

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO E
SISTEMAS

DAYANE FILAKOSKI DOS SANTOS

**ANÁLISE DA REDUÇÃO DE EXCESSOS NO DESENVOLVIMENTO DE
SOFTWARE ATRAVÉS DA METODOLOGIA LEAN E ÁGIL JÁ APLICADAS EM
EMPRESAS DO ARRANJO PRODUTIVO LOCAL DE TECNOLOGIA DO
SUDOESTE DO PARANÁ**

DISSERTAÇÃO

PATO BRANCO

2018

DAYANE FILAKOSKI DOS SANTOS

ANÁLISE DA REDUÇÃO DE EXCESSOS NO DESENVOLVIMENTO DE
SOFTWARE ATRAVÉS DA METODOLOGIA LEAN E ÁGIL JÁ APLICADAS EM
EMPRESAS DO ARRANJO PRODUTIVO LOCAL DE TECNOLOGIA DO
SUDOESTE DO PARANÁ

Dissertação apresentada como requisito parcial à
obtenção do título de Mestre em Engenharia de
Produção e Sistemas, do Programa de Pós-
Graduação em Engenharia de Produção e Sistemas
da Universidade Tecnológica Federal do Paraná,
Câmpus Pato Branco.

Orientador: Prof. Dr. Gilson Adamczuk Oliveira

Pato Branco

2018

S327a Santos, Dayane Filakoski dos.
Análise da redução de excessos no desenvolvimento de software através da metodologia Lean e ágil jpa aplicadas nas empresas do arranjo produtivo local de tecnologia do sudoeste do Paraná / Sayane Filakoski dos Santos. – 2018.
79 f. : il. ; 30 cm.

Orientador: Prof. Dr. Gilson Adamczuk Oliveira
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná.
Programa de Pós-Graduação em Engenharia de Produção e Sistemas.
Pato Branco, PR, 2018.
Bibliografia: f. 71 - 79.

1. Software - Desenvolvimento. 2. Produção enxuta. 3. Engenharia de produção. I. Oliveira, Gilson Adamczuk, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas. III. Título.

CDD 22. ed. 670.42

Ficha Catalográfica elaborada por
Suélem Belmudes Cardoso CRB9/1630
Biblioteca da UTFPR Campus Pato Branco



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Diretoria de Pesquisa e Pós-Graduação
*Programa de Pós-Graduação em Engenharia de Produção e
Sistemas*



TERMO DE APROVAÇÃO DE DISSERTAÇÃO Nº 28

A Dissertação de Mestrado intitulada “**Análise da redução de excessos no desenvolvimento de software através da metodologia Lean e ágil já aplicadas em empresas do arranjo produtivo local de tecnologia do sudoeste do Paraná**”, defendida em sessão pública pela candidata **Dayane Filakoski dos Santos**, no dia 29 de janeiro de 2018, foi julgada para a obtenção do título de Mestre em Engenharia de Produção e Sistemas, área de concentração Gestão dos Sistemas Produtivos, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

BANCA EXAMINADORA:

Prof. Dr. Marcelo Gonçalves Trentin - Presidente - UTFPR

Prof. Dr. Guilherme Luz Tortorella - UFSC

Prof. Dr. Sergio Luiz Ribas Pessa - UTFPR

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Pato Branco, 29 de janeiro de 2018.

Carimbo e assinatura do Coordenador do Programa.

Dedico este trabalho aos meus pais José Leonço dos Santos e Tereza Filakoski.

Minhas maiores fontes de inspiração e coragem.

AGRADECIMENTOS

Primeiramente agradeço à Deus, autor da vida, pelo milagre de não permitir que as controvérsias me fizessem desistir.

Gratidão aos meus pais, José Leonço dos Santos e Tereza Filakoski, pelo incentivo diário, por sempre acreditarem em mim e torcerem pelo meu sucesso.

Ao meu orientador, Gilson Adamczuk Oliveira, pela paciência, por não desistir de mim, mesmo quando eu já não tinha mais forças para continuar.

Ao professor Marcelo Gonçalves Trentin por sempre colaborar quando precisei, e aceitar participar da banca examinadora.

Aos professores, Guilherme Tortorella e Sérgio Luiz Ribas Pessa, pelo aceite ao convite de participação da banca examinadora.

Ao professor Guilherme Tortorella também por providenciar data e local para a banca.

À secretária do programa, Adriani Michelin, pelas orientações e cordialidade em todos os momentos que busquei auxílio.

SANTOS, Dayane F.; **ANÁLISE DA REDUÇÃO DE EXCESSOS ATRAVÉS DA METODOLOGIA LEAN E ÁGIL JÁ APLICADAS EM EMPRESAS DESENVOLVEDORAS DE SOFTWARE NO ARRANJO PRODUTIVO LOCAL DE TECNOLOGIA DO SUDOESTE DO PARANÁ**. 2018. 79 folhas. Dissertação (Mestrado em Engenharia de Produção e Sistemas) – Universidade Tecnológica Federal do Paraná, Pato Branco.

RESUMO

O crescimento da competitividade entre as empresas de desenvolvimento de *software*, trouxe uma maior pressão no aumento da produtividade, adaptação rápida à mudanças e entregas cada vez mais rápidas de produtos cada vez melhores e mais baratos. Baseado nesse contexto, surgiram as metodologias ágeis, cujo enfoque principal está relacionado à adaptabilidade com o propósito de entregar produtos com maior valor agregado no menor espaço de tempo possível. A fim de preencher algumas lacunas de gestão ainda apresentadas pelas metodologias ágeis, houve um desdobramento da Filosofia Lean, dentro da perspectiva do desenvolvimento de *software*, o *Lean Software Development* (LSD). As implementações de LSD são recentes, e a partir de pesquisas em diversos periódicos, foi possível observar que pouco se falou até então sobre essas aplicações em empresas de desenvolvimento de *software* inseridas em diferentes contextos. Este estudo também se preocupou com os excessos produzidos no desenvolvimento de *software*, acima e além das necessidades ou recursos disponíveis. A literatura possui poucas referências nesta questão, sendo o assunto importante para novas discussões. Tratando-se do Arranjo Produtivo Local (APL) de Tecnologia da Informação do Sudoeste do Paraná, há oportunidades de pesquisa a serem desenvolvidas. Apesar das metodologias ágeis já estarem inseridas, ainda existem diversas lacunas da sua aplicação e processos ainda sem gestão. A partir dessa concepção de ideias, este trabalho se propôs a levantar as possibilidades de inserção dessas metodologias nos processos já trabalhados nas empresas do APL, analisando as áreas de processos que sofrem excessos no

desenvolvimento, bem como convergindo para a ideia de melhorar o desempenho e produtividade das empresas analisadas. Para essa finalidade, foi realizado estudo de caso múltiplo, abrangendo três empresas de desenvolvimento de *software*. Com o levantamento de dados durante a pesquisa, observou-se crescimento emergente das metodologias ágeis no setor. As três empresas analisadas já buscam inserir ferramentas e métodos ágeis e/ou *Lean* no processo. No entanto, esse crescimento ainda é embrionário em sua maioria. Apenas uma das empresas teve contato com a filosofia *Lean* para *software*, enquanto as outras duas inseriram métodos ágeis há apenas um ano, uma com uma evolução e integração da equipe bem formalizada, a outra com um pouco menos de consistência. É interessante salientar que os excessos ainda ocorrem mesmo quando o uso da metodologia *Lean* ou Ágil já está inserida. Sendo possível observar melhorias, mas ainda lacunas em que o processo pode ser mais refinado e os desperdícios menos impactantes ou até inexistentes.

Palavras-chave: Desenvolvimento de *Software*, *Lean*, Ágil, LSD, Excesso de Desenvolvimento de *Software*.

SANTOS, Dayane F.; **ANALYSIS OF THE REDUCTION OF EXCESSES IN THE DEVELOPMENT OF SOFTWARE THROUGH LEAN AND AGILE METHODOLOGY ALREADY APPLIED IN COMPANIES OF THE TECHNOLOGY CLUSTER OF THE PARANÁ SOUTHWEST**. 2018. 79 folhas. Dissertation (Master's Degree in Production Engineering and Systems) – Federal Technology University of Paraná. Pato Branco.

ABSTRACT

The growth of competitiveness between software development companies has brought greater pressure on increasing productivity, quickly adaptation to changes and ever faster deliveries of ever better and cheaper products. Based on this context agile methodologies have emerged, with the mainly focus related to adaptability with the purpose of delivering products with higher added value in the shortest time possible. In order to fill some management gaps still presented by the agile methodologies, there was an unfolding from Lean Philosophy, inside of the perspective of software development, the Lean Software Development (LSD). The LSD implementations are recent, and through surveys at various periodic intervals, it was possible to observe that little has been said about applications in software development companies inserted in different contexts. This study also concerned itself with the excesses produced in software development, above and beyond the needs or resources available. The literature has few references in this issue, being the subject important for new discussions. In the case of the Local Productive Arrangement (LPA) of Information Technology of the SouthWest of Paraná, there are research opportunities to be developed. Although agile methodologies are already inserted, there're still several gaps in its application and processes that aren't managed yet. Based on this conception of ideas, this work aims to raise the possibilities of insertion of these methodologies in the processes already worked in the LPA companies, analyzing as areas of processes that suffer excesses in the development as well as converging on the idea of improving the performance and production of the companies analyzed. For this purpose, a multiple case study was carried out, covering three software development companies. With the data collection during the research, there was an

emerging growth of the agile methodologies in the sector. The three companies analyzed already seek to insert agile and / or Lean tools and methods in the process. However, this growth is still embryonic in its majority. Only one of the companies had contact with the Lean philosophy for software, while the other two introduced agile methods only a year ago, one with a well-formed team evolution and integration, the other with a little less consistency. It's interesting to note that excesses still occur even when the use of the Lean or Agile methodology is already inserted. It's possible to observe improvements but still gaps, in which the process can be more refined and the wastes less impacting or even none existent.

Keywords: Software Development, Lean, Agile, LSD, Development Software over.

LISTA DE FIGURAS

Figura 1: Etapas Proknow – C.....	32
Figura 2: Procedimento para seleção de artigo.....	33
Figura 3: Relevância dos periódicos do Portfólio.	40
Figura 4 Análise combina de autores	41
Figura 5: Lente 1 - Filosofia Enxuta (<i>Lean</i>)	43
Figura 6 Fluxo metodológico dos estudos de caso	58

LISTA DE QUADROS

Quadro 1 Resumo dos capítulos 2 e 3 desenvolvidos na pesquisa. Fonte: o autor. .24	
Quadro 2 Contrastes entre atributos do LSD e o Modelo Tradicional de <i>Software</i> (BELL <i>et al.</i> , 2011)	31
Quadro 3 Portfólio Bibliográfico	38
Quadro 4 Desperdícios na área de desenvolvimento de <i>software</i> : Fonte: o autor. ...	54
Quadro 5 Categorias e termos de desperdício. Adaptado de Shmueli <i>et al.</i> (2017)..	57
Quadro 6 Análise Resumida dos Estudos de Caso	66

LISTA DE SIGLAS

AN	- Além das Necessidades
AP	- Além do Planejamento
APL	- Arranjo Produtivo Local
AR	- Além dos Recursos
DSDM	- <i>Dynamic Software Development Method</i>
JIT	- <i>Just-in-Time</i>
LabMCDA	- Laboratório de Metodologias Multicritério em Apoio à Decisão
LSD	- <i>Lean Software Development</i>
MIT	- <i>Massachusetts Institute of Technology</i>
PMES	- Pequenas e Médias Empresas
Proknow-C	- <i>Knowledge Development Process-Constructivist</i>
RUP	- <i>Rational Unified Process</i>
TI	- Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	16
1.1	CONTEXTUALIZAÇÃO	16
1.2	PROBLEMA DE PESQUISA	19
1.3	OBJETIVOS	20
1.3.1	Objetivo Geral	20
1.3.2	Objetivos Específicos	20
1.4	JUSTIFICATIVA	21
1.5	ESTRUTURA DA DISSERTAÇÃO	22
1.6	METODOLOGIA DA DISSERTAÇÃO	22
2	PROCESSO ENXUTO PARA DESENVOLVIMENTO DE SOFTWARE: UM MAPEAMENTO DA LITERATURA	25
2.1	INTRODUÇÃO	25
2.2	REFERENCIAL TEÓRICO	27
2.3	METODOLOGIA DA PESQUISA BIBLIOMÉTRICA	32
2.4	ANÁLISE DOS ARTIGOS DO PORTFÓLIO BIBLIOGRÁFICO E SUAS REFERÊNCIAS	39
2.5	ANÁLISE SISTÊMICA	41
2.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO 2	46
3	ANÁLISE DE DESPERDÍCIOS EM EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE DO SUDOESTE DO PARANÁ	47
3.1	INTRODUÇÃO	48
3.2	REFERENCIAL TEÓRICO	50
3.3	METODOLOGIA DE PESQUISA	55
3.4	ESTUDOS MULTI-CASO	59
3.5	RESULTADOS E DISCUSSÕES	64
3.6	LIMITAÇÕES E PESQUISAS FUTURAS	68

3.7	CONSIDERAÇÕES FINAIS DO CAPÍTULO 3	69
4	CONSIDERAÇÕES FINAIS	70
5	REFERENCIAL BIBLIOGRÁFICO	72

1 INTRODUÇÃO

Neste capítulo, apresentam-se os seguintes tópicos: (i) Contextualização; (ii) Problema de pesquisa; (iii) Objetivo geral e objetivos específicos; (iv) Justificativa; (v) Estruturação da Dissertação; (vi) Metodologia da Dissertação.

1.1 CONTEXTUALIZAÇÃO

A economia moderna tem sofrido constantes transformações e conforme Sučić (2010) a base econômica atual está voltada aos serviços ao invés de produtos, sendo que as organizações, governos e universidades em todo o mundo recentemente despertaram para a compreensão de que os serviços dominam a economia global e o crescimento econômico (CACM, 2007).

No setor de serviços, encontram-se as empresas de serviços de Tecnologia da Informação (TI), que é considerada uma das atividades com forte potencial de crescimento. As TI estão situadas no centro da chamada nova economia e configuram-se como dispositivos fundamentais para o acesso à informação e à sociedade do conhecimento (IBGE, 2015). Sendo, dessa forma, um campo de pesquisa em extensão, e de muito envolvimento social e financeiro no país.

O *software* tornou-se uma ferramenta indispensável em todas as atividades da sociedade atual, apoiando-se nos progressos da tecnologia na academia e na indústria. A produção de novos equipamentos eletrônicos cresce cada vez mais e de forma mais rápida, alinhada à crescente expansão destes dispositivos que tem como consequência a determinação de um desenvolvimento de *software* cada vez mais especializado e avançado. Isto sugere diretamente o aumento da necessidade por programas e implica de maneira indireta na busca por formas mais apropriadas de desenvolvê-los. Partindo deste princípio, o mercado de *software* tem se tornado um tanto competitivo, fazendo com que a Engenharia de *Software* desenvolva a capacidade de se adaptar a estas mudanças frequentes (BASSI FILHO, 2008).

Com o significativo crescimento na demanda por *software*, alinhado com a alta velocidade com que seus requisitos evoluem, torna-se evidente que o seu

desenvolvimento exige flexibilidade, visto que muitas das decisões precisam ser tomadas ao longo do projeto. Além disso, as dificuldades presentes na etapa de desenvolvimento vão além de questões técnicas (ANJOS *et al.*, 2008).

Segundo o Standish Group (2009), 32% dos projetos de *software* são bem-sucedidos, sendo entregues no prazo, cumprindo o orçamento e de acordo com especificações prévias. Por outro lado, 24% destes projetos são entregues com falhas e 44% com deficiências (orçamento ultrapassado, e/ou com menos ou mais funções e funcionalidades requeridas, entre outras). O mesmo grupo, em 2012 expôs, que em um *software* personalizado, cerca de 45% das funcionalidades implementadas nunca são utilizadas e 19% são raramente utilizadas (JOHNSON, 2012).

Organizações de desenvolvimento de *software* estão percebendo que seu principal desafio para produzir produtos de qualidade é gerenciar efetivamente o seu processo de desenvolvimento (NIAZI *et al.*, 2006). A qualidade de produtos de *software* deve ser considerada em três dimensões: (i) qualidade do produto em relação aos seus requisitos de qualidade; (ii) qualidade do processo que produz o produto; (iii) qualidade do produto no contexto do ambiente em que será utilizado (homologação) (KITCHENHAM *et al.*, 1995).

Durante um projeto de desenvolvimento, as funcionalidades em excesso são incluídas pelo cliente ou empresa desenvolvedora como forma de garantir que todas as necessidades dos clientes serão atendidas. Porém, desenvolver mais que o necessário resulta em um esforço desnecessário, logo, percebe-se o quão comum são os desperdícios em um processo de desenvolvimento de sistemas (POPPENDIECK *et al.*, 2006).

O mercado de *software* entrou em uma fase de alta competição, principalmente no início do século XXI, sendo que as mudanças constantes, prazos curtos e exigência por produtos feitos de forma rápida e eficiente se tornam aspectos prioritários no processo produtivo. Essas exigências também vinham impactando na indústria tradicional (COCKBURN *et al.*, 2001; TAKEUCHI *et al.*, 1986).

Com relação a estas mudanças, no contexto da indústria de *software*, surgem, então, as metodologias ágeis, que ganham maior destaque no início do ano de 2001 com o Manifesto Ágil (BECK *et al.*, 2000). Nestas abordagens o processo de desenvolvimento recebe um novo enfoque, sendo priorizada as entregas aceleradas,

simplicidade e rápido retorno a mudanças. Estes valores são alcançados via um replanejamento constante, proporcionado por meio de diversos *feedbacks* recebidos durante as interações (COCKBURN; HIGHSMITH, 2001).

A abordagem utilizando retornos periódicos durante o processo de desenvolvimento é necessária para este tipo de produto (*software*), pois, de acordo com Schwaber (2004), trata-se de uma atividade complexa que facilmente se comporta de forma imprevisível tornando inviável, inclusive, prever as causas que tornarão o problema ainda mais complexo e imprevisível.

As metodologias ágeis possuem grande ligação com a filosofia *Lean*, e seus modelos de desenvolvimento, que por sua vez é a aplicação dos princípios da *Toyota Product Development System* (POPPENDIECK, 2003). O pensamento *Lean* tem como por finalidade o exímio atendimento ao cliente, evitando o desperdício no processo, com entregas rápidas e de qualidade, com respeito às pessoas que participam do processo e com compromisso de difundir conhecimento. Liker (2004) complementa que agilidade é um conceito frequentemente associado ao desenvolvimento *Lean* por possuir configurações e objetivos mútuos.

Originalmente o desenvolvimento *Lean* de *software* foi visto como mais um método ágil por muitos autores (DYBA; DINGSOYR, 2009). Poppendieck e Poppendieck (2003) consideram que a abordagem *Lean* fornece uma base teórica para os métodos ágeis, porém tornam-se cada vez mais frequentes os estudos que consideram o método de desenvolvimento *Lean* independente e diferente dos métodos ágeis (HIBBS; JEWETT; SULLIVAN, 2009; MIDDLETON; JOYCE, 2012; PETERSEN; WOHLIN, 2010). Vale salientar que também existe a vertente de explorar a lógica do uso de práticas de produção *Lean* para a melhoria dos métodos de desenvolvimento ágil, como o exemplo do método chamado “Scrumban” que usa o método ágil *scrum* com o método *kanban* da abordagem *Lean* (NIKITINA; KAJKOMATTSSON; STRALE, 2012).

A eliminação de desperdício é o principal elemento do sistema de produção *Lean*, que é uma filosofia de gerenciamento que procura otimizar a organização de forma a atender as necessidades do cliente no menor prazo possível, na mais alta qualidade e ao mais baixo custo, ao mesmo tempo em que aumenta a segurança e o

moral de seus colaboradores, envolvendo e integrando não só manufatura, mas todas as partes da organização (GHINATO, 2000).

Este trabalho tem por objetivo executar uma revisão do histórico de Metodologias Ágeis e do *Lean Software Development* (LSD) através da revisão bibliográfica e de seus principais autores. E, ainda, considerando os excessos presentes no desenvolvimento de *software*, essa pesquisa preocupa-se em analisar empresas desenvolvedoras e identificar os processos que geram excessos na produção de sistemas. Bem como analisar a forma com que metodologias ágeis e o *Lean* podem influenciar na melhoria da qualidade do *software*.

No Sudoeste do Paraná, a existência de um grupo de empresas permitiu a constituição do Arranjo Produtivo Local (APL) de TI. Pereira (2011) salienta que foi a partir do envolvimento de agentes locais e iniciativas de instituições de apoio que ações foram sendo desenvolvidas com o passar do tempo, gerando condições favoráveis para a criação e estruturação de um ambiente tecnológico propício ao desenvolvimento do setor de TI em Pato Branco e região.

O APL TI é formado por empresas estabelecidas principalmente nos municípios de Pato Branco, Francisco Beltrão e Dois Vizinhos, que atuam, sobretudo, no segmento de atividades de software e serviços de TI. Está organizado através de uma mesa diretora, câmaras temáticas e grupos de trabalho. A mesa diretora tem como principal função coordenar os trabalhos da governança do APL (NTI, 2015).

O APL TI encontra-se constituído por 52 empresas, sendo que de acordo com o Sebrae (2012), no ano de 2008 compunha-se de 14 empresas. Há que se apontar a dificuldade de obtenção de dados precisos sobre o número de empresas vinculadas ao núcleo, no geral, devido a não participação das empresas, pois apesar de considerarem-se todas pertencentes, muitas não se relacionam, não aderindo formalmente aos trabalhos do núcleo gestor.

1.2 PROBLEMA DE PESQUISA

Com o crescimento emergente do setor de desenvolvimento de *software* no sudoeste do Paraná, o presente estudo busca analisar a aplicabilidade das ferramentas de desenvolvimento ágil, em que o *Lean Software Development* (LSD)

está inserido, com foco no excesso e desperdício, características fortemente tratadas pelo *Lean*. A abordagem se dará, mais especificamente, a cerca de empresas que produzem *softwares* de pequeno e médio porte. Pois para estes projetos é possível acompanhar todo o processo de desenvolvimento, observando a aplicação das metodologias já utilizadas em cada fase do escopo, promovendo uma análise detalhada das possíveis aplicações e melhorias através do *Ágil/Lean*, além do que já está inserido propriamente.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Este trabalho tem como objetivo abordar e analisar os excessos e desperdícios no processo de desenvolvimento de software, a partir de conceitos obtidos na revisão sistemática da literatura incorporando a aderência de ferramentas e metodologias LSD no desenvolvimento de *software* nas empresas desenvolvedoras inseridas no APL de TI do Sudoeste do Paraná.

1.3.2 Objetivos Específicos

A partir do objetivo geral apresentado, foram definidos os seguintes objetivos específicos:

- Realizar pesquisa bibliográfica afim de buscar base e fundamento teórico sobre desenvolvimento *ágil/Lean* na literatura, estabelecendo ligação com outras pesquisas empíricas em empresas de diferentes portes e com diferentes escopos de trabalho na área de desenvolvimento de *software*, fortalecendo, assim, a necessidade e importância de estudos de casos para esta área;
- Caracterizar o APL de TI investigado, a partir de suas demandas, desafios e dificuldades, analisando a forma com que os métodos *ágeis/Lean* podem

colaborar em melhores resultados e no melhor atendimento das respectivas demandas;

- Realizar estudo de casos múltiplos com a finalidade de analisar, a partir da implementação de processos ágeis/*Lean* já inseridos no desenvolvimento de *software* das empresas pesquisadas, a redução dos excessos e desperdícios em todo seu processo de desenvolvimento, bem como as tangentes que ainda passar por melhorias em implementação.

1.4 JUSTIFICATIVA

A TI tem tido cada vez mais relevância na vida moderna, com aplicativos e *hardware* embutidos no dia a dia da medicina, do transporte, das comunicações, sendo fundamental para realização de qualquer trabalho moderno. Assim como ocorrido na indústria automobilística, o *lean thinking* em TI pode contribuir muito para o aumento de produtividade e de qualidade, mudando radicalmente a indústria (CANTANHEDE, 2014).

Existe um número limitado de publicações da aplicação do *lean thinking* na área de TI e no desenvolvimento de *software*, conforme pode-se observar na pesquisa bibliográfica. Identificou-se uma prática de integração de conceitos de vários métodos e culturas, em busca de ganhos de produtividade e qualidade. Alguns desses métodos e práticas podem ser inclusive conflitantes, trazendo assim resultados inesperados. Desta forma, existe uma oportunidade de contribuir com a pesquisa de aplicação do *lean thinking* na área de TI, e gerar conhecimento para melhorar as práticas de desenvolvimento de *software*.

Segundo Baskerville *et al.* (2001), invariavelmente qual seja o nicho de mercado, as empresas têm sido constantemente pressionadas quanto à entrega mais rápida de uma nova versão do *software*. O ciclo de desenvolvimento existente na década passada (com duração típica de 24 a 36 meses) foi reduzido (12 a 18 meses) para as empresas que não tem internet como foco principal. Já para as empresas envolvidas com comércio eletrônico, criação e manutenção de portais web, o ciclo é ainda mais curto (3 a 6 meses de duração). Mesmo que esta observação não seja uma novidade, ainda é possível observar, no trabalho diário das empresas, que a

pressão por reduzir o prazo de entrega dos softwares ainda é uma realidade. Para atingir estes ciclos curtos, as empresas buscam adotar técnicas do *Lean manufacturing production* com uso de ferramentas, métodos e gestão de projetos para eliminar a ineficiência, desperdício, retrabalho em todas as fases do desenvolvimento. Ainda assim, muitas empresas sentem-se pressionadas e optam pela metodologia tradicional, a fim de manter a qualidade do produto final, e por insegurança às práticas de inovação.

A região na qual esta pesquisa está inserida tem um cunho forte ligado ao desenvolvimento de *software*, sendo tratada como um polo regional dentro da área. Porém ainda conserva muitos pontos dos processos mais tradicionais de desenvolvimento devido a formação menos recente de seus idealizadores. Entretanto, atentos à necessidade de crescimento e a demanda estimada, as empresas adaptam processos tradicionais com características de desenvolvimento ágil, mesmo sem ter total compreensão da dinâmica utilizada. Sendo necessário estabelecer esta ligação através de modelos práticos e concisos da metodologia.

1.5 ESTRUTURA DA DISSERTAÇÃO

A presente dissertação é apresentada é composta por dois capítulos consecutivos promovendo a ligação das etapas da pesquisa. Desta forma, o Capítulo 1 apresenta os elementos introdutórios a esta pesquisa, contexto, objetivo e justificativa, apresentando ainda a metodologia da dissertação. Os capítulos 2 e 3 irão contemplar os objetivos específicos da pesquisa, por fim o capítulo 4 irá apresentar as considerações finais acerca do trabalho realizado, avaliando os resultados, bem como considerando as pesquisas futuras.

1.6 METODOLOGIA DA DISSERTAÇÃO

A descrição da natureza da pesquisa se propõe a classificá-la quanto aos fins e quanto aos meios. Quanto aos fins, a pesquisa inicialmente se enquadra como descritiva que, segundo Cervo (2007), caracteriza-se por observar, registrar, analisar

e correlacionar fatos ou fenômenos sem manipulá-los. Trata-se então do estudo e da descrição das características, propriedades ou relações existentes na comunidade, grupo ou realidade pesquisada. Essa natureza será atendida através dos estudos de caso empíricos, que serão realizados nos capítulos 2 e 3 deste trabalho.

A abordagem empírica usa métodos indutivos para alcançar os seus resultados a partir da observação de casos reais, diferente das abordagens analíticas que usam métodos dedutivos. Os estudos empíricos também permitem que se investigue um grande número de variáveis para identificar novas relações entre eles. Além disto, tal procedimento metodológico é sugerido para fenômenos sociais complexos que envolvem uma grande quantidade de variáveis em contextos únicos (WACKER, 1998; EISENHARDT *et al.*, 2007).

Quanto aos meios, a pesquisa se caracteriza por ser bibliográfica. Para Cervo (2007), uma pesquisa é considerada bibliográfica quando se busca explicar o problema a partir de referências teóricas publicadas em artigos, livros, dissertações, testes, entre outros. Dessa forma, é feito um levantamento sobre contribuições passadas relacionado a um determinado assunto, tema ou problema, método abordado pelo Artigo 1 da dissertação.

A seguir apresenta-se (Quadro 1) as etapas da pesquisa. Conforme mencionado anteriormente, a próxima etapa, Capítulo 2, utiliza uma abordagem de mapeamento da literatura a fim de levantar lacunas de pesquisa, bem como a bibliografia que apoiará o desenvolvimento do trabalho.

O Capítulo 3 irá realizar estudo de casos múltiplos. O intuito da pesquisa será analisar o APL em que a pesquisa está inserida, bem como se aprofundar no reconhecimento do processo de desenvolvimento de *software* em que as empresas identificando pontos de excesso e desperdício.

ETAPA	OBJETIVO	METODOLOGIA	REFERENCIAL TEÓRICO
Capítulo 02,; Processo Enxuto para Desenvolvimento de um <i>Software</i> em	Efetuar busca nos estudos publicados nos últimos 10 anos sobre metodologias	Revisão sistemática da literatura realizada através do método <i>Proknow-C</i> , <i>Knowledge</i>	Levantamento das metodologias ágeis/ <i>Lean</i> utilizadas

Empresas do APL de TI do Sudoeste do Paraná: Um Mapeamento da Literatura.	ágeis/ <i>Lean</i> no desenvolvimento de <i>software</i> afim de estabelecer relação com o comportamento atual da área.	<i>Development Process-Constructivist</i> (ENSSLIN <i>et al.</i> , 2010)	no desenvolvimento de <i>software</i> .
Capítulo 03,: Estudo de caso múltiplo: relato da situação prática de desenvolvimento de um <i>software</i> de ciclo rápido, identificando problemas de excessos e desperdícios através da ótica de desenvolvimento <i>Lean</i> de Software.	Análise, de forma pontual e prática, do processo de desenvolvimento de <i>software</i> , desde sua concepção até entrega ao cliente, sinalizando e tratando as formas de desperdício e excessos que ocorrem durante o processo de desenvolvimento.	Estudo de casos múltiplos, identificando características do processo de desenvolvimento que correspondam aos princípios básicos de desenvolvimento <i>Lean</i> de <i>software</i> , com ênfase em eliminar desperdícios e excessos no processo de desenvolvimento.	Conceitos gerais ágil/ <i>Lean</i> . Metodologias ágeis/ <i>Lean</i> com foco em pequenas e médias empresas (PMES). Retomar o foco das práticas ágeis, estudando com maior profundidade a <i>Lean Software Development</i> , que é utilizada para promover as melhorias apresentadas no processo. Identificar e promover sugestões de melhorias quanto ao item em estudo com foco em eliminar desperdício e excessos no processo de desenvolvimento de <i>software</i> .

Quadro 1 Resumo dos capítulos 2 e 3 desenvolvidos na pesquisa. Fonte: o autor.

Para finalizar, no capítulo 4, são apresentadas as considerações finais, delimitações da pesquisa, como também, os trabalhos futuros já vislumbrados e a dissertação encerra-se com o capítulo 5 que contempla as referências utilizadas nas seis etapas anteriores.

2 PROCESSO ENXUTO PARA DESENVOLVIMENTO DE SOFTWARE: UM MAPEAMENTO DA LITERATURA

O presente capítulo trata do *Lean Thinking* (pensamento enxuto) cuja aplicação já é uma realidade na indústria de manufatura. Para o contexto no uso de *Lean* para outras áreas, destaca-se o desenvolvimento de *software* utilizando a abordagem *Lean Software Development* (LSD). O trabalho teve como propósito fazer uma análise bibliométrica e sistêmica de um portfólio bibliográfico alinhado a este tema. O método utilizado para a construção e proposição do portfólio bibliográfico e posteriores pesquisas bibliométricas e sistêmicas foi o *Knowledge Development Process-Constructivist* - Proknow-C (ENSSLIN *et al.*, 2010). De um total de 2263 artigos, após a aplicação de filtros, obteve-se 23 artigos que compuseram o portfólio bibliográfico, o qual passou por uma análise bibliométrica e por fim, pode-se observar que o tema *Lean Software Development* ainda é um conceito muito recente, portanto, não há um consenso comum sobre a sua definição. Observa-se algumas questões empíricas à cerca de sua aderência nos estudos de caso resultantes da pesquisa, sendo esta a principal lente deste trabalho. Ainda que existam diversas lacunas a serem preenchidas, quanto a pesquisas sobre LSD e outros métodos ágeis, a revisão apresentou dados importantes para alavancar futuras pesquisas no que se refere ao uso da filosofia *Lean* em diferentes formatos de empresas de desenvolvimento de *software*.

2.1 INTRODUÇÃO

Pode-se observar que a necessidade de desenvolvimento e implantação de *software* continua crescendo significativamente nos dias atuais. Grande parte dos projetos de *software* se esforçam para alcançar um equilíbrio entre três objetivos básicos: qualidade, custo direto e entrega pontual, independentemente do método utilizado: ágil ou tradicional (PINTO, 2015).

Entretanto, as abordagens ágeis têm sido questionadas quanto ao processo de documentação, gerenciamento visual de *software*, métricas de desenvolvimento, entre outras características essenciais durante o processo de

desenvolvimento e para a entrega do produto final (BIFFL, *et al.*, 2005). Com isso, a fim de melhorar os métodos e processos de desenvolvimento de *software*, as empresas começaram a observar a abordagem *Lean* da Manufatura com o viés para desenvolvimento de *software*, criando o conceito de *Lean Software Development* (LSD).

Partindo da importância que têm o conhecimento de metodologias ágeis, dentre elas a LSD para desenvolvimento de sistemas computacionais, salienta-se que a pesquisa bibliográfica, especialmente dentro da área de Engenharia de *Software* ligada à Engenharia de Produção, é indispensável. Para tal, inicia-se um processo metodológico estruturado e consistente para selecionar materiais de cunho científico, que permite ao pesquisador analisar a relevância da base bibliográfica, e especialmente identificar as lacunas presentes na sua linha de pesquisa (CAUCHICK *et al.*, 2010).

A especial motivação para esta pesquisa é a carência de material e informações relevantes na área de engenharia e desenvolvimento de *software* alinhados aos métodos ágeis, e especialmente com referência à filosofia *Lean*, observada através de pesquisas dos autores em bases de dados reconhecidas. De acordo com Poppendieck (2006), o desenvolvimento de *software Lean* é a aplicação dos princípios da Toyota *product development system* para o desenvolvimento de *software*. Quando ele é aplicado corretamente, o desenvolvimento agrega melhoria na qualidade, além de ser realizado rapidamente e buscando reduzir os custos de desenvolvimento.

Levando em consideração os assuntos expostos acima, criou-se então o principal problema que motivou este capítulo: Como manter os benefícios obtidos com as práticas ágeis na otimização dos métodos e processos de desenvolvimento de *software* e qual sua adesão dentro da realidade local? Para responder a essa pergunta, faz-se essencial um levantamento do estado da arte da temática que circunda este trabalho, o *Lean Software Development*.

Pra este capítulo, o objetivo foi apresentar uma melhor compreensão sobre o tema LSD, além de possibilitar a identificação de lacunas para continuidade da pesquisa. Para isso, tem-se os objetivos específicos: (i) Selecionar um portfólio bibliográfico sobre desenvolvimento *Lean* de *software* em publicações internacionais;

(ii) evidenciar o portfólio os artigos, autores, periódicos e palavras-chaves de maior relevância e (iii) analisar o portfólio de artigos, identificando suas lentes de abordagem.

Sendo assim, este trabalho visa contribuir para a comunidade acadêmica, uma vez que alcançado o resultado da pesquisa, busca-se ampliar o entendimento do conhecimento, especialmente LSD. Como limitação dessa pesquisa, destaca-se que somente foram considerados estudos que enfatizavam as práticas ágeis de desenvolvimento de *software*, ou seja, estudos que tratavam das metodologias tradicionais foram desconsiderados, devido à não relevância à temática abordada nesse trabalho.

2.2 REFERENCIAL TEÓRICO

2.2.1 Desenvolvimento Ágil de *Software*

À medida que os métodos tradicionais passaram a não obter o nível de sucesso necessário na concretização dos projetos, houve a necessidade de se criar novas práticas para guiar as equipes de desenvolvimento de *software*. Essas novas práticas são orientadas às pessoas pertencentes ao projeto e procuram trazer flexibilidade para sobreviver a um ambiente com mudanças constantes (COCKBURN; HIGHSMITH, 2001).

Os métodos ágeis surgem como respostas aos métodos tradicionais, que são baseados no seguimento de um plano bem definido, excesso de documentação e rigorosa padronização (NERUR *et al.*, 2005; KAJKO-MATTSSON, 2008).

Dessa forma, no início do ano de 2001, um grupo de profissionais se reuniu com o intuito de debater sobre as formas que por eles eram utilizadas como alternativa aos processos excessivamente baseados em documentação e formalismo. Desse encontro se origina o Manifesto pelo Desenvolvimento Ágil de *Software*, ou simplesmente Manifesto Ágil, o qual possui os seguintes valores (BECK *et al.*, 2001): Indivíduos e interações acima de processos e ferramentas; *Software* funcionando acima de documentação abrangente; Colaboração com o cliente acima de negociação de contratos; e responder a mudanças acima de seguir um plano.

Segundo Dyba e Dingsoyr (2008), os principais métodos ágeis são: *Crystal methodologies*, *Dynamic software development method* (DSDM), *Feature-driven development*, *Lean software development*, *Scrum* e *Extreme programming*, sendo que este artigo irá abordar mais especificamente apenas o *Lean Software Development*.

2.2.2 Metodologia de desenvolvimento ágil: SCRUM

O método de desenvolvimento SCRUM foi proposto em 1995 por Ken Schwaber (2004), num momento em que ficou claro para a maioria dos profissionais que o desenvolvimento de *software* não era algo que poderia ser planejado, estimado e concluído com sucesso usando um método comum e burocrático. O método SCRUM baseia-se no trabalho de Pittman (1993) e Booch (1995) e adere aos princípios do desenvolvimento ágil de *software*.

Para o SCRUM a centralidade da atividade está na maioria dos processos durante o desenvolvimento que não pode ser previsto. Por isso, aborda o *software* em desenvolvimento de maneira flexível. As únicas duas partes que estão totalmente definidas durante um projeto de desenvolvimento de *software* são a primeira e a última fase (planejamento e fechamento). No centro, o produto final é desenvolvido por várias equipes em uma série de entregas periódicas e flexíveis chamadas '*Sprints*'. Não há novos requisitos anexados durante as *sprints*. Isso garante que o produto final esteja sendo desenvolvido com uma alta probabilidade de sucesso, mesmo dentro de uma mudança constante no ambiente inserido. Este ambiente, que inclui fatores como concorrência, tempo e pressão financeira, mantém sua influência em desenvolvimento até a fase de encerramento.

2.2.3 *Lean Software Development* (LSD)

Segundo Womack e Jones (1994), os principais princípios da produção enxuta podem ser aplicados em qualquer organização, ainda que esta tenha sido originada na indústria automobilística. A seguir apresenta-se como se dá a incorporação da filosofia *Lean* ao desenvolvimento de *software*, demonstrando, desta maneira como os conceitos do Sistema Toyota são apresentados ao contexto da indústria de *software*. Sendo assim, o trabalho de Poppendieck e Poppendieck (2003)

se torna a principal referência utilizada, pois, a literatura nessa área é, em sua maioria, obra desses autores, ou baseada em suas ideias.

A fim de melhor compreender como os benefícios da produção enxuta podem ser aplicados no desenvolvimento de *software* é necessário entender os principais pilares desta filosofia. Elenca-se então, os referidos princípios, (POPPENDIEK et al.; CUSUMANO, 2012):

- 1-Eliminar Desperdícios;
- 2-Amplificar o aprendizado;
- 3-Adiar decisões ao máximo;
- 4-Entregar o mais rapidamente possível;
- 5-Delegar poder à equipe;
- 6-Incorporar integridade;
- 7-Ver o todo.

Eliminar desperdícios consiste em um dos pilares da filosofia *Lean* que trata dos sete desperdícios identificados por Shingo (1996) – Estoque; Processos Extras; Excesso de Produção; Transporte; Espera; Movimento; Defeitos – sendo, nesta ótica, tudo que não agrega valor ao cliente, tendo-o sempre no foco de todo o processo. O cliente é responsável por dar início ao procedimento, prover feedback relacionado ao andamento do mesmo e ser o termômetro sobre a real necessidade de qualquer esforço empregado durante o desenvolvimento do produto. Em relação ao princípio de amplificar o aprendizado, tem-se que o processo de desenvolvimento de *software* é uma atividade complexa, pode-se afirmar que nem sempre o desenvolvedor tem em mente a melhor maneira de fazer o trabalho. Sendo assim, a realização de experimentos com entregas de ciclos rápidos aumenta o *feedback* sobre o que está acontecendo, facilitando a adaptação às mudanças que podem acontecer durante o processo de desenvolvimento do sistema.

Outro grande benefício gerado por avaliações constantes do trabalho reside no fato de corrigir-se qualquer coisa que não esteja nos conformes antes que um grande esforço seja demandado. Sendo que essas interações estejam gerando valor agregado, e cada subproduto gerado deve ser uma parte testada, integrada e funcional do produto.

Por razão de algumas características que são intrínsecas do *software*, é considerado bastante difícil representar, de forma satisfatória, como o produto ficará quando estiver pronto, pois trata-se de uma abstração, e nem sempre é possível validar se a ideia contida na mente dos diversos *stakeholders* foi captada corretamente (Brooks, 1997).

A imensa maleabilidade do *software* também dá maior respaldo para exigências por mudanças. O sistema em produção deve ser capaz de se adaptar a novas características que podem ser necessárias (Brooks, 1997).

Sendo assim, ao se adiar decisões ao máximo, o cliente passa a tomar decisões baseando-se em opções, não apenas especulações. A partir do momento em que o desenvolvimento evolui, é possível aumentar o entendimento sobre os aspectos que realmente agregam valor.

O modelo mais tradicional de *software*, conhecido como Modelo Cascata busca planejar tudo que deve ser feito antes de iniciar o trabalho com objetivo de prevenir riscos e evitar mudanças durante o projeto. Porém, segundo Cockburn e Highsmith (2001) a necessidade por mudanças é uma constante em *softwares*. Elas podem vir de uma melhor compreensão do problema a ser resolvido, de novas necessidades do trabalho, novas legislações, entre outros.

Bell e Orzen (2011) analisam e afirmam que talvez possa existir um conflito entre as práticas lean e as práticas do modelo tradicional. No quadro 2 são apresentados alguns conceitos no lean e no desenvolvimento tradicional. Essa área de *software* tende a mover-se com cuidado e devagar para evitar instabilidade e interrupção do negócio. O lean encoraja cada indivíduo a identificar e ajustar os problemas fazendo pequenas melhorias todos os dias. As diferenças não significam que não haja conciliação entre o lean e o modelo tradicional, mas que existe uma tendência natural ao conflito e que deve ser um ponto de atenção.

	LSD	Modelo Tradicional
Gestão da Mudança	Orgânica, incremental e contínua	“Engineered” e através de grandes eventos planejados
Organização	Equipe multi-funcional	Controles e comando central

Métricas	Métricas ligadas ao resultado de cima para baixo e de baixo para cima, conectando iniciativas de melhoria com os objetivos estratégicos	Contenção de custos e “ <i>uptime</i> ”
Gestão do Conhecimento	Generalização	Especialização
Educação	Foco no processo	Foco na tarefa
Definição de Sucesso	Velocidade e Agilidade	Estabilidade

Quadro 2 Contrastes entre atributos do LSD e o Modelo Tradicional de *Software* (BELL *et al.*, 2011)

Em vez de tentar prevenir que mudanças ocorram, o ideal é entregar de forma tão rápida que o cliente não possa solicitar alterações no escopo.

Quando se trata de um ambiente ágil, com entregas que devem ser efetuadas de forma rápida e constante, não pode haver perda de tempo com tomadas de decisões. As informações fluindo constantemente fazem com que os próprios trabalhadores façam ajustes no processo. Com a comunicação constante entre os principais atores do processo produtivo, e tarefas delegadas não existe necessidade de uma unidade centralizadora para tomadas de decisões.

Na ótica do desenvolvimento de *software* utilizando os princípios *Lean*, existem dois tipos de integridade, segundo Poppendieck e Poppendieck (2003), sendo elas: integridade percebida e integridade conceitual.

A integridade percebida está relacionada à entrega daquilo que o cliente deseja e necessita. O cliente pode necessitar de algo, mas pode não saber explicar essa necessidade ou até mesmo desconhecer-la. Dessa forma, uma constante comunicação permite que se alcance uma melhor compreensão do que realmente é preciso ser produzido.

Já a integridade conceitual diz respeito à harmonia e coesão entre as diversas partes que compõe o produto. Deve-se evitar, assim, que o usuário tenha a percepção de estar utilizando um conjunto de *softwares* diferentes.

Os autores como Poppendieck *et al.* (2011) e Poppendieck e Cusumano (2012) descrevem o LSD como um conjunto de princípios e classificação de

desperdícios. Jonsson et al. (2013), com uma publicação recente, explicam que mesmo com a recente popularização do LSD, ainda não existe um conceito geral, aceito, claro e definido no detalhe do que seria o significado (modelo) do LSD.

Jonsson et al. (2013) executaram um estudo dos principais autores do LSD. Os principais conceitos foram identificados e mapeados contra a definição de cada um dos autores. Essa publicação de Jonsson et al. (2013) demonstra a tentativa de se criar um modelo de LSD que possa ser aplicado e replicado na indústria de *software*.

O desafio de melhor compreender o LSD, o nível de utilização e aplicação dentro da indústria de *software*, e a sua relação com outras práticas como desenvolvimento ágil, trouxe a necessidade da pesquisa ferramentas de avaliação do lean na área de desenvolvimento de *software* e o aprofundamento da sua aplicação dentro de empresas de desenvolvimento.

2.3 METODOLOGIA DA PESQUISA BIBLIOMÉTRICA

A fim de formar um portfólio bibliográfico, com condições de prover uma análise bibliométrica e sistêmica, a metodologia utilizada para a criação do respectivo trabalho científico, é o Proknow-C (*Knowledge Development Process – Constructivist*) (ENSSLIN *et al.*, 2010). Criado pelo LabMCDA (Laboratório de Metodologias Multicritério em Apoio à Decisão), o respectivo método, é composto de 3 principais etapas, conforme ilustrado na figura 1: 1 - Seleção dos artigos; 2 - Análise Bibliométrica e 3 - Revisão Sistêmica.

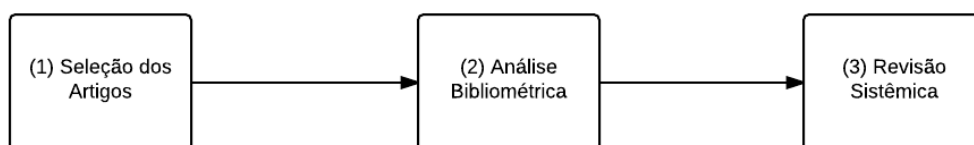


Figura 1: Etapas Proknow – C.
Fonte: Adaptado de Tasca (2010)

A seguir cada uma das etapas desenvolvidas no presente trabalho serão descritas em maiores detalhes.

2.3.1 Procedimentos Para Seleção dos Artigos

A seleção dos artigos contempla uma série de atividades executadas que tem como artefato final, o portfólio bibliográfico. É importante ressaltar que, para a realização do presente trabalho, optou-se pela utilização das seguintes bases de dados: Web of Science (ISI) e Scopus, pela sua reconhecida abrangência no mundo acadêmico.

A Figura 2 apresenta um resumo do processo para chegar até o portfólio final da pesquisa. E em seguida, algumas partes do processo são detalhadas para melhor compreensão.

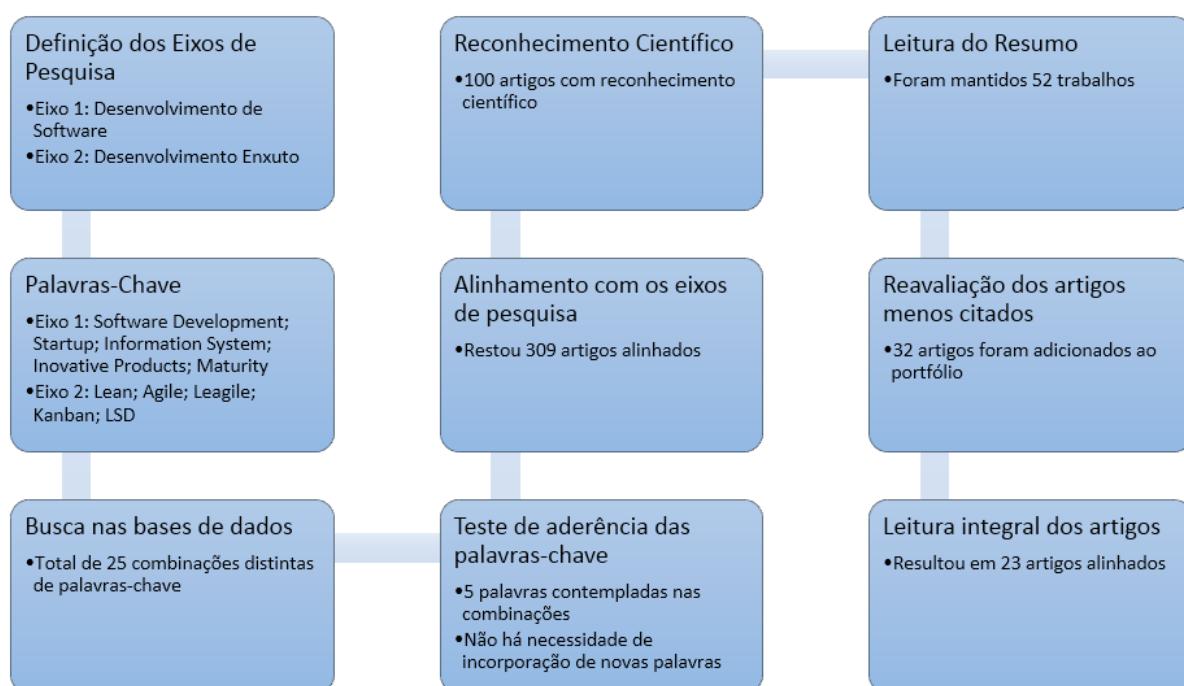


Figura 2: Procedimento para seleção de artigo
Fonte: Adaptado de Bortoluzzi *et al.*, 2011

A atividade inicial, previamente realizada antes das buscas nas bases de dados já citadas, foi a definição dos eixos principais que norteiam as palavras chave. Bortoluzzi *et al.*, 2011 indicam a definição de três eixos para a pesquisa. No presente trabalho foram utilizados dois, devido à grande restrição de informação que o outro

eixo causaria. A inclusão de um terceiro eixo relacionado a clusters, diminuiu mais de 50% o retorno da pesquisa, tornando-se inviável para o trabalho.

Sendo assim, foram definidos os dois principais eixos: (1) Desenvolvimento de *Software* e (2) Desenvolvimento Enxuto. Para o primeiro, as seguintes palavras-chave foram definidas: *Software Development*, *Startup*, *Information System*, *Inovative Products* e *Maturity*. E as seguintes palavras-chave foram adotadas para o segundo eixo: *Lean*, *Agile*, *Leagile*, *Kanban* e *LSD*. Todas as palavras foram transcritas para a língua inglesa a fim de atingir o escopo da publicação internacional qualificada.

A partir das considerações descritas acima, iniciou-se o processo de busca nas bases de dados. As buscas foram feitas a partir da combinação cruzada de cada uma das palavras chave previamente definidas. A partir das 10 palavras, foram criadas um total de 25 combinações distintas, cujas foram utilizadas em pesquisas nas bases de dados, totalizando 2263 artigos. Para este momento, foi definido a criação de duas delimitações: descarte dos artigos publicados há mais de 10 anos e consideração de somente artigos científicos de periódicos do domínio das Ciências Sociais e Exatas.

Após o primeiro resultado, o processo prescreve a realização de um teste de aderência nas palavras-chave previamente definidas. A etapa é importante para definir se há a necessidade da incorporação de novos termos não inseridos anteriormente. O processo resume-se à escolha aleatória de 2 a 3 artigos, leitura dos seus resumos a fim de verificar se está enquadrado em algum dos eixos da pesquisa. Caso sim, é feito uma análise das palavras chave de cada um dos trabalhos a fim de verificar se os termos anteriormente escolhidos estão contemplados nos referentes. E a seguir, realiza-se o teste de aderência das palavras-chave, gerando uma compatibilidade entre os artigos selecionados e as palavras-chave já inseridas, não sendo necessário a incorporação de nenhuma outra.

O passo seguinte do processo é fazer uma análise do título de cada um dos artigos, a fim de garantir que estão enquadrados em algum dos eixos de pesquisa do presente trabalho.

Com o objetivo de auxiliar no processo de análise bibliográfica, os 2263 artigos foram importados na ferramenta de gerenciamento bibliográfico EndNote e então foi realizado o processo de eliminação de artigos duplicados. A eliminação foi então realizada na própria ferramenta, cuja encontrou um total de 397 artigos

repetidos. Fazendo a eliminação dos referentes, teve-se um total de 1866 artigos. Após a leitura de cada um dos títulos, decidiu-se por eliminar 1555 artigos, restando um total de 309 trabalhos, cujos estavam alinhados com a temática do trabalho.

Com o propósito de manter no portfólio, artigos com maior relevância acadêmica, a respectiva etapa objetiva eliminar trabalhos com pouco ou nenhum reconhecimento científico e manter aqueles com maior relevância. Para tanto, tabulou-se o número de citações que cada um dos trabalhos tem, ordena-os do maior ao menor e estabelece um ponto de corte, que determinará o conjunto de trabalhos com maior representatividade científica.

Como entrada da referente etapa, teve-se os 309 artigos já alinhados. Utilizou-se a ferramenta Google Scholar para encontrar o número de citações de cada trabalho. As pesquisas foram realizadas entre os meses outubro de 2015 a novembro do mesmo ano. Após a tabulação dos resultados, estabeleceu-se um ponto de corte em 90,15% do total de citações, que implica em dizer que: artigos com 20 citações ou menos são considerados, neste escopo, sem reconhecimento científico e consecutivamente, excluídos do portfólio bibliográfico.

Como resultado desta fase, 100 artigos foram aqueles que obtiveram reconhecimento científico dentro do percentual de representatividade necessário (90,15%), os outros 209 trabalhos possuíam um número de citações não representativo (9,85%).

É importante ressaltar que os 209 trabalhos considerados com baixa representatividade ainda passarão por uma nova análise sob outros critérios, em uma próxima fase deste processo de revisão.

Uma vez selecionado os artigos com maior representatividade científica, os seus resumos foram analisados com o propósito de identificar se os mesmos estão alinhados com o foco da pesquisa em questão. Dos 100 artigos analisados, 48 deles foram descartados e 52 trabalhos foram mantidos.

Com o propósito de reavaliar aqueles artigos descritos com pouco ou nenhum reconhecimento científico, levantou-se os autores de cada um dos 52 trabalhos, identificando-os como autores de relevância à temática da presente pesquisa, ou seja, autores com mais publicações e maior visibilidade em outros trabalhos.

A seguinte fase do Proknow-C, prevê a reavaliação daqueles artigos menos citados (2090. Para isso, os referidos passaram pelas seguintes análises: (1) Tempo de publicação menor que 2 anos, (2) Autor do trabalho presente no banco de autores com relevância e (3) Alinhamento à temática da pesquisa a partir da leitura do resumo. Somente trabalhos que passaram das três etapas, são reinseridos ao portfólio bibliográfico.

Como entrada desta etapa, tiveram-se os 209 trabalhos considerados com pouca ou nenhuma representatividade científica e após cada uma das três etapas, 177 trabalhos foram mais uma vez descartados e consecutivamente 32 foram adicionados ao portfólio bibliográfico, totalizando assim, 84 artigos.

A última etapa da concepção do portfólio bibliográfico, é a leitura integral de cada um dos trabalhos, a fim de garantir a sua relação à temática da pesquisa em questão. Como entrada desta etapa, teve-se 84 trabalhos.

O próximo passo foi procurar nas respectivas bases de dados, cada um dos artigos na íntegra e prosseguir com a leitura. A partir desta etapa, 26 trabalhos foram descartados por não estarem alinhados ao tema do trabalho ou por não terem sido encontrados nas bases de dados utilizadas. Por fim, após a leitura integral, onde foram descartados outros 35 artigos, formulou-se o portfólio bibliográfico com um total de 23 trabalhos que foi objeto das próximas etapas do processo: (2) Análise Bibliométrica e (3) Revisão Sistemática.

2.3.2 Procedimentos Para Análise Bibliométrica e Sistemática

Segundo Ensslin *et al.* (2010), bibliometria é um processo quantitativo de mensuração de dados estatísticos a partir de um conjunto pré-definido de artigos, com o propósito de identificar periódicos, autores e artigos mais relevantes no contexto de uma pesquisa científica.

O conjunto de artigos selecionados para a análise bibliográfica é o portfólio bibliográfico contendo 23 artigos que serão citados ao longo da análise e estão dispostos no quadro 3 a seguir.

Nº	Artigo
----	--------

1	Baskerville, R. Ramesh, B. Levine, L. Pries-Heje, J. (2006). "High-speed software development practices: What works, what doesn't". <u>IT Professional: 29-36</u>
2	Baptista, G. L. Vanalle, R. M. Salles, J. A. A. (2015). "A Software Development Process Model Integrated to a Performance Measurement System". <u>IEEE Latin America Transactions: 739 - 745</u>
3	Chow, T. Cao, D. B. (2008). "A survey study of critical success factors in agile software projects". <u>Journal of Systems and Software: 961-971.</u>
4	Da Silva, I. F. Da Mota Silveira Neto, P. A. O'Leary, P. De Almeida, E. S. De Lemos Meira, S. R. (2011). "Agile software product lines: A systematic mapping study". <u>Software - Practice and Experience: 899-920</u>
5	Dybå, T. Dingsøyr, T. (2008) "Empirical studies of agile software development: A systematic review". <u>Information and Software Technology: 9-10</u>
6	Ebert, C. Abrahamsson, P. Oza, N. (2012). "Lean software development". <u>IEEE Software: 22-2</u>
7	Fogelström, N. D. Svahnberg, T. G. M. Olsson, P. (2010). "The impact of agile principles on market-driven software product development". <u>Journal of Software Maintenance and Evolution: 53-80</u>
8	Hansson, C. Dittrich, Y. Gustafsson, B. Zarnak, S. (2005). "How agile are industrial software development practices?". <u>Journal of Systems and Software: 1295-1311</u>
9	Kettunen, P. (2009). "Adopting key lessons from agile manufacturing to agile software product development-A comparative study". <u>Technovation: 408-422</u>
10	Middleton, P. Joyce, D. (2012). "Lean software management: BBC worldwide case study". <u>IEEE Transactions on Engineering Management: 20-32</u>
11	Mishra, D. Mishra, A. (2011). "Complex software project development: Agile methods adoption". <u>Journal of Software Maintenance and Evolution: 549-564</u>
12	Misra, S. C. Kumar, V. Kumar, U. (2009). "Identifying some important success factors in adopting agile software development practices". <u>Journal of Systems and Software: 1869-1890</u>
13	Nerur, S. Mahapatra, R. Mangalaraj, G. (2005). "Challenges of migrating to agile methodologies". <u>Communications of the ACM: 72-78</u>
14	Petersen, K. Wohlin, C. (2011). "Measuring the flow in lean software development". <u>Software - Practice and Experience: 975-996</u>
15	Petersen, K. Wohlin, C. (2010). "Software process improvement through the Lean Measurement (SPI-LEAM) method". <u>Journal of Systems and Software: 1275-1287</u>
16	Poppendieck, M. Cusumano, M. A. (2012). "Lean software development: A tutorial". <u>IEEE Software: 26-32</u>
17	Zhi, J. Garousi-Yusifollu, V. Sun, B. Garousi, G. Shahnewaz, S. Ruhe, G. (2015). "Cost, benefits and quality of software development documentation: A systematic mapping". <u>Journal of Systems and Software: 175-198.</u>

18	Torrecilla-Salinas, C. J. Sedeño, J. Escalona, M. J. Mejías, M. (2015). "Estimating, planning and managing Agile Web development projects under a value-based perspective". <u>Information and Software Technology: 124-144</u>
19	Suomalainen, T. Kuusela, R. Tihinen, M. (2015). "Continuous planning: An important aspect of agile and lean development". <u>International Journal of Agile Systems and Management: 132-162</u>
20	Serrador, P. Pinto, J. K. "Does Agile work? - A quantitative analysis of agile project success". (2015). <u>International Journal of Project Management: 1040 – 1051</u>
21	Sharp, H. Robinson, H. Petre, M. (2009). "The role of physical artefacts in agile software development: Two complementary perspectives". <u>Interacting with Computers: 108 - 116</u>
22	Van Waardenburg, G. Van Vliet, H. (2013). "When agile meets the enterprise". <u>Information and Software Technology: 2154-2171</u>
23	Wang, X. Conboy, K. Cawley, O. (2012). "'Leagile' software development: An experience report analysis of the application of lean approaches in agile software development". <u>Journal of Systems and Software: 1287-1299</u>

Quadro 3 Portfólio Bibliográfico

O processo de bibliometria, foi dividido em três principais etapas: (1) Análise dos artigos do portfólio bibliográfico, (2) Análise das referências dos artigos do portfólio e (3) Análise combinada dos artigos e das referências. A seguir será apresentado, sucintamente, cada uma das etapas e respectivas conclusões.

Após a definição do portfólio bibliográfico e a análise quantitativa de dados estatísticos dos trabalhos selecionados, desenvolveu-se a etapa de revisão do conteúdo, ou revisão sistêmica, que também terão os resultados apresentados no desenvolver desse trabalho. Segundo Bortoluzzi *et al.*, 2011, o processo de análise sistêmica permite identificar as lacunas encontradas na literatura sobre um determinado tema de pesquisa, que poderão torna-se possíveis outros trabalhos científicos.

A revisão sistêmica é um processo científico utilizado para analisar o conteúdo do portfólio bibliográfico, utilizando para isso, várias lacunas encontradas nos trabalhos selecionados. Essas lacunas também são chamadas de lentes, cujo objetivo é evidenciar os destaques e oportunidades sobre o contexto da pesquisa em questão. Para o trabalho em questão, foi feito uma adaptação do método Proknow-c, pois o original utiliza outras lentes (Ensslin *et al*, 2010).

Para este trabalho, foram definidas quatro lentes. Que serão descritas após à análise bibliométrica. São elas:

1. Apresentação da Filosofia *Lean*;
2. Coleta de dados e análise;
3. Aplicação prática dos estudos sobre abordagens ágeis;
4. Aderência das práticas *Lean* e Ágil no processo de desenvolvimento de *software*.

2.4 ANÁLISE DOS ARTIGOS DO PORTFÓLIO BIBLIOGRÁFICO E SUAS REFERÊNCIAS

A primeira análise realizada a partir dos artigos selecionados, é a busca pelo grau de relevância dos periódicos. Como resultado, nota-se que um periódico se destaca dos demais, trata-se do *Journal of System and Software*. Dos 23 trabalhos selecionados, sete deles foram publicados no periódico destacado, ou seja, quase 30% dos trabalhos do portfólio foram expostos no *journal* em questão. A Figura 3 ilustra o resultado acima descrito.

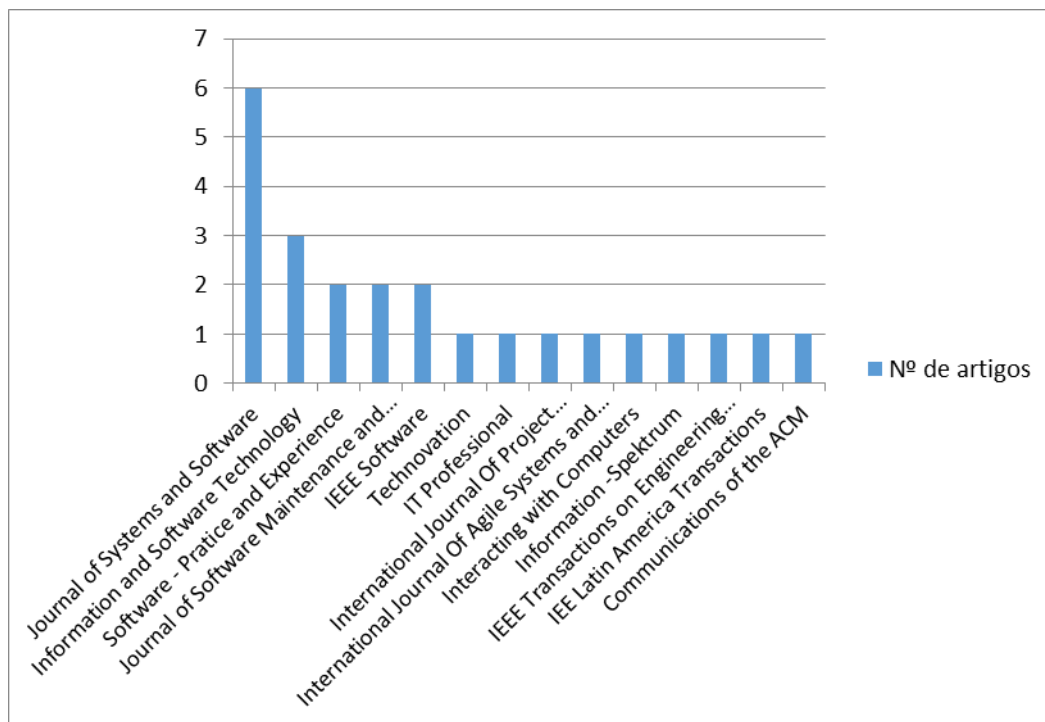


Figura 3: Relevância dos periódicos do Portfólio.

Na próxima etapa do processo de análise bibliométrica, foi feito a análise do reconhecimento científico dos artigos presentes no portfólio bibliográfico. Destaca-se o trabalho *Empirical studies of agile software development: A systematic review*. Esse trabalho apresentou mais de 1100 citações em pesquisa realizada entre os meses outubro e novembro de 2015. Essa quantidade representa cerca de 38% do total de citações de todos os artigos contidos no portfólio.

Outra etapa do processo, é a relevância dos autores do portfólio bibliográfico, conforme Figura 4 abaixo. A referente análise revelou um número considerável de autores distintos. Do total de 23 trabalhos, 68 autores foram encontrados, média de 2.9 autores por artigo. Conclui-se dessa forma, que as publicações estão espalhadas entre diferentes autores. Destacam-se dois autores: Petersen K. e Wohlin, C., P., cujos possuem dois artigos cada. Um trabalho de cada um foi publicado no periódico *Journal of System and Software* destacado como aquele de maior relevância dentre os trabalhos do portfólio bibliográfico.

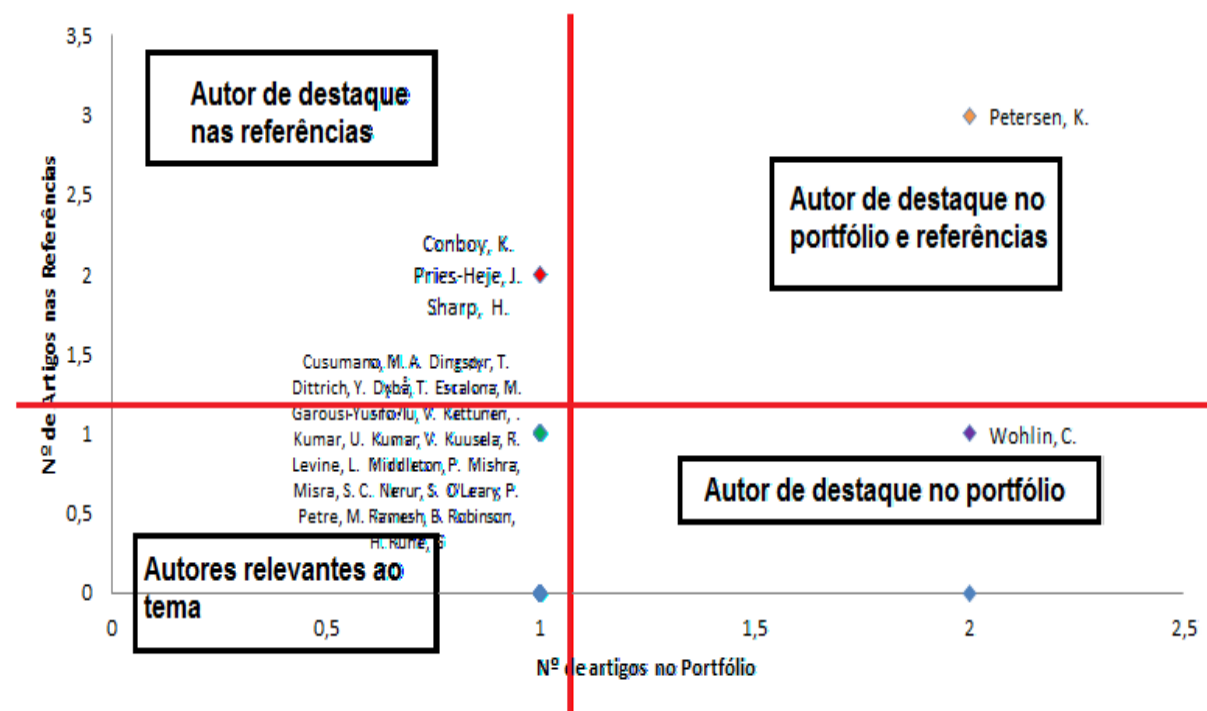


Figura 4 Análise combina de autores

A última fase desta primeira etapa, é a análise da relevância das palavras-chave. Com o propósito de identificar termos mais importantes para o contexto da pesquisa em questão, as palavras-chave de todos os trabalhos do portfólio bibliográfico são tabuladas e o número de vezes que são utilizadas é também catalogado. Dessa forma, as seguintes palavras-chave se destacaram: *Software Design*, *Agile* e *Software Development*, aparecendo, respectivamente, em 16, 8 e outras 8 vezes nos artigos selecionados. É importante ressaltar que dos termos mais relevantes, 2 deles foram contemplados no processo de definição dos eixos e palavras-chave, apresentados na seção 2.3.1 deste trabalho.

Entre os periódicos mais citados nas referências, foi possível abstrair os seguintes: *IEE Computer*, *IEE Software* e *Journal of System and Software*, cujos apresentam, respectivamente 28, 26 e 19 publicações cada, representando 31% do total das publicações.

Com relação aos autores mais vezes referenciados, destacaram-se: Cockburn A. e Wohlin C., ambos foram citados 6 vezes cada.

2.5 ANÁLISE SISTÊMICA

Definidas as lentes da pesquisa, as próximas seções, descrevem os resultados da análise sistêmica dos artigos do portfólio.

2.5.1 Lente 1 - Filosofia Enxuta (*Lean*)

Nos últimos anos, o desenvolvimento de *software* tem evoluído principalmente em torno do processo de fabricação, aplicando práticas ágeis a fim de aperfeiçoá-lo. (Dyba e Dingsøyr, 2008). Entretanto, as abordagens ágeis têm sido cada vez mais questionadas quando o escopo do projeto é variado. (Biffl, *et al.*, 2005). Com o propósito de melhorar os métodos e processos de desenvolvimento de *software*, as empresas começam a observar a abordagem *Lean* da Manufatura como uma estratégia interessante. A aplicação dos conceitos *Lean* para o contexto de desenvolvimento de *software* não é tão simples uma vez que a sua aplicação depende muito do contexto e dos objetivos do ambiente a ser introduzido. (Poppendieck e Poppendieck, 2007). Contudo, através do ajuste das práticas a se utilizar, melhorias substanciais podem ser atingidas no processo de fabricação (Wang, 2012).

O conceito *Lean* para desenvolvimento de *software* é um termo novo e não muito difundido na literatura, porém destaques e resultados interessantes na prática já foram e estão sendo observados. (Wang, 2012).

Dessa forma, a lente em questão, objetiva analisar, dentre os 23 artigos selecionados, quais destacam a filosofia *Lean* nas práticas de desenvolvimento de *softwares* e respectivos estudos científicos. De todos os trabalhos selecionados 61% deles (14 trabalhos) destacam ou citam a filosofia *Lean* na sua respectiva pesquisa, sendo esses: Torrecilla-Salinas, C. J et. al 2015, Ebert, C. et.al 2012, Da Silva, I. F. et. al 2011, Petersen, K. et. al 2011, Petersen, K. et. al 2010, Poppendieck, M. et. al 2012, Fogelström, N. D. et. al 2010, Middleton, P. et. al 2012, Kettunen, P. et. al 2009, Nerur, S. et. al 2005, Dybå, T. et. al 2008, Chow, T. et. al 2008, Van Waardenburg, G. et. al 2013 e Wang, X. et. al 2012.

Dentre os 14 trabalhos, seis deles destacam *Lean Software Development*, como uma metodologia ágil, por possuir características semelhantes àquelas já encontradas em modelos ágeis mais conhecidos, como Scrum, XP, entre outros.

Destacam-se outros dois trabalhos dos seguintes autores: Ebert, C. 2012 e Poppendieck, M. 2012. Trata-se de estudos científicos que descrevem o conceito do *Lean Software Development*. Poppendieck, M. 2012 descreve o LSD não como uma

metodologia da engenharia de *software* tradicional e sim como uma síntese de práticas dos princípios e da filosofia *Lean* para a construção de *software*.

Os outros seis trabalhos que mencionam a filosofia *Lean*, tratam-na como uma evolução dos processos ágeis de desenvolvimento de *software*, onde as lacunas apresentadas pelas práticas ágeis poderiam ser resolvidas com a aplicação de algumas ferramentas do pensamento enxuto. Entretanto, os autores destacam que existem muitos poucos estudos sobre a sua adoção prática, como é o caso do artigo de Wang, X. et. al 2012. A Figura 5 ilustra os resultados acima mencionados.

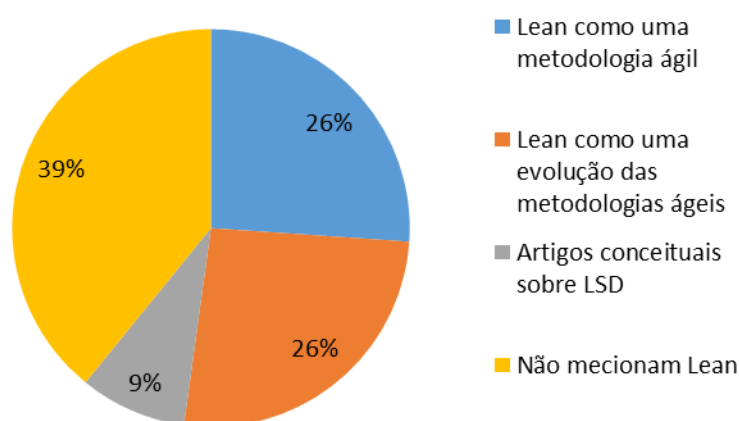


Figura 5: Lente 1 - Filosofia Enxuta (*Lean*)

2.5.2 Lente 2 – Coleta e Análise dos Dados

Para considerar a análise sistêmica desta lente, é necessário analisar de que forma ela é interpretada. Segundo Marconi *et al.* (2007), uma das fases mais importantes de uma pesquisa é a coleta de dados. Esta ajuda a analisar ponto a ponto os fatos ou fenômenos que estão ocorrendo em uma organização, sendo o ponto de partida para a elaboração e execução de um trabalho. Sendo assim, pode-se afirmar que problemas nesta fase, irão dificultar a assertividade nas demais.

Com relação ao portfólio obtido para esta pesquisa, há dois artigos (Chow *et al.*, 2007; Serrador *et al.*, 2015) utilizando o método Survey; 15 são estudos de caso – único e múltiplo – (Kettunen, 2009; Misra *et al.*, 2009; Torrecillas-Salinas *et al.*, 2015; Hansson *et al.*, 2005; Baskerville *et al.*, 2006; Wang *et al.*, 2012; Middleton *et al.*, 2012; Petersen *et al.*, 2010; Petersen *et al.*, 2009; Sharp *et al.*, 2008; Fogelström *et al.*, 2010; Waardenburg *et al.*, 2013, Mishra *et al.*, 2011; Suomalainen *et al.*, 2015; Baptista *et*

al.; 2015) e os demais (Silva *et al.*, 2011; Nerur *et al.*, 2005; Zhi *et al.*, 2014; Dyba *et al.*, 2008; Ebert *et al.*, 2012; Poppendieck *et al.*, 2012) são revisões bibliográficas.

Para garantir melhor acompanhamento e conhecimento dos processos orientados dentro dos projetos de desenvolvimento de *software*, observa-se maior aproveitamento com os estudos de caso (Yin, 2005). Entretanto, é necessário que a equipe disponível para o acompanhamento do estudo esteja preparada para oferecer informações coerentes. Ao analisar os artigos que utilizam este método para coletar dados foi possível observar que apenas dois (Hansson *et al.*, 2006; Mishra *et al.*, 2009) aplicam este processo em conjunto com profissionais das áreas interessadas do projeto em andamento, sendo assim, a maioria dos trabalhos analisados oferece problemas e divergências nas informações obtidas.

Como oportunidade de pesquisa para esta lente, ressalta-se a importância que há no processo de coleta de dados ser realizado com profissionais especialistas de cada área designada da pesquisa.

2.5.3 Lente 3 - Aplicação Prática dos Estudos sobre Abordagens Ágeis

Para considerar a análise da respectiva lente, faz-se extremamente necessário entender a sua importância. Para entender o tema em questão, é preciso uma breve contextualização sobre senso comum e ciência. Segundo Cauchick *et al.* 2010, o senso comum também conhecido como “sabedoria popular”, é obtido a partir de experiências práticas, costumes e hábitos, sendo que a ciência se diferencia principalmente, na utilização de um rigor metodológico. “Geralmente, a palavra rigor metodológico, é utilizada para demarcar a diferença entre a ciência e senso comum”.

Principalmente na Engenharia de Produção, o senso comum e a ciência têm uma forte interação, devido ao fato de que muitos problemas tiveram sua solução primeiramente na prática e depois tornaram interessante para a comunidade científica a partir dos resultados obtidos. De acordo com Cauchick *et al.* 2010, um dos maiores exemplos disso é o Sistema Toyota de Produção.

Assim sendo, uma análise foi feita sobre cada um dos trabalhos selecionados, a fim de identificar quais deles apresentam alguma aplicação na prática, sejam elas estudos de caso práticos, entrevistas com especialistas, entre outras. Assim sendo, dos 23 trabalhos do portfólio, 15 artigos apresentam estudos práticos,

ou seja 61% deles: Torrecilla-Salinas *et al.*, 2015, Hansson, C. *et al.*, 2005, Suomalainen *et al.*, 2015, Poppendieck *et al.*, 2012, Da Silva *et al.*, 2011, Petersen *et al.*, 2011, Fogelström *et al.*, 2010, Serrador, P *et al.* 2015, Kettunen, P, 2009, Baskerville, R. *et al.*, 2006, Nerur, S. *et al.* 2005, Dybå, T. *et al.* 2008, Chow, T. *et al.*, 2008, Wang, X. *et al.*, 2012. Dos 15 trabalhos, quatro deles destacam a aplicação de entrevistas presenciais com especialistas das respectivas áreas, oito destacaram que utilizaram análise de dados secundários, como observações direta, análise de documentos, entre outros. Destes oito trabalhos, vale ressaltar o artigo de Petersen K. 2011, cujo utilizou dos dados coletados para a proposição de um modelo. Os outros 3 trabalhos que ainda não foram citados, tratam-se da realização de pesquisas bibliográficas.

A importância dessa lente é notória tendo em vista que, a grande maioria dos artigos destacaram a sua contribuição prática, o que na área de Engenharia de Produção, como foi destacado pelo autor Cauchick 2010, é uma medida de alta contribuição para a pesquisa.

2.5.4 Lente 4 – Aderência das práticas Lean e Ágil

Antes de analisar os artigos quanto a esta lente, é necessário tornar clara a ideia de aderência para este estudo. No que diz respeito a este termo, não estão sendo levadas em consideração análises estatísticas e quantitativas para testes de aderência, e sim, o que se refere a abordagem utilizada nos processos de desenvolvimento avaliados. Essa característica pode ser observada a partir do cruzamento das práticas *Lean* e *Ágil* já conhecidas e destacadas, com as estabelecidas e realizadas em cada estudo de caso.

Novamente, para compor essa lacuna, são necessárias investigações realizadas a partir de estudos de caso, afim de garantir o índice de coincidências entre os processos utilizados, bem como os resultados obtidos, sendo estes, então, os 15 artigos já citados anteriormente.

Dentre os trabalhos analisados, é possível afirmar que todos apresentam uma pesquisa com posição positiva quanto ao uso de metodologias ágeis no processo de desenvolvimento de *software*, sendo possível demonstrar através de medições,

conforme apresentado em quatro artigos (Mishra *et al.*, 2011; Torrecilla-Salinas *et al.*, 2015; Petersen *et al.*, 2010; Middleton *et al.*, 2012).

Porém, afim de disciplinar a pesquisa quanto aos processos de comparação e aderência de etapas do processo, ou seja, a forma com que as metodologias ágeis podem estar presentes nos processos de desenvolvimento de *software*, desde pequenas empresas, até projetos de desenvolvimento industriais e complexos, é possível observar em apenas cinco dos artigos a ligação entre as etapas de desenvolvimento associadas às metodologias *Lean*/ágil (Hansson *et al.*, 2006; Petersen *et al.*, 2009; Fogelstrom *et al.*, 2009; Van Waardenburg *et al.*, 2012; Misra *et al.*, 2008).

Sendo assim, esta quinta lente se configura como uma lacuna na literatura, um campo de pesquisa a ser explorado com mais profundidade. Pouco se analisou concretamente entre as ligações e associações das metodologias mais praticadas há décadas no desenvolvimento de *software* e os métodos ágeis que estão emergindo em todo mercado. Apesar de vários autores concordarem que o uso eficiente das novas metodologias depende do projeto, pouco se analisou sobre as semelhanças e adesões nas diferentes etapas de projetos de escopos variados com o tema *Lean* e Ágil.

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO 2

O desenvolvimento deste trabalho buscou servir como auxílio para construção de conhecimento acerca do Desenvolvimento *Lean* de *Software*, apoiando-se nas demais metodologias ágeis como fontes de pesquisa, afim de promover investigações futuras sobre este tema. Considerando, também, o interesse dos autores no entendimento da temática LSD em diferentes contextos, para posteriormente identificá-la como possível solução para as lacunas encontradas na implantação de metodologias ágeis no desenvolvimento de *software*.

O processo empregado para a formação da análise bibliométrica e análise sistêmica compôs um portfólio bibliográfico de 23 artigos que permitiu analisar: i) o grau de relevância dos periódicos, apresentando o *Journal of System and Software* com maior destaque com relação aos demais, bem como palavras-chave (*Agile*, *Software Development*, *Leagile*, *Scrum* e *Lean Software Development*) e autores mais

relevantes (Petersen K. e Wohlin, C., P.,); ii) e através da filiação teórica foi possível observar as principais características entre os trabalhos analisados.

Como limitação do presente trabalho, destaca-se a não inclusão de trabalhos que utilizam de outras metodologias de desenvolvimento de software diferente das que utilizam a metodologia Ágil. Vale ressaltar que elas podem apresentar resultados importantes para o contexto da problemática principal da pesquisa em questão, sendo que muitos contextos de desenvolvimento de software ainda se estabelecem com métodos tradicionais, desfrutando apenas de alguns artefatos das novas metodologias, que não chegaram a ser contemplados neste artigo.

Em análise a todo o contexto estudado neste trabalho, é possível afirmar que há duas grandes lacunas presentes no meio científico para o tema Desenvolvimento *Lean* de *Software* e que se apoiam, justamente, na deficiência na apresentação de conhecimento acerca de práticas de modelos aplicadas na gestão e desenvolvimento de *software*, e de não existir ligações de comparação ou aderência entre processos tradicionais e ágeis. Ainda é possível observar que, quando houve aplicação de um modelo específico, o mesmo não acompanhou a documentação necessária para validação científica, dificultando a criação de modelos baseados em estudos de caso empíricos.

Sendo assim, a literatura possui oportunidades para trabalhos futuros a cerca deste tema, considerando especialmente as questões de aplicação do conhecimento teórico em ambiente empresarial, juntamente com a documentação necessária para o registro do mesmo. Como trabalho futuro, estudos empíricos utilizando as práticas descritas dentro do contexto LSD.

3 ANÁLISE DE DESPERDÍCIOS EM EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE DO SUDOESTE DO PARANÁ

Este capítulo tem por finalidade apresentar resultados de estudos de caso múltiplos em três empresas de desenvolvimento de *software* do sudoeste do Paraná, com intuito de observar os casos de excessos e desperdício mais frequente. Analisando, inclusive, a utilização de metodologias ágeis e *Lean* na concepção de seus produtos, focando na melhoria de processo que pode ser vislumbrada a partir da

inserção destes métodos, e os pontos que ainda podem melhorar ao utilizar mais ferramentas de diagnóstico *Lean*.

3.1 INTRODUÇÃO

Atualmente as organizações investem altas quantias de recursos financeiros no desenvolvimento de sistemas de informação. Muitos destes sistemas não geram os resultados prometidos ou esperados e criam um enorme excesso de informação de pouca utilidade (FERRO, 2008). Estima-se que em um *software*, cerca de 45% das funcionalidades implementadas nunca são utilizadas e 19% são raramente utilizadas (JOHNSON, 2002). Tom e Mary Poppendieck (2006), atestam que os dados apresentados por Jim Johnson, apesar de alarmantes, têm se comprovado em levantamentos realizados com clientes e parceiros.

Durante um projeto de desenvolvimento de *software*, as funcionalidades em excesso são incluídas pelo cliente ou empresa desenvolvedora como forma de garantir que todas as necessidades dos clientes serão atendidas. Porém, desenvolver mais que o necessário resulta em um esforço desnecessário, o qual se reflete em desperdícios de recursos no processo de desenvolvimento de *software* (POPPENDIECK, M. *et al.*, 2006).

Qualquer atividade que não gera valor ao cliente é um foco de desperdício e deve ser eliminado (SHINGO, 1989). A eliminação de desperdício é o foco principal do sistema de produção *Lean* inspirado nas práticas na empresa japonesa de automóveis Toyota. Este sistema de produção ganhou destaque em todo mundo por ter como base a eliminação de desperdícios e busca constante pela qualidade e por aplicação do just-in-time (POPPENDIECK, M. *et al.*, 2006).

Deve-se atentar que a aplicação de práticas e princípios criados especialmente para um sistema de produção de automóveis não é trivialmente aplicado na indústria de desenvolvimento de *software*. *Software* é um produto intelectual com um grau de customização maior quando comparado à indústria automobilística. O grande benefício que o legado da Toyota pode trazer para a área de desenvolvimento de *software* é a disseminação do pensamento *Lean* que pode modificar radicalmente a forma como a maioria das empresas desenvolve *software*.

Conclui-se, então, que *Lean* estará na agenda das empresas de desenvolvimento de *software* dos próximos anos (ABRAHAMSSON *et al.*, 2010).

O primeiro passo para aplicar o desenvolvimento de *software Lean* é executar um trabalho intenso de identificação dos focos de desperdícios e tratar de eliminá-los. Os processos de desenvolvimento de *software* e a forma como eles são gerenciados podem estar repletos de excessos. Coman e Ronen (2009) estimam que há mais de 30% de excesso nos recursos de desenvolvimento de *software* para aplicações financeiras que servem organizações como bancos ou companhias de seguros. Mais de 25% dos esforços de desenvolvimento de *software* são dedicados a questões e atividades que não agregam valor (COMAN *et al.*, 2010). Considerando a estimativa conservadora de que 25% do escopo é supérfluo (BATTLES *et al.*, 1996; COMAN *et al.*, 2010), deve-se saber quais são os custos de desenvolvimento excessivo, bem como danos à qualidade e integridade do sistema.

Embora seja muito importante explorar todos os riscos das práticas excessivas em desenvolvimento de *software*, melhorar o conhecimento e consciência destes, bem como apontar práticas para mitigação do problema são áreas com pouca pesquisa relevante. Este trabalho propõe-se analisar três empresas de desenvolvimento de *software*, inseridas no APL de tecnologia do sudoeste do Paraná, quanto às suas práticas de desenvolvimento, com foco no desperdício encontrado. Ainda, neste trabalho, são apontados os benefícios da utilização de metodologias Ágil, reforçando os conceitos de *Lean* para desenvolvimento de *software*, como estratégia relevante na melhoria do processo e na eliminação do desperdício.

O método científico de uma pesquisa corresponde a um caminho para se chegar ao fim de um determinado trabalho (GIL, 1999).

Existem várias maneiras de se classificar uma pesquisa, sendo os pontos mais tradicionais: quanto à natureza da pesquisa, a forma de abordagem do problema, os seus objetivos e os procedimentos técnicos (SILVA E MENEZES, 2000).

Em relação à natureza, este trabalho pode ser classificado como aplicada ou tecnológica, uma vez que o objetivo é gerar conhecimentos para a aplicação prática, dirigidos à solução de um problema específico (JUNG, 2004).

Quanto aos objetivos da pesquisa, este trabalho pode ser considerado como exploratória uma vez que visa à descoberta de teorias e práticas que modificarão a forma de desenvolver *software* na organização.

Com relação à abordagem do problema, a classificação da pesquisa é considerada como qualitativa. Para Penna (2004), “a pesquisa qualitativa caracteriza-se como uma abordagem interpretativa e compreensiva dos fenômenos, buscando seus significados e finalidades”.

O procedimento para realização do trabalho em questão foi caracterizado como estudos de caso múltiplos, sendo que a pesquisa foi realizada em três diferentes empresas do APL de Tecnologia do Sudoeste do Paraná.

Em relação ao método para coleta de dados, este trabalho foi do tipo observação, já que o pesquisador participou de reuniões de Sprint com a equipe de desenvolvimento sem opinar sobre possíveis melhorias, também houveram algumas reuniões com líderes de equipe a fim de reconhecer algumas das maiores dificuldades com relação ao tema abordado na pesquisa.

Esta pesquisa tem interesse em investigar focos de desperdícios, tratados ou não, nas empresas atuantes do arranjo produtivo local (APL) de *software* do sudoeste do Paraná (Brasil), representadas nesse trabalho por três empresas membros do APL. E identificar formas que a aderência às metodologias ágeis, bem como a filosofia *Lean*, pode auxiliar diretamente na melhoria deste ponto no processo.

3.2 REFERENCIAL TEÓRICO

3.2.1 *Lean Software Development*

O termo *Lean* foi cunhado ao final da década de 80 em um projeto de pesquisa do Massachusetts Institute of Technology (MIT) sobre a indústria automobilística mundial. A pesquisa revelou que a Toyota havia desenvolvido um novo e superior paradigma de gestão nas principais dimensões dos negócios (FADEL *et al*, 2010).

Lean é uma gestão de negócios visando aumentar a satisfação dos clientes através da melhor utilização dos recursos. A Gestão *Lean* procura fornecer consistentemente valor aos clientes com os custos mais baixos (Propósito) através da identificação de melhoria dos fluxos de valor primários e de suporte (Processos) por meio do envolvimento das pessoas qualificadas, motivadas e com iniciativa (Pessoas).

O foco da implementação deve estar nas reais necessidades dos negócios e não na simples aplicação das ferramentas *Lean* (Lean Institute Brasil, 2010).

A Toyota procura pelo desperdício que geralmente não é notado porque está inserido como uma parte natural do trabalho diário nas organizações. Por exemplo, a superprodução é comum em processos de várias empresas, pois, garante que metas e prazos serão alcançados com folga. Essas empresas preferem produzir mais que o necessário para compensar os itens fabricados com defeito. Na visão da Toyota a superprodução é um grande foco de desperdício. A solução tomada pela Toyota é encontrar uma maneira de zerar a ocorrência de defeitos, ou seja, focar na qualidade é um caminho para a eliminação de desperdício. Também é possível perceber que a superprodução é quando certo produto é concluído muito antes da data de entrega, neste caso acontece um desperdício pela antecipação, ou seja, foi feito um esforço para concluir algo que ainda não é necessário. Para eliminar este desperdício foi elaborado o *just-in-time* (JIT) que quer dizer no momento certo (SHINGO, 1989).

Como uma forma de lidar com os problemas de fluxo, o just-in-time (JIT) foi criado de forma pioneira pela Toyota por volta de 1950, chegando ao ocidente por volta de 1980. Essa técnica foi criada por Taiichi Ohno com uma forma de harmonizar o fluxo. JIT pode trabalhar de forma efetiva apenas se os tempos e/ou o custo para as mudanças de configuração de máquinas, ou mudanças em geral, forem significativamente reduzidos. O objetivo é que as etapas do processo produzam apenas pequenas quantidades necessárias para dar seguimento nas próximas etapas. A prática do nivelamento da produção (level scheduling) também é importante, para que se possa ter um alinhamento do fluxo, evitando as variações que podem ser causadas pela variação de demanda no dia a dia.

Para se entender o conceito *Lean* de desenvolvimento de *software*, primeiro remete-se ao desenvolvimento de produtos em geral. Existem duas escolas de pensamento sobre como lidar com esta transformação: pode-se chamar uma de escola de pensamento determinística e a outra de escola empírica de pensamento. A escola determinística começa criando uma definição completa do produto e então cria uma realização a partir desta definição. A escola empírica começa com um conceito de produto em alto nível e então estabelece ciclos de feedback bem definidos que ajustam as atividades para criar uma interpretação ótima do conceito.

O pensamento *Lean* trabalha com base na segunda escola, onde o desenvolvimento incremental é necessário para que o produto de *software* seja desenvolvido de acordo com as necessidades do cliente (PRESSMAN, 2006). O primeiro passo para aplicar o desenvolvimento de *software Lean* é executar um trabalho intenso de identificação dos focos de desperdícios e tratar de eliminá-los. Os processos de desenvolvimento de *software* e a forma como eles são gerenciados são repletos de desperdícios (WHEATCRAFT, 2011).

É muito comum, no desenvolvimento de *software*, a preferência por processos baseados no ciclo de vida em cascata. Neste ciclo de vida as etapas acontecem linearmente e a próxima etapa somente é iniciada quando a etapa anterior for concluída. Sendo assim, o teste de uma funcionalidade pelo cliente somente acontecerá após todos os requisitos do sistema terem sido levantados e analisados; e todo o sistema ter sido projetado e implementado (FRANCO, 2007). Nesta abordagem o *software* é tratado de forma determinística. Compreendê-lo assim é uma grande falha. O *software* deve ser compreendido como um produto com requisitos mutáveis e o ciclo de vida em cascata é visivelmente um foco de desperdício, por esconder defeitos e falhas de requisitos por um bom período de tempo, que somente será reconhecido no final do processo. Evitar o ciclo de vida em cascata é fundamental para se aproximar do desenvolvimento de *software Lean*.

Também se deve prestar atenção com o processo de desenvolvimento de *software* RUP (Rational Unified Process), processo da engenharia de *software* que é amplamente adotado no mercado, provê diversas fontes de desperdícios como documentações excessivas e desnecessárias, várias atividades de gerenciamento, especificação com muita antecendência da implementação e troca de conhecimento através de documentos (POPPENDIECK, *et al.*, 2006). Este processo costuma ser aplicado com bastante burocracia, está repleto de documentações e atividades que de fato não geram nenhum valor ao cliente.

Fadel e Silveira (2010), observaram uma série de fontes de desperdícios, que podem ser visualizadas no quadro 4, bastante comum no desenvolvimento de *software*.

Desperdício	Exemplos	Fonte
-------------	----------	-------

Trabalho Incompleto	<ul style="list-style-type: none"> • Documentação não codificada: geralmente um grande esforço é feito no início do projeto para documentar todos os requisitos do <i>software</i>, porém no momento da implementação alguns requisitos foram modificados. • Código não testado: um código foi escrito, mas ele não foi testado. Então este código está inacabado, pois um código somente deve ser considerado pronto quando ele for testado. Isso leva ao desperdício de tempo, pois um defeito corrigido imediatamente se consome bem menos tempo que corrigido tardiamente. 	Kettunen, 2008
Funcionalidades a mais	<ul style="list-style-type: none"> • O desperdício aqui é evidente. Desenvolver o que o cliente não precisa ou o que ele não irá utilizar é por si só um desperdício. • Código não utilizado introduz complexidade ao <i>software</i> que irá dificultar as manutenções. 	Chow, 2008
Reaprendizado	<ul style="list-style-type: none"> • Ter que aprender novamente algo que já foi vivenciado é o que pode se chamar de reaprendizado e de fato é um desperdício. Comumente criar um documento em que se registram os conhecimentos obtidos em um projeto de desenvolvimento de <i>software</i> é a solução adotada por muitas empresas. O problema é que este documento nunca ou raramente é lido. Pensar alternativas melhores para manter o conhecimento na empresa pode ser interessante para evitar este desperdício. 	Chow, 2008; Conboy, 2009
Troca de Tarefas	<ul style="list-style-type: none"> • O desenvolvimento de <i>software</i> é uma atividade que exige concentração. Fazer mais de uma tarefa de cada vez prejudica o raciocínio e a velocidade na realização das tarefas. 	Wang, 2012; Pernstal, 2013

	<ul style="list-style-type: none"> Quando se aloca mais de uma tarefa para apenas um recurso corresse mais risco de que nenhuma tarefa fique pronta no prazo definido. 	
Atrasos	<ul style="list-style-type: none"> Programadores precisam tomar decisões a cada 15 minutos. Acreditar que todas as respostas às suas dúvidas serão respondidas em um documento de especificação de requisitos é utópico e ele precisará aguardar até que um analista de negócio tire sua dúvida. Uma forma de evitar este desperdício é dar ao programador as informações necessárias para desenvolver uma funcionalidade no momento em que realmente for pertinente. 	Chow, 2008; Nerur, 2005

Quadro 4 Desperdícios na área de desenvolvimento de *software*: Fonte: o autor.

Desperdício é qualquer atividade executada que não agrega valor ao produto. Para o ambiente de desenvolvimento de *software*, POPPENDIECK (2003), destaca os seguintes: o desenvolvimento de um componente que não está sendo usado, trata-se de um desperdício; um requisito coletado, documentado e não foi utilizado pela equipe e nem entregue ao cliente final, trata-se também de uma atividade desnecessária; desenvolvimento de rotinas de códigos que não estão e não serão executadas, é uma atividade que gerou desperdício para todo o ciclo de vida do *software*.

O quer que esteja no caminho entre entregar um produto que satisfaça o cliente e a sua entrega propriamente dita, é então considerada uma atividade de desperdício (FOGELSTRÖM, 2010).

Na literatura atual, os conceitos “ágeis” e “*Lean*”, quando aplicados no mesmo contexto, são pouco difundidas e a utilização desses termos é muitas vezes considerada ambígua e inconsistente (CONBOY, 2009). Alguns autores veem os dois termos como nomes diferentes para tratarem a mesma ocorrência, por exemplo, nos estudos de Kettunen e Chow (2008), nenhuma distinção significativa é feita entre as duas práticas. Entretanto, outros estudos, como por exemplo, Wang (2012), Pernstal (2013), destacam a prática *Lean* para desenvolvimento de *software* como uma

metodologia independente da engenharia de *software*, utilizando uma composição dos princípios da filosofia enxuta para a construção de *softwares*.

Este trabalho buscou, como objeto final, trabalhar com características mais específicas do LSD como abordagem independente da metodologia ágil, conceito adotado por diferentes autores (WANG 2012; PERNSTÅL 2013; CHOW 2008; NERUR 2005; MIDDLETON 2012; POPPENDIECK 2007; PETERSEN 2010; PETERSEN 2011; EBERT 2012), mesmo que na realidade das empresas pesquisadas os conceitos ainda sejam muito próximos na execução.

3.3 METODOLOGIA DE PESQUISA

Segundo Jung (2009), a metodologia de pesquisa se caracteriza por um conjunto de métodos, técnicas e procedimentos que objetivam tornar viável a execução da pesquisa, que por sua vez resulta em um novo produto, processo ou conhecimento.

No próximo tópico será descrito o tipo de pesquisa adequado e os procedimentos metodológicos realizados para efetivação deste trabalho.

3.3.1 Estratégia da Pesquisa

A primeira fase foi escolher as empresas que apresentassem características que atendessem a pesquisa, tais como desenvolvimento de software de pequeno e médio porte, equipe de trabalho em projetos de desenvolvimento de até 10 pessoas e interesse em novos métodos de desenvolvimento, segundo o coordenador geral do APL, além disso, que pudessem receber a pesquisa no tempo em que ela foi projetada para ser realizada, sendo neste caso, dois meses para coleta e análise. Após esta escolha, foram agendadas as visitas, inicialmente com o líder da equipe para apresentação da proposta de estudo.

O roteiro do estudo nas empresas sobre práticas de desperdício de *software* foi programado a partir de pontos levantados na revisão bibliográfica sobre especificar, projetar e desenvolver um sistema de *software* além das necessidades reais do cliente ou do mercado, gerando um *software* além das necessidades do

mercado (Boehm e Papaccio, 1988; Markus e Keil, 1994; Ronen e Passe, 2008; Ropponen e Lyytinen, 2000), também casos de mudança de escopo com projeto em andamento (Elliott, 2007; Feiler, 2000) e a sobrecarga de recursos na concepção do *software* (Buschmann, 2010). Para melhor apresentar as análises executadas durante a pesquisa, o quadro 5 a seguir destaca os termos principais analisados, enaltecendo esses três pontos de desperdício.

Categorias Práticas	Termos Utilizados	Autores
AN - Além das Necessidades: Especificar, projetar e desenvolver um <i>software</i> além das necessidades reais do cliente ou mercado.	Excesso de Requisitos	Pass and Ronen, 2014; Shmueli <i>et al.</i> , 2015a, 2015c
	Excesso de Especificação	Abrahams, 1988; Coman and Ronen, 2010; Ronen and Pass, 2008
	Excesso de Design	Coman and Ronen, 2009a, 2010
AP - Além do Planejamento: mudando e adicionando continuamente recursos e funcionalidades além do planejado.	Bells-and-whistles	Ropponen and Lyytinen, 2000
	Mission Creep	Elliott, 2007
	Desvio de Rigor	Elliott, 2007; McConnell, 1997; Rust <i>et al.</i> , 2006
	Desvio de Objetivos	Buschmann, 2009; Feiler, 2000; Lang and Fitzgerald, 2005; Murphy, 2001; Wheatcraft, 2011
	Desvio de Requisições	Damian and Chisan, 2006; Elliott, 2007; Jones, 1996; Wheatcraft, 2011

	Aumento de Recursos na fase de entrega	Buschmann, 2010; Elliott, 2007
AR - Além dos recursos: Definir escopo de um projeto além dos recursos disponíveis, possuindo mais funções que podem realmente ser implementadas no tempo proposto.	Sobrecarga de escopo	Shmueli <i>et al.</i> , 2015
	Aumento de escopo	Bjarnason <i>et al.</i> , 2012

Quadro 5 Categorias e termos de desperdício. Adaptado de Shmueli *et al.* (2017).

Nas empresas investigadas, bem como é característico para este tipo de empreendimento, não é comum o armazenamento de históricos que tragam informações sobre tarefas empreendidas, recursos utilizados, duração. Com a inserção dos métodos ágeis, menos documentação é mantida. O desenvolvimento é trabalhado em relação às lições aprendidas e o conhecimento difundido em reuniões diárias. Em geral as empresas optam por um *software* de controle de versão que guarda um *log* de histórico de edições, porém este é gerado de forma automática com dados pontuais como data e usuário que desenvolveu a tarefa. E ainda outra ferramenta para armazenar as lições aprendidas no processo de desenvolvimento.

A partir da realidade do APL de tecnologia do Sudoeste do Paraná, a pesquisa foi dirigida seguindo etapas conforme Figura 6 abaixo:

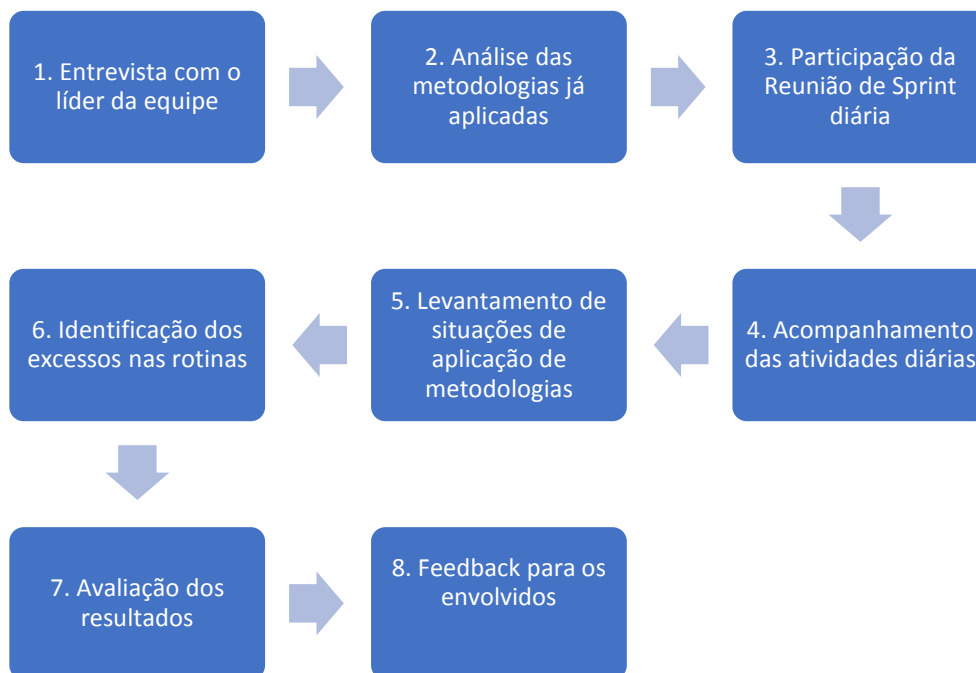


Figura 6 Fluxo metodológico dos estudos de caso

A etapa 1 apresenta-se como primeiro contato direto com a empresa em que a pesquisa está sendo aplicada. Essa entrevista tem por objetivo entender a evolução do processo de desenvolvimento pelo qual a empresa já passou e/ou ainda está passando. Com esse entendimento foi possível analisar as metodologias já aplicadas desenvolvimento de *software* (etapa 2).

A etapa 3 alicerça um melhor acompanhando das resoluções do dia anterior e as tarefas para o dia distribuídas para equipe. Logo após a reunião foi realizado o acompanhamento das atividades de desenvolvimento do dia, resolução de problemas, administração do tempo e de recursos, identificado como etapa 4 da pesquisa.

Acompanhando a forma de trabalho e a realidade da equipe, na próxima etapa se fez o levantamento de situações em que as metodologias estão bem aplicadas e situações com necessidade de melhoria, seguindo da identificação dos excessos apresentados no processo de desenvolvimento nas empresas analisadas (etapa 6). Tendo esse material a disposição como fonte de informação, houve o processamento dos mesmos e avaliação dos resultados, como etapa 7 do processo.

Por fim, após o término de todas as pesquisas e o aprofundamento em relação aos excessos observados, foi possível fazer um feedback da pesquisa. Esse retorno aconteceu como forma de dar uma visão externa de como a empresa está trabalhando e incentivar a equipe a melhorias, dentro da realidade de cada organização.

3.4 ESTUDOS MULTI-CASO

Esta sessão apresentará o resultado obtido em cada uma das empresas. É importante salientar que a pesquisa não tem por objetivo um comparativo de metodologias empregadas pelas mesmas, mas sim a identificação e análise de melhorias executadas e em execução, bem como as lacunas identificadas no processo, especialmente no tratamento e ocorrência de desperdícios e excessos nas atividades de desenvolvimento.

3.4.1 Estudo de Caso 1

A primeira empresa analisada está no mercado desde 2011. Trata-se de um empreendimento de pequeno porte, possuindo 193 clientes ativos espalhados em todo o Brasil.

Há aproximadamente um ano, a equipe de desenvolvimento passou a adotar métodos ágeis como forma de trabalho. Como este tempo é relativamente curto, foi possível ter mais informações, por parte da equipe, sobre como era o trabalho antes e o que realmente gerou mudança na inclusão de metodologias ágeis de desenvolvimento.

O estudo iniciou em uma entrevista com o líder da equipe que resultou em alguns dados sobre o perfil da equipe, cronogramas de reuniões periódicas e projetos em andamento. As estratégias metodológicas de desenvolvimento empregadas pela empresa são Scrum e Kanban. A primeira com objetivo de agilizar e organizar o processo de construção do *software*, e a segunda para distribuir e controlar tarefas e recursos. Antes da adoção dessas práticas, a equipe se reunia a cada 15 dias para definir o que seria implementado, ocasionando acúmulo de tarefas, mal planejamento

e como não havia acompanhamento, os atrasos se tornavam comuns e o desperdício de tempo e esforço sem *feedback* das atividades, também era alto. No estado atual, as reuniões acontecem da forma que o método *Scrum* indica. São reuniões diárias, chamadas de *Daily* em que a equipe repassa o andamento do dia anterior e é feito o planejamento diário individual de cada membro do grupo. Além dessas reuniões diárias, sempre que se inicia uma nova *Sprint*, ou seja, um novo fluxo de trabalho para entrega, é feita uma reunião de alinhamento e distribuição geral de tarefas.

Mesmo considerando grande melhoria na assertividade das atividades realizadas, as reuniões diárias não garantem a eliminação do desperdício. Ainda é muito comum, na empresa, serem recebidas solicitações de desenvolvimento do cliente para customização do processo que emprega grande tempo para produzir e que facilmente pode se tornar obsoleto.

Dentro das três categorias elucidadas anteriormente na metodologia, alguns excessos são observados claramente na empresa 1. Em uma situação real, presenciada durante a pesquisa, um cliente abre um chamado no departamento de suporte da empresa relatando um problema encontrado no sistema. O problema relatado possuía forte aspecto processual, ou seja, a demanda era específica para atender o cliente em questão. Por se tratar de um cliente de grande porte, e o problema atrasar o funcionamento do negócio, o suporte trata o chamado como prioritário e é feito repasse imediato para o departamento de desenvolvimento, gerando uma *Sprint*. A análise do chamado é feita através de contato com o cliente para levantamento de requisitos. Neste momento é comum o cliente, devido à sua sobrecarga de trabalho, salientar algumas necessidades que parecem ser obrigatórias, mas que poderiam ser solucionadas de outra forma se melhor analisadas.

Nesta situação identifica-se desperdício na categoria AN, levado pelo excesso de requisitos, seguido do excesso de especificação tratado na análise. E também é possível identificar o excesso de recursos que são demandados para esse tipo de atendimento, entrando na categoria AR.

Mesmo os excessos existindo, o líder da equipe salientou que, ainda que com pouco tempo da implantação de novas metodologias, o tempo para as *Sprints* serem homologadas, bem como sua gestão e desempenho dos funcionários melhorou

em 30%, trazendo para a empresa maior agilidade no processo, e permitindo que um novo produto possa ter entrado em produção.

Outro aspecto ainda evidente salientado na pesquisa, é que o desperdício ainda acontece consideravelmente com o aumento do escopo do projeto durante sua execução, ou seja, ainda na situação exposta acima, em conversa com o cliente para implantar as alterações, surgem novas necessidades de procedimentos que devem ser implementados para que a solicitação inicial seja completamente atendida.

Foi observado também a importância de manter um *software* para ajudar na aplicação dos métodos. Existem diversidades destes sistemas disponíveis no mercado, grande parte são plataformas gratuitas que atendem a essa demanda de pequenas e médias empresas.

No processo realizado na primeira empresa já foi possível perceber que a aplicação de instrumentos e métodos para desenvolvimento de *software*, como Scrum, Kanban, entre outros, já proporcionariam um crescimento importante de toda a equipe, bem como um ganho real em tempo de desenvolvimento, e principalmente, como foco desse trabalho, a queda exponencial da quantidade de retrabalho, excesso e desperdício durante todo o processo de desenvolvimento de *software*.

3.4.2 Estudo de Caso 2

A segunda empresa pesquisada está em atividade há mais tempo em relação a primeira (desde 2003), quando ainda estava incubada através de projetos de empreendedorismo públicos municipais, quando a partir de 2005, passou a agir de forma autônoma. Inseriu-se no mercado atendendo uma demanda específica de *software*, mas seu crescimento proporcionou a busca por outras áreas de desenvolvimento e também a entrada de novas tecnologias para a concepção das mesmas. Hoje possui vários produtos desenvolvidos e mais de 500 clientes (incluindo matrizes e filiais).

Nesta companhia, a implantação de metodologias ágeis vem sendo trabalhada desde 2012, sendo inicialmente feita através do *Scrum*. Antes disso todo o formato de desenvolvimento era com base no modelo Melhoria de Processo de *Software* Brasileiro (MPS-BR), que também se trata de uma metodologia focada em

qualidade do produto (Pires, 2012), porém não se insere diretamente nas práticas ágeis.

A mudança de um modelo para outro gerou grandes resultados operacionais e de interatividade nas equipes. Mais do que a mudança na forma de gerir o trabalho, gerou uma mudança de comportamento e pensamento nos recursos humanos inseridos nas áreas de desenvolvimento, conforme relato do supervisor.

Neste estudo foi possível observar diretamente a interação com a metodologia LSD, mesmo que ainda de forma embrionária. A empresa está em processo de estruturação do modelo *Lean* MVP (Produto Minimamente Viável) ou *Lean* Startup, que investiga a viabilidade do produto revelando se a ideia corresponde a algo interessante (comercialmente) ou se trata-se apenas de uma expectativa utópica, sem lastro com as demandas praticadas no mercado (Lippoldt and Strykowski, 2009), que ajuda a ampliar práticas ágeis e *Lean* dentro da organização. Em meio aos pilares deste modelo, a empresa se insere no *Lean* Inception, muito utilizado na criação de Startups, desenvolvimento fortemente defendido pela companhia atualmente.

Durante entrevista com o líder da área de criação e desenvolvimento, algumas informações foram colocadas em destaque. Hoje a empresa está, ainda, fortemente dividida em sua área de produção em duas grandes equipes. Uma delas com foco em novas tecnologias e desenvolvimentos rápidos e soluções inteligentes, enquanto outra equipe trabalha com objetivo de manter o funcionamento do primeiro sistema que entrou no mercado, para cooperativas de crédito. Não apenas o ambiente é dividido bem como a forma de trabalho. Exatamente neste ponto é possível evidenciar grandes problemas com relação às práticas excessivas no desenvolvimento. Esta equipe atende os clientes sob demanda, ou seja, é focada em resolver problemas com urgência. Neste processo, é comum haver excesso de requisitos, abordado na categoria AN, pois o problema a ser resolvido gera uma demanda urgente, que para solucionar é mais prático fazer o que se pede sem análise adequada do que é solicitado. Sendo que deste modo muitos processos são repetidos dentro do desenvolvimento, há excesso no código, aumento do escopo e também nos recursos alocados para resolver a demanda, afirma o supervisor da equipe analisada quanto aos problemas mais comuns enfrentados, excessos caracterizados na categoria AP e AR. Por se tratar de problemas relacionados às cooperativas de

crédito, as soluções devem acontecer com urgência, por exemplo, o cliente cooperativa “x” abre um chamado apontando um defeito no sistema referente ao cadastro de um cliente que está sendo atendido. O chamado é caracterizado como urgente, e torna-se prioridade no atendimento. Como a solução deve ser oferecida rapidamente, o problema acaba sendo resolvido de modo paliativo, atendendo exatamente o que o cliente ‘cooperativa’ solicitou, o que paradoxalmente pode ser incorreto (o cliente às vezes pode solicitar algo inadequado). Neste caso é comum alocar mais recursos e diminuir o rigor com que a correção é efetuada, gerando excessos característicos das categorias AP e AR.

Ainda que hajam esses gargalos pelo excesso de desenvolvimento em parte da empresa, a introdução de métodos ágeis, pensamento *Lean*, melhorou significativamente a qualidade do produto final. As reuniões, que com o MPS-BR eram apenas processo burocrático e esporádicas, tornaram-se semanais, com trocas de ideias sobre as demandas, atualização do trabalho realizado, lições aprendidas e troca colaborativa de conhecimento. E além das reuniões, para troca de informações e aprendizado, os gestores apresentaram algumas estratégias visuais que são utilizadas para melhorar o trabalho em equipe. Algumas dessas ferramentas são: WordPlace, que funciona como uma rede social interna para compartilhamento de ciência; Jira, que coordena as atividades de cada time; Trello, como ferramenta online para gestão de projetos; entre outras.

3.4.3 Estudo de Caso 3

A terceira empresa a ser pesquisada está em atividade desde 2003, tem como foco o desenvolvimento de soluções específicas prezando pela qualidade, garantia de informações íntegras, e suporte técnico dedicado, possuindo em torno de 390 clientes.

Trata-se de uma empresa que, mesmo estando presente no mercado há um bom tempo, apresentou pouca evolução quanto às práticas e metodologias ágeis ou até mesmo, métodos semelhantes.

Nesta empresa, o contato foi direto com o gestor da área de desenvolvimento que se dispôs a discorrer sobre a realidade de processo em que toda

a equipe trabalha. É possível identificar, já nesse primeiro contato, a inexistência de uma metodologia fortemente agregada ao desenvolvimento de *software*. Ainda baseado nos primeiros modelos de construção de sistemas, pode-se observar o modelo Cascata ainda sendo fortemente defendido e utilizado.

Neste formato de trabalho, os excessos são amplamente reconhecidos. Há muitas solicitações de cliente que não agregam valor ao produto, caracterizando AR. Há excesso no desenvolvimento a fim de atender a demanda com praticidade e rapidez, mas com operações mais imediatistas, ou seja, muito trabalho é feito, e muitas rotinas tornam-se redundantes no processo.

O escopo também sofre pelo excesso, pois nem sempre o cliente tem a ideia clara do que é necessário ser desenvolvido para atender suas necessidades, tornando o desenvolvimento categorizado como AN. E com propósito de garantir a satisfação do mesmo, acaba-se gerando um aumento considerável de demanda, abstraindo mais recursos e aumentando o custo para o desenvolvimento (AR).

O supervisor da produção reitera que o pensamento da equipe é bem voltado para um desenvolvimento tradicional. Em geral as mudanças propostas não são bem acolhidas, pois levam a mudança no formato de trabalho e o perfil da equipe não está preparado para isso.

Hoje está sendo trabalhada a inserção de metodologia Scrum para difundir conhecimento e proporcionar reuniões periódicas, a fim de criar um ambiente de troca de ideias e melhoria no aprendizado. Pouco se evoluiu, mas é possível observar uma abertura por parte dos recursos em participar dessas reuniões.

Nesta empresa, especificamente, a maior resistência na utilização de novos métodos ainda é por parte dos colaboradores. As mudanças geram muitas dúvidas e inseguranças e o processo acaba sendo retardado. Foi possível analisar que a sobrecarga de tarefas e demanda é muito comum, pois a troca de informações não é assertiva e periódica, logo grande parte fica concentrada a um único colaborador, gerando entregas atrasadas, excesso de requisitos, entre outros.

3.5 RESULTADOS E DISCUSSÕES

No momento em que a pesquisa foi iniciada, as empresas foco foram indicadas pelo Coordenador Geral do APL de TI do sudoeste do Paraná. Sendo assim, o fato de três empresas bem distintas terem sido investigadas tornou-se uma estratégia para dar maior amplitude à pesquisa, observando é claro que estudos de caso não permitem generalização de resultados.

Este trabalho se concentrou em analisar três práticas excessivas no desenvolvimento de *software*: Além das Necessidades (NA), Além do Planejamento (AP) e Além dos Recursos (AR), abordadas na literatura através de alguns termos que podem se sobrepor e até mesmo receberem tratamentos confusos. Os estudos de caso buscam apresentar situações reais que pertençam a essas categorias de excesso. Apresentam, também, práticas das empresas para melhoria da qualidade do produto gerado, bem como áreas de processos em que a aplicação de LSD pode ajudar a diminuir o desperdício e aumentar a produtividade com qualidade, conforme ainda será comentado nessa seção.

O quadro 6 a seguir busca realizar uma aproximação entre as informações tratadas em cada estudo, com o objetivo de identificar pontos em comum entre as empresas, mas também as possibilidades de evolução que podem haver dentro do APL.

	Empresa 1	Empresa 2	Empresa 3
Tempo de Empresa	Desde 2011	Desde 2003	Desde 2003
Número de Clientes	193	Mais de 500	390
Uso de metodologia ágil	Há um ano	Desde 2012	Há um ano
Estratégias adotadas	Scrum e Kanban	Scrum, Kanban, Lean MVP	Iniciando no Scrum
Excessos e/ou Problemas Identificados	<ul style="list-style-type: none"> Gargalos no atendimento de suporte (AR); 	<ul style="list-style-type: none"> Divisão das equipes de desenvolvimento para atender 	<ul style="list-style-type: none"> Equipe pouco adepta à mudanças no processo de desenvolvimento;

	<ul style="list-style-type: none"> • Dificuldade em efetuar testes de qualidade (AP); • Retrabalho nas customizações (AN e AR). 	demanda especial (AR); <ul style="list-style-type: none"> • Gargalos por falta de metodologias em parte da equipe (AN e AP) 	<ul style="list-style-type: none"> • As informações ficam concentradas em poucas pessoas (AP e AR);
Pontos Favoráveis	<ul style="list-style-type: none"> • Equipe engajada com novas tecnologias; • Adoção de ferramentas ágeis com facilidade; • Processo bem definido. 	<ul style="list-style-type: none"> • Equipe totalmente interagida; • Equipe interessada em novas tecnologias; • Adoção de diversas ferramentas para organizar a troca de informações e metodologias da equipe; • Processo bem definido 	<ul style="list-style-type: none"> • Equipe com bastante experiência em desenvolvimento
Melhorias após visita	<ul style="list-style-type: none"> • Coordenador da equipe iniciou formação em práticas Lean; 	<ul style="list-style-type: none"> • Aproximação dos processos em toda a empresa; • Disponibilidade para mais funcionários se especializarem em técnicas Lean de desenvolvimento. 	<ul style="list-style-type: none"> • Equipe mais participativas nas reuniões de Sprint; • Colaboradores estudando metodologias ativas para adoção.

Quadro 6 Análise Resumida dos Estudos de Caso

Durante a coleta de dados, foi possível observar que as empresas analisadas não demandam históricos numéricos dos trabalhos realizados. O que eles costumam manter é realmente o acervo de práticas aprendidas e conhecimento disponibilizado. Dessa forma não foi possível mensurar de forma objetiva qual a evolução e crescimento de cada empresa mediante a empregabilidade de metodologias ágeis.

O excesso é característica forte e evidente em praticamente todas as empresas de desenvolvimento de *software*. A pesquisa apresenta três casos que enaltecem, de formas diferentes, essa “patologia” organizacional no desenvolvimento de *software*. Porém da mesma forma que é evidente sua existência, também foi possível observar com clareza como a introdução de uma metodologia bem trabalhada, concisa e consistente durante a produção de sistemas, poderá ajudar a resolver grande parte desses problemas.

Um dos pontos relevantes do método LSD é justamente o controle dos excessos no desenvolvimento. A empresa 2 é a única, nesta pesquisa, que já insere diretamente essas técnicas dentro do formato de trabalho. E é possível afirmar, através de relatos do coordenador e da equipe, a melhoria significativa no tempo de desenvolvimento, na qualidade do produto ofertado, bem como no crescimento pessoal e profissional de todos os envolvidos. A empresa 1, ao empregar métodos ágeis de concepção de *software*, utiliza de meios que melhoram o processo de desenvolvimento e minimizam problemas, porém as situações observadas durante o estudo mostram que o emprego do pensamento enxuto através do LSD pode tornar o processo ainda mais evoluído. Steffen (2011) apresenta uma relação de ferramentas e métodos, referentes à produção de *software*, que podem ser associados a cada princípio praticado pelo *Lean Software Development*, não apenas na eliminação do desperdício, mas para todas as fases. Focando-se nos excessos observados dentro das empresas, é possível afirmar que a utilização de VSM (Value Stream Mapping), técnica *Lean* que consiste em criar representações gráficas dos processos, fluxos e informações para identificar situações específicas (WANG *et al.*, 2012; STEFFEN, 2011), ajuda a identificar os pontos de excesso e desperdício que passam despercebidos. Ao criar um mapeamento de fluxo de valor das atividades mais críticas, é possível apontar as fases do sistema que merecem maior demanda de tempo e recursos, e com isso planejar o desenvolvimento. Inclusive quando trata-se de correções de *bugs*. A ferramenta também ajudaria na melhoria do processo da empresa 03, agregando maior valor ao produto e diminuindo tempo e gastos na produção.

Após o repasse do feedback para as três empresas analisadas, houve interesse em todas em formação da equipe para atualização do processo de trabalho. Como demonstrado no quadro 6 anteriormente, todas buscaram realizar alterações

no desenvolvimento a fim de melhorar a qualidade do produto e diminuir os excessos gerados ao longo do processo. Isso qualifica a pesquisa como forma de contribuição para a realidade na qual ela foi aplicada.

3.6 LIMITAÇÕES E PESQUISAS FUTURAS

Ainda que as três empresas pesquisadas produzam essencialmente sistemas, não é possível realizar um comparativo direto e específico entre elas, ou entre quaisquer outras que participassem da pesquisa, pois cada uma possui características próprias de trabalho, formatos de equipes diferenciados, maturidade empresarial distintas e os produtos desenvolvidos são diversificados. Logo, foi plausível identificar focos que precisam de maior atenção, áreas em desenvolvimento, mudanças ao longo da adoção de novas metodologias e enquadrá-las em alguns aspectos mais fáceis de analisar, conforme capítulo anterior.

Todavia, o pensamento *Lean*, especialmente empregado no desenvolvimento de *software*, abrange todas as diferenças encontradas entre as empresas, e oferece soluções nas mais diferentes vertentes de pensamento e desenvolvimento. Pois não se trata apenas de ferramentas, mas sim de conceitos e forma de trabalho, que associado a realidade já trabalhada, pode gerar inúmeros benefícios e melhoras significativas.

Existem lacunas muito grandes quanto a pesquisa acadêmica aplicada e empírica na área do desenvolvimento *Lean* de *software*. Portanto ainda é possível realizar inúmeros trabalhos, abrangendo pesquisa longitudinal com maior acompanhamento em um ciclo inteiro de desenvolvimento, a fim de apontar estratégias para cada fase, bem como aguçar o levantamento de informações nos diferentes tipos de estratégias empregadas, benefícios e dificuldades encontradas.

É conveniente, também, executar mais estratégias de pesquisa bibliográfica sobre desenvolvimento *Lean* de *software* e métodos ágeis, com aderência e foco nas ferramentas disponíveis para utilização.

Conclui-se, então, que o campo de pesquisa ainda é vasto e com inúmeras possibilidades para trabalhos posteriores. Pois várias questões ainda puderam ser

levantadas e áreas interessantes para pesquisa surgiram através das lacunas criadas nestes estudos.

3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO 3

Este trabalho se propôs a investigar três empresas desenvolvedoras de *software* inseridas no Sudoeste do Paraná com o objetivo de analisar as metodologias de desenvolvimento de produto inseridas nessas empresas, e concentrando o foco em considerar os excessos praticados, e a influência que métodos ágeis e o LSD podem exercer na melhoria dos processos.

Cada empresa apresentou um grau de inserção em metodologias bem diferenciado e com focos e pontos diferentes no planejamento. Como isso faz parte de práticas empresariais específicas, não foi possível compará-las através de um referencial específico, porém foram investigadas individualmente, considerando o que cada uma poderia oferecer de informações e práticas empregadas no trabalho.

Após estudos nas três empresas foi possível perceber que os excessos, em suas três dimensões (AR, AN e AR), estão fortemente presentes em diferentes etapas do processo de desenvolvimento em todas elas. Em contrapartida, também foi possível observar significativas melhorias com a inserção de metodologias que buscam minimizar esses desperdícios, como a LSD. Isso se aplica mais diretamente a empresa 2 apontada na pesquisa.

Essa área de pesquisa oferece muitos campos a serem ainda desbravados com afinco. Por isso existem várias opções de pesquisas futuras, entre elas a possibilidade de investigar processos de *startups* desde a concepção até a entrega ao cliente, tratando os excessos com a inserção da metodologia LSD no processo. Também é possível desenvolver uma pesquisa mais longa, com um produto de maior complexidade e ao final apresentar propostas de melhorias em diferentes etapas da construção do sistema. Inúmeras são as oportunidades a serem investigadas, pois trata-se de um assunto que ainda não foi totalmente pesquisado.

4 CONSIDERAÇÕES FINAIS

A motivação para este trabalho foi identificar nas empresas do APL de TI do sudoeste do Paraná, os excessos ocasionados no desenvolvimento de *software*, bem como as estratégias e metodologias já utilizadas e como elas podem melhorar o processo de produção de sistemas, especialmente no que diz respeito a eliminação do desperdício.

Como mostrado no início deste trabalho, apenas 32% dos projetos de *software* são bem sucedidos, com isto novas propostas para implementação de *software* devem ser exploradas pelas organizações.

O trabalho desenvolveu uma base teórica sobre o *Lean Thinking*, *Lean Software Development (LSD)* e o desenvolvimento Ágil. O *Lean Thinking* foi conceituado em um conjunto de princípios e ampliado sua aplicação para fora da área da manufatura (Womack *et al.*, 2003). Poppendieck *et al.* (2003; 2007; 2011) estudaram a aplicação do *Lean* para a área de desenvolvimento de *software* e, seguindo a mesma linha de Womack *et al.* (2003), desenvolveram um conjunto de princípios para o *Lean Software Development*. Poppendieck *et al.* (2003; 2007) e Poppendieck e Cusumano (2012) são amplamente citados por vários autores que estudam *Lean* em *software*. O desenvolvimento Ágil foi apresentado com forte relação com as práticas *Lean*. Alguns dos autores estudados também relacionam as duas práticas e alguns de seus conceitos.

A revisão da literatura estruturada e sistemática deixou evidente a existência de lacunas acerca de estudos empíricos da aplicação de metodologia *Lean* na área de sistemas de informação. Todavia, vários trabalhos identificaram a importância da filosofia *Lean* juntamente com as práticas ágeis como resposta às problemáticas encontradas no desenvolvimento de *software*. Estas lacunas e conclusões justificaram esta pesquisa e sua área de extensão. Nesta primeira etapa obteve-se o portfólio bibliográfico que nutriu o restante dos estudos, identificando os principais autores, bem como as áreas emergentes a serem estudadas.

A segunda parte deste trabalho foi acerca, diretamente, dos estudos de caso em três empresas desenvolvedoras de *software*, previamente escolhidas por indicação do APL. Sendo a problemática da pesquisa já destacada, com intuito de

observar os casos de excesso encontrados no desenvolvimento e também a inserção de metodologias ágeis e *Lean* nas empresas, a pesquisa dedicou-se a explorar situações que ainda precisam ser corrigidas e apontar como o *Lean* pode solucionar estes pontos.

As três empresas já apresentam algum tipo de inserção em aplicações de metodologias ágeis e *Lean*, porém em níveis bem diferenciados. A empresa 1 em torno de um ano de práticas ágeis, a empresa 2 já há mais de cinco anos envolve-se com práticas ágeis, e já inserem argumentos e métodos LSD em seu processo de produção de sistemas. Já a empresa 3 pertence, ainda, a um estágio mais precoce de evolução em termos de metodologias de desenvolvimento. Porém em todas elas há identificação de excessos pertencentes às três áreas identificadas: Além do Planejamento (AP); Além das Necessidade (AN); Além dos Recursos (AR). Práticas excessivas no desenvolvimento sobrecarregam os projetos de *software* com funcionalidades e capacidades extras e excessivas, e tem sido considerada o principal risco no processo (Boehm, 1991; Brooks, 1975; Jones, 1994).

O *Lean* para a área de desenvolvimento de *software* é uma realidade e pode trazer ganhos importantes para essa indústria. A base teórica vem se desenvolvendo, principalmente a partir do início deste século, e mais recentemente com publicações sobre modelos e ferramentas. Através da revisão bibliográfica e do contato com as empresas é possível também observar o quanto que o assunto *Lean* para a área de *software* é recente na área acadêmica e na indústria de TI. Apesar de recente, é um tema que desperta um grande interesse em ambas as áreas.

As lacunas deste trabalho tornam-se oportunidades de trabalhos futuros. Como o acompanhamento de uma startup ou outro desenvolvimento desde sua solicitação até sua entrega, afim de validar cada processo e suas possibilidades de melhoria. É possível sugerir uma pesquisa-ação, que possa ir solucionando alguns problemas que aparecem durante a construção do sistema, especialmente os excessos encontrados. E a fim de melhorar o acervo estruturado de pesquisa na área, ainda é possível realizar novas pesquisas estruturadas na bibliografia, com novas revisões sistêmicas identificando mais lacunas para outras pesquisas.

5 REFERENCIAL BIBLIOGRÁFICO

ABRAHAMMS, P. President's letter. Commun. ACM 31, p. 480–481, 1988.

ABRAHAMMS, P.; OZA, N. Lean Enterprise Software and Systems. Helsinki: Springer, 2010.

ANJOS, S. J. G; ABREU, A. F. La medición de la calidad de servicio: una aplicación en empresas hoteleras. Revista Europea de Dirección y Economía de la Empresa, v. 18, n° 6, p. 175-186, 2008.

BAPTISTA, G. L.; VANALLE, R. M.; SALLES, J. A. A. A Software Development Process Model Integrated to a Performance Measurement System. IEEE Latin America Transactions: p. 739 – 745, 2015.

BASKERVILLE, R.; RAMESH, B.; LEVINE, L.; PRIES-HEJE, J. “High-speed software development practices: What works, what doesn't”. IT Professional: p. 29-36, 2006.

BASSI FILHO, D. L. Experiências com desenvolvimento Ágil. São Paulo, 2010.

BATTLES, B.E.; MARK, D.; RYAN, C. An open letter to CEOs: how otherwise good managers spend too much on information technology. McKinsey Q. p. 116–127, 1996.

BECK, K. eXtreme Programming Explained: Embrace Change. Addison-Wesley, San Francisco, 2000.

BIFFL, S.; AURUM, A.; GRÜNBACHER, P.; BOEHM, B. Value Based Software Engineering. Springer, 2005.

BJARNASON, E.; WNUK, K.; REGNELL, B. Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering. Inf. Softw. Technol. 54 (10), p. 1107–1124, 2012.

BOEHM, B.; PAPACCIO, P. Understanding and controlling software costs. IEEE Trans. Softw. Eng. 14 (10), p. 1462–1477, 1988.

BOOCH, G. Object Solutions: Managing the Object-Oriented Project. Addison-Wesley. 1995.

- BORTOLUZZI, S.C. *et al.* Avaliação de desempenho em redes de pequenas e médias empresas: Estado da arte para as delimitações postas pelo pesquisador. *Estratégia & Negócios*. V. 04, n. 02, p. 202-222, jun/dez. 2011
- BUSCHMANN, F. Learning from failure, part 1: scoping and requirements woes. *IEEE Softw.* 26 (6), p. 68–69, 2009.
- BUSCHMANN, F. Learning from failure, part 2: featuritis, performitis, and other diseases. *IEEE Softw.* 27 (1), p. 10–11, 2010.
- CACM Journal - Communications of the ACM, Special Issue on “Services Science,” 49/7 (July 2006); Journal of Retailing, Special Issue on “Service Excellence,” 83/1, 2007.
- CAUCHICK et al. Metodologia da pesquisa em engenharia de produção e gestão de operações. Rio de Janeiro: Elsevier, 2010.
- CERVO, A. L.; SILVA, R.; BERVIAN, P. A. Metodologia Científica. Pearson Education - Br. 6ª ed. 2007.
- CHOW, T.; CAO, D. B. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*: p. 961-971, 2008.
- COCKBURN, A. Agile Software Development. Addison-Wesley, UK. 2002.
- COMAN, A.; RONEN, B. Focused SWOT: diagnosing critical strengths and weaknesses. *Int. J. Prod. Res.* 47 (20), p. 5677–5689, 2009a.
- COMAN, A.; RONEN, B. Overdosed management: how excess of excellence begets failure. *Hum. Syst. Manag.* 28 (3), p. 93–99, 2009b.
- COMAN, A.; RONEN, B. Icarus' predicament: managing the pathologies of overspecification and overdesign. *Int. J. Proj. Manag.* 28 (3), p. 237–244, 2010.
- CONBOY K., Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development, Information Systems Research. 2009.
- DA SILVA, I. F.; DA MOTA S. N.; O'LEARY P. A.; P; DE ALMEIDA, E. S; DE LEMOS MEIRA, S. R. Agile software product lines: A systematic mapping study. *Software - Practice and Experience*: p. 899-920, 2011.
- DAMIAN, D.; CHISAN, J. An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in

productivity, quality, and risk management. *IEEE Trans. Softw. Eng.* 32 (7), p. 433–453, 2006.

DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*: p. 9-10, 2008.

DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. 50 (9-10), p. 833-859, 2008.

EBERT, C.; ABRAHAMSSON, P.; OZA, N. Lean software development. *IEEE Software*: p. 22-25, 2012.

ELLIOTT, B. Anything is possible: managing feature creep in an innovation rich environment. *Proceedings of the IEEE International Engineering Management Conference*, July 29–Aug 1, Piscataway, New Jersey. 2007.

ENSSLIN, L.; ENSSLIN, S. R.; LACERDA, R. T. O.; TASCA, J. E. Processo de seleção de portfólio bibliográfico: Processo técnico com patente de registro pendente junto ao INPI. Brasil, 2010a.

FADEL, A. C.; SILVEIRA, H. M. Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean. Limeira, 2010. Disponível em: <http://www.ceset.unicamp.br/liag/Gerenciamento/monografias/Lean%20Agil_v8.pdf> Acesso em: 18 set. 2017.

FEILER, E. Evaluating accounting software consultants. *CPA J.* 70 (6), p. 46–51, 2000.

FERRO, J. R. O Movimento Lean no Brasil e no mundo. *LeanM@il*, 2008. Disponível em: <<http://www.Lean.org.br/Leanmail/19/o-movimento-Lean-no-brasil-e-nomundo.aspx>>. Acesso em: 23 ago. 2017.

FOGELSTRÖM, N. D.; SVAHNBERG, T. G. M.; OLSSON, P. The impact of agile principles on market-driven software product development. *Journal of Software Maintenance and Evolution*: p. 53-80, 2010.

FRANCO, E. F. Um modelo de gerenciamento de projeto baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta. São Paulo, 2007. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-09012008-155823/pt-br.php>>. Acesso em: 20 set. 2017.

GIL, A. C. Métodos e técnicas de pesquisa social. 5. ed. São Paulo: Atlas, 1999.

IMPLEMENTING KANBAN FOR SUSTAINING ENGINEERING. Disponível em <<http://blogs.mulesoft.org/implementing-kanban-for-sustaining-engineering/>>. Acesso em: 20 out. 2017.

GHINATO, P. Elementos Fundamentais do Sistema Toyota de Produção. Disponível em: <<http://pt.scribd.com/doc/294131/Fundamentos-do-Sistema-Toyota-de-Producao>>. Acesso em: 27 jul. 2017.

HANSSON, C.; DITTRICH, Y.; GUSTAFSSON, B.; ZARNAK, S. How agile are industrial software development practices?. Journal of Systems and Software: p. 1295-1311, 2005.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2015. Disponível em: <<http://www.ibge.gov.br>>. Acesso em: 20 dez 2017.

IPARDES. Arranjo Produtivo Local de Software de Pato Branco, Dois Vizinhos e Região Sudoeste: estudo de caso. Curitiba, 2006.

JOHNSON, J.; ROI, It's Your Job. In: Third International Conference on eXtreme Programming (XP2002), 2002. Disponível em: <<http://agile.pols.co.uk/tracks/1/posts/15>>. Acesso em: 23 ago. 2017.

JONES, C. Strategies for managing requirements creep. Computer 29 (6), p. 92–94, 1996.

JUNG. C. F. Metodologia para pesquisa & desenvolvimento aplicada a novas tecnologias, produtos e processo. Rio de Janeiro: Axcel Books, xvi. 321 p., 2009.

JUNIOR, E. D. B.; ENSSLIN, L.; ENSSLIN, R. S. Proposta de processo para seleção, bibliometria e revisão sistêmica de artigos sobre a avaliação de desempenho na cadeia de suprimentos. Revista Produção Online, Florianópolis, SC, v.12, n. 4, p. 876-903, out./dez. 2012.

KETTUNEN, P. Adopting key lessons from agile manufacturing to agile software product development-A comparative study. Technovation: p. 408-422, 2009.

KITCHENHAM, B. Procedures for performing systematic reviews, Keele University, Keele, UK. 2004.

LANG, M.; FITZGERALD, B. Hypermedia systems development practices: a survey. *IEEE Softw.* 22 (2), 68. 2005.

LEAN INSTITUTE BRASIL. Desenvolvimento Lean de produtos. 2010. Disponível em: <http://www.Lean.org.br> . Acesso em: 18 set. 2017.

MARAFON, A. D. et al. Avaliação de desempenho na gestão de P&D – Revisão sistêmica literária. *Revista P&D em Engenharia de Produção*. V.10, n. 02, p. 171-194, 2012.

MARCONI, M. A.; LAKATOS E. M. *Fundamentos de Metodologia Científica*. 6. Ed. São Paulo: Atlas, 2007.

MARKUS, M. L.; KEIL, M. If we build it, they will come: designing information systems that people want to use. *Sloan Manage. Rev.* 35 (4), p. 11–25, 1994.

MCCONNELL, S. Achieving Leaner software. *IEEE Softw.* 14 (6), p. 127–128, 1997.

MIDDLETON, P.; JOYCE, D. Lean software management: BBC worldwide case study. *IEEE Transactions on Engineering Management*: p. 20-32, 2012.

MISHRA, D.; MISHRA, A. Complex software project development: Agile methods adoption. *Journal of Software Maintenance and Evolution*: p. 549-564, 2011.

MISRA, S. C.; KUMAR, V.; KUMAR, U. Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*: p. 1869-1890, 2009.

MOZZATO, A. R.; GRZYBOVSKI, D. Análise sistêmica como Técnica de Análise de Dados Qualitativos no Campo da Administração: Potencial e Desafios. *Revista de Administração Contemporânea*, Curitiba, v. 15, n. 4, p. 731-747, Jul./Ago. 2011.

MURPHY, L. Using software project “should-cost” models. *Trans. AACE Int.* 4, 1–4.3. 2001.

NERUR, S.; MAHAPATRA, R.; MANGALARAJ, G. Challenges of migrating to agile methodologies. *Communications of the ACM*: p. 72-78, 2005.

NIAZI, M.; WILSON, D.; ZOWGHI, D. "Implementing Software Process Improvement Initiatives: An Empirical Study", Springer, v. 2006, p. 222-233, LNCS. 2006.

NTI. Notícias. Disponível em: <<http://www.ntipr.org.br/noticias>>. Acesso em: 05 jun. 2018.

- PASS, S.; RONEN, B. Reducing the software value gap. *Commun. ACM* 57(5), p. 80–87, 2014.
- PENNA, E. M. D. O Paradigma Junguiano no contexto da metodologia qualitativa de pesquisa. *Psicologia USP*, 2004.
- PEREIRA, PATRIZIA K. B. Análise do aglomerado de tecnologia da informação em Pato Branco: dimensões produtiva e institucional. 2011. 101 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Desenvolvimento Econômico. Curitiba, 2011.
- PERNSTAL, J.; FELDT, R.; GORSHEK, T. The Lean gap: A review of Lean approaches to large-scale software systems development. *Journal of Systems and Software*: p. 2797 – 2821. 2013.
- PETERSEN, K.; WOHLIN, C. Software process improvement through the Lean Measurement (SPI-LEAM) method. *Journal of Systems and Software*: p. 1275-1287, 2010.
- PETERSEN, K.; WOHLIN, C. Measuring the flow in Lean software development. *Software - Practice and Experience*: p. 975-996, 2011.
- PITTMAN, M. Lessons Learned in Managing Object-Oriented Development. *IEEE Software*, 45-53, (1993)
- PRESSMAN, R. S. Engenharia de software. 6. Ed. São Paulo: McGraw-Hill, 2006 xxxi, 720p.
- POPPENDIECK, M.; CUSUMANO, M. A. “Lean software development: A tutorial”. *IEEE Software*: p. 26-32, 2012.
- POPPENDIECK, M.; POPPENDIECK, T. Implementing Lean Software Development, from concept to cash. Boston: Pearson Education. 2007.
- POPPENDIECK, M.; POPPENDIECK, T. Implementing Lean Software Development from Concept to Cash. Addison Wesley Professional. 2006.
- QUMER, A.; HENDERSON-SELLERS, B. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(3), p. 280-295, 2008.
- RONEN, B.; PASS, S. Focused Operations Management: Achieving More with Existing Resources. John Wiley & Sons, Hoboken, New Jersey. 2008.

ROPPONEN, J.; LYYTINEN, K. Components of software development risk: how to address them? A project manager survey. *IEEE Trans. Softw. Eng.* 26 (2), p. 98–112, 2000.

RUST, R. T.; THOMPSON, D. V.; HAMILTON, R. W. Defeating feature fatigue. *Harv. Bus. Rev.* 84 (2), p. 98–107, 2006.

SEBRAE. Panorama do Setor de Software e Serviços de TI do Paraná. Curitiba, PR: Sebrae, 2010; 2011, 2012.

SERRADOR, P.; PINTO, J. K. Does Agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*: p. 1040 – 1051, 2015.

SHARP, H.; ROBINSON, H.; PETRE, M. The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers*: p. 108 – 116, 2009.

SCHWABER, K. *Agile Project Management with Scrum*. Redmond/Washington: Microsoft Press, 2004.

SHINGO, S. *A study of the Toyota production system from a industrial engineering viewpoint*. Cambridge: Rudra Press, 1989.

SHMUELI, O.; PLISKIN, N.; FINK, L. Can the outside-view approach improve planning decisions in software development projects? *Inf. Syst. J.* 2015. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1111/isj.12091/full>>. Acesso em: 20 jul. 2017.

SHMUELI, O.; RONEN, B. Excessive software Development: Practices and Penalties. *International Journal of Project Management*. 35. p. 13-27, 2017.

SILVA, E.L.; MENEZES, E.M. *Metodologia da pesquisa e elaboração de dissertação*. Florianópolis: [S. n.], 2000.

STANDISH GROUP. New Standish Group report shows more Project failing and less successful projects. Disponível em: http://www.standishgroup.com/newsroom/chaos_2009.php, 2009. Acesso em: Out, 2017.

STEFFEN, J. B. *Lean para Desenvolvimento de Software. Afinal o que é isto?*. 2011. Disponível em: <www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/Lean_para_des

envolvimento_de_sw_o_que__c3_a9_isso_afinal12?lang=en>. Acesso em: 17 jul. 2017.

SUČIĆ, M. Hotel product quality through the innovation system (case study of ritz carlton). *Tourism & Hospitality Management*, nº0, p. 639-645, 2010. SUOMALAINEN, T.; KUUSELA, R.; TIHINEN, M. Continuous planning: An important aspect of agile and Lean development. *International Journal of Agile Systems and Management*: p. 132-162, 2015.

TAKEUCHI, T. et al. Pathological changes observed in the finger biopsy of patients with vibration-induced white finger. *Scandinavian Journal of Work Environment & Health*, v. 12, p. 280-283, 1986.

TASCA, J. E.; ENSSLIN, L.; ENSSLIN, S. R.; ALVES, M. B. M. An approach for selecting a theoretical framework for the evaluation of training programs. *Journal of European Industrial Training*, 2010.

TORRECILLA-SALINAS, C. J.; SEDEÑO, J.; ESCALONA, M. J.; MEJÍAS, M. Estimating, planning and managing Agile Web development projects under a value-based perspective. *Information and Software Technology*: p. 124-144, 2015.

VALLON, R.; MÜLLER-WERNHART, M.; SCHRAMM, W.; GRECHENIG, T. Combination of agile and Lean in software development: Concept and realization. *Informatik-Spektrum*: p. 28-35, 2014.

VAN WAARDENBURG, G.; VAN VLIET, H. When agile meets the enterprise. *Information and Software Technology*: p. 2154-2171, 2013.

WANG, X.; CONBOY, K.; CAWLEY, O. Leagile software development: An experience report analysis of the application of Lean approaches in agile software development. *Journal of Systems and Software*: p. 1287-1299, 2012.

YIN, R. K. *Estudo de Caso – Planejamento e Métodos*. 3. Ed. São Paulo: Bookman, 2005.

WHEATCRAFT, L.S. Triple your chances of project success risk and requirements. *INCOSE International Symposium*, Denver, CO, June 23. 2011.

WOMACK, J. P.; JONES, D. T. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York, NY: Free Press, 2003.

WOMACK, Jame P.; JONES, Daniel T.; ROSS, Daniel. The machine that changed the world. New York: Rawson Associates, 1990.

ZHI, J.; GAROUSHI-YUSIFOGLU, V.; SUN, B.; GAROUSHI, G.; SHAHNEWAZ, S.; RUHE, G. Cost, benefits and quality of software development documentation: A systematic mapping. Journal of Systems and Software: p. 175-198, 2015.