foi elaborada a Tabela 2. De acordo com esta tabela, pode-se notar um aumento da publicação de artigos na área de ES com o método de GT na base pesquisada.



**Tabela 2** – Ano de publicação dos trabalhos encontrados pela revisão sistemática

<b>Ano de publicação</b>	<b>Número de artigos em cada ano</b>
2007	2
2008	0
2009	3
2010	1
2011	1
2012	8
2013	5
2014	5
2015	0

**Fonte:** Dados da pesquisa

Por fim, foi objetivo deste trabalho caracterizar o uso da GT nos trabalhos encontrados, considerando as etapas e técnicas de coleta de dados utilizadas e documentadas e ainda o uso de software de apoio. Coleman e O'Connor (COLEMAN & O'CONNOR, 2007) afirmam que entrevista, tanto formal quanto informal, é a principal técnica de coleta de dados usada em pesquisas com GT. Tal afirmação foi corroborada pelos resultados da revisão sistemática realizada que mostrou que dos 25 trabalhos analisados, 17 explicitamente disseram ter usado entrevistas, conforme mostrado na Tabela 3. Outras técnicas utilizadas incluem questionários, análise documental e grupo focal. Uma observação interessante é a variedade de técnicas de coleta de dados possível de ser utilizada no contexto de uma pesquisa com GT.

**Tabela 3** – Técnicas utilizadas nos trabalhos encontrados pela revisão sistemática

<b>Nome da técnica de coleta de dados utilizada</b>	<b>Número de artigos reportando uso da técnica</b>
Entrevistas	17
Questionário	3
Análise documental	3
Grupo focal	2
Workshops	1
Observação	1
Análise de gravação de vídeo	1
Revisão sistemática	1
Não identificada	2

**Fonte:** Dados da pesquisa

Como afirma Coleman & O'Connor (2007), a apresentação dos resultados obtidos com o uso da GT é um desafio ao pesquisador em termos de estrutura, nível de detalhe apresentado e como os dados são mostrados para evidenciar os relacionamentos entre as categorias identificadas. Os artigos obtidos pelo processo de revisão sistemática foram lidos e analisados quanto ao uso das

etapas da GT, segundo reportado pelos próprios autores de cada trabalho. A Tabela 4 exhibe o número de artigos que reportou ter alcançado cada uma das etapas do método, na ordem prevista de sua realização (codificação aberta, seguida por axial e seletiva, e por fim, a teoria enunciada). Pode-se observar que poucos trabalhos procuraram enfatizar ter chegado a enunciar uma teoria. A maioria dos trabalhos parece ter utilizado a GT para organizar as categorias e conceitos encontrados nos estudos. Quando os trabalhos chegaram a enunciar ter identificado uma categoria central, foi considerada a conclusão da etapa de codificação seletiva. Quando os trabalhos reportaram apenas a identificação de conceitos e categorias e seus relacionados, considerou-se que a etapa de codificação axial foi concluída. Dois trabalhos não apresentaram evidência da conclusão de nenhuma etapa do processo.

**Tabela 4** – Etapas da *Grounded Theory* alcançadas nos trabalhos encontrados

<b>Etapa da Grounded Theory alcançada (reportada)</b>	<b>Número de artigos</b>
Codificação aberta	2
Codificação axial	8
Codificação seletiva	8
Teoria enunciada	4
Não identificada	3

**Fonte:** Dados da pesquisa

Por fim, buscou-se entender se os trabalhos que aplicam GT em ES utilizaram algum software de apoio. Por ser um método de pesquisa que gera muitas informações, um software pode ajudar o pesquisador a organizar e registrar não só o processo de coleta de dados, mas como a organização e análise destes dados em conceitos e categorias. Dos 25 trabalhos recuperados, 6 afirmaram ter usado software de apoio, sendo que 4 deles utilizaram o Atlas.ti e 2 o QSR N Vivo. Segundo Lili Yu et al. (2012), o Atlas.ti é uma ferramenta desenhada especificamente para o uso com a GT. Já Coleman & O'Connor (2007) dizem que o Atlas.ti permite a conexão, pesquisa e ordenação de dados. O software permite ainda registrar transcrições de entrevistas, gerenciar a lista de códigos e memorandos, gerar famílias de códigos relacionados e criar diagramas para representar estes códigos e categorias.

## 5 CONCLUSÕES

Apesar da afirmação de que na ES estudos de GT são relativamente raros (MONTONI & ROCHA, 2010), nos últimos tempos, tem havido um crescimento no número de pesquisadores explorando aspectos humanos e sociais da Engenharia de Software e que muitos destes têm usado a GT (HODA et al., 2012).

Este trabalho corroborou esta última afirmação ao realizar uma revisão sistemática com o objetivo de analisar e caracterizar o uso da *Grounded Theory* em pesquisas da área de ES. Foram identificados na base de publicações IEEEExplore 25 trabalhos que atendiam aos critérios de inclusão da revisão sistemática, após uma pesquisa realizada que retornou 204 possíveis trabalhos. A leitura dos trabalhos identificados foi realizada com o objetivo de verificar quais temas da ES são trabalhados com o uso de GT e caracterizar como os pesquisadores têm usado este método.

Como temas mais pesquisados com o uso de GT foram identificados trabalhos que envolvem projetos de desenvolvimento de software com métodos ágeis e ainda sobre desenvolvimento distribuído de software. A técnica de coleta de dados mais utilizada, segundo os resultados da revisão sistemática, foi entrevista, utilizada em 68% dos trabalhos analisados. O uso do software Atlas.ti foi reportado como apoio à realização de GT em 4 dos 25 trabalhos estudados.

Por fim, procurou-se caracterizar como os pesquisadores têm utilizado o método de pesquisa GT. Observou-se que, na maioria dos artigos, os pesquisadores reportaram explicitamente ter executado o processo somente até a fase de codificação axial ou ainda até a etapa de codificação seletiva, quando uma categoria central é identificada. Apenas em 4 artigos foi reportado explicitamente ter se chegado a enunciar uma teoria. Claramente, por se tratar de um método que procura uma teoria que emerja dos dados, e por não haver compromisso em se chegar a tal teoria, entende-se que não necessariamente há falhas na utilização deste método. Entretanto, pode-se argumentar que a diferenciação para outros métodos qualitativos de pesquisa, como a análise do conteúdo, pode ficar menos evidente.

Uma limitação desta pesquisa está no fato de a revisão sistemática ter sido realizada consultando somente uma base científica, a IEEEExplore. Entretanto, tal fato é justificado pelo número razoável de artigos encontrados que atendiam aos

critérios de inclusão da pesquisa (25 trabalhos), pela importância científica desta base na área de ES e de ter sido possível já com a análise destes artigos caracterizar o uso da GT em pesquisas de ES e responder às perguntas de pesquisa deste trabalho.

Outra ameaça às conclusões reportadas aqui está no fato de a leitura dos resultados dos artigos retornados pela revisão sistemática ter sido feito sob risco de viés de interpretação do próprio pesquisador. Em alguns textos, os autores não são explícitos em relação ao processo adotado da GT ou ainda não mostram exemplos de conceitos e categorias obtidos em seus trabalhos. Como esta pesquisa procurou caracterizar os trabalhos quanto às etapas utilizadas da metodologia da GT, pode ter havido desvios entre a interpretação deste pesquisador e a execução das pesquisas em GT de fato, considerando a possível existência de dados não explícitos nos trabalhos analisados.

Como trabalhos futuros, pretende-se expandir esta revisão sistemática incluindo mais bases científicas e ampliando as palavras-chave usadas na busca. Pretende-se também aprofundar o trabalho no estudo do uso de *Grounded Theory* especificamente em projetos que usam métodos ágeis, uma vez que este foi o principal tema pesquisado nos trabalhos encontrados nesta revisão.

## REFERÊNCIAS

ADOLPH, S.; KRUCHTEN, P. Generating A Useful Theory of Software Engineering. In: 2ND WORKSHOP ON A GENERAL THEORY OF SOFTWARE ENGINEERING, 2013, San Francisco. **Anais...** San Francisco: [s.n.], 2013. <https://doi.org/10.1109/GTSE.2013.6613870>

BADREDDIN, O. Empirical evaluation of research prototypes at variable stages of maturity. In: 2ND INTERNATIONAL WORKSHOP ON USER EVALUATIONS FOR SOFTWARE ENGINEERING RESEARCHERS (USER), 2013, [S.I.] . **Anais...** [S.I.]: IEEE, p. 1–4, 2013. <https://doi.org/10.1109/USER.2013.6603076>

BADREDDIN, O.; LETHBRIDGE, T. C. Combining experiments and grounded theory to evaluate a research prototype: Lessons from the umple model-oriented programming technology. In: 1ST INTERNATIONAL WORKSHOP ON USER EVALUATION FOR SOFTWARE ENGINEERS RESEARCHERS (USER), 2012, [S.I.] . **Anais...** [S.I.]: IEEE, p. 1–4, 2012. <https://doi.org/10.1109/USER.2012.6226575>

BANDEIRA DE MELLO, R.; CUNHA, C. **Grounded theory**: pesquisa qualitativa em estudos organizacionais: paradigmas, estratégias e métodos. São Paulo: Saraiva, 2006.

BERRY, D. M. et al. Requirements specifications and recovered architectures as grounded theories. **The Grounded Theory Review**, v. 12, n. 1, 2013.

BIANCHI, E. M. P. G.; IKEDA, A. A. Usos e aplicações da grounded theory em administração. **Revista Eletrônica de Gestão Organizacional**, v. 6, n. 2, p. 231–248, 2008.

FRANÇA, A. C.; FELIX, A. De L. C.; SILVA, F. Q. B. Towards an explanatory theory of motivation in software engineering: a qualitative case study of a government organization. In: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 2012, , [S.l.] . **Anais...** [S.l.]: [s.n.], 2012. <https://doi.org/10.1049/ic.2012.0010>

CHAKRABORTY, S.; DEHLINGER, J. Applying the grounded theory method to derive enterprise system requirements. In: 10TH ACIS INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCES, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING (SNPD), 2009, , [S.l.] . **Anais...** [S.l.]: IEEE, 2009. p. 333–338. <https://doi.org/10.1109/SNPD.2009.102>

CLEAR, T.; HUSSAIN, W.; MACDONELL, S. G. The many facets of distance and space: the mobility of actors in globally distributed project teams. In: 7TH INTERNATIONAL CONFERENCE ON GLOBAL SOFTWARE ENGINEERING (ICGSE), 2012, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 144–148, 2012. <https://doi.org/10.1109/ICGSE.2012.14>

COLEMAN, G.; O'CONNOR, R. Using grounded theory to understand software process improvement: A study of Irish software product companies. **Information and Software Technology**, v. 49, n. 6, p. 654–667, jun. 2007. <https://doi.org/10.1016/j.infsof.2007.02.011>

CONTE, T.; CABRAL, R.; TRAVASSOS, G. H. Aplicando Grounded Theory na Análise Qualitativa de um Estudo de Observação em Engenharia de Software – Um Relato de Experiência. In: V WORKSHOP UM OLHAR SOCIOTÉCNICO SOBRE A ENGENHARIA DE SOFTWARE – WOSES, 2012, , [S.l.] . **Anais...** [S.l.]: [s.n.], 2012.

DAGENAIS, B. et al. A qualitative study on project landscapes. In: ICSE WORKSHOP ON COOPERATIVE AND HUMAN ASPECTS ON SOFTWARE ENGINEERING (CHASE), 2009, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 32–35, 2009. <https://doi.org/10.1109/CHASE.2009.5071407>

DORAIRAJ, S.; NOBLE, J.; ALLAN, G. Agile Software Development with Distributed Teams: Senior Management Support. In: IEEE 8TH INTERNATIONAL CONFERENCE ON GLOBAL SOFTWARE ENGINEERING, 2013, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 197–205, 2013. <https://doi.org/10.1109/ICGSE.2013.33>

DORAIRAJ, S.; NOBLE, J.; MALIK, P. Understanding lack of trust in distributed agile teams: a grounded theory study. In: 16TH INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING (EASE), 2012, , [S.l.] . **Anais...** [S.l.]: IET, p. 81–90, 2012. <https://doi.org/10.1049/ic.2012.0011>

FRANCA, A. C. C.; CARNEIRO, D. E. S.; SILVA, F. Q. B. Da. Towards an Explanatory Theory of Motivation in Software Engineering: A Qualitative Case Study of a Small Software Company. In: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 2012, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 61–70, 2012. <https://doi.org/10.1109/SBES.2012.28>

GANDOMANI, T. J.; ZULZALIL, H.; NAFCHI, M. Z. Agile transformation: What is it about? In: 8TH MALAYSIAN SOFTWARE ENGINEERING CONFERENCE (MYSEC), 2014, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 240–245, 2014. <https://doi.org/10.1109/MySec.2014.6986021>

GLASER, B. G.; STRAUSS, A. L. **The discovery of grounded theory: strategies for qualitative research**. New Bnmswick: Aldine Transaction, 2006.

GOULDING, C. Grounded Theory: A magical formula or a potential nightmare. **The Marketing Review**, v. 2, p. 21–34, 2001. <https://doi.org/10.1362/1469347012569409>

GREILER, M.; DEURSEN, A. VAN; STOREY, M.-A. Test confessions: A study of testing practices for plug-in systems. In: 34TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2012, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 244–254, 2012. <https://doi.org/10.1109/icse.2012.6227189>

HODA, R.; NOBLE, J.; MARSHALL, S. Developing a grounded theory to explain the practices of self-organizing Agile teams. **Empirical Software Engineering**, v. 17, n. 6, p. 609–639, dez. 2012. <https://doi.org/10.1007/s10664-011-9161-0>

KASURINEN, J.; TAIPALE, O.; SMOLANDER, K. Analysis of Problems in Testing Practices. In: ASIA-PACIFIC SOFTWARE ENGINEERING CONFERENCE (APSEC), 2009, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 309–315, 2009. <https://doi.org/10.1109/APSEC.2009.17>

KHAN, H. H.; MAHRIN, M. N. Bin; CHUPRAT, S. Bt. Situational factors affecting Requirement Engineering process in Global Software Development. In: IEEE CONFERENCE ON OPEN SYSTEMS (ICOS), 2013, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 118–122, 2013. <https://doi.org/10.1109/ICOS.2013.6735059>

KITCHENHAM, B. **Procedures for performing systematic reviews**. Keele University Technical Report TR/SE-0401.

LILI YU et al. Using grounded theory to understand testing engineers' soft skills of third-party software testing centers. In: 3RD INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE (ICSESS), 2012, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 403–406, 2012. <https://doi.org/10.1109/ICSESS.2012.6269490>

MARTINI, A.; BOSCH, J.; CHAUDRON, M. Architecture Technical Debt: Understanding Causes and a Qualitative Model. In: 40TH EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS (SEAA), 2014, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 85–92, 2014. <https://doi.org/10.1109/SEAA.2014.65>

MONTONI, M. A.; ROCHA, A. R. Applying Grounded Theory to Understand Software Process Improvement Implementation. In: 7TH INTERNATIONAL CONFERENCE ON THE QUALITY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY (QUATIC), 2010, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 25–34, 2010. <https://doi.org/10.1109/QUATIC.2010.20>

PARIZI, R. M.; GANDOMANI, T. J.; NAFCHI, M. Z. Hidden facilitators of agile transition: Agile coaches and agile champions. In: 8TH MALAYSIAN SOFTWARE ENGINEERING CONFERENCE (MYSEC), 2014, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 246–250, 2014. <https://doi.org/10.1109/MySec.2014.6986022>

PARRY, W. Grounded theory and social process: a new direction for leadership research. **Leadership Quarterly**, v. 9, n. 1, p. 85–106, 1998. [https://doi.org/10.1016/S1048-9843\(98\)90043-1](https://doi.org/10.1016/S1048-9843(98)90043-1)

SALINGER, S.; ZIERIS, F.; PRECHELT, L. Liberating pair programming research from the oppressive driver/observer regime. In: 35TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2013, , [S.I.] . **Anais...** [S.I.]: IEEE, p. 1201–1204, 2013. <https://doi.org/10.1109/ICSE.2013.6606678>

SAMPAIO, R. F.; MANCINI, M. C. Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. **Revista Brasileira de Fisioterapia**, v. 11, n. 1, p. 83–89, fev. 2007. <https://doi.org/10.1590/S1413-35552007000100013>

SEAMAN, C. B. Qualitative Methods in Empirical Studies of Software Engineering. **IEEE TRANSACTIONS ON SOFTWARE ENGINEERING**, v. 25, n. 4, p. 557–572, 1999. <https://doi.org/10.1109/32.799955>

STEINMACHER, I. *et al.* Preliminary Empirical Identification of Barriers Faced by Newcomers to Open Source Software Projects. *In: BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING (SBES)*, 2014, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 51–60, 2014. <https://doi.org/10.1109/SBES.2014.9>

STOJANOV, Z.; DOBRILOVIC, D.; JEVTIC, V. Identifying properties of software change request process: Qualitative investigation in very small software companies. *In: 9TH INTERNATIONAL SYMPOSIUM ON INTELLIGENT SYSTEMS AND INFORMATICS (SISY)*, 2011, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 47–52, 2011. <https://doi.org/10.1109/SISY.2011.6034369>

STRAUSS, A.; CORBIN, J. **Basics of qualitativa research: techniques and procedures for developing grounded theory**. London: Sage Publications, 1998.

VALENTIM, N. M. C.; CONTE, T. Improving a Usability Inspection Technique Based on Quantitative and Qualitative Analysis. *In: BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING (SBES)*, 2014, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 171–180, 2014. <https://doi.org/10.1109/SBES.2014.23>

VERGARA, S. C. **Métodos de pesquisa em administração**. 1. ed. São Paulo: Atlas, 2005.

WHITWORTH, E.; BIDDLE, R. The Social Nature of Agile Teams. *In: AGILE CONFERENCE (AGILE)*, 2007, , [S.l.] . **Anais...** [S.l.]: IEEE, p. 26–36, 2007. <https://doi.org/10.1109/AGILE.2007.60>

YUNES, M. Â. M.; SZYMANSKI, H. Entrevista Reflexiva & Grounded-Theory: Estratégias Metodológicas para Compreensão da Resiliência em Famílias. **Interamerican Journal of Psychology**, v. 39, n. 3, 2005.



Artigo recebido em 22/01/2016 e aceito para publicação em 15/12/2016

DOI: <http://dx.doi.org/10.14488/1676-1901.v17i1.2326>