

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA ARQUITETURA E URBANISMO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**MÉTODO DE APLICAÇÃO DO *LEAN THINKING* PARA
DESENVOLVIMENTO DE PROJETOS DE *SOFTWARE* COM
METODOLOGIA SCRUM**

CAMILO CÉSAR PERUCCI

ORIENTADOR: PROF. DR. FERNANDO CELSO DE CAMPOS

SANTA BÁRBARA D'OESTE

2016

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA ARQUITETURA E URBANISMO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**MÉTODO DE APLICAÇÃO DO *LEAN THINKING* PARA
DESENVOLVIMENTO DE PROJETOS DE *SOFTWARE* COM
METODOLOGIA SCRUM**

CAMILO CÉSAR PERUCCI

ORIENTADOR: PROF. DR. FERNANDO CELSO DE CAMPOS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção, da Faculdade de Engenharia Arquitetura e Urbanismo, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Engenharia de Produção.

SANTA BÁRBARA D'OESTE

2016

Ficha Catalográfica elaborada pelo Sistema de Bibliotecas da UNIMEP
Bibliotecária: Juliete Susann Ferreira de Souza CRB-8/9843

P471m	<p>Perucci, Camilo César</p> <p>Método de aplicação do lean thinking para desenvolvimento de projetos de software com metodologia scrum. – 2016. 126 f. : il. ; 30 cm.</p> <p>Orientador: Prof. Dr. Fernando Celso de Campos. Dissertação (mestrado) – Universidade Metodista de Piracicaba, Engenharia de Produção, Santa Bárbara D'oeste, 2016.</p> <p>1. Scrum (Desenvolvimento de software). I. Campos, Fernando Celso de. II. Título.</p> <p style="text-align: right;">CDU – 681.3.06</p>
-------	---

MÉTODO DE APLICAÇÃO DO *LEAN THINKING* PARA DESENVOLVIMENTO DE PROJETOS DE *SOFTWARE* COM METODOLOGIA SCRUM

CAMILO CÉSAR PERUCCI

Dissertação de Mestrado defendida e aprovada, em 06 de dezembro de 2016,
pela Banca Examinadora constituída pelos Professores:

Prof. Dr. Fernando Celso de Campos (Orientador e Presidente)
(PPGEP – FEAU/UNIMEP)

Prof. Dr. Mauro Luiz Martens
(PPGEP - FEAU/UNIMEP)

Prof. Dr. Moacir Godinho Filho
(DEP/UFSCar)

Dedico esse trabalho à minha esposa Delma e
minha filha Lorena, pela compreensão nos
momentos de ausência.

AGRADECIMENTOS

O maior agradecimento a Deus, por me conceder saúde e perseverança para realizar mais esta jornada.

Ao Professor Dr. Fernando Celso de Campos, pela oportunidade, orientação, confiança e amizade.

Aos professores Dr. Mauro Luiz Martens e Dr Moacir Godinho Filho, por aceitarem participar da banca de qualificação e defesa, contribuindo em pontos fundamentais para o amadurecimento da pesquisa.

Ao corpo docente, colaboradores, colegas e comissão de bolsa do Programa de Pós-Graduação que fizeram parte deste trabalho, em especial à Marta Helena T. Bragaglia, por toda a sua prontidão, carinho e amizade.

À Instituição de Ensino, UNIMEP por toda disponibilidade de estrutura e infraestrutura e pelo apoio.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), pelo apoio fornecido em forma de bolsa de fomento para o desenvolvimento deste trabalho.

PERUCCI, Camilo César. **Método de Aplicação do *Lean Thinking* para Desenvolvimento de Projetos de Software com Metodologia Scrum**. 2016. 126 f. Dissertação (Mestrado em Engenharia de Produção) – Faculdade de Engenharia Arquitetura e Urbanismo, Universidade Metodista de Piracicaba, Santa Bárbara d'Oeste.

RESUMO

Na sociedade moderna *softwares* são essenciais para a vida das pessoas e das empresas, ajudando em sua rotina cotidiana, promovendo entretenimento e comunicação, controlando o funcionamento de equipamentos, apoiando a tomada de decisão, entre outros. Assim como qualquer outro produto, *software* também possui um processo de desenvolvimento (produção) que é influenciado por diferentes recursos como: pessoas, equipamentos, informações e energia. Esses processos estão em constante evolução para se adaptar às novas tecnologias, demandas do mercado e necessidades dos clientes. Nesse contexto, esse projeto de pesquisa apresenta uma proposta para contribuir com essa evolução, avaliando a integração das abordagens *Lean Thinking* (Pensamento Enxuto) e Scrum para produção de *softwares* com melhor qualidade e produtividade. A aplicação do *Lean Thinking* na área de desenvolvimento de *software* é conhecida como *Lean Software Development*. Portanto, o objetivo é avaliar a aplicação dos princípios e práticas *Lean Software Development* em equipes que utilizam metodologia Scrum para desenvolvimento de *software* propondo um método para integração das duas abordagens. O método de pesquisa adotada foi a pesquisa-ação, pois o foco do pesquisador foi acompanhar a aplicação do método, identificar oportunidades de melhoria e adaptá-lo com a participação colaborativa da equipe de projeto. O método foi aplicado em 2 (duas) empresas que desenvolvem *software*. Os resultados evidenciam que a integração das abordagens Scrum e *Lean Thinking* é possível e que são complementares. Além disso, apresentou resultados positivos como integração e motivação da equipe, qualidade dos produtos, aumento da produtividade, redução da fila de pendências, maior velocidade nas respostas a problemas, alinhamento dos produtos com as necessidades dos clientes e redução do tempo de ciclo de produção e entrega.

PALAVRAS-CHAVE: Scrum, *Lean Thinking*, *Lean Software*, Metodologias Ágeis.

PERUCCI, Camilo César. **Lean Thinking Application Method for Software Project Development with Scrum Methodology**. 2016. 126 f. Dissertation (Master in Production Engineering) – Faculdade de Engenharia Arquitetura e Urbanismo, Universidade Metodista de Piracicaba, Santa Bárbara d'Oeste.

ABSTRACT

In modern society software is essential to the lives of people and businesses, helping in their daily routine, promoting entertainment and communication, controlling the operation equipment, supporting decision-making, among others. Similarly any other product, software also has a development process (production) which is influenced by different resources such as people, equipment, information and energy. These processes are constantly evolving to adapt to new technologies, market demands and customer needs. In this context, this research project presents a proposal to contribute to this development, assessing the integration of approaches Lean Thinking and Scrum for software production with better quality and productivity. The application of Lean Thinking in software development area is known as Lean Software Development. Therefore, the aim of this work is to evaluate the application of the principles and practices of Lean Software Development in teams that use Scrum methodology for software development by proposing a method for integrating the two approaches. The adopted research was the action-research methodology, because the focus of the researcher was to monitor the implementation of the method, identifying opportunities for improvement and adapt it with the collaborative team participation. The method was applied in two software development companies. The results show that the integration of Scrum and Lean Thinking approach it is possible and that are complementary. Furthermore, it presented positive results as team integration and motivation, product quality, increased productivity, reduction of the pending queue, greater speed in responding to problems, alignment the products with the interests of customers and reducing the lead time.

PALAVRAS-CHAVE: Scrum, Lean Thinking, Lean Software, Agile Methodologies.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	I
LISTA DE TABELAS E QUADROS.....	II
LISTA DE ABREVIATURAS E SIGLAS	III
1 INTRODUÇÃO.....	1
1.1 JUSTIFICATIVA E RELEVÂNCIA	3
1.2 QUESTÃO DE PESQUISA.....	4
1.3 OBJETIVOS DE PESQUISA.....	5
1.4 VISÃO GERAL DA PESQUISA.....	5
1.5 ESTRUTURA DO TRABALHO	8
2 FUNDAMENTAÇÃO TEÓRICA.....	10
2.1 <i>LEAN THINKING</i> (PENSAMENTO ENXUTO)	10
2.1.1 TOYOTA <i>PRODUCTION SYSTEM</i> (TPS)	11
2.1.2 PRINCÍPIOS DO <i>LEAN THINKING</i>	14
2.1.3 APLICAÇÃO <i>LEAN THINKING</i> EM DIVERSAS ÁREAS	16
2.1.4 APLICAÇÃO <i>LEAN THINKING</i> PARA DESENVOLVIMENTO DE <i>SOFTWARE</i>	17
2.2 <i>LEAN SOFTWARE DEVELOPMENT</i> (LSD)	18
2.2.1 PRINCÍPIOS DO <i>LEAN SOFTWARE DEVELOPMENT</i> (LSD)	20
2.2.2 ROTEIRO DE IMPLANTAÇÃO <i>LEAN SOFTWARE DEVELOPMENT</i> (LSD)	24
2.3 MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE <i>SOFTWARE</i>	26
2.3.1 APLICAÇÃO DE MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE <i>SOFTWARE</i>	28
2.4 METODOLOGIA SCRUM	29
2.4.1 ESTRUTURA DO SCRUM	30
2.4.2 ROTEIRO DE IMPLANTAÇÃO SCRUM	37
2.4.3 APLICAÇÃO DA METODOLOGIA SCRUM	40
2.5 METODOLOGIAS ÁGEIS, <i>LEAN SOFTWARE DEVELOPMENT</i> (LSD) E SCRUM.....	43
2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO	44
3 PROCEDIMENTOS METODOLÓGICOS.....	47
3.1 ASPECTOS METODOLÓGICOS.....	47
3.2 PESQUISA BIBLIOGRÁFICA	49
3.3 PESQUISA-AÇÃO	50
4 MÉTODO DE APLICAÇÃO <i>LEAN THINKING</i> EM PROJETOS SCRUM (MALS).....	57
4.1 MODELO TEÓRICO CONCEITUAL DO MÉTODO DE APLICAÇÃO <i>LEAN THINKING</i> EM PROJETOS SCRUM (MALS).....	57
4.2 MÉTODO DE APLICAÇÃO <i>LEAN THINKING</i> EM PROJETOS SCRUM (MALS) CONSOLIDADO	59
4.3 AVALIAÇÃO DA PROPOSTA DO MALS COM A PESQUISA-AÇÃO	64
5 APLICAÇÃO DO MALS E RESULTADOS	67
5.1 SELECIONAR AS UNIDADES DE ANÁLISE	67
5.2 DEFINIÇÃO DOS CICLOS DE PESQUISA E COLETA DE DADOS	68
5.3 DISCUSSÃO DOS RESULTADOS	68
5.3.1 EMPRESA A: ESCOPO E RESULTADOS	69
5.3.2 EMPRESA B: ESCOPO E RESULTADOS	76
5.4 RESULTADOS IDENTIFICADOS COM A IMPLANTAÇÃO DO MALS	83
6 CONCLUSÃO	86

6.1 PROPOSTA PARA TRABALHOS FUTUROS	89
REFERÊNCIAS BIBLIOGRÁFICAS	90
APÊNDICES	97
APÊNDICE A – QUADRO RESUMO DA PESQUISA-AÇÃO EMPRESA A	97
APÊNDICE B – QUADRO RESUMO DA PESQUISA-AÇÃO EMPRESA B	108

LISTA DE ILUSTRAÇÕES

FIGURA 1 – VISÃO GERAL DA PESQUISA	7
FIGURA 2 – A ENGENHARIA DO SCRUM.....	35
FIGURA 3 – EXEMPLO DE GRÁFICO <i>BURNDOWN</i>	36
FIGURA 4 – EXEMPLO DE QUADRO SCRUM	39
FIGURA 5 – LINHA DO TEMPO: REVISÃO DA LITERATURA	45
FIGURA 6 – ESTRUTURA DA PESQUISA-AÇÃO.....	52
FIGURA 7 – MÉTODO DE APLICAÇÃO <i>LEAN THINKING</i> EM PROJETOS SCRUM (MALS) – MODELO TEÓRICO-CONCEITUAL.	58
FIGURA 8 – MÉTODO DE APLICAÇÃO <i>LEAN THINKING</i> EM PROJETOS SCRUM (MALS) – MODELO CONSOLIDADO	60
FIGURA 9 – EVOLUÇÃO DA PRODUÇÃO POR CICLO SCRUM: EMPRESA A	76
FIGURA 10 – EVOLUÇÃO DA PRODUÇÃO POR CICLO SCRUM: EMPRESA B	81
FIGURA 11 – EVOLUÇÃO DA FILA DE PENDÊNCIAS POR CICLO SCRUM: EMPRESA B	82

LISTA DE TABELAS E QUADROS

QUADRO 1 – DESCRIÇÃO GERAL DA ABORDAGEM METODOLÓGICA	8
QUADRO 2 – CARACTERÍSTICAS DO TPS.....	13
QUADRO 3 – PRINCÍPIOS DO <i>LEAN THINKING</i>	15
QUADRO 4 – PRINCÍPIOS DO <i>LEAN SOFTWARE DEVELOPMENT (LSD)</i>	21
QUADRO 5 – OS SETE DESPERDÍCIOS.....	22
QUADRO 6 – ROTEIRO IMPLANTAÇÃO <i>LEAN SOFTWARE DEVELOPMENT</i>	24
QUADRO 7 – ROTEIRO IMPLANTAÇÃO PROJETO SCRUM.....	37
QUADRO 8 – LINHA DO TEMPO: REVISÃO DA LITERATURA	46
QUADRO 9 – PLANEJAMENTO DA PESQUISA-AÇÃO.....	54
QUADRO 10 – QUADRO-RESUMO PARA PESQUISA-AÇÃO	56
QUADRO 11 – DETALHAMENTO DO MALS.....	61
QUADRO 12 – QUESTIONÁRIO DE ADEQUAÇÃO SCRUM.....	62
QUADRO 13 – EVOLUÇÃO DO MALS.	65
QUADRO 14 – PRÁTICAS SUGERIDAS PELA EMPRESA A PARA MELHORIA PROCESSO SCRUM.....	71
QUADRO 15 – AÇÕES DE MELHORIA APLICADAS PELA EMPRESA A BASEADAS NA ABORDAGEM LSD	73
QUADRO 16 – PRÁTICAS SUGERIDAS PELA EMPRESA B PARA MELHORIA PROCESSO SCRUM.....	78
QUADRO 17 – AÇÕES DE MELHORIA APLICADAS PELA EMPRESA B BASEADAS NA ABORDAGEM LSD	80
QUADRO 18 – EQUIVALÊNCIA SCRUM E PRÁTICAS LSD	83
QUADRO 19 – EQUIVALÊNCIA SCRUM E PRÁTICAS LSD	84

LISTA DE ABREVIATURAS E SIGLAS

ASD – *Adaptative Software Development*

DSDM – *Dynamic System Development Method*

ED – Departamentos de tratamento de Saúde de Emergência

FDD – *Feature Driven Development*

GM – General Motors

GSD – Desenvolvimento de *Software* Global

IMVP – *Internacional Motor Vehicle Program*

ITIL – *Information Technology Infrastructure Library*

JIT – *Just-in-Time*

LSD – *Lean Software Development*

MALS – Método de Aplicação de *Lean Thinking* em Projetos Scrum

MIT – Instituto de Tecnologia de Massachusetts

PO – *Product Owner*

SM – *Scrum Master*

TDD – Desenvolvimento Guiado por Testes

TFS – *Team Foundation Server*

TI – Tecnologia da Informação

TPS – *Toyota Production System*

XP – *Extreme Programming*

1 INTRODUÇÃO

O mundo moderno não poderia existir sem o *software* (SOMMERVILLE, 2011). De fato, o *software* controla a produção e distribuição industrial, sistemas financeiros, sistemas de controle de tráfego aéreo, sistemas de saúde, a maioria dos equipamentos elétricos e eletrônicos inclui um *software* que os controla, portanto, televisores, automóveis, aviões. O *software* possibilitou a conquista do espaço e a criação da internet, o mais importante sistema de comunicação da história da humanidade. Os serviços e a infraestrutura do planeta, como energia, comunicações e transporte dependem de sistemas computacionais complexos e confiáveis (SOMMERVILLE, 2011).

Atualmente enfrentam-se novos desafios, como mudanças climáticas, a competitividade das empresas, terrorismo internacional, crescimento da população mundial. Para resolver esses problemas é preciso desenvolver novas tecnologias e o *software* tem um papel fundamental nesse sentido (PRESSMAN, 2006).

Portanto, produtos de *software* confiáveis e que sejam produzidos de forma econômica, rápida e que se adaptem às necessidades dos clientes é uma exigência cada vez maior de nossa sociedade no formato como está organizada (PRESSMAN, 2006; SOMMERVILLE, 2011).

Para isso a evolução de metodologias e aplicação de práticas de produção de *software* tem se mostrado constante, dentre elas surge a proposta da aplicação do Pensamento Enxuto ou *Lean Thinking*, utilizado originalmente no setor automobilístico, para a área de tecnologia da informação (TI) (BELL; ORZEN, 2011).

O *Lean Thinking* é uma metodologia aplicada ao processo de produção que busca fazer cada vez mais utilizando cada vez menos recursos, ou seja, menos esforço humano, menos equipamento, menos tempo, menos movimentação e menos espaço, eliminando, assim, desperdícios, diminuindo a geração de

resíduos e aumentando os benefícios para a corporação por meio de atividades que apenas agreguem valor ao produto final, com foco nos clientes, para ofertar exatamente o que eles desejam (WOMACK; JONES, 2004).

O *Lean Thinking* tem sido utilizado como uma forma da indústria de TI se diferenciar e buscar solução para alguns dos seus problemas. Uma das áreas de aplicação do *Lean Thinking* é a área de desenvolvimento de *software*, conhecida como *Lean Software Development* (LSD) (POPPENDIECK; CUSUMANO, 2012).

O LSD se popularizou com o esforço de associação com o termo “ágil” para o desenvolvimento de produtos de *software* (POPPENDIECK; CUSUMANO, 2012). Trabalhos científicos evidenciam o sucesso da integração das duas abordagens, em que o *Lean* pode ser utilizado para melhoria do processo ágil.

Poppendieck e Cusumano (2012) analisam que os preceitos do Manifesto Ágil, documento criado por um grupo de profissionais que encoraja o uso de melhores métodos para desenvolver sistemas de *software*, são compatíveis com os princípios *Lean*. Além disso, recomendam que as organizações utilizem o *Lean* integrado com práticas de outras metodologias como Scrum.

Scrum é utilizado para gerenciar o desenvolvimento de um projeto utilizando práticas iterativas e incrementais (FONSECA; CAMPOS, 2008). De acordo com Agile Alliance (2016) a metodologia Scrum é uma das metodologias ágeis mais utilizadas para gerenciamento de projetos de *software*, porém não é uma metodologia completa e carece de práticas técnicas (POPPENDIECK; CUSUMANO, 2012).

O LSD propõe essas técnicas aplicadas a desenvolvimento de *software* com foco na criação de valor para o cliente, eliminando desperdícios, otimizando os fluxos de valor, capacitando as pessoas e, continuamente, melhorando (EBERT; ABRAHAMSSON; OZA, 2012).

Estudos apresentaram poucas iniciativas com objetivo de integração entre LSD e Scrum. Alguns casos de sucesso, como os estudos de Wang, Conboy e

Cawley (2012) e Starr (2013) evidenciam isso, porém os mesmos autores alertam que estudos mais aprofundados são necessários.

Diante desse contexto, novos esforços de pesquisa são necessários para consolidar a integração entre as abordagens LSD e Scrum.

1.1 JUSTIFICATIVA E RELEVÂNCIA

Diante da evidente importância do *software* e a necessidade de evoluir metodologias e práticas de produção de *software* para se adaptar às necessidades da sociedade moderna, bem como, da tendência de se buscar a integração entre *Lean Software Development* (LSD) e Scrum, justifica-se essa tentativa de pesquisa como uma alternativa para melhoria de processo de produção de *software* visando-se obter maior produtividade e qualidade.

Scrum é uma metodologia ágil para gerenciar o desenvolvimento de um projeto com práticas iterativas e incrementais e é uma excelente maneira de introduzir o conceito *Lean* de “fluxo” para desenvolvimento de *software* (FONSECA; CAMPOS, 2008; POPPENDIECK; CUSUMANO, 2012). Além disso, é boa prática para organizar o desenvolvimento do *software*, mas insuficiente para a construção do projeto da solução do produto, pois carece de práticas técnicas (POPPENDIECK; CUSUMANO, 2012). O LSD apresenta essas práticas e estabelece um conjunto de princípios que envolvem todo o ciclo de vida do produto, com uma abordagem interfuncional, dessa forma é mais indicado para orientar a combinação de projeto, desenvolvimento, implantação e validação em ciclo contínuo de descoberta e fornecimento de valor (POPPENDIECK; CUSUMANO, 2012).

Sendo assim, o Scrum foi escolhido devido a seu estágio consolidado junto a equipes de desenvolvimento evidenciado por estudos como de Mann e Maurer (2005) e Gannon (2013), que apresentaram bons resultados em projetos de desenvolvimento de *software*.

Também, é preciso destacar a pesquisa da Forrester, publicada em 2011, sobre a aplicação da abordagem ágil nas organizações. Essa pesquisa realizou um *survey* com 500 participantes em 70 países. Identificou que o Scrum lidera o segmento de abordagens ágeis, pois 40% das empresas entrevistadas relataram que utilizam em suas atividades principais, seguido por *Kanban* com 15% e *Lean* com 11% (SCRUM ALLIANCE, 2013).

De acordo com Wang, Conboy e Cawley (2012), a abordagem *Lean* para desenvolvimento de *software* é uma área de estudo emergente e é considerada “a próxima onda de processo de *software*”. O estudo de Starr (2013), comprova que usar o *Lean Thinking* com Scrum produz retornos elevados e equipes mais eficazes. Porém, é consenso dos autores Poppendieck e Cusumano (2012), Wang, Conboy e Cawley (2012), Ebert *et al.* (2012) e Starr (2013) que os estudos de como os “*lean methods*” impactam o desenvolvimento de *software* ainda estão em fases iniciais. Starr (2013), afirmou: “equipes Scrum ainda estão aprendendo como aplicar *Lean*”.

Portanto, observada a popularidade da utilização da metodologia Scrum, e a necessidade de estudos mais aprofundados sobre aplicação de *Lean Thinking* em desenvolvimento de *software* (LSD), esse trabalho propõe avaliar a integração das duas abordagens, como algo relevante e que se justifica academicamente pelas possibilidades de se trazer avanços nessa área de apoio da TI.

1.2 QUESTÃO DE PESQUISA

Portanto, a partir do contexto e da lacuna identificada, a questão de pesquisa é:

“Como integrar a aplicação de princípios do *Lean Thinking* em equipes que utilizam Scrum, visando melhoria dos processos de produção de *software*?”

1.3 OBJETIVOS DE PESQUISA

Portanto o objetivo geral é:

Propor um método para integrar a aplicação dos princípios *Lean Thinking* em equipes que utilizam metodologia Scrum, visando melhorias do processo de produção de *software*.

Os objetivos específicos são:

- Identificar os princípios, conceitos e aplicação da metodologia Scrum;
- Coletar informações e experiências da aplicação da metodologia Scrum em equipes de desenvolvimento de *software*;
- Implantar a metodologia Scrum e avaliar os resultados obtidos;
- Identificar os princípios e práticas da abordagem *Lean Software Development* (LSD);
- Coletar informações e experiências da aplicação LSD em equipes de desenvolvimento de *software*;
- Implantar o LSD em projetos de desenvolvimento de *software* que utilizam Scrum e avaliar os resultados obtidos.

1.4 VISÃO GERAL DA PESQUISA

O desafio que se apresenta é avaliar se a integração das práticas de gerenciamento de projetos de *software*, definidas na metodologia Scrum e os princípios *Lean Software Development* (LSD), pode ser uma alternativa para uma melhoria de processo de produção de *software* para se obter maior produtividade e qualidade.

Portanto, esse projeto de pesquisa apresenta a avaliação por um método de aplicação dos princípios LSD em 2 (dois) projetos de desenvolvimento de *software* que utilizam Scrum. O método proposto, referido anteriormente, sistematiza a implantação e a integração das duas abordagens.

A partir das definições do problema da pesquisa e dos objetivos, o método de pesquisa utilizado nesse estudo, foi a pesquisa-ação, que possibilitou alcançar os resultados conforme o *outline* descrito na Figura 1.

Para elaboração desse trabalho, realizou-se uma Revisão Sistemática de Literatura (RSL). Os detalhes da RSL são apresentados na seção 3.2 Pesquisa Bibliográfica.

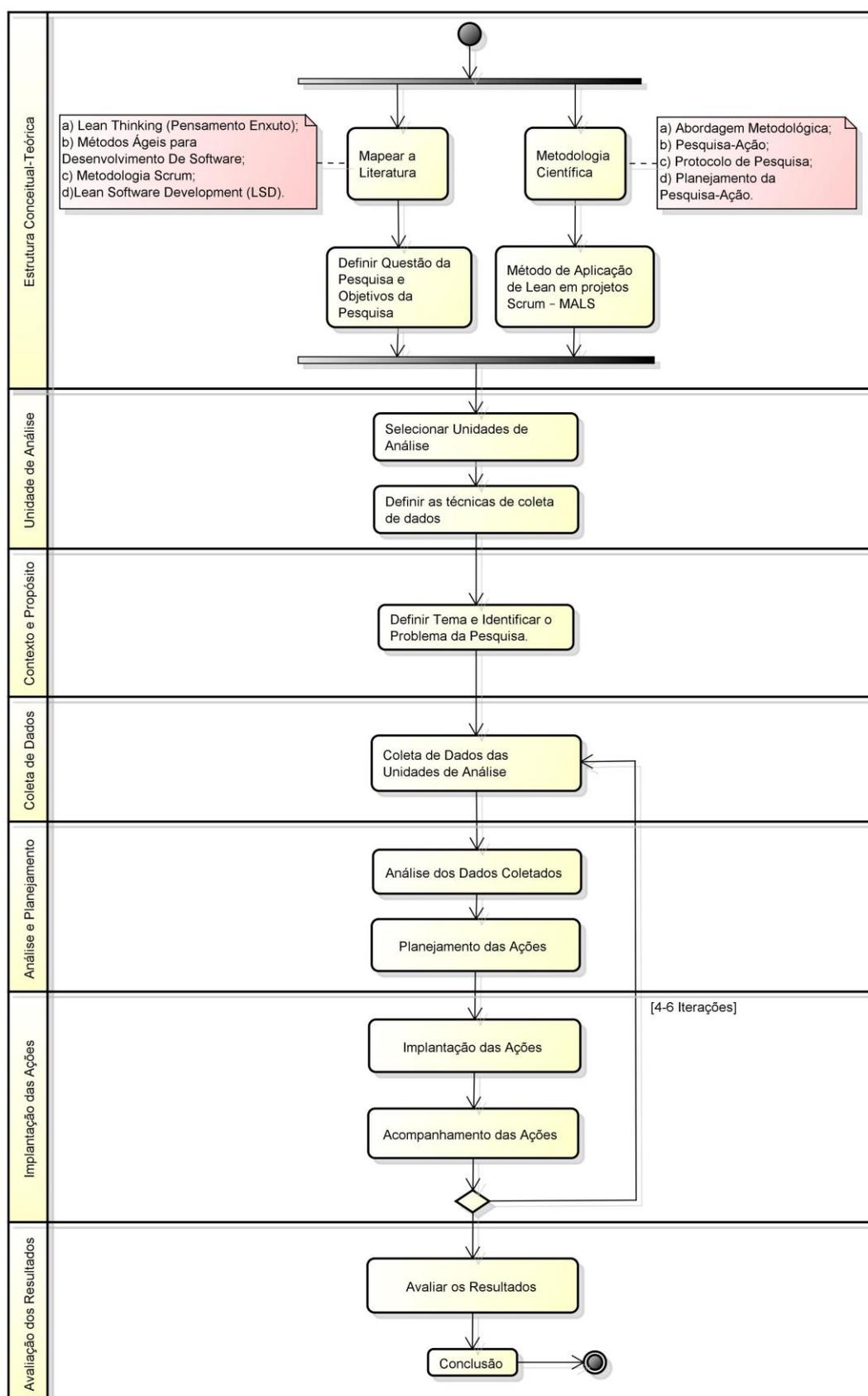


FIGURA 1 – VISÃO GERAL DA PESQUISA. FONTE: ELABORADO PELO AUTOR.

Os detalhes do projeto metodológico da pesquisa serão abordados no capítulo 4 Procedimentos Metodológicos, no entanto a descrição geral está no Quadro 1.

Definições	Fonte	Descrição
Abordagem Geral	MARTINS (2010)	Qualitativa
Quanto aos Objetivos	GIL (2009)	Exploratória
Quanto ao Método de Pesquisa	NAKANO (2010); MIGUEL (2007); DEMO (1995)	Teórico-empírico
Procedimentos Técnicos	GIL (2009); THIOLLENT (2000)	Pesquisa bibliográfica e Pesquisa-Ação
Técnica de Coleta de Dados	GIL (2011)	Levantamento bibliográfico, entrevista coletiva e observação.
Técnica de Análise dos Dados	MELLO <i>et al.</i> (2012) e GIL (2011).	Análise comparativa; Tabulação e Relatórios.

QUADRO 1 – DESCRIÇÃO GERAL DA ABORDAGEM METODOLÓGICA.
FONTE: ELABORADO PELO AUTOR.

1.5 ESTRUTURA DO TRABALHO

Este trabalho de dissertação de mestrado apresenta no Capítulo 1 a introdução do desenvolvimento do trabalho, detalhando as justificativas e relevâncias, a questão de pesquisa, contextualizando os objetivos do trabalho e a visão geral da pesquisa.

No Capítulo 2 é apresentada a fundamentação teórica, abordando os temas relacionados com *Lean Thinking* (Pensamento Enxuto) buscando apresentar o histórico e elementos para sua aplicação, *Lean Software Development* e o modelo de aplicação proposto por Poppendieck e Poppendieck (2011), Métodos Ágeis para Desenvolvimento de *Software* e sua relação com o Pensamento Enxuto e por fim a metodologia Scrum, o modelo de aplicação de Sutherland (2014) e trabalhos que relacionam as abordagens *Lean* e Scrum.

No Capítulo 3 é apresentada a abordagem metodológica geral, objetivos, método de pesquisa, procedimentos técnicos gerais, técnica de coleta, análise dos dados e o planejamento da pesquisa.

No Capítulo 4 é apresentada a proposta do modelo teórico conceitual do Método de Aplicação *Lean Thinking* em Projetos Scrum (MALS) e o modelo consolidado do MALS, aplicando o método de pesquisa.

No Capítulo 5 é apresentada a discussão dos resultados do monitoramento e da aplicação do MALS e no Capítulo 6 a conclusão da pesquisa realizada, bem como as propostas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão revisados os temas relacionados com *Lean Thinking* (Pensamento Enxuto) buscando apresentar o histórico e elementos para sua aplicação, *Lean Software Development* e o modelo de aplicação proposto por Poppendieck e Poppendieck (2011), Métodos Ágeis para Desenvolvimento de *Software* e sua relação com o Pensamento Enxuto e por fim a metodologia Scrum e o modelo de aplicação de Sutherland (2014) e trabalhos que relacionam as abordagens *Lean* e Scrum.

2.1 LEAN THINKING (PENSAMENTO ENXUTO)

A definição “*Lean Thinking*” (Pensamento Enxuto), começou a ser utilizada a partir da publicação do livro “A máquina que Mudou o Mundo” de Womack *et al.* (1990). Esta obra apresenta um trabalho acadêmico realizado por Jim Womack, Dan Jones e Dan Roos, gerentes seniores do *Internacional Motor Vehicle Program* (IMVP), relatando as pesquisas realizadas pelo Instituto de Tecnologia de Massachusetts (MIT). A publicação faz uma comparação entre o tipo de indústria predominante na época, conhecida por produção em massa, com essa nova visão que foi denominada como produção enxuta, termo traduzido da expressão inglesa *lean manufacturing* ou *lean production* (WOMACK *et al.*, 2004; HOLWEG, 2006; CANTANHEDE, 2014).

O “*Lean Thinking*” tem origem no ambiente de produção das indústrias japonesas, em especial relacionado às inovações da *Toyota Motor Corporation*. Essas inovações, resultantes da escassez de recursos e intensa concorrência interna no mercado japonês para automóveis, incluíam o sistema *just-in-time* (JIT), o método *Kanban* de produção puxada e o respeito e grande envolvimento dos funcionários na solução de problemas. A Toyota denominava seu conjunto de técnicas, ferramentas e filosofias de Sistema Toyota de Produção (*Toyota Production System* - TPS) (HINES, 2004; HOLWEG, 2006).

De uma forma geral, a produção enxuta consiste na aplicação de práticas que visam identificar e eliminar perdas ou desperdícios no sistema produtivo, além de buscar incessantemente o bom atendimento ao cliente por meio de melhor qualidade nos produtos e serviços, custos mais baixos e maior flexibilidade (WOMACK *et al.*, 2004; CARDOSO, 2010).

Com o objeto de identificar como as mudanças da indústria automobilística poderiam afetar outras atividades econômicas, Womack e Jones (2004) publicaram, em 1996, o livro “A mentalidade enxuta nas empresas - *Lean Thinking: Elimine desperdício e crie riqueza*”, explicando como ir além dos jogos financeiros, para criar valor real e duradouro em qualquer empresa (WOMACK *et al.*, 2004; WOMACK; JONES, 2004).

Womack e Jones (2004) definiram o *Lean Thinking* como uma forma de trabalho para fazer mais com cada vez menos: menos esforço humano, menos equipamento, menos tempo e menos espaço, ao mesmo tempo, aproximar cada vez mais dos clientes oferecendo exatamente o que eles desejam. Além disso, o *Lean Thinking* é uma forma nova de trabalhar, mais satisfatória, com *feedback* imediato e esforços contínuos para eliminar desperdícios e criar processos que agreguem valor.

Nas próximas seções são apresentados o histórico do TPS e *Lean Production*, o detalhamento dos princípios do *Lean Thinking* e a aplicação do *Lean Thinking* em diversas áreas. O objetivo não é o aprofundamento de cada conceito, mas fundamentar o *Lean Thinking* e buscar elementos para sua aplicação.

2.1.1 TOYOTA *PRODUCTION SYSTEM* (TPS)

A fundação da Toyota Motor Company remonta a 1918, quando o empresário Sakichi Toyoda vendeu as patentes de seu avançado tear automático. Esses fundos foram a base para seu filho Kiichiro realizasse a visão de fabricação de automóveis (HOLWEG, 2006).

De acordo com Holweg (2006), na época o mercado japonês era dominado pelas subsidiárias da Ford e General Motors (GM), porém mesmo com dificuldades financeiras e a morte de Sakichi em 1930, a Toyota's *Automotive* se estabeleceu e começou a produzir seu primeiro modelo, utilizando componentes da Ford e da GM. Para facilitar a pronúncia, a empresa alterou seu nome para Toyota. A produção de caminhões e carros iniciou em 1935 e em 1937 a empresa Toyota Motor Company foi formalmente constituída. Porém, com a Segunda Guerra Mundial a produção foi interrompida e a dificuldade financeira do pós-guerra resultou no aumento de estoque de veículos não vendidos e problemas trabalhistas que dividiram a empresa, culminando com a renúncia de Sakichi. Seu primo, Eiji Toyoda, assumiu o negócio e foi para os Estados Unidos estudar o sistema de produção em massa, contudo por restrições financeiras e os baixos volumes do mercado japonês, não conseguiu implantar essa forma de produção.

De acordo com Ohno (1997), o Sistema Toyota de Produção evoluiu da necessidade, devido a restrições no mercado que exigiram a produção de pequenas quantidades e muitas variedades com baixas demandas.

O grande impulso para o TPS ocorreu com Taiichi Ohno, que observou as seguintes falhas no sistema de produção em massa: a produção em lotes grandes resultava em grande quantidade de estoque, necessitando de capital e espaço; e a incapacidade de acomodar as preferências dos clientes a variedade de produtos. Gradualmente Ohno implantou o conceito de produção em lotes pequenos. O principal foco é reduzir custos eliminando o desperdício. Logo após implantou o sistema *kanban* para controlar a reposição de material e conceito *just-in-time* (JIT), que consistia em ter as peças para montagem no lado da linha apenas no momento de sua utilização. O resultado foi a capacidade de produzir uma grande variedade de veículos, com baixos volumes e a custos competitivos, contrapondo a lógica da produção em massa (HOLWEG, 2006; WOMACK; JONES, 2004).

Holweg (2006) conclui que o novo conceito de produção implementado pela Toyota foi resultado de um ciclo contínuo de aprendizagem que durou décadas.

Considera que o sucesso da TPS é fundamentado nessa capacidade de aprendizagem dinâmica.

De acordo com Womack *et al.*(2004), a Toyota havia delineado os princípios da produção enxuta (*lean production*) no início da década de 1960 e nessa década conquistado enorme vantagem sobre os produtores em massa, conseguindo aumentar continuamente sua participação na produção mundial de veículos.

De acordo com Shingo (1996) as principais características do TPS estão apresentadas no Quadro 2.

Princípio TPS	Descrição
Minimizar custos	O princípio da minimização de custos é fundamental, a sobrevivência da empresa depende da redução de custos, por meio da eliminação completa de desperdícios.
Estoque Zero	<i>Just-in-Time</i> (no tempo certo). A melhor resposta à demanda é a produção contrapedido, abandonando produção convencional em grandes lotes. As exigências da produção contrapedido (alta diversidade, produção em baixas quantidades, entrega rápida e manejo da flutuação da carga), somente podem ser satisfeitas a partir da contínua eliminação dos desperdícios por superprodução.
Operações de Fluxo	As demandas da produção contrapedido geraram soluções para diversos problemas, a primeira dessas soluções consiste nas operações de fluxo.
Reduzir os Tempos de Trocas de Ferramentas e Matrizes (<i>setup</i>)	Tempo de troca reduzidos é um pré-requisito indispensável para produção contrapedido.
Eliminar as Quebras e Defeitos (<i>jidoka</i>)	Quebras e Defeitos gera a necessidade de estoque, deve ser adotada a interrupção de uma linha ou máquina sempre que ocorre uma situação anormal.
Fusão do Balanceamento com a Produção com Estoque Zero	As quantidades e as variedades de produção por hora são balanceadas para todos os processos.
Operações de Fluxo Totalmente Integradas	Superar as barreiras criadas pela divisão do trabalho em plantas e seções.
Reduzir custos de mão de obra	A redução de mão de obra é um comprometimento no TPS e reconhece a vantagem de utilizar máquinas.
Automação	Transferir as funções mentais humanas às máquinas.
Manter e Desenvolver Operações-Padrão	Padronização das operações em cada etapa do processo de produção.
Gerenciar utilizando Recursos Visuais	Ferramentas para que possa ser compreendido o funcionamento do sistema de produção rapidamente, como exemplo o <i>Kanban</i> e <i>Andon</i> .

QUADRO 2 – CARACTERÍSTICAS DO TPS. FONTE: SHINGO (1996)

De acordo com SUGIMORI *et al.* (1977), Toyota implementou seu sistema de produção fundamentado em dois conceitos básicos: redução de custos por meio da redução do desperdício, incorporando conceitos de *just-in-time production* e *jidoka*; e plena utilização da capacidade dos trabalhadores, para isso, eliminando desperdício com a movimentação, focando na segurança, mecanizando as operações e valorizando as capacidades individuais dos trabalhadores dando-lhes maior responsabilidade e autoridade.

De acordo com Womack *et al.* (2004), a Toyota havia delineado os princípios da produção enxuta (*lean production*) no início da década de 1960 e nessa década conquistado enorme vantagem sobre os produtores em massa, conseguindo aumentar continuamente sua participação na produção mundial de veículos.

O termo *lean production*, é utilizado por Womack *et al.* (1990), como forma de nominar a aplicação do TPS em outras empresas diferentes da Toyota. O termo *Lean Thinking* caracteriza a aplicação do *lean production* em outras atividades econômicas, além da indústria automobilística (WOMACK *et al.*, 2004; WOMACK; JONES, 2004).

2.1.2 PRINCÍPIOS DO *LEAN THINKING*

Womack e Jones (2004) acreditam que o sistema de produção implantado na Toyota, poderia ser aplicado fora da indústria automobilista e da manufatura, originando a definição *Lean Thinking*. Para isso os autores agrupam os conceitos *Lean* em princípios norteadores das empresas que buscam a aplicação da filosofia.

De acordo com Womack e Jones (2004) os cinco princípios do *Lean Thinking* são: Determinar Valor, Fluxo de Valor, Fluxo, Sistema Puxado e Perfeição. O detalhamento de cada princípio é apresentado no Quadro 3.

Princípios Lean Thinking	Descrição
Determinar Valor	<p>Especificar o valor é o primeiro passo essencial no pensamento enxuto. O valor deve ser definido pelo cliente final, ou seja, um bem ou serviço que atenda suas necessidades a um “preço específico no momento específico”. O conceito de valor da maioria das empresas está invertido, pois elas olham para dentro, definindo o valor pelo nível de sofisticação e tecnologia de seus produtos ou serviços, muitas vezes irrelevantes aos verdadeiros desejos do cliente.</p> <p>Portanto, o pensamento enxuto “deve começar com uma tentativa consciente de definir precisamente valor em termos de produtos específicos com capacidades específicas, oferecidos a preços específicos, por meio do diálogo com clientes específicos”.</p> <p>Para isso, é preciso ignorar as tecnologias existentes e repensar a empresa com base em uma linha de produto dedicada às reais necessidades do cliente. “Oferecer o bem ou serviço errado da forma certa é desperdício”.</p>
Fluxo de Valor	<p>Fluxo de Valor são todas as ações necessárias para entregar um produto ou serviço (ou ambos) ao cliente final. Para isso, deve-se analisar as três tarefas gerenciais críticas: a tarefa de solução de problemas, que vai da concepção até a entrega do produto; a tarefa de gerenciamento da informação, que corresponde ao recebimento do pedido até a entrega; e a tarefa de transformação física, que vai da matéria-prima ao produto acabado na mão do cliente.</p> <p>A identificação de fluxo de valor inteiro consiste em avaliar os processos que envolvem os fornecedores, portanto, o pensamento enxuto deve ir além da empresa, deve analisar o conjunto inteiro de atividades que envolvem a criação e fabricação de um produto específico, da venda inicial à entrega, mas também da matéria-prima produzida fora da empresa.</p>
Fluxo	<p>Para fazer o fluxo de valor “fluir”, a primeira etapa a ser realizada é focar o objeto, como exemplo o projeto, o produto ou o pedido específico a ser desenvolvido. A segunda etapa é eliminar as fronteiras departamentais, profissionais e de funções com objetivo de evitar retrabalhos e filas, consequentemente desperdícios. A terceira etapa é planejar as práticas e ferramentas de trabalhos específicas, de forma a eliminar retrabalhos, paralisações e sucata.</p> <p>Como uma forma de lidar com o fluxo em uma linha de produção, a Toyota implementou o <i>just-in-time</i>, que significa que os componentes necessários à montagem estejam disponíveis na linha de montagem no momento e quantidade que serão utilizados.</p> <p>Combater o pensamento departamentalizado, em lotes, pois existe quase sempre uma forma mais eficiente e precisa para realizar as tarefas quando se trabalha o produto continuamente, ou seja, é necessário focar o produto e suas necessidades de forma que atividades necessárias para projetar, pedir e fornecer um produto ocorram em um fluxo contínuo. Para isso, um exemplo de uma estratégia é criar equipes de produtos realmente dedicadas com todas as habilidades para especificar valor, projetar, comprar e planejar a produção, na mesma sala, utilizando a metodologia de processo decisório em equipe.</p>

(CONTINUA)

QUADRO 3 – PRINCÍPIOS DO LEAN THINKING. FONTE: WOMACK E JONES (2004)

(CONTINUAÇÃO)

Princípios <i>Lean Thinking</i>	Descrição
Sistema Puxado	<p>“Puxar”, significa que um processo deve produzir um bem ou um serviço apenas quando o cliente tenha solicitado. O objetivo do sistema é poder fabricar qualquer produto, em qualquer combinação, de modo a acomodar as mudanças da demanda.</p> <p>Na prática a projeção de vendas deve ser substituída pela produção baseado na solicitação do cliente. A produção deve ser realizada em lotes pequenos, com trocas rápidas de ferramentas (<i>set ups</i>) e balanceamento da produção utilizando recursos visuais como o <i>Kanban</i>.</p> <p>O resultado é o rápido atendimento de um pedido, mesmo que seja muito específico, eliminando estoques de produtos que o cliente não deseja.</p> <p>“Puxar” implica toda a cadeia de valor, envolvendo inclusive o fornecedor.</p>
Perfeição	<p>A aplicação dos quatro princípios iniciais - determinar o valor com precisão, identificar o fluxo de valor, fazer o valor fluir sem interrupções e deixar o cliente puxar o valor – as etapas de um fluxo de valor podem ser melhorados. Portanto a perfeição, que é o sinônimo de eliminação total do desperdício deve ser o foco de uma empresa enxuta.</p> <p>Em cada etapa os gerentes precisam aprender a “ver”: ver o fluxo de valor, ver o valor fluir e ver o valor sendo puxado pelo cliente. Ver, resulta na identificação dos objetivos de melhoria que resultam na perfeição. Esses objetivos devem ser claros e tangíveis para a empresa como um todo. A “perfeição é como o infinito”, tentar imagina-lo é impossível, porém o esforço para fazê-lo “oferece a inspiração e a direção essenciais para o progresso ao longo do caminho”.</p>

QUADRO 3 (CONT.) – PRINCÍPIOS DO LEAN THINKING. FONTE: WOMACK E JONES (2004)

2.1.3 APLICAÇÃO LEAN THINKING EM DIVERSAS ÁREAS

Womack e Jones (2004) apresentam em seu livro “A mentalidade enxuta nas empresas – *Lean Thinking: Elimine desperdício e crie riqueza*”, publicado em 1996, exemplos dos esforços necessários para transformar as organizações em massa em organizações enxutas. Os autores apresentam casos de sucesso em áreas além da automobilística, como construção civil (*lean construction*), hospitalar (*lean healthcare*) e serviços (*lean service*).

O trabalho de Portioli-Staudacher (2010) analisa a aplicação *Lean Thinking* na área de serviços, em três empresas de logística e sete empresas da área financeira. Identifica que empresas de logística que processam um produto físico estão mais avançadas do que bancos e empresas de seguros. Destaca que a principal diferença da aplicação *lean* em serviços são os processos relacionados com cliente. Conclui que a aplicação de *Lean Thinking* na área de serviços ainda está no início.

De acordo com Staats, Brunner e Upton (2011), a implementação de um sistema de produção enxuta no trabalho do conhecimento é possível e muda como a organização aprende. Para isso o processo principal deve ser alterado e os resultados ajudam a simplificar a comunicação, as tarefas e a solução dos problemas.

Holden (2011) apresenta uma revisão de literatura sobre a aplicação de *Lean Thinking* em Departamentos de tratamento de Saúde de Emergência (ED). Relata que na tentativa de reduzir custos, aumentar a qualidade e a segurança do serviço, as organizações de saúde estão cada vez mais adotando *Lean Thinking*. Nos Estados Unidos, 60% das ED relataram ter implantado os princípios *Lean*. Conclui que *Lean* oferece uma significativa oportunidade de melhoria nos processos de atendimento de saúde, porém necessita de estudos adicionais de como realizar a adaptação do *Lean* para essa área.

O trabalho de Middleton e Joyce (2012) analisa como as ideias *Lean* podem ser aplicadas para gerenciamento de projetos de *software*, conclui que o desempenho do desenvolvimento de *software* com adoção da abordagem *Lean*, após 12 meses, reduziu o tempo de entrega dos projetos em 37% e os defeitos caíram 24%. Mehta, Anderson e Raffo (2008), também confirmam que os princípios *Lean* trazem benefícios para melhores práticas de engenharia de *software*, para melhorar o fluxo de trabalho e a estratégia de gerenciamento de projetos.

2.1.4 APLICAÇÃO LEAN THINKING PARA DESENVOLVIMENTO DE SOFTWARE

Esta seção apresenta uma análise de trabalhos científicos relacionados à aplicação do *Lean Thinking* para desenvolvimento de *software*. O objetivo é comprovar sua relevância e avaliar sua aplicação em projetos de *software*.

Middleton, Flaxel e Cookson (2005) apresentam em seu trabalho um estudo de caso da empresa Timberline Inc. e a aplicação da iniciativa *Lean* para desenvolvimento de *software*. O estudo demonstrou que o *Lean Thinking* pode trabalhar com sucesso para desenvolvedores de *software*.

Perera e Fernando (2007) apresentam as diferenças entre as práticas *Lean* e *Agile* e conclui que os princípios *Lean* podem ser utilizados para aprimorar o paradigma *Agile* e ressalta que apesar das duas abordagens apresentarem modelos de processos similares, não possuem processos redundantes.

Petersen e Wohlin (2010) propõe uma nova abordagem para unificar o processo de qualidade e as práticas de desenvolvimento de *software Lean* chamado método SPI-LEAM. Afirmam que desenvolvimento de *software Lean* propaga dois princípios importantes que ajudam a melhoria do processo: a identificação de resíduos e a interação entre as pessoas no processo de desenvolvimento com uma perspectiva de ponta a ponta. Em outro trabalho de Petersen e Wohlin (2011), é apresentando uma abordagem de fluxo de processo para desenvolvimento de *software Lean* com objetivo de aumentar o rendimento e reduzir o *lead-time*, para alcançar respostas rápidas ao cliente e fornecer um sistema de rastreamento que mostra o progresso do desenvolvimento de produtos de *software*.

O trabalho de Wang, Conboy e Cawley (2012) analisa 30 publicações com relatos da experiência da aplicação da abordagem *Lean* para desenvolvimento ágil de *software*. Os resultados deste estudo mostram que *Lean* pode ser aplicado em processos ágeis de diferentes maneiras e para diferentes propósitos e que os conceitos, princípios e práticas *Lean* são mais frequentemente utilizados para melhoria contínua do processo ágil.

2.2 LEAN SOFTWARE DEVELOPMENT (LSD)

A aplicação dos princípios e filosofias do *Lean Thinking* para área de Tecnologia da Informação (TI) ficou conhecida como *Lean IT* (BELL; ORZEN, 2011; CANTANHEDE, 2014).

De acordo com Bell e Orzen (2011), *Lean Software Development* (LSD), é uma das áreas de aplicação e estudo do *Lean IT* e envolve o ambiente que o *software* opera e todo o fluxo de valor. LSD não segue um modelo prescritivo,

mas uma abordagem iterativa e adaptativa, dessa forma os envolvidos rapidamente aprendem e resolvem problemas.

Lean Software Development (LSD) é um paradigma de desenvolvimento de produtos com foco na criação de valor para o cliente, eliminando desperdícios, otimizando os fluxos de valor, capacitando as pessoas e continuamente melhorando (EBERT; ABRAHAMSSON; OZA, 2012).

Lean Software Development (LSD) é a aplicação dos princípios do Toyota *Production System* (TPS) para o desenvolvimento de *software*. O termo *Lean Software Development* (LSD), começou a ser utilizado com a publicação do livro com o mesmo nome, pelos autores Mary Poppendieck e Tom Poppendieck, em 2003 (POPPENDIECK, 2007).

De acordo com Poppendieck (2007), LSD quando aplicado corretamente, resulta em produtos de *software* de alta qualidade, desenvolvidos de forma rápida e com baixo custo.

Poppendieck e Cusumano (2012) definem que LSD não é um processo de desenvolvimento. O foco do LSD é um conjunto de princípios para todo o ciclo de vida do produto com objetivo de agregar valor ao cliente. Esses princípios são utilizados para discutir práticas de desenvolvimento de produto mais apropriadas, dependendo da situação única de cada organização.

Poppendieck e Poppendieck (2011) esclarecem que princípios “são verdades subjacentes que não mudam no tempo ou espaço”, e práticas “são a aplicação dos princípios a uma situação particular”.

Poppendieck e Cusumano (2012) enumeram os sete princípios do LSD: 1) Otimizar o todo; 2) Respeitar as Pessoas; 3) Entregar rápido; 4) Adiar comprometerimentos; 5) Criar conhecimento; 6) Integrar Qualidade; 7) Eliminar o desperdício.

2.2.1 PRINCÍPIOS DO *LEAN SOFTWARE DEVELOPMENT* (LSD)

De acordo com Poppendieck e Poppendieck (2011), a implantação de novas práticas para melhorar o processo de desenvolvimento de *software*, tem maiores chances de sucesso se previamente são entendidos os princípios fundamentais da abordagem que está sendo utilizada. Copiar práticas, sem entendimento dos princípios tem um histórico de resultados medíocres. Práticas enxutas de produção e de gerenciamento da cadeia de suprimentos não são facilmente traduzidas para desenvolvimento de *software*, pois são distintos em operações e logística.

Poppendieck e Poppendieck (2011) acreditam que qualquer processo de desenvolvimento, que lida com um ambiente de mudanças deveria ser um processo empírico, pois tem maior capacidade de adaptação. Um processo empírico começa com um conceito de produto de alto nível e estabelece ciclos de *feedback* bem definidos que ajustam as atividades para criar uma interpretação ótima do conceito. Por isso, o desenvolvimento de *software* deveria ser um processo empírico.

Portanto o entendimento dos conceitos *Lean* é fundamental para a melhoria do processo de desenvolvimento de *software*, que deve ser empírico para permitir a adaptação às características específicas de cada produto de *software*.

O Quadro 4 apresenta o detalhamento dos princípios *Lean Software Development* (LSD).

Princípios Lean Software Development	Descrição
Princípio 1: Otimizar o Todo	<p>Uma organização <i>Lean</i> deve otimizar todo o fluxo de valor, desde o recebimento da necessidade dos clientes até o <i>software</i> implantado e a necessidade atendida.</p> <p>Para otimizar o todo é necessário implementar os princípios <i>Lean</i> em todo o fluxo de valor. Para isso, é necessário nomear um líder ou uma equipe responsável pelo fluxo de valor inteiro; reestruturar as medidas, substituindo medidas locais por medidas para todo o fluxo de valor; e reduzir o custo por atrasos no fluxo de valor que ocorrem na passagem de fronteiras de departamento ou de empresas.</p>
Princípio 2: Respeitar as Pessoas	<p><i>Lean</i> é um sistema de gerência que tem objetivo de criar pessoas pensantes e comprometidas em todos os níveis da organização, principalmente na “linha de frente”, que fazem o produto. Para isso, as pessoas devem ser profundamente respeitadas, que é a base do Sistema Toyota de Produção.</p> <p>Em LSD as práticas que devem ser implantadas envolvem o engajamento das pessoas, encorajando o trabalho em equipe, treinando líderes de equipe (e não de processos), transferindo a responsabilidade e a tomada de decisão para os mais baixos níveis possíveis e alimentando o orgulho dos membros da equipe.</p>
Princípio 3: Entregar Rápido	<p>Entregar rápido significa reduzir o desperdício. Processos de desenvolvimento de <i>software</i> que buscam a melhoria contínua, eliminando os desperdícios (defeitos, complexidade, baixa produtividade, intolerância a mudança, funcionalidades adicionais), aumenta o percentual de tempo gasto adicionando valor. A “alta velocidade” só pode ser sustentada se integrada com qualidade.</p> <p>Em LSD, “precisamos descobrir como entregar <i>software</i> tão rápido que os clientes não tenham tempo de mudar de ideia”. Para LSD uma prática para garantir a entrega rápida é minimizar as transferências de pessoas da equipe. Processos rápidos e flexíveis exige que as pessoas estejam engajadas e atuantes, tomando decisões e resolvendo problemas de forma independente.</p> <p>Entregar rápido significa reduzir o tempo de ciclo.</p>
Princípio 4: Adiar Comprometimentos	<p>Em desenvolvimento de <i>software</i> as decisões irreversíveis devem ser tomadas no último momento possível, quando se tem mais informações. Diante da incerteza, a abordagem mais bem-sucedida é combater problemas difíceis experimentando soluções diversas e deixar as opções críticas pendentes até que uma decisão precisar ser tomada.</p>
Princípio 5: Criar Conhecimento	<p>As empresas que exibem excelência em longo prazo em desenvolvimento de produtos geram novo conhecimento por meio de experimentação e compartilham este conhecimento com a organização. A chave para a criação do conhecimento organizacional está nos processos de conversão do conhecimento explícito e tácito (NONAKA; TAKEUCHI, 1997).</p> <p>O desenvolvimento de <i>software</i> é um processo de criação de conhecimento. É importante que o aprendizado sistemático seja encorajado durante todo o ciclo de desenvolvimento, mas também possibilite uma melhoria constante desse processo. Para LSD, uma prática que deve ser adotada é realizar o projeto detalhado durante o desenvolvimento do produto, com feedback constante das partes interessadas e dos clientes.</p>

(CONTINUA)

QUADRO 4 – PRINCÍPIOS DO LEAN SOFTWARE DEVELOPMENT (LSD).
 FONTE: POPPENDIECK E POPPENDIECK (2011).

(CONTINUAÇÃO)

Princípios Lean Software Development	Descrição
Princípio 6: Integrar Qualidade	<p>O LSD deve ser muito disciplinado, construindo qualidade no código desde o início, evitando defeitos e não os descobrindo depois no processo de testes do produto.</p> <p>Existem dois tipo de inspeção: inspeção após os defeitos ocorrerem e inspeção para prevenir defeitos. Para se obter qualidade o objetivo é controlar as condições para evitar os defeitos, evitando a inspeção após a ocorrência dos defeitos. O trabalho de inspeção (testes) é prevenir defeitos e não encontrá-los (SHINGO, 1996).</p> <p>Portanto, uma boa prática é inspecionar pequenas partes para identificar os defeitos o mais cedo possível, e se encontra-los corrigir imediatamente evitando filas de serviço parcialmente concluído, considerado retrabalho. Para o paradigma Lean, “filas são pontos de acúmulo e desperdício”. Para LSD deve-se perseguir “fazer certo na primeira vez”, para isso práticas como Desenvolvimento Guiado por Testes (TDD) e integração contínua podem ser utilizadas.</p>
Princípio 7: Eliminar o Desperdício	<p>O foco do Sistema Toyota de Produção (TPS) é eliminar o desperdício, essa é a essência da produção lean. Para isso, o primeiro passo é entendimento de valor pelo cliente. Desperdício é qualquer coisa que seja feito que não adicione valor ao cliente, ou atraso que impeça o cliente de obter valor no momento que deseja. Em desenvolvimento de software as maiores causas de desperdício são trabalho executado parcialmente; funcionalidades adicionais desnecessárias; requisitos especificados muito antes ou testados muito depois do desenvolvimento; Perda de Conhecimento.</p>

QUADRO 4 (CONT.) – PRINCÍPIOS DO LEAN SOFTWARE DEVELOPMENT (LSD).
 FONTE: POPPENDIECK E POPPENDIECK (2011).

O Quadro 5 apresenta uma análise comparativa de desperdício na produção a partir de Shingo (1996) com os desperdícios no desenvolvimento de *software* a partir de Poppendieck e Poppendieck (2011).

Produção	Desenvolvimento de Software
Estoques no Processo	Trabalho Inacabado
Superprodução	Funcionalidades Adicionais
Processo Adicional	Reaprendizagem
Transporte	Transferência de Controle
Movimentação	Troca de Tarefas
Esperas	Atrasos
Defeitos	Defeitos

QUADRO 5 – OS SETE DESPERDÍCIOS.

FONTE: POPPENDIECK E POPPENDIECK (2011).

Trabalho Inacabado: Na produção estoque é desperdício. Em desenvolvimento de *software* equivale aos trabalhos parcialmente acabados. O objetivo é entregar o *software* completamente pronto (integrado, testado,

documentado), em um fluxo contínuo e rápido. O meio para se alcançar isso é dividindo o trabalho em pequenos lotes, ou tarefas.

Funcionalidades Adicionais: “Se não existe uma necessidade econômica clara e atual para uma funcionalidade, ela não deve ser desenvolvida”. Essa é principal fonte de desperdício. O processo ideal de desenvolvimento de *software* dever permitir entregar 80% do valor do produto com 20% do código. (POPPENDIECK; POPPENDIECK, 2011).

Reaprendizagem: Redescobrir algo que se sabia, mas esqueceu, é talvez a melhor definição de “retrabalho”, outro ponto é ignorar o conhecimento que as pessoas trazem para o trabalho, por falta de engajamento no processo de desenvolvimento.

Transferências de Controle: O conhecimento tácito, que está na mente das pessoas é difícil de transferir, portanto uma grande quantidade de informação é perdida. Reduzir o número de transferências, utilizar equipes multifuncionais e comunicação face a face, são algumas formas de reduzir o desperdício da transferência de controle.

Troca de Tarefas: Profissionais de conhecimento necessitam de tempo para reiniciar uma nova tarefa. Trabalho com excessiva transição de tarefas gera distração e normalmente prejudicam os resultados de ambas as tarefas.

Atrasos: Esperar por pessoas que estão trabalhando em outras áreas é uma grande causa do desperdício proveniente do atraso. O conhecimento tem que estar disponível exatamente quando e onde for necessário, por exemplo, com equipes completas, compartilhando a mesma sala e realizando iterações curtas com *feedback* frequente podem diminuir drasticamente os atrasos.

Defeitos: Toda base de código deve incluir um conjunto de testes à prova de falha que não permita defeitos no código. Testes devem acontecer mais cedo e mais frequentemente para encontrar as falhas inesperadas, e sempre que um defeito for encontrado, um teste deve ser criado para que ele não ocorra novamente.

2.2.2 ROTEIRO DE IMPLANTAÇÃO *LEAN SOFTWARE DEVELOPMENT* (LSD)

O Quadro 6 apresenta um roteiro com 21 etapas para implantação *Lean Software Development*, baseado nos 7 (sete) princípios criado por Poppendieck e Poppendieck (2011).

Etapa	Descrição
Princípio 1: Otimizar o todo	
1.Implementar os princípios <i>Lean</i> ao longo do fluxo de valor inteiro e do produto completo	Nomear um líder/equipe de liderança fluxo valor; Desenhar o fluxo de valor e procurar interrupções no fluxo; Corrigir processos falhos e abastecer capacidades ausentes no sistema.
2.Reestruturar as Medidas	Substituir as medidas locais por medidas de fluxo de valor.
3.Reduzir o custo de passagem de limites	Acelerar o fluxo de valor eliminando as fronteiras departamentais ou da empresa.
Princípio 2: Respeitar as Pessoas	
4.Treinar líderes de equipe/supervisores	Treinar, orientar e liberar líderes de equipe para implementar os princípios <i>lean</i> .
5.Transferir a responsabilidade e tomada de decisão para os mais baixos níveis possíveis	Oferecer condições para as equipes de trabalho criar valor; Fazer com que as equipes de trabalho projetem seus próprios processos <i>lean</i> , sob orientação dos líderes.
6.Alimentar o Orgulho dos membros da equipe	Criar incentivos para a equipe e não individuais; Automatizar tarefas de rotina; Dar autonomia para equipe definir a melhor abordagem para seu trabalho; Eliminar: espaços e práticas de trabalho descuidadas, prazos impossíveis, falta de tempo para refatoração adequada, imposição de processos.
Princípio 3: Entregar Rápido	
7.Trabalhar em Lotes Pequenos	Reduzir o tamanho do projeto; Encurtar os ciclos de <i>release</i> ; Estabilizar; Repetir; Esvaziar cada lista e fila, limitando de forma agressiva seu tamanho.
8.Limitar o tamanho conforme a capacidade	Trabalhar em uma cadência repetível para estabelecer a capacidade; Esperar que as equipes puxem as filas com base em sua velocidade comprovada para terminar completamente o trabalho antes de começarem com mais trabalho; Limitar o tamanho das filas e não aceitar trabalho a menos que exista espaço vazio na fila.
9. Concentrar-se no ciclo de tempo e não na utilização	Medir a resposta do processo pelo tempo de chegada ao mercado (<i>time to market</i>) ou ao cliente; Evitar o controle de alocação/utilização de recursos.

(CONTINUA)

QUADRO 6 – ROTEIRO IMPLANTAÇÃO *LEAN SOFTWARE DEVELOPMENT*.
 FONTE: POPPENDIECK E POPPENDIECK (2011).

(CONTINUAÇÃO)

Etapa	Descrição
Princípio 4: Adiar o Comprometimento	
10. Acabar com a noção de que é uma boa prática começar o desenvolvimento com uma especificação completa	Praticar desenvolvimento concorrente, ou seja, permitir que a especificação ocorra durante o processo de desenvolvimento.
11. Quebrar pendências	Quebrar as dependências para permitir que as funcionalidades sejam feitas em qualquer ordem.
12. Manter opções	Desenvolver múltiplas opções para todas as decisões irreversíveis e não fechar qualquer opção até o último momento de resposta.
Princípio 5: Criar Conhecimento	
13. Criar equipes construtoras de design	Garantir que a arquitetura de sistemas permita aos produtos serem quebrados em módulos que podem ser endereçados a equipes multifuncionais representando os interesses de todos os passos do fluxo de valor; Fornecer a cada equipe a liderança e os incentivos adequados para manter o engajamento, a transparência e o <i>feedback</i> intensivo; Encorajar a equipe para compartilhar mais cedo e mais frequentemente, a falhar rápido e a aprender constantemente.
14. Manter uma cultura de melhoria constante	Criar o tempo e a expectativa de que cada equipe deve melhorar seus processos; Promover eventos multi-equipes e multifuncionais para identificar acomodações e restrições no fluxo de valor; Substituir práticas e políticas que melhorarão resultados globais.
15. Ensinar métodos de resolução de problemas	Ensinar o ciclo PDCA ou método científico ou outra variação desses métodos para resolução de problemas; Esperar que as equipes estabeleçam hipóteses, conduzam experimentos rápidos, criem documentação concisa e implementem mudanças justificadas.
Princípio 6: Integrar Qualidade	
16. Sincronizar	Reduzir agressivamente o trabalho inacabado; Escrever os testes primeiro; Testar o código mais cedo possível; Corrigir os defeitos no momento que forem detectados; Integrar o código de forma contínua e extensiva; Utilizar sincronização aninhada para integrar sistemas maiores e mais complexos.
17. Automatizar	Automatizar cada processo que puder o mais breve possível. Automatizar testes, <i>builds</i> , instalação e rotinas para permitir que as pessoas mantenham pensando como fazer melhor as coisas.
18. Refatorar	Manter o código base limpo e simples; Refatorar o código, os testes e a documentação para reduzir a complexidade no momento que as duplicações aparecerem.

(CONTINUA)

QUADRO 6 (CONT.) – ROTEIRO IMPLANTAÇÃO LEAN SOFTWARE DEVELOPMENT.
 FONTE: POPPENDIECK E POPPENDIECK (2011).

(CONTINUAÇÃO)

Etapa	Descrição
Princípio 7: Eliminar Desperdícios	
19. Fornecer liderança de mercado e liderança técnica	Garantir que exista uma responsabilidade e o entendimento do que o cliente valoriza associada ao entendimento do que a tecnologia pode entregar; Certificar de que a equipe construa a coisa certa.
20. Não criar nada, exceto valor	Garantir que todos os passos em todos os processos estão concentrados nas atividades que criam valor; Medir a eficácia do ciclo de processo e manter melhorando.
21. Escrever menos código	Limitar de forma agressiva as funcionalidades em um sistema para apenas aquelas que são absolutamente necessárias para agregar valor. Desenvolver uma intolerância organizacional à complexidade.

QUADRO 6 (CONT.) – ROTEIRO IMPLANTAÇÃO LEAN SOFTWARE DEVELOPMENT.
 FONTE: POPPENDIECK E POPPENDIECK (2011).

2.3 MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

Os conceitos relacionados ao desenvolvimento ágil de *software* nasceram na década de 1990, decorrentes da insatisfação de abordagens pesadas de engenharia de *software*, caracterizado pelo formalismo nas documentações e regulamentações. A abordagem ágil, conhecida inicialmente por “métodos leves” (*Lightweight Methods*), permitia que a equipe de desenvolvimento focasse no *software* e não na documentação (SOMMERVILLE, 2011; SBROCCO; MACEDO, 2012).

De acordo com Sbrocco e Macedo (2012), métodos ágeis para desenvolvimento de *software* foi fortemente influenciado por boas práticas adotadas pela indústria automobilística japonesa, destacando a Toyota.

Métodos ágeis baseiam-se em uma abordagem iterativa e incremental para a especificação, desenvolvimento e entrega do produto, sendo mais adaptados a desenvolvimento de produtos quando os requisitos mudam durante o processo de desenvolvimento. O objetivo é entregar *software* funcionando rapidamente aos clientes, os quais podem solicitar alterações e novos requisitos a serem incluídos em iterações posteriores do sistema (SOMMERVILLE, 2011).

A filosofia que sustenta os métodos ágeis é refletida no manifesto ágil, acordado em uma reunião realizada em 2001 onde estavam presentes 17 profissionais de *software* que trabalhavam com métodos leves utilizados na época, como Scrum, *Extreme Programming* (XP), *Dynamic System Development Method* (DSDM), *Adaptative Software Development* (ASD), Crystal, *Feature Driven Development* (FDD) (HIGHSMITH; COCKBURN, 2001).

Esse manifesto pode ser considerado um marco para expansão da abordagem ágil, principalmente na indústria de *software*. Seu enfoque está na descrição de 4 (quatro) valores e 12 princípios para o desenvolvimento de *software* (CONFORTO, 2013).

De acordo com AGILE ALLIANCE (2016), os valores do Manifesto são: i) Indivíduos e interações mais que processos e ferramentas; ii) *Software* em funcionamento mais que documentação abrangente; iii) Colaboração com o cliente mais que negociação de contratos; iv) Responder a mudanças mais que seguir um plano.

Os princípios dos métodos ágeis podem ser sintetizados: i) Envolvimento do cliente no processo de desenvolvimento, fornecendo e priorizando requisitos do sistema; ii) Entrega Incremental, o *software* é desenvolvido em incrementos; iii) Pessoas, não processos, foco na habilidade da equipe de desenvolvimento as quais devem ser reconhecidas e exploradas, a equipe deve ser auto-gerenciada, sem definição de processos prescritivos; iv) Aceitar as mudanças, alterações em requisitos são aceitos e o sistema deve acomodar essas mudanças; v) Manter a simplicidade, foco na simplicidade do produto de *software* a ser desenvolvido, assim como no processo de desenvolvimento (SOMMERVILLE, 2011; HIGHSMITH; COCKBURN, 2001).

Métodos ágeis geralmente utilizam equipes pequenas de desenvolvimento para garantir a integração e facilitar o autogerenciamento e comunicação. De acordo com Highsmith e Cockburn (2001), a criatividade da equipe, e não a volumosa quantidade de regras é o caminho para gerenciar problemas complexos durante o processo de desenvolvimento de *software*.

Satisfazer o cliente é o foco das equipes de desenvolvimento ágil, para isso o envolvimento do cliente, das equipes de negócio e desenvolvimento devem trabalhar juntas, permitindo rápida identificação de falhas de requisitos e adaptação às mudanças que gerem vantagem competitiva ao cliente (SBROCCO; MACEDO, 2012).

Outra característica importante em métodos ágeis é a preocupação com a excelência técnica e melhoria contínua. A equipe deve realizar reflexões periódicas sobre seu desempenho, para identificar como se tornar mais eficaz (AGILE ALLIANCE, 2016).

2.3.1 APLICAÇÃO DE MÉTODOS ÁGEIS PARA DESENVOLVIMENTO DE SOFTWARE

De acordo com Dingsoyr *et al.* (2012), a comunidade de pesquisa tem concedido atenção sobre as questões relacionadas com o desenvolvimento ágil de *software*, desde que o manifesto ágil foi criado, principalmente focado em eXtreme Programming (XP) e Scrum.

O trabalho de Misra, Kumar V. e Kumar U. (2010) apresenta uma *survey*, com objetivo de identificar as principais mudanças necessárias para a transição de práticas tradicionais de produção de *software* para práticas ágeis. Identificou que as principais mudanças estão relacionadas com as características pessoais da equipe, mudanças de atitude dos clientes e mudanças no processo de gestão do conhecimento.

Lagerberg *et al.* (2013) apresenta uma pesquisa sobre a aplicação de métodos ágeis em projetos de larga escala. A pesquisa foi aplicada em dois projetos da empresa Ericsson com 420 e 120 membros respectivamente. Os resultados mostraram que o tamanho do projeto não afeta o *stress* da equipe, porém é fundamental o compartilhamento do conhecimento organizacional para facilitar a gestão dos projetos e aumentar a produtividade.

O trabalho de Hoda, Noble e Marshall (2013) apresenta uma pesquisa com 58 praticantes de métodos ágeis em 23 empresas diferentes durante um período

de 4 (quatro) anos e identificou que as práticas de auto-organização da equipe, característica dos métodos ágeis, geraram regras diferentes e “espontâneas” para cada equipe, resultados do aprendizado coletivo e da gestão do conhecimento. Entender essas regras ajudam as equipes a executar melhor as atividades e facilitam o gerenciamento do projeto.

2.4 METODOLOGIA SCRUM

Scrum é um arcabouço para desenvolver e manter produtos complexos, composto por eventos, papéis, artefatos e as regras que os mantêm integrados (SUTHERLAND; SCHWABER, 2013).

A metodologia Scrum segue os princípios do manifesto ágil e baseia-se em seis características: flexibilidade dos resultados, flexibilidade dos prazos, times pequenos, revisões frequentes, colaboração e orientação a objetos (SBROCCO; MACEDO, 2012).

Scrum é utilizado para gerenciar o desenvolvimento de um projeto utilizando práticas iterativas e incrementais. É composto por um conjunto de boas práticas de gerenciamento que aceita ajustes rápidos, além de acompanhamento e visibilidade constantes (FONSECA; CAMPOS, 2008).

O objetivo do Scrum é proporcionar um método pelo qual as pessoas possam resolver problemas complexos e adaptativos, de forma criativa e produtiva com o mais alto valor possível.

Scrum não é um processo ou técnica para construir produtos, porém é um arcabouço dos quais podem ser empregados vários processos e técnicas (SUTHERLAND; SCHWABER, 2013).

Ken Schwaber e Jeff Sutherland desenvolveram o Scrum, inspirados pelo artigo publicado pelos professores de administração japoneses, Hirotaka Takeuchi e Ikujiro Nonaka, com o título “*The New New Product Development Game*”. Takeuchi e Nonaka analisaram algumas das empresas mais produtivas

e inovadoras, como Honda, Fuji Xerox, 3M, Hewlett-Packard, Toyota, entre outras. Essas empresas utilizavam práticas de manufatura enxuta, como processo de desenvolvimento de produtos de sobreposição, que era mais rápido e flexível, com equipes pequenas, multifuncionais e autônomas, nas quais os executivos agiam apenas como facilitadores focados em eliminar os obstáculos, mas não determinavam como a equipe deveria desenvolver o produto (SUTHERLAND, 2014).

Jeff Sutherland, vice-presidente de engenharia da empresa Easel, percebeu que seu time de desenvolvimento precisava de uma metodologia de desenvolvimento mais adequada para proporcionar rápido desenvolvimento. Decidiu, portanto, implantar as práticas identificadas no artigo de Takeuchi e Nonaka para desenvolvimento de *software*, nascendo assim o Scrum. Em 1995, com a ajuda de Ken Schwaber, foi apresentado um artigo intitulado “*SCRUM Development Process*”, o qual sistematizava essas práticas, em uma conferência de pesquisa da *Association for Computing Machinery* (SUTHERLAND, 2014).

A denominação Scrum surgiu a partir da comparação, dos professores Takeuchi e Nonaka, do trabalho das equipes de alta produtividade com um time de rugby e diziam que as melhores equipes agiam como se houvesse um *scrum*, ou seja, como uma unidade integrada, cada integrante desenvolvendo um papel específico e todos se ajudando em busca de um benefício comum (TAKEUCHI; NONAKA, 1986; SBROCCO; MACEDO, 2012).

De acordo com Poppendieck e Poppendieck (2012), ao longo dos anos, o método ágil Scrum se tornou cada vez mais popular para substituir a gestão tradicional de projetos, considerado uma forma enxuta (*lean*) de desenvolvimento de *software*.

2.4.1 ESTRUTURA DO SCRUM

O desenvolvimento de *software* em projetos que utilizam o Scrum como método é dividido em ciclos chamados de *Sprints*, que representam iterações

de trabalho com duração variável, geralmente ocorre em intervalos entre 2 (duas) e 4 (quatro) semanas. As equipes são pequenas e formadas por projetistas, programadores, engenheiros de *software* e gerentes de qualidade que desenvolvem suas tarefas a partir dos requisitos do *Backlog* do Produto (*Product Backlog*) (SCHWABER, 2004).

O *Backlog* do Produto é uma lista ordenada de tudo o que deve ser necessário no produto, é uma origem única dos requisitos do produto, que pode ser alterada conforme necessidade do *Product Owner* (SUTHERLAND; SCHWABER, 2013).

De acordo com Sutherland e Schwaber (2013), os papéis e responsabilidades do Scrum são:

a) *Product Owner (PO)*: representa o cliente e é responsável por garantir que a equipe Scrum agregue valor ao negócio. Suas principais responsabilidades são:

- Definir claramente os requisitos do produto (*Backlog* do Produto);
- Priorizar os itens do *Backlog* do Produto;
- Garantir o valor do trabalho realizado pelo time de desenvolvimento;
- Garantir que o *Backlog* do Produto seja visível, transparente e claro para todos, apresentando o que o Equipe Scrum (*Scrum Team*) vai trabalhar a seguir; e
- Garantir que o time de desenvolvimento entenda os itens do *Backlog* do Produto.

b) *Scrum Master (SM)*: é responsável por garantir que o Scrum seja entendido e aplicado. Deve repassar o conhecimento sobre Scrum a todos os envolvidos no projeto, ajustando o método de modo que seja melhor adequado à cultura da empresa. Deve garantir que todos sigam as regras e práticas do Scrum. É responsável por remover os impedimentos do projeto. Suas ações incluem suporte ao *Product Owner*, à Equipe Scrum e à organização:

- *Scrum Master* deve apoiar o *Product Owner* para:

- i. Encontrar técnicas para o gerenciamento do efetivo do *Backlog* do Produto;
 - ii. Comunicar a visão, objetivo e itens do *Backlog* do Produto para a Equipe Scrum;
 - iii. Compreender e praticar a agilidade; e
 - iv. Facilitar os eventos Scrum conforme necessários.
- Scrum *Master* deve apoiar a Equipe Scrum para:
 - i. Treinar a Equipe Scrum em autogerenciamento e interdisciplinaridade;
 - ii. Orientar e liderar a Equipe Scrum na criação de produtos de alto valor;
 - iii. Remover impedimentos para o progresso da Equipe Scrum;
 - iv. Facilitar os eventos Scrum conforme necessários; e,
 - v. Treinar a Equipe Scrum em ambientes organizacionais nos quais o Scrum não é totalmente adotado e compreendido.
 - Scrum *Master* deve apoiar a organização para:
 - i. Liderar e treinar a organização na adoção do Scrum;
 - ii. Planejar as implementações Scrum dentro da organização;
 - iii. Ajudar funcionários e partes interessadas a compreender e tornar aplicável o Scrum;
 - iv. Implantar mudanças que aumentam a produtividade da Equipe Scrum; e,
 - v. Interagir com outros Scrum *Masters* para aumentar a eficácia da aplicação do Scrum nas organizações.
- c) Equipe Scrum (*Scrum Team*):** Representa a equipe responsável pelo desenvolvimento do projeto, geralmente composto de um grupo de quatro a nove integrantes, devendo analisar, desenvolver e testar cada um dos itens das *sprints*. Suas principais características são:
- Auto-organizados. Ninguém (nem mesmo o Scrum *Master*) diz a Equipe Scrum como transformar o *Backlog* do Produto em incrementos de funcionalidades potencialmente utilizáveis;

- Multifuncionais. Deve possuir todas as habilidades necessárias, enquanto equipe, para criar o incremento do produto;
- O Scrum não reconhece títulos para os integrantes da Equipe Scrum que não seja o Desenvolvedor, independentemente do trabalho que está sendo realizado pela pessoa;
- Individualmente os integrantes da Equipe Scrum podem ter habilidades especializadas e área de especialização, mas a responsabilidade da entrega das funcionalidades pertence à Equipe Scrum como um todo; e,
- Não contém subequipes dedicados a domínios específicos de conhecimento, tais como teste ou análise de negócios.

Conforme Schwaber (2004) o Scrum possui ciclo de vida composto por três fases:

- a) *Pre-game*:** Nesta fase se estabelece a visão do projeto e alocação de recursos para a sua execução. São criadas as versões iniciais do *Backlog* do Produto, o plano de instalação e a arquitetura de negócio. As equipes são formadas e mecanismos de comunicação e coordenação são definidos;
- b) *Game*:** Esta fase é contemplada por meio das *sprints*, nela ocorre o desenvolvimento iterativo e incremental dos itens do *Backlog* do Produto priorizados no *Pre-game*;
- c) *Post-game*:** Nesta fase ocorre a entrega do produto final ao cliente.

De acordo com Sutherland e Schwaber (2013) os eventos (ou cerimônias) do Scrum estão confinados nos *sprints* que é parte central do Scrum.

A *Sprint* é uma iteração, com tempo fixo definido, normalmente de 2-4 semanas e são compostas por uma reunião de planejamento, reuniões diárias, uma reunião para revisão e uma reunião para retrospectiva da *Sprint* (SUTHERLAND; SCHWABER, 2013). A seguir são detalhados os eventos Scrum:

- a) **Planejamento da *Sprint*:** Primeira reunião do projeto na qual todos precisam participar. Nessa reunião o *Product Owner* (PO) deve planejar e elaborar a lista de prioridades que devem ser cumpridas pelo projeto. Os itens do *Product BackLog* do Produto, selecionados para a *Sprint*, juntamente com o plano de entrega é chamado de *Sprint BackLog*. A reunião de Planejamento da *Sprint* deve responder as seguintes questões:
- O que pode ser entregue como resultado do incremento da próxima *Sprint* ?
 - Como o trabalho necessário (tarefas) para entregar o incremento será realizado ?
- b) **Reunião Diária:** Reunião rápida, geralmente de 15 minutos, realizada com objetivo de sincronizar as atividades e criar um plano para as próximas 24 horas. A Reunião Diária deve ser realizada com a Equipe Scrum. Os objetivos da Reunião Diária são facilitar a comunicação da Equipe Scrum, identificar e remover os impedimentos para o desenvolvimento, tomar decisões rápidas e melhorar o nível de conhecimento de todos os envolvidos.
- c) **Revisão da *Sprint*:** Reunião executada ao término de cada *sprint* com objetivo de avaliar o produto entregue, na qual a equipe deve apresentar o que foi realizado e responder as questões sobre o incremento. A Revisão da *Sprint* deve também validar a entrega do produto com as partes interessadas, garantindo a entrega de valor ao cliente com maior frequência. A Equipe Scrum deve ainda discutir sobre o andamento das atividades realizadas durante a *sprint*, analisar problemas enfrentados e como eles foram resolvidos. A reunião Revisão da *sprint* gera informações de extremo valor para as próximas versões do produto e reuniões de Planejamento de *Sprint*, ao iniciar novo ciclo de desenvolvimento (SCHWABER, 2004).
- d) **Retrospectiva da *Sprint*:** A reunião denominada Retrospectiva da *Sprint* tem o objetivo de avaliar a última *sprint* e criar um plano de ações de melhoria para a próxima. A Retrospectiva da *Sprint* deve participar a Equipe Scrum inclusive o *Scrum Master* e deve ocorrer depois da reunião de

Revisão da *Sprint* e antes da reunião de planejamento da próxima *sprint*. O Scrum Master deve garantir que a reunião ocorra e que todos da equipe entendam o objetivo, além, disso deve encorajar a Equipe Scrum a melhorar, revisando seu processo de desenvolvimento de forma a torná-lo mais produtivo para a próxima *sprint*. Neste momento deve ser verificado como ocorreu a interação entre as pessoas, processos e ferramentas durante a última *sprint*. Ao final dessa etapa a equipe deve ter identificado melhorias factíveis que deverão ser incorporadas na próxima *sprint* (SCHWABER, 2004).

A engrenagem do Scrum pode ser observada na Figura 2.

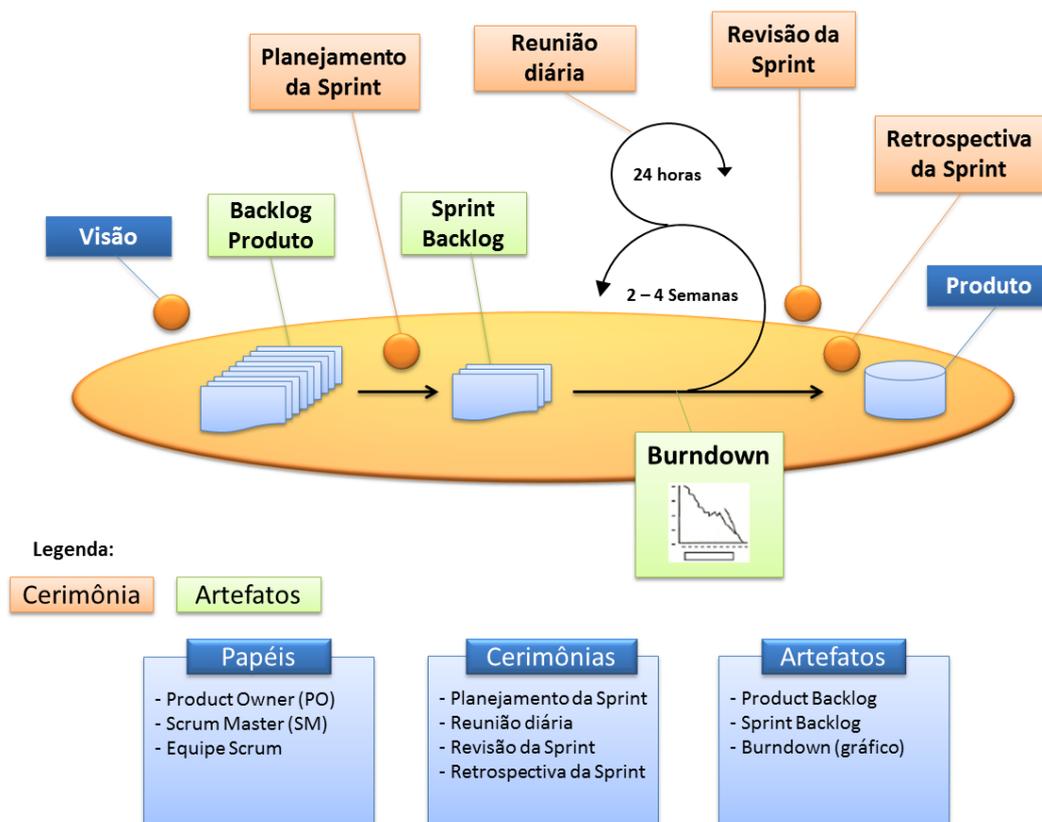


FIGURA 2 – A ENGRENAGEM DO SCRUM. FONTE: ETTINGER (2012)

Os artefatos propostos pela metodologia Scrum são produzidos em diferentes momentos durante a *sprint* e podem ser resumidos a três: *Backlog* do Produto, *Sprint BackLog* e *Burndown Chart* (SBROCCO; MACEDO, 2012; ETTINGER, 2012).

- a) **Backlog do Produto (Product Backlog):** Representa a visão do produto, “contendo uma lista ordenada de tudo o que deve ser necessário para o projeto” (SUTHERLAND; SCHWABER, 2013). Criado e mantido pelo *Product Owner* (PO) e gerado durante o “pre-game”, antes dos ciclos iniciais dos *sprints*.
- b) **Sprint BackLog:** Gerado na Reunião de Planejamento da *Sprint* e representa o conjunto de itens do *Backlog* do Produto que devem ser desenvolvidas durante a *sprint*. Durante a reunião as tarefas devem ser detalhadas e o esforço estimado. Ao longo da *sprint* a Equipe Scrum pode modificar o *Sprint Backlog*, a medida que ela aprende mais sobre o trabalho a ser realizado. Essas mudanças são apresentadas e entendidas na Reunião Diária e não devem alterar o plano definido para a *sprint* (SUTHERLAND; SCHWABER, 2013).
- c) **Gráfico Burndown:** Tem como objetivo apresentar o esforço restante para a conclusão da iteração e o acompanhamento da produtividade do time em relação ao planejado (SBROCCO; MACEDO, 2012). A Figura 3 demonstra um exemplo de Gráfico de *Burndown*.

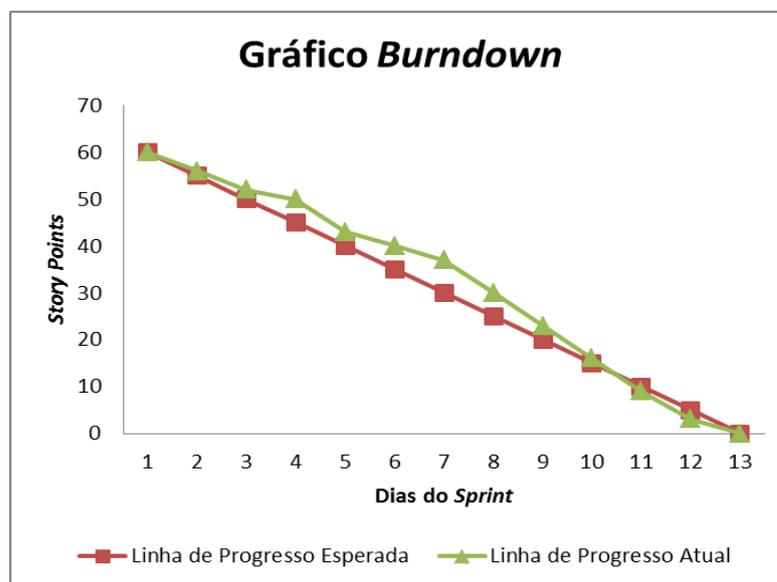


FIGURA 3 – EXEMPLO DE GRÁFICO BURNDOWN.
 FONTE: ADAPTADO DE NOOR ET AL. (2012).

Conforme observado na Figura 3 o eixo horizontal do Gráfico *Burndown* exibe o tempo e o eixo vertical mostra a quantidade de trabalho (ponto das histórias - *story points* ou *user points*, horas de trabalho, dias de equipe) restante (NOOR *et al.*, 2012). Portanto o Gráfico *Burndown* possibilita medir a velocidade, o progresso da equipe e avaliar os riscos do projeto (ETTINGER, 2012).

2.4.2 ROTEIRO DE IMPLANTAÇÃO SCRUM

O Quadro 7 apresenta um roteiro com 10 (dez) etapas para implantação de um projeto Scrum. De acordo com Sutherland (2014) esse resumo determina o “como” iniciar um projeto Scrum.

Etapas	Descrição
1. Escolher o Dono do Produto (<i>Product Owner - PO</i>)	Essa é a pessoa responsável pela “visão” do projeto. Ela deve levar em consideração os riscos e os benefícios, o que pode ser feito e o que desperta motivação na equipe.
2. Escolher a Equipe	Selecionar as pessoas que trabalharão no projeto. Essa equipe deve ter as habilidades necessárias para transformar a “visão” do PO em um produto. As equipes devem ser pequenas, entre quatro e nove integrantes.
3. Escolher o Scrum Master	Essa pessoa vai orientar o restante da equipe em relação à estrutura do Scrum e eliminar qualquer problema que possa deixar a equipe mais lenta.
4. Criar e Priorizar o Backlog do Produto	Criar uma lista detalhada de tudo o que precisar ser feito ou construído para transformar a “visão” do PO em realidade.
5. Aperfeiçoar e Estimar os requisitos do Backlog do Produto	As estimativas de esforço dos requisitos do <i>Backlog</i> do Produto devem ser realizadas pelas pessoas que irão realmente participar da construção dos itens.
6. Planejar o Sprint	Essa é a primeira reunião Scrum, deve ser feita com todos os envolvidos (Equipe, Scrum Master e PO). O objetivo é estimar quantas tarefas do <i>Backlog</i> do Produto pode ser realizada no <i>Sprint</i> .
7. Tornar o trabalho visível	Utilizar técnicas visuais para todos os envolvidos acompanharem o trabalho que está sendo realizado.
8. Realizar reuniões diárias	As reuniões diárias determinam o ritmo Scrum. Todos os dias, no mesmo horário a equipe Scrum deve se reunir para avaliar sua evolução.
9. Demonstrar o trabalho da Sprint	Essa reunião, chamada de Revisão da <i>Sprint</i> tem como objetivo apresentar o que a equipe conseguiu fazer durante o <i>Sprint</i> .
10. Avaliar a Sprint	Essa reunião, conhecida como Retrospectiva da <i>Sprint</i> , tem como objetivo aprimorar o processo Scrum, avaliando o que poder ser melhorado. Esse aprimoramento também é conhecido com <i>Kaizen</i> .

QUADRO 7 – ROTEIRO IMPLANTAÇÃO PROJETO SCRUM. FONTE: SUTHERLAND (2014)

O Backlog do Produto (etapa 4) é a origem única dos requisitos ou pendências do produto, porém podem ser alterados e evoluídos durante o desenvolvimento do projeto. Esses requisitos são responsabilidade do *Product Owner* e ele deve definir sua prioridade (SUTHERLAND, 2014).

As estimativas de esforço (etapa 5) devem ser realistas e realizadas pelas pessoas envolvidas na construção dos requisitos do *Backlog* do Produto. A equipe deve analisar se é possível ser construído, se existem informações suficientes para entender, estimar e se gera valor para o cliente. De acordo com Sutherland (2014) as estimativas não devem ser realizadas em horas, mas utilizando uma classificação relativa ao tamanho, como pequeno, médio ou grande. Ou, usando a sequência de Fibonacci¹, realizando as estimativas por pontos (1, 2, 3, 5, 8, 13, 21, etc.).

O planejamento da *Sprint* (etapa 6) deve definir quantas tarefas podem ser realizadas partindo do topo da lista (*backlog*) priorizada pelo PO. As *Sprints* não devem durar mais que um mês, normalmente 1 (uma) ou 2 (duas) semanas e a duração deve ser sempre a mesma. A equipe deve “pegar” as tarefas que totalizem o mesmo número de pontos da *Sprint* anterior, totalmente entregue. Esse número é conhecido como “Velocidade da Equipe”. De acordo com Sutherland (2014), a equipe deve tentar aumentar o número de pontos a cada *Sprint*. O resultado da reunião é a lista de tarefas a serem executadas e o entendimento de como essas tarefas irão atender à visão do projeto e o objetivo da *Sprint*. A lista de tarefas não pode ser alterada durante o *Sprint* e a equipe deve trabalhar de forma autônoma para concluir o que foi previsto.

De acordo com Sutherland (2014), a melhor forma de tornar o trabalho visível no Scrum (etapa 7) é criar o Quadro Scrum (*Task Board*), com três colunas: A fazer, Fazendo/Executando e Feito. Nesse quadro as tarefas são representadas por *post-its* que são movimentados a medida que são concluídos. A Figura 4 apresenta um exemplo de Quadro Scrum. Outra forma de tornar o trabalho visível é criar o Gráfico *Burndown* (vide figura 3).

¹ É uma sequência de números inteiros, na qual a soma dos números inteiros corresponde a soma dos dois anteriores. Recebeu o nome do matemático Leonardo de Pisa, conhecido como Fibonacci.



FIGURA 4 – EXEMPLO DE QUADRO SCRUM.
 FONTE: ADAPTADO DE SUTHERLAND (2014).

Na Figura 4 podem-se observar os *post-its* que representam as tarefas a serem executadas no *Sprint*. Na coluna “A fazer” estão as tarefas que ainda não foram iniciadas, na coluna “Fazendo” estão as tarefas que estão sendo executadas no momento, a coluna “Testes” é opcional no Quadro Scrum, nela estão as tarefas que necessitam verificação e na coluna “Feito” estão todas as tarefas prontas. Durante uma *Sprint* as tarefas devem ser movimentadas em direção a coluna “Feito” que deve conter todas as tarefas no final da *Sprint*. O responsável pela tarefa deve movimentar no Quadro Scrum. O Quadro Scrum deve estar visível e disponível para toda equipe acompanhar a evolução das tarefas. O Quadro Scrum pode ser um quadro físico (*post-its*) ou utilizando ferramentas computacionais específicas.

As reuniões diárias (etapa 8) devem ser executadas todos os dias no mesmo horário com duração definida, não ultrapassando 15 (quinze) minutos, nas quais todos da equipe devem responder às perguntas:

- O que você fez ontem que ajudou a Equipe Scrum a atender a meta da *Sprint* ?
- O que você fará hoje para ajudar a Equipe Scrum a atender a meta da *Sprint* ?
- Existe algum obstáculo que impeça a você ou a Equipe Scrum no atendimento da meta da *Sprint* ?

O *Scrum Master* é responsável por resolver qualquer obstáculo ou impedimento para o progresso da equipe.

A reunião de revisão ou demonstração da *Sprint* (etapa 9) deve ser realizada no final da *Sprint*, nesse momento os resultados são apresentados. Qualquer pessoa pode participar: PO, *Scrum Master*, Equipe e outros envolvidos como o cliente. De acordo com Sutherland (2014), a equipe deve apresentar o que está totalmente concluído e pode ser entregue sem qualquer trabalho adicional.

2.4.3 APLICAÇÃO DA METODOLOGIA SCRUM

Esta seção apresenta uma análise de trabalhos científicos relacionados ao Scrum, com objetivo de comprovar sua relevância e avaliar sua aplicação em projetos de *software*.

O artigo apresentado por Rising e Janoff (2000), apresenta a experiência com aplicação do Scrum na empresa AG Communication Systems. Na empresa as equipes de desenvolvimento de sistemas variavam de apenas 2 (dois) integrantes até algumas centenas de integrantes. Metodologias que funcionavam bem com grandes equipes não funcionavam bem em pequenas equipes. Diante disso, buscou-se uma abordagem que se adaptasse para equipes com até 10 (dez) integrantes e o Scrum foi adotado em 3 projetos piloto. O relato das experiências evidenciou um aumento no “voluntarismo”, rapidez para tomada de decisões e as reuniões permitiram vantagens com a troca de experiências da equipe, melhorando a competência coletiva.

De acordo com Rising e Janoff (2000), o Scrum é apropriado para projetos em que os requisitos não estão claramente definidos no início do projeto e quando condições caóticas são esperadas ao longo do ciclo de vida de desenvolvimento do produto.

O trabalho desenvolvido por Mann e Maurer (2005), apresenta um estudo de caso sobre o impacto da implantação do Scrum em relação à satisfação do cliente e as atividades de horas extras realizadas pela equipe de desenvolvimento de *software*. O objetivo do projeto de *software* era o desenvolvimento de um *Web Site* para um indústria de óleo e gás, na cidade de Calgary / Canadá. O estudo de caso apresenta o resultado de 2 (dois) anos de trabalho, analisando dados que antecederam a implantação do Scrum e sucederam a implantação do Scrum. Foi utilizada análise quantitativa para avaliar o volume de horas extras utilizadas pela equipe de desenvolvimento para entregar os projetos no prazo combinado com o cliente, antes da implantação do Scrum e após a implantação do Scrum, observou-se uma redução de 19% para 7% em média de tempo gasto com horas extras após a implantação do Scrum. Os resultados da análise qualitativa avaliou a satisfação do cliente e da equipe de desenvolvimento com a adoção do Scrum. Os clientes ficaram muito satisfeitos com a adoção do Scrum e as mudanças que foram introduzidas, pois proporcionou maior envolvimento do cliente e entendimento das ações do projeto, facilitando a comunicação. A equipe de desenvolvimento relatou que o Scrum foi útil, pois ficou mais claro quais os requisitos do projeto que a equipe deveria trabalhar e quando entregar. Uma lição importante relatada é que os clientes também devem ser treinados no Scrum, pois nas primeiras *sprints* eles ficaram confusos sobre como seria sua atuação no projeto.

Hossain (2009) apresenta uma revisão sistemática de literatura sobre o uso do Scrum em Desenvolvimento de *Software* Global (GSD), que consiste em equipes de desenvolvimento distribuídas em várias partes do mundo. O artigo analisou 20 trabalhos científicos e observou um crescimento de publicações acerca do assunto. Nos anos de 2003 a 2005 apenas 1 (um) artigo publicado

por ano, nos anos de 2006 a 2009 uma média de 5 (cinco) artigos publicados por ano. Esse trabalho ainda revelou uma necessidade de adaptações do Scrum em projetos GSD, embora evidenciou que as práticas Scrum podem ser usadas em projetos distribuídos, mesmo com vários números de equipes. Os principais desafios para uso do Scrum em projetos GSD são a distribuição dos projetos, comunicação, colaboração e coordenação.

O trabalho de Gannon (2013) apresenta um estudo de caso da aplicação da metodologia Scrum para desenvolvimento de um projeto para Johns Hopkins University, aplicado ao laboratório de Física. O objetivo do *software* era controlar remotamente dispositivos de um satélite a partir de uma estação tripulada. A equipe de projeto foi composta por 19 envolvidos, incluindo o *Product Owner* e o *Scrum Master*. O Backlog do Produto iniciou com 53 histórias de usuário e 2200 *story points*. Foi utilizado *sprints* de 30 dias. Como ferramenta de apoio a equipe adotou o uso do Quadro *Kanban*. As métricas utilizadas foram capacidade, que corresponde ao total de horas de dedicação para o projeto; e velocidade, que corresponde ao total de *story points* consumidas em cada *sprint*. Todos os eventos do Scrum foram realizados rigorosamente, liderados pelo *Scrum Master*.

Segundo Gannon (2013), as principais dificuldades encontradas estavam relacionadas à disponibilidade da equipe, pois nem todos estavam dedicados exclusivamente ao projeto, além disso, nos primeiros *sprints* a dificuldade para estimar o esforço. Porém, a implantação do Scrum para esse projeto foi considerado um sucesso, com entregas sucessivas de produtos de *software* e melhoria da qualidade, devido principalmente ao esforço coletivo, aumento de produtividade e capacitação da equipe.

2.5 METODOLOGIAS ÁGEIS, *LEAN SOFTWARE DEVELOPMENT (LSD)* E SCRUM

Esta seção apresenta uma análise de trabalhos científicos relacionados à integração e aplicação de Metodologias Ágeis, *Lean Software Development (LSD)* e Scrum. O objetivo é identificar casos que utilizaram a integração das abordagens.

O LSD pode ser integrado com metodologias ágeis de desenvolvimento, como o Scrum. O Scrum é uma excelente maneira de introduzir o conceito *Lean* de “fluxo” para desenvolvimento de *software*, porém não é uma metodologia completa e carece de práticas técnicas (POPPENDIECK; CUSUMANO, 2012).

De acordo com Poppendieck e Cusumano (2012), metodologias ágeis são boas práticas para organizar o desenvolvimento do *software*, porém são insuficientes para a construção do projeto da solução do produto, pois as atividades de projeto e arquitetura do sistema ocorrem fora da equipe de desenvolvimento do produto. LSD estabelece um conjunto de princípios que envolvem todo o ciclo de vida produto, com uma abordagem interfuncional, dessa forma é mais indicado para orientar a combinação de projeto, desenvolvimento, implantação e validação em ciclo contínuo de descoberta e fornecimento de valor.

De acordo do Starr (2013), usar o *Lean Thinking* com Scrum produz retornos elevados e é uma ótima maneira de manter uma cultura *Kaizen*. Equipes Scrum ainda estão aprendendo como aplicar *Lean* para Scrum, mas muitas práticas estão ganhando popularidade, pois tem provado fazer equipes Scrum mais eficazes.

O trabalho de Starr (2013) apresenta uma extensão da metodologia Scrum aplicando práticas *Lean* e *Kanban*. A aplicação das práticas consiste em 5 (cinco) estudos de caso que analisam a limitação da equipe trabalho durante o processo de desenvolvimento de *software*; o tempo de espera entre os processos e dependências; o adiamento do comprometimento; a construção de códigos de qualidade no início do processo de desenvolvimento, impactando em redução de defeitos; e a aplicação do *Kanban* para visualização do fluxo de

trabalho. Todos os estudos de caso apresentaram resultados positivos e redução do desperdício.

De acordo com Wang, Conboy e Cawley (2012), técnicas híbridas são utilizadas para desenvolvimento de *software*, e as mais comuns na última década é a integração de XP e Scrum. Porém, relata que o foco está mudando, e abordagem *Lean* para desenvolvimento de *software* é uma área de estudo emergente e é considerada “a próxima onda de processo de *software*”. Apresenta ainda estudos de caso, como a empresa Capital One, uma das 500 maiores empresas financeiras, que utilizaram os princípios do LSD, integrados com Scrum, para melhorar a qualidade do produto, ampliar o conhecimento da equipe e entregar mais rápido.

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Conforme apresentado na fundamentação teórica, as abordagens Ágil e *Lean Thinking* (Pensamento Enxuto) são utilizadas atualmente e trazem benefícios para várias áreas, incluindo a área de desenvolvimento de *software*. Analisando a origem das abordagens observa-se uma familiaridade dos princípios que as fundamentam.

O *Lean Thinking* tem origem nas indústrias automobilísticas japonesas e consiste na aplicação de práticas visando eliminar desperdícios do sistema produtivo, desenvolver produtos alinhados às necessidades dos clientes, com melhor qualidade e custos mais baixos.

Para a área de aplicação de desenvolvimento de *software* o *Lean Thinking* é conhecido como *Lean Software Development* (LSD).

A abordagem ágil surgiu com o objetivo de entregar produtos mais rapidamente, baseando em uma abordagem iterativa e incremental para desenvolvimento e entrega do produto, permitindo assim melhor adaptação às mudanças.

Trabalhos científicos tais como Perera e Fernando (2007), Petersen e Wohlin (2010) e Wang, Conboy e Cawley (2012), evidenciam a tendência da utilização Lean Thinking para melhoria do processo ágil.

A metodologia ágil mais conhecida e utilizada atualmente, conforme pesquisa de Forrester, publicado em 2011, é a metodologia Scrum.

Trabalhos científicos tais como Rising e Janoff (2000), Mann e Maurer (2005), e Gannon (2013), evidenciam bons resultados com a implantação da metodologia Scrum, melhorando a produtividade, a qualidade dos produtos, o aprendizado coletivo e a satisfação do cliente.

Além disso, outros trabalhos como Starr (2013) e Wang, Conboy e Cawley (2012) e Ebert *et al.* (2012) discutem a integração entre o LSD e o Scrum e destacam que iniciativas nesse sentido e estudos mais aprofundados são necessários.

A Figura 5 (cinco) e o Quadro 8 (oito) apresenta uma linha do tempo, identificando o histórico da revisão de literatura pesquisada:

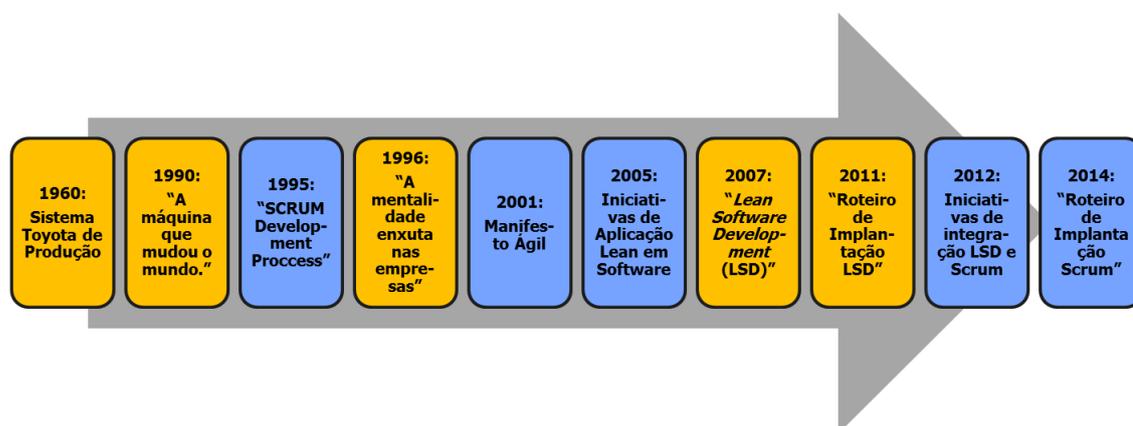


FIGURA 5 – LINHA DO TEMPO: REVISÃO DA LITERATURA.
FONTE: ELABORADO PELO AUTOR.

Ano	Descrição
1960	Consolidação do Sistema Toyota de Produção, principais princípios: Eliminar Desperdício; Produção Puxada; <i>Just-in-Time</i> ; <i>Kanban</i> ; Respeito e envolvimento dos funcionários.
1990	Publicação da obra “A máquina que mudou o mundo” de Womack e Jones: Consolidação do termo “ <i>Lean Thinking</i> ”; Comparação Produção em Massa x Produção Enxuta.
1995	Publicação “ <i>Scrum Development Proccess</i> ”; Principais autores: Jeff Sutherland e Ken Schwaber; Características: Flexibilidade dos resultados; Flexibilidade dos prazos; Times pequenos; Revisões frequentes; Colaboração.
1996	Publicação da obra “ <i>A mentalidade enxuta nas empresas</i> ” de Womack e Jones. Definição dos princípios “ <i>Lean Thinking</i> ”: Determinar Valor; Determinar o Fluxo de Valor; Fluxo; Sistema Puxado; Perfeição; Aplicação <i>Lean Thinking</i> em diversas áreas: Saúde, Logística e Tecnologia da Informação.
2001	Publicação do Manifesto Ágil: Acordo de 17 profissionais que aplicavam “Métodos Leves”; 4 valores e 12 princípios; Valorização da equipe, Proximidade com cliente, Flexibilidade às mudanças, Rapidez na entrega dos produtos.
2005	Publicações de iniciativas de aplicação <i>Lean Thinking</i> em <i>Software</i> : Middleton, Flaxel e Cookson (2005); Perera e Fernando (2007); Petersen e Wohlin (2010).
2007	Publicação “ <i>Lean Software Development - LSD</i> ” de Poppendieck e Poppendieck.
2011	Publicação da obra “Implementando o Desenvolvimento <i>Lean</i> de <i>Software</i> ” de Poppendieck e Poppendieck; Roteiro de Implantação LSD: 7 princípios / 21 práticas para desenvolvimento de <i>software</i> .
2012	Publicações de iniciativas de integração LSD e Scrum: Wang, Conboy e Cawley (2012); Starr (2013).
2014	Publicação da obra “Scrum: a arte de fazer o dobro do trabalho na metade do tempo” de Jeff Sutherland; Roteiro de Implantação Scrum: 10 etapas.

QUADRO 8 – LINHA DO TEMPO: REVISÃO DA LITERATURA.
 FONTE: ELABORADO PELO AUTOR.

3 PROCEDIMENTOS METODOLÓGICOS

Nesse capítulo será descrito o Projeto Metodológico da Pesquisa, quanto aos aspectos da abordagem, objetivos, método de pesquisa, procedimentos técnicos (visão geral, classificação de revisão bibliográfica, técnica de coleta de dados), técnicas de análise dos dados, planejamento da pesquisa-ação e também a caracterização das empresas investigadas.

3.1 ASPECTOS METODOLÓGICOS

A pesquisa está orientada, pelo que relata Gil (2009) quanto ao plano geral, como sendo um processo formal e sistemático de desenvolvimento do método científico, pois para o conhecimento ser considerado como científico é necessário identificar (e declarar em registros adequados) as operações mentais e técnicas que vão possibilitar uma futura verificação. Mesmo existindo uma diversidade de procedimentos de pesquisa, existe um consenso nos processos adotados tais como: planejamento, coleta de dados, análise e interpretação e redação do relatório.

Para tanto, a partir deste ponto, é apresentado um detalhamento das etapas que foram desenvolvidas na execução desse Projeto Metodológico da Pesquisa.

A **abordagem geral** dessa pesquisa, pelo que apresenta Martins (2010) foi classificada como QUALITATIVA, pois apresenta um delineamento do contexto do ambiente da pesquisa, é uma abordagem não muito estruturada, tem múltiplas fontes de evidências e destaca a importância da concepção da realidade organizacional. De acordo com Turrioni e Mello (2012), o ambiente natural é a fonte direta para a coleta de dados e o pesquisador é o instrumento-chave, buscando compreender os fenômenos, observando-os e descrevendo-os.

A pesquisa **quanto aos objetivos** é classificada como EXPLORATÓRIA, porque proporciona maior familiaridade com o problema, aprimorando ideias ou a descoberta/confirmação de intuições. Além disso, este tipo de método tem um planejamento flexível, de modo a considerar os mais variados aspectos relativos ao fato estudado. De acordo com GIL (1999), as pesquisas exploratórias têm como principal finalidade desenvolver, esclarecer e modificar conceitos e ideias para formular novos problemas ou hipóteses para serem estudados em novas pesquisas.

O **método de pesquisa** adotado é classificado, a partir de Nakano (2010), Miguel (2007) e Demo (1995), como sendo TEÓRICO-EMPÍRICO, parte-se de discussões conceituais a partir da literatura, revisões bibliográficas e modelagens conceituais para codificar a face mensurável da realidade social.

Quanto ao **procedimento técnico** será utilizado a PESQUISA BIBLIOGRÁFICA e PESQUISA-AÇÃO. Em relação à pesquisa bibliográfica, Gil (2009) menciona como o elemento mais importante para a identificação de um delineamento. Foi aplicado o procedimento PESQUISA-AÇÃO, de acordo com Thiollent (2000), é um tipo de pesquisa concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo no qual os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo e participativo.

Quanto à **técnica de coleta de dados**, nos moldes descritos por Gil (2011), utilizou-se: LEVANTAMENTO BIBLIOGRÁFICO, ENTREVISTAS COLETIVAS E OBSERVAÇÃO PARTICIPANTE. O levantamento bibliográfico já foi justificado anteriormente; quanto às entrevistas, esse mesmo autor apresenta uma caracterização de que ela deve ser feita com um roteiro previamente definido e que, este roteiro poderá conter perguntas estruturadas ou questões abertas; e quanto à observação, é definido como o uso dos sentidos com vistas a adquirir os conhecimentos necessários para o cotidiano.

Quanto à **técnica de análise dos dados**, para abordagem qualitativa desenvolvida pela pesquisa-ação, uma das etapas de seu ciclo de execução

prevê a análise comparativa entre os dados tabulados com a teoria envolvida no tema pesquisado. A etapa final dessa análise dos dados deve gerar um plano de ação com recomendações para a solução dos problemas, responsáveis e prazo de implantação (MELLO *et al.*, 2012).

A tabulação dos dados é o processo de agrupar e contar as situações que foram desenvolvidas ao longo da aplicação do Modelo (GIL, 2011). Portanto, para a apresentação dos resultados dessa pesquisa foram utilizados o quadro resumo da pesquisa-ação e o relatório de pesquisa.

3.2 PESQUISA BIBLIOGRÁFICA

Pesquisa Bibliográfica é desenvolvida a partir de um material já elaborado, constituído principalmente de livros e artigos científicos, nos estudos exploratórios. É comum a utilização dessa estratégia de pesquisa por permitir que o pesquisador ganhe mais cobertura do que faria pesquisando diretamente (GIL, 2011).

Para elaboração desse trabalho, realizou-se uma Revisão Sistemática de Literatura (RSL). Uma RSL é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma determinada questão, ou tópico de uma área, ou fenômeno de interesse (KITCHENHAM; CHARTERS, 2007; TRANFIELD; DENYER; SMART, 2003).

Esse projeto foi elaborado como principais fontes, as publicações científicas disponíveis nas bases de dados via Portal de periódicos da CAPES, com destaque para as bases de dados Web of Science (WoS) e SCOPUS.

O período de pesquisa considerou publicações entre 2004 a 2015, focando nas palavras chaves: *Lean Thinking*, *Scrum*, *Software*, *Metodologias Ágeis* e *Lean Software Development* (LSD). Após seleção dos artigos relevantes, a base de análise resultou 40 publicações diferentes. Todos os trabalhos selecionados para discussão abordaram diretamente a aplicação de *Lean* em processos de produção de *software*, apoiando o gerenciamento de projetos, reduzindo

prazos de entrega, melhorando a qualidade, o fluxo de processos e a interação entre as pessoas, facilitando assim a comunicação e o aprendizado (PERUCCI; CAMPOS, 2015).

3.3 PESQUISA-AÇÃO

Pesquisa-Ação é a metodologia de pesquisa com base empírica e que está associado com uma ação ou com a resolução de um problema coletivo, em que o pesquisador está envolvido na pesquisa de modo cooperativo ou participativo. O termo pesquisa se refere à produção do conhecimento e o termo ação, a uma modificação intencional de uma dada realidade (THIOLLENT, 2003; MELLO *et al.*, 2012).

O objetivo da pesquisa-ação é aumentar a base de conhecimento científico, obtendo informações que seriam de difícil acesso por outros procedimentos; e resolver um problema prático, com o levantamento de soluções e proposta de ações para auxiliar o pesquisador na sua atividade transformadora da situação (THIOLLENT, 2003; MELLO *et al.*, 2012).

A pesquisa-ação nas organizações ajuda os autores tornando-os capazes de identificar seus problemas e encontrar possíveis soluções, com a ajuda de seus funcionários (FERRANCE, 2000; THIOLLENT, 2009; COUGHLAN; COGHLAN, 2002; GIL, 2011). A definição de problema segundo Gil (2011) na acepção científica é uma questão não resolvida, que é o objeto de discussão, em qualquer domínio do conhecimento. A definição de organização segundo Thiollent (2009) é qualquer entidade que agregue grupos sociais cujas atividades são estruturadas em processos com objetivo definido.

De acordo com Coughlan e Brannick (2008), a pesquisa-ação é apropriada quando a questão de pesquisa está relacionada e descreve o desdobramento de uma sequência de ações ao longo do tempo em dado grupo, comunidade ou organização; para explicar como e porque a ação de um membro do grupo

pode mudar ou melhorar o trabalho executado; e para entender e aprender com o processo de mudança ou de melhoria.

A pesquisa-ação pode ser vista de duas maneiras: uma sequência de eventos; e uma abordagem para a resolução de problemas. Uma sequência de eventos é constituída por ciclos iterativos que tem os processos divididos em: coleta de dados, realimentação desses dados para os interessados, análise dos dados, planejamento das ações, tomada de ação e avaliação da ação, levando a nova coleta de dados. Uma abordagem para resolução de problema é a aplicação de uma abordagem científica para a localização dos fatos que requererem uma ação na busca de solução, em que o pesquisador é membro do grupo (COUGHLAN; COGHLAN, 2002; TURRIONI; MELLO, 2012).

Mello *et al.* (2012) desenvolveu a estrutura para o planejamento da pesquisa-ação agregando no planejamento da pesquisa a definição da estrutura conceitual como parte da pesquisa-ação. O planejamento de Mello *et al.* (2012) é composto por sete passos, conforme ilustrado na Figura 6.

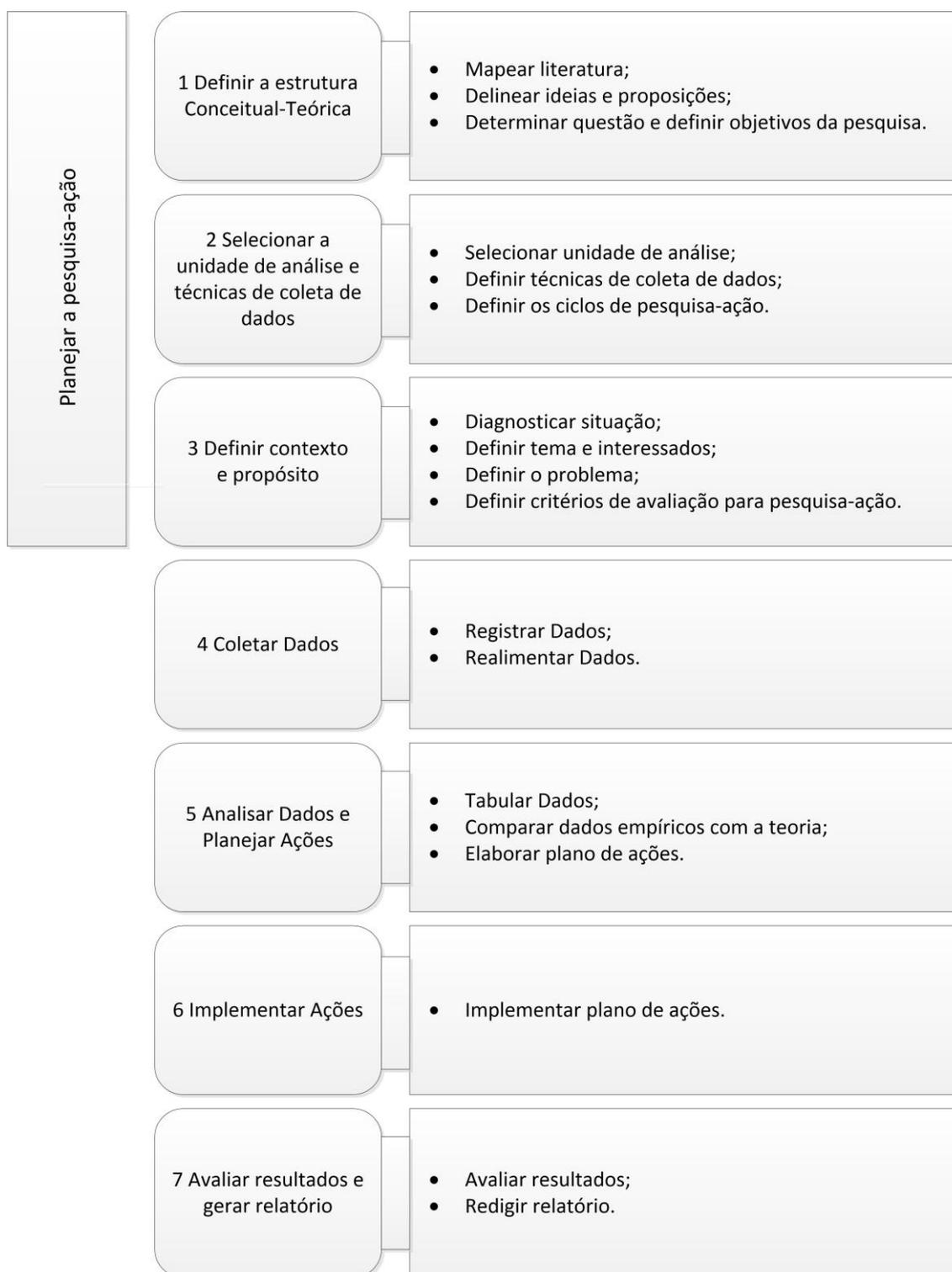


FIGURA 6 – ESTRUTURA DA PESQUISA-AÇÃO.
FONTE: ADAPTADO DE MELLO ET AL. (2012).

A Figura 6 detalha o planejamento da pesquisa-ação e descrevem os sete passos sendo eles:

- Definir a estrutura conceitual e teórica: procura mapear a pesquisa bibliográfica por meio da determinação do objetivo da pesquisa;
- Selecionar a unidade de análise e técnicas de coleta de dados: define a organização a ser pesquisada, bem como as técnicas de coleta de dados, como exemplo: entrevistas e observação;
- Definir contexto e propósito: pretende diagnosticar o ambiente a ser pesquisado, definindo o tema de interesse e delimitando os critérios de avaliação para o problema;
- Coletar dados: acontece de diferentes formas dependendo do objetivo, podendo ser utilizados relatórios, estatística operacional, observação, discussões e entrevistas;
- Analisar dados e planejar ações: deve ser feita junto com os membros da pesquisa-ação;
- Implementar ações: deve ocorrer conforme o planejamento;
- Avaliar resultados e gerar relatório: envolve uma reflexão dos resultados obtidos, essa reflexão é a chave do aprendizado.

Os objetivos do método de pesquisa-ação para este projeto de pesquisa são:

- Avaliar a proposta do MALS; e
- Avaliar os resultados com a aplicação do MALS.

O Quadro 9 apresenta o detalhamento do Planejamento da Pesquisa-Ação aplicando a proposta de condução baseado em Mello *et al.* (2012).

<p>Passo 1. Definir a estrutura conceitual-teórica</p> <ul style="list-style-type: none"> • Mapear literatura: O mapeamento da literatura é apresentado na seção 3.2. • Delinear ideias e proposições: Proposta do Método de Aplicação <i>Lean Thinking</i> em Projetos Scrum (MALS). • Determinar questão e definir objetivos da pesquisa: A questão de pesquisa e o detalhamento dos objetivos são apresentados nas seções 1.2 e 1.3.
<p>Passo 2. Selecionar a unidade de análise e técnicas de coleta de dados</p> <ul style="list-style-type: none"> • Selecionar unidade de análise: Para seleção da unidade de análise desse projeto foram definidos os seguintes critérios: <ul style="list-style-type: none"> ○ Empresa: a) Empresa deve possuir equipes que desenvolvem <i>softwares</i>, seja para consumo próprio ou para terceiros, como fábrica de <i>software</i>; b) Aceitação e apoio pela alta administração da empresa. ○ Equipe: a) Disponibilidade para o projeto de no mínimo 1 (uma) equipe, com mínimo de 4 (quatro) e máximo 9 (nove) integrantes; b) Disponibilidade de um membro da equipe, para que além das tarefas cotidianas possa liderar o processo de implantação do MALS. • Definir técnicas de coleta de dados: A coleta de dados para avaliar a proposta do MALS, deve ocorrer antes de iniciar os ciclos de pesquisa e durante as reuniões da retrospectiva da <i>Sprint</i>, por meio de entrevista coletiva, aplicando o questionário: <ul style="list-style-type: none"> ○ A equipe compreende as atividades do MALS ? ○ A equipe compreende as fases do MALS (Scrum/LSD) ? ○ A equipe identifica algum impedimento para implantação do MALS ? ○ A equipe avalia oportunidade de melhorias no MALS ? <p>A coleta de dados para avaliar a aplicação do MALS deve ser realizada na reunião da retrospectiva da <i>Sprint</i>, por meio de entrevista coletiva, discussões e observações documentais.</p> <p>Nas entrevistas coletivas devem ser efetuadas as seguintes perguntas específicas:</p> <ul style="list-style-type: none"> ○ O que deu certo na última <i>Sprint</i> ? ○ O que poderia melhorar para a próxima <i>Sprint</i> ? <p>As observações documentais devem ser realizadas utilizando os gráficos <i>Burndown</i> (vide seção 2.4.1), Quadro Scrum (vide seção 2.4.2) e quando aplicável, a ferramenta de gestão de configuração utilizada pela empresa.</p> • Definir os ciclos de pesquisa-ação: Os ciclos de pesquisa devem ocorrer de acordo com os ciclos da metodologia Scrum, definidos para cada unidade de análise.
<p>Passo 3. Definir Contexto e Propósito</p> <ul style="list-style-type: none"> • Diagnosticar Situação: O diagnóstico da situação problema e análise de relevância da pesquisa são apresentados na seção 1.1. • Definir tema e interessados: Pesquisadores da comunidade acadêmica e empresas que produzem software e buscam maior agilidade, qualidade e produtividade. • Definir o problema: O problema técnico a ser solucionado é validar um método para o processo de desenvolvimento de <i>software</i> e avaliar os resultados com sua aplicação. • Definir critérios de avaliação para pesquisa-ação: <ul style="list-style-type: none"> ○ Os critérios para avaliar a proposta do MALS deve utilizar o questionário definido na técnica de coleta de dados; ○ Os critérios para avaliar a aplicação do MALS devem ser definidos pela equipe em cada ciclo do processo de pesquisa.

(CONTINUA)

(CONTINUAÇÃO)

Passo 4. Coletar Dados
<ul style="list-style-type: none"> • Registrar e Realimentar Dados: Os dados da avaliação e aplicação do MALS devem ser registrados no Quadro-Resumo para Pesquisa-ação (Quadro 10).
Passo 5. Analisar dados e Planejar Ações
<ul style="list-style-type: none"> • Tabular Dados: Os dados da proposta e da aplicação do MALS devem ser tabulados e registrados no Quadro-Resumo para Pesquisa-ação (Quadro 10); • Comparar os dados empíricos com a teoria: Os dados coletados devem ser comparados com a teoria definida no MALS; • Elaborar Plano de Ações: A reunião de retrospectiva da <i>Sprint</i> deve gerar um Plano de Ação para ser implementado pela equipe e o pesquisador. O Plano de Ação deve ser registrado no Quadro-Resumo para Pesquisa-ação (Quadro 10).
Passo 6. Implementar Ações
<ul style="list-style-type: none"> • Implementar Plano de Ações: <ul style="list-style-type: none"> ○ As ações de melhoria da proposta do MALS devem ser implementadas pelo pesquisador e avaliadas no próximo ciclo de pesquisa. ○ As ações de melhoria da aplicação do MALS devem ser implementadas pela equipe. O acompanhamento das ações deve ser realizado nas reuniões diárias definidas no Scrum e avaliado no próximo ciclo de pesquisa.
Passo 7. Avaliar Resultados e Gerar Relatório
<ul style="list-style-type: none"> • Avaliar resultados: <ul style="list-style-type: none"> ○ Os resultados das ações de melhoria da proposta do MALS devem ser avaliados pelo pesquisador. ○ Os resultados das ações de melhoria da aplicação do MALS devem ser avaliados de acordo com os critérios definidos na fase de coleta de dados; Essa avaliação deve ser realizada na reunião de retrospectiva da <i>Sprint</i> com a equipe envolvida e a alta administração da empresa. • Redigir relatório: <ul style="list-style-type: none"> ○ O relatório da pesquisa deve registrar a fundamentação teórica, baseado nos desdobramentos de cada ciclo de pesquisa; ○ A narrativa da pesquisa deve apresentar uma síntese da coleta de dados e monitoramento dos ciclos da Pesquisa-Ação.

QUADRO 9 (CONT.) – PLANEJAMENTO DA PESQUISA-AÇÃO.

FONTE: FONTE: ELABORADO PELO AUTOR.

De acordo com Coughlan e Coughlan (2002), o monitoramento é uma metafase que deve ocorrer em todos os ciclos da pesquisa. Os pesquisadores podem encontrar dificuldades para realizar esse monitoramento, devido a natureza cíclica da pesquisa-ação. Para superar essa dificuldade, Mello *et al.* (2012) propõe a elaboração de um quadro resumo, sendo que cada ciclo da pesquisa-ação é descrito resumidamente. O Quadro 10 apresenta um modelo de quadro resumo para pesquisa-ação.

Ciclo	Período	Fase da pesquisa	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
...
...

*QUADRO 10 – QUADRO-RESUMO PARA PESQUISA-AÇÃO.
 FONTE: ADAPTADO DE MELLO ET AL. (2012).*

O Quadro 10 apresenta os campos: Ciclo, identificar o ciclo da pesquisa; Período, data inicial e final do ciclo; Fase da Pesquisa, identificar a fase da pesquisa conforme modelo de Mello *et al.* (2012), conforme Figura 6; Critérios de avaliação, descrever os critérios de avaliação identificados no ciclo; Atividades executadas, descrever as atividades executadas no ciclo; Avaliação, descrever como foram realizadas as avaliações do ciclo e critérios definidos; Melhoria e aprendizagem, descrever o plano de ações, melhorias e lições aprendidas no ciclo.

4 MÉTODO DE APLICAÇÃO *LEAN THINKING* EM PROJETOS SCRUM (MALS)

Neste capítulo é apresentado na seção 4.1 o modelo teórico conceitual do MALS e na seção 4.2 o modelo consolidado e detalhado do MALS, relatando os resultados com aplicação nas situações reais.

4.1 MODELO TEÓRICO CONCEITUAL DO MÉTODO DE APLICAÇÃO *LEAN THINKING* EM PROJETOS SCRUM (MALS)

A revisão bibliográfica identificou trabalhos que relatam benefícios com a integração das abordagens *Lean* e Scrum (vide seção 2.5), porém não foi identificado na literatura um método para orientar o processo de implantação e integração dessas abordagens no contexto de desenvolvimento de *software*.

Portanto, propõe-se um método para orientar o processo de implantação e integração *Lean* e Scrum, utilizando como base o roteiro de implantação do Scrum de Sutherland (2014) e o roteiro de implantação do *Lean Software Development* (LSD) de Poppendieck e Poppendieck (2011).

A Figura 7 apresenta a proposta do modelo teórico conceitual do Método de Aplicação *Lean Thinking* em Projetos Scrum (MALS), antes da aplicação do método de pesquisa nas situações reais.

Nessa versão ficaram definidas duas fases: Implantação Scrum e Implantação LSD. Cada uma dessas fases tem sequências de etapas para serem implementadas, sendo da 1 (um) a 7 (sete), para Scrum e da 8 (oito) a 12, para LSD.

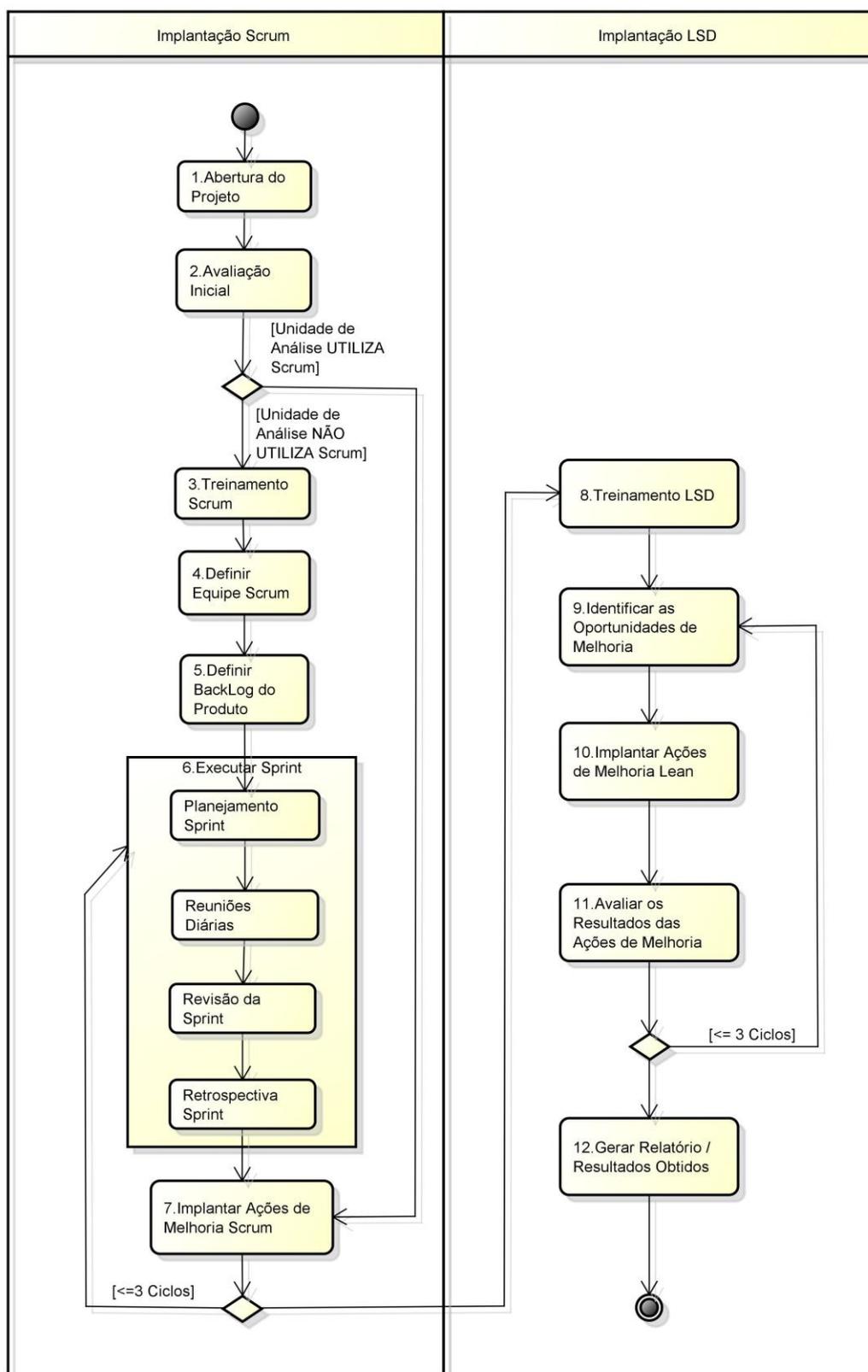


FIGURA 7 – MÉTODO DE APLICAÇÃO LEAN THINKING EM PROJETOS SCRUM (MALS) – MODELO TEÓRICO-CONCEITUAL. FONTE: ELABORADO PELO AUTOR.

4.2 MÉTODO DE APLICAÇÃO LEAN THINKING EM PROJETOS SCRUM (MALS) CONSOLIDADO

A aplicação do MALS em situações reais utilizando a Pesquisa-Ação possibilitou avaliar o modelo teórico conceitual, resultando em sua versão final apresentado nesta seção.

Para esse projeto o roteiro de implantação do Scrum de Sutherland (2014), é identificado como “Roteiro Scrum” (Quadro 7); e o roteiro de implantação do LSD de Poppendieck e Poppendieck (2011), é identificado como “Roteiro LSD” (Quadro 6).

A Figura 8 apresenta a proposta do MALS consolidado, com objetivo de orientar o processo de implantação.

O MALS foi dividido em 13 (treze) etapas e em 2 (duas) fases principais:

- Fase 1. Implantação do Scrum; e
- Fase 2. Implantação do LSD.

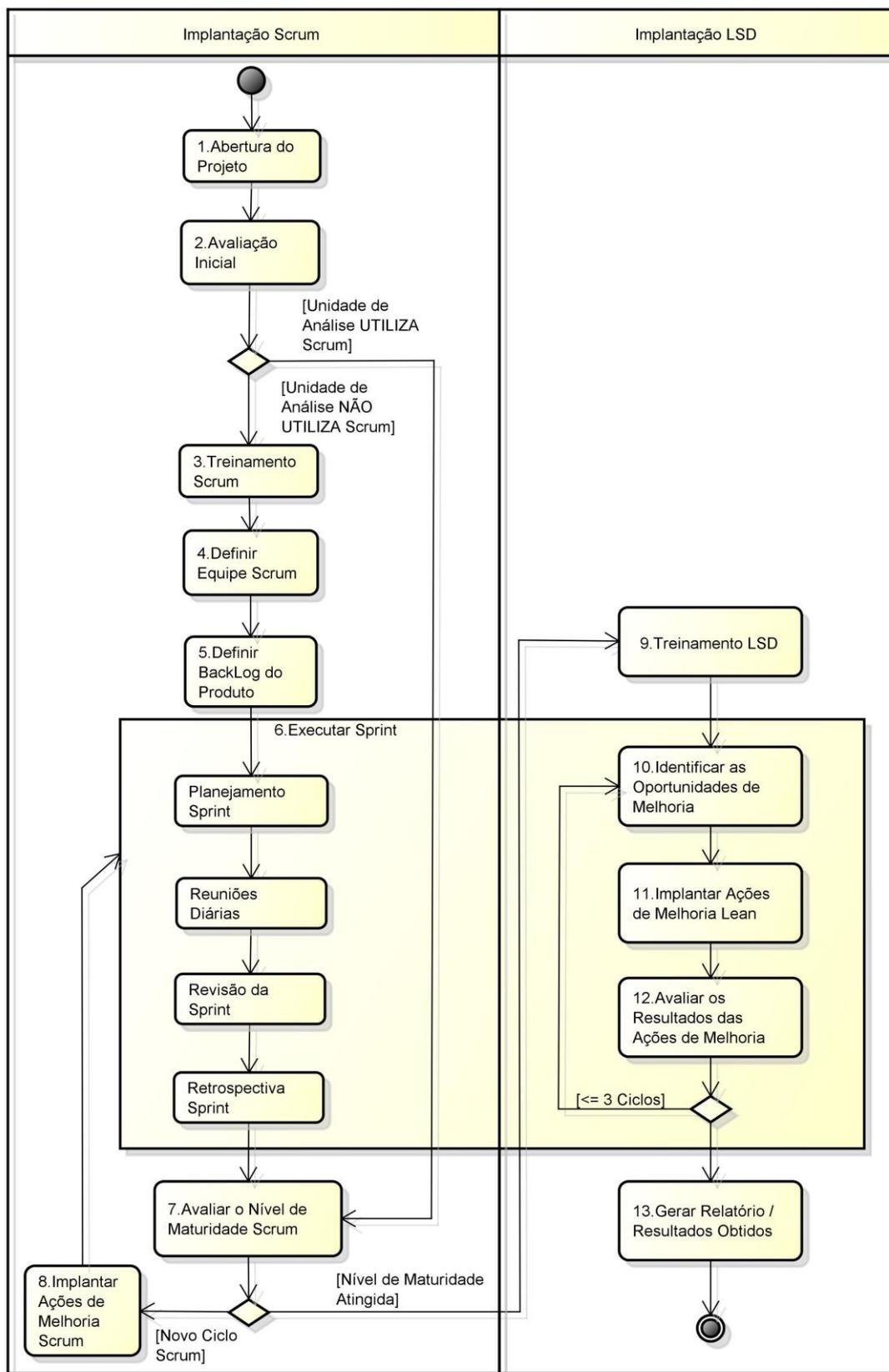


FIGURA 8 – MÉTODO DE APLICAÇÃO LEAN THINKING EM PROJETOS SCRUM (MALS) – MODELO CONSOLIDADO. FONTE: ELABORADO PELO AUTOR.

O Quadro 11 apresenta o detalhamento de cada etapa do MALS.

Etapa	Descrição	Ações/Comentários	Proposta de Pesquisa-Ação de Mello et al. (2012)
1	Abertura do Projeto	Apresentar a proposta MALS para os envolvidos; Analise viabilidade e condições organizacionais.	Coletar Dados
2	Avaliação inicial Scrum	Identificar o nível de adequação Scrum.	Coletar Dados
3	Treinamento Scrum	Treinar equipe: Métodos Ágeis e metodologia Scrum (conforme Roteiro Scrum); Definir técnicas para visualização e acompanhamento do trabalho; Definir as práticas de engenharia <i>software</i> para especificação e estimativa de esforço.	Implementar Ações
4	Definir equipe Scrum	Executar as etapas 1, 2 e 3 do Roteiro Scrum.	Implementar Ações
5	Definir o <i>BackLog</i> do Produto	Executar as etapas 4 e 5 do Roteiro Scrum.	Implementar Ações
6	Executar a <i>Sprint</i>	Executar as etapas 6 a 10 do Roteiro Scrum.	Coletar Dados; Analisar Dados e Planejar Ações
7	Avaliar o Nível Maturidade Scrum	Avaliar o nível de maturidade Scrum a cada ciclo.	Coletar Dados
8	Implantar Ações de Melhoria Scrum	Implantar as ações de melhoria identificadas na Retrospectiva da <i>Sprint</i> .	Implementar Ações
9	Treinamento Princípios LSD	Treinar equipe conforme Roteiro LSD.	Implementar Ações
10	Identificar as oportunidades de Melhoria	Identificar com a equipe as oportunidades de melhoria baseando-se nos princípios LSD.	Coletar Dados; Analisar Dados e Planejar Ações
11	Implantar Ações de melhoria LSD	As ações de melhoria baseadas no LSD devem ser executadas conjuntamente com as atividades executadas no ciclo Scrum.	Implementar Ações
12	Avaliar os resultados das ações de melhoria LSD	As ações de melhoria devem ser avaliadas na reunião de Retrospectiva da <i>Sprint</i> .	Avaliar os resultados e gerar relatório
13	Gerar relatório	Gerar relatório de fechamento do estudo registrando os resultados do estudo realizado.	Avaliar os resultados e gerar relatório

QUADRO 11 – DETALHAMENTO DO MALS. FONTE: ELABORADO PELO AUTOR.

Na etapa 1 do MALS deve ser apresentada a proposta do método para os envolvidos e deve ser analisada a viabilidade da implantação de acordo com as seguintes condições organizacionais:

- Na etapa 1 do MALS deve ser apresentada a proposta do método para os envolvidos e deve ser analisada a viabilidade da implantação de acordo com as seguintes condições organizacionais:
- Apoio da alta administração da empresa. A implantação do MALS gera impacto no processo de trabalho dos envolvidos, podendo ocasionar distúrbios iniciais, inerentes ao processo de mudança. A alta administração da empresa deve aceitar e apoiar as mudanças geradas pelo MALS;
- Disponibilidade de projetos. A implantação do MALS ocorre durante o desenvolvimento de um projeto de um novo produto ou parte de um produto existente. A equipe deve eleger o projeto que deve ser o foco do estudo e implantação do MALS;
- Equipe. Disponibilidade da equipe tempo integral ou parcial para o projeto. A equipe deve possuir no mínimo 4 (quatro) e no máximo 9 (nove) integrantes.

Na etapa 2 (dois) do MALS deve ser avaliado o nível de adequação da equipe em relação ao Scrum. Para isso deve ser respondido o questionário, conforme Quadro 12:

- | |
|---|
| <ol style="list-style-type: none">1. O Dono do Produto (<i>Product Owner</i> – PO) está definido ?2. A Equipe do projeto está definida ?3. O Scrum <i>Master</i> está definido ?4. O <i>BackLog</i> do Produto está criado e priorizado ?5. O esforço dos requisitos do <i>BackLog</i> do Produto estão estimados ?6. O planejamento da <i>Sprint</i> está sendo realizado em períodos definidos ?7. Existem técnicas para visualização e acompanhamento do trabalho que está sendo realizado ?8. Reunião de acompanhamento diário estão sendo realizadas ?9. O resultado dos trabalhos realizados estão sendo demonstrados para equipe e o PO ?10. O <i>Sprint</i> está sendo avaliado sistematicamente, promovendo ciclos de melhoria contínua ? |
|---|

QUADRO 12 – QUESTIONÁRIO DE ADEQUAÇÃO SCRUM.
FONTE: ELABORADO PELO AUTOR.

Deve ser considerado que a equipe “Utiliza Scrum” se todas as etapas do Roteiro Scrum estão implantadas, ou seja, se todas as respostas forem positivas, nesse caso a próxima etapa deve ser a etapa 7 (sete) “Avaliar o Nível de Maturidade Scrum”, porém se alguma das respostas for negativa a próxima etapa deve ser a etapa 3 (três), “Treinamento do Scrum”.

Na etapa 3 (três) do MALS a equipe envolvida deve ser treinada na metodologia Scrum, conforme Roteiro Scrum. Deve ser definido as técnicas para visualização e acompanhamento do trabalho que será realizado, como exemplo as ferramentas geração do Gráfico *BurnDown* (vide seção 2.4.1) e para visualização do Quadro Scrum (vide seção 2.4.2). Além disso, deve ser definido práticas de engenharia de *software* para especificação e estimativa de esforço dos requisitos.

As etapas 4 (quatro), 5 (cinco) e 6 (seis) do MALS devem ser executadas conforme definido no Roteiro Scrum.

Na etapa 7 (sete) do MALS deve ser avaliado o nível de adequação do Scrum para iniciar a segunda fase do MALS, a Implantação do LSD. A adequação do Scrum deve ser avaliada conforme etapa 2 (dois) do MALS aplicando-se o questionário de adequação Scrum (Quadro 12). Considera-se que a equipe esteja preparada para iniciar a fase de implantação LSD se todas as etapas do Roteiro Scrum estejam implantadas e se no mínimo 2 (dois) ciclos/*sprints* estejam concluídos.

Na etapa 8 (oito) do MALS deve ser implantado as ações de melhoria Scrum identificadas nas reuniões da retrospectiva da *Sprint*. O acompanhamento das ações deve ser realizado nas reuniões diárias definidas na etapa 8 (oito) do Roteiro Scrum.

A partir da etapa 9 (nove) do MALS inicia-se a implantação do LSD. A equipe envolvida deve ser treinada nos princípios LSD, conforme Roteiro LSD.

Após o treinamento dos princípios LSD, a equipe deve identificar as oportunidades de melhoria (etapa 10 do MALS) baseando-se nos princípios

LSD. Para isso devem ser utilizadas as técnicas de entrevista com a equipe e discussão.

Na etapa 11 do MALS devem ser implantadas as ações de melhoria LSD identificadas no treinamento dos princípios LSD e nas reuniões da retrospectiva da *sprint*, nos ciclos que sucederem o início da segunda fase. O acompanhamento das ações de melhoria deve ser realizado conjuntamente com as ações de melhoria do Scrum, definidas na etapa 8 (oito) do MALS.

Na etapa 12 do MALS devem ser avaliados os resultados das ações de melhoria LSD. A avaliação dos resultados deve ser executada nas reuniões da retrospectiva da *Sprint* utilizando as técnicas de entrevista com a equipe, discussão e observações documentais.

Na etapa 13 do MALS deve ser gerado o relatório de fechamento do estudo registrando os resultados baseado nos desdobramentos de cada ciclo do processo de implantação do MALS.

4.3 AVALIAÇÃO DA PROPOSTA DO MALS COM A PESQUISA-AÇÃO

Esta seção descreve a evolução da proposta do modelo conceitual teórico com os desdobramentos da Pesquisa-Ação.

A entrevista para avaliação inicial do MALS foi realizada na etapa 1 (um) Abertura do Projeto e durante os ciclos de pesquisa nas reuniões de retrospectiva da *Sprint*. O questionário para coleta dos dados para avaliar o MALS está definido no Quadro 9. Os resultados do questionário são detalhados no Quadro Resumo de Pesquisa-Ação, apêndices A e B respectivamente.

A avaliação do MALS possibilitou identificar oportunidades de melhoria no modelo conceitual teórico. O Quadro 13 descreve a evolução do MALS com a aplicação da Pesquisa-Ação.

Avaliação	Resultados do Questionário de Avaliação	Plano de Ação	Resultados da Ação
Inicial	Identificação de oportunidade de melhoria: Identificar as técnicas para visualização e acompanhamento do trabalho realizado e práticas para especificação de requisitos e estimativa de esforço.	Incluído na etapa 3 Treinamento do Scrum, a definição das técnicas para visualização e acompanhamento do trabalho; e as práticas para especificação de requisitos e estimativa de esforço.	Definição nas fases iniciais do MALS as técnicas para visualização e acompanhamento do trabalho; e as práticas para especificação de requisitos e estimativa de esforço.
Ciclo de Pesquisa	A equipe identificou que 3 (três) ciclos do Scrum eram insuficientes para consolidar a implantação da primeira fase do MALS. Identificação de oportunidade de melhoria: Avaliar o nível de maturidade da primeira fase do MALS antes de iniciar a segunda fase.	Criado etapa 7 para avaliar o nível de maturidade do Scrum e identificar o momento mais apropriado para iniciar a fase de implantação do LSD. Foi retirado o limite de 3 ciclos fixos de implantação da primeira fase.	O momento de transição para a segunda fase do MALS foi definida pela equipe.
Ciclo de Pesquisa	Identificação de oportunidade de melhoria: Avaliar o nível de adequação do Scrum.	Incluído na etapa 2 (Avaliação Inicial Scrum) e na etapa 7 (Avaliar o Nível Maturidade Scrum) a avaliação de adequação Scrum, por meio do questionário, conforme apresentado no Quadro 12.	A avaliação de adequação inicial do Scrum na etapa 2 avalia se a unidade de análise atende todos os requisitos do Scrum, para decidir se a equipe deve ser treinada conforme etapa 3 (Treinamento Scrum). A avaliação de adequação do Scrum na etapa 7 definiu os critérios para transição para segunda fase do MALS.
Ciclo de Pesquisa	A equipe não compreende as fases do MALS: A equipe não compreendeu claramente que os ciclos Scrum continuam sendo executados na segunda fase do MALS.	Alterado a estrutura visual do MALS.	O questionário de avaliação do MALS, aplicado no ciclo seguinte, evidenciou o entendimento da alteração da estrutura visual realizada.

QUADRO 13 – EVOLUÇÃO DO MALS. FONTE: ELABORADO PELO AUTOR.

A alteração na etapa 3 (Treinamento Scrum) possibilitou identificar as técnicas para visualização e acompanhamento do trabalho e as práticas para especificação de requisitos e estimativa de esforço antes de iniciar os ciclos Scrum, facilitando a organização das tarefas e a estimativa de esforço.

Além disso, a criação e adaptações na etapa 7 (Avaliar o Nível Maturidade Scrum) definiram os critérios para transição para segunda fase do MALS, dessa forma a implantação do LSD deve ocorrer após todos os requisitos Scrum estarem compreendidos pela equipe.

A alteração da estrutura visual do MALS facilitou o entendimento que as fases 10, 11 e 12 ocorrem durante o ciclo Scrum, ou seja, durante a implantação do LSD os ciclos Scrum continuam sendo executados.

5 APLICAÇÃO DO MALS E RESULTADOS

Este capítulo apresenta a discussão dos resultados (relatório de pesquisa) baseada na aplicação do MALS nas unidades de análise, aplicando o protocolo de pesquisa-ação do Quadro 9.

5.1 SELECIONAR AS UNIDADES DE ANÁLISE

Baseando-se nos critérios definidos (Quadro 9), foram selecionadas 2 (duas) unidades de análise, nesse projeto identificadas como Empresa A e Empresa B.

A **Empresa A** é uma fábrica de *software*, com sede na cidade de Leme/SP. Atuação principal na área comercial, no seguimento de saúde, atualmente com 450 clientes aproximadamente.

A equipe de análise da Empresa A possui 9 (nove) integrantes, sendo que 3 (três) tem como função principal controle de qualidade e 6 (seis) integrantes tem como função principal desenvolvimento (codificação) de *software*. Um membro da equipe exerce a função de gerente de projetos e possui especialização em Engenharia de *Software*. Esse membro da equipe foi nomeado como líder para implantação do projeto MALS e tem autoridade final para uma ação na unidade de análise. O pesquisador é cooperativo e participativo na equipe, atuando de forma consultiva, ou seja, recomendando aos tomadores de decisão as ações implementadas.

A **Empresa B** é uma empresa do ramo educacional. Com equipe destinada para desenvolvimento de *softwares* para consumo próprio, com sede na cidade de Araras/SP.

A equipe de análise da Empresa B possui 9 (nove) integrantes, sendo que 2 (dois) tem como função principal controle de qualidade, 5 (cinco) integrantes tem como função principal desenvolvimento (programação) de *software*, 1 (um)

gerente de projetos e 1 (um) analista de sistemas. Todos os envolvidos possuem nível graduação e 4 (quatro) possuem nível pós-graduação. O pesquisador é cooperativo e participativo na equipe, e tem responsabilidade de executar ações recomendadas. O gerente de projetos possui autoridade para aprovar a implementação das ações na unidade de análise.

5.2 DEFINIÇÃO DOS CICLOS DE PESQUISA E COLETA DE DADOS

De acordo com a metodologia Scrum, os ciclos de trabalho devem ocorrerem em intervalos entre 2 (duas) e 4 (quatro) semanas (SCHWABER, 2004). Portanto, os ciclos de pesquisa definidos para Empresa A e Empresa B foram de 14 dias, ou 2 (duas) semanas.

Os ciclos de implantação do MALS para as 2 (duas) unidades de análise (Empresa A e Empresa B) são detalhados no Quadro Resumo de Pesquisa-Ação, apêndices A e B respectivamente.

Para ambas as unidades de análise, Empresa A e Empresa B, a coleta de dados foi realizada por meio de entrevista coletiva, discussões e observações. A entrevista coletiva e as discussões ocorreram na reunião da Retrospectiva da *Sprint*, incentivando todos a participar (as questões a serem efetuadas para avaliar a aplicação do MALS estão definidas no Quadro 9). As observações documentais devem ser realizadas utilizando os gráficos *BurnDown* (vide seção 2.4.1), *TaskBoards* (vide seção 2.4.2) e quando aplicável a ferramenta de gestão de configuração utilizada pela empresa. Dessa forma foi possível realizar a triangulação dos dados gerando informações para realimentação do ciclo de melhoria e aprendizagem.

5.3 DISCUSSÃO DOS RESULTADOS

Essa seção apresenta os resultados obtidos com a aplicação do Método de Aplicação *Lean Thinking* em Projetos Scrum (MALS) nas Empresas A e B.

5.3.1 EMPRESA A: ESCOPO E RESULTADOS

Empresa A: Escopo

Na Empresa A, a aplicação do MALS ocorreu no período de 12/05/2016 a 30/09/2016. Nesse período foram avaliados 9 (nove) ciclos de pesquisa, correspondendo a 5 (cinco) ciclos na primeira fase do MALS (Implantação Scrum) e 4 (quatro) ciclos na segunda fase do MALS (Implantação LSD).

A Empresa A não utilizava a metodologia Scrum, portanto foi necessário iniciar a implantação com a primeira fase do MALS, que corresponde a implantação e avaliação do Scrum, conforme as etapas 3 (três) a 8 (oito) descritas no Quadro 11.

A participação do pesquisador na primeira fase do MALS teve o objetivo de treinar a equipe e orientar sobre a utilização da metodologia Scrum, conforme Roteiro Scrum.

Empresa A: Resultados

Os critérios de avaliação da aplicação do MALS definidos pela Empresa A nos ciclos de pesquisa foram: a) Práticas de gestão de projetos; b) Práticas de Engenharia de *Software*; c) Integração e Comunicação da Equipe; d) Qualidade do Produto; e) Prazo de entrega de *releases* estáveis; f) Produtividade.

A **primeira fase de implantação do MALS**, referente à implantação e avaliação do Scrum apresentou os resultados:

- a) **Práticas de Gestão de Projetos:** A sistematização das reuniões de planejamento (reunião de planejamento da *sprint*) e acompanhamento (reuniões diárias) padronizou o *feedback* do trabalho, definindo para todos os mesmos prazos de entrega, facilitando a gestão e planejamento das tarefas;
- b) **Práticas de Engenharia de Software:** Foi definido o uso do *Team Foundation Software* (TFS) como ferramenta para visualização e

acompanhamento do trabalho (*TaskBoard*) e a técnica de *User Stories* para especificação de requisitos e *User Points* como métrica de produtividade e estimativa de esforço. Essas práticas facilitaram as operações de gerenciamento e controle dos eventos Scrum e permitiram a geração dos gráficos de controle (*BurnDown*);

- c) **Integração e Comunicação da Equipe:** Em todas as entrevistas coletivas realizadas com a equipe, nas reuniões de retrospectiva da *Sprint*, foi relatado maior compreensão dos objetivos do trabalho e das atividades realizadas pela equipe, facilitando assim o planejamento das tarefas individuais. Esse planejamento reduziu os atrasos por falta de informação ou desconhecimento das dependências das tarefas;
- d) **Qualidade do Produto:** Evidenciado a partir da 4ª (quarta) *Sprint* que os *releases* entregues não apresentaram falhas identificadas pelo cliente;
- e) **Prazo de entrega de *releases* estáveis:** A equipe determinou como objetivo, um *release* estável (pronto para entrega para o cliente) por ciclo/*sprint* (duas semanas). A partir da 4ª (quarta) *sprint*, esse objetivo foi atingido. O ciclo de entrega de *releases* antes do Scrum ultrapassava 3 (três) meses;
- f) **Produtividade:** As práticas sugeridas pela equipe (Quadro 14) com objetivo de melhoria de processo permitiram identificar a capacidade produtiva da equipe, média de **148 *User Points*** (média até o 5º ciclo/*sprint*);
- g) **Melhorias identificadas pela equipe:** Reação rápida diante de um impedimento ou problema; “Coragem para resolver problemas”, requisitos que estavam pendentes por muito tempo foram incluídos no planejamento do trabalho; Identificação da capacidade produtiva da equipe, evitando super ou sub dimensionamento de trabalho;
- h) **Cultura de Melhoria Contínua:** As reuniões quinzenais de retrospectiva da *sprint* implantaram a cultura de melhoria contínua, por meio do

questionamento dos problemas enfrentados e das ações de melhoria sugeridas pela equipe.

O Quadro 14 apresenta um resumo das práticas sugeridas pela equipe para melhoria do processo.

Ciclo	Práticas Sugeridas pela Equipe	Melhorias do Processo
1	a) Detalhar as <i>User Stories</i> e dividir em tarefas menores;	Facilitar o planejamento e estimativa de esforço.
2	b) Bloquear alterações em requisitos depois de liberado para testes;	Sincronizar as atividades de desenvolvimento e controle de qualidade (testes) para permitir liberar um <i>release</i> estável por <i>Sprint</i> .
	c) Liberar <i>release</i> por <i>Sprint</i> ;	
3	d) Identificar a capacidade produtiva da equipe para facilitar o planejamento;	Melhorar a estimativa de esforço para facilitar o planejamento.
	e) Classificar as <i>User Stories</i> por nível de complexidade (baixa, média, alta) para facilitar o entendimento da tarefa e facilitar o planejamento;	
4	f) Antecipar o Plano de Testes e disponibilizar para desenvolvimento;	Reduzir o número de falhas entregues ao controle de qualidade na fase de implementação do <i>software</i> .
5	g) Retirar atribuição da equipe de controle de qualidade por realizar demonstração de novos produtos;	Aumentar a disponibilidade da equipe de controle de qualidade para sincronizar as atividades.

QUADRO 14 – PRÁTICAS SUGERIDAS PELA EMPRESA A PARA MELHORIA PROCESSO SCRUM. FONTE: ELABORADO PELO AUTOR.

A principal dificuldade encontrada na primeira fase de implantação do MALS foi a sincronização das atividades de desenvolvimento e controle de qualidade, para permitir que todas as tarefas sejam totalmente finalizadas dentro do ciclo. Essa dificuldade foi contornada com a utilização de *sprints* com tarefas focadas apenas no controle de qualidade e correções de falhas. Porém essa ação reduziu a capacidade produtiva de novas funcionalidades dentro do ciclo.

Diante disso, o *Product Owner* (PO) relatou que não observou aumento de produtividade com a implantação do Scrum (até o 5º ciclo), mesmo observando que ocorreram melhorias no processo de gerenciamento da produção de *software* e controle de qualidade dos produtos, com entregas de *releases* estáveis e que agregam valor ao cliente mais rapidamente.

Outra dificuldade relatada pela equipe foi realizar a estimativa de esforço das tarefas. A métrica utilizada é *User Points*, obtidas utilizando a técnica de *Planning Poker*. Para facilitar a compreensão do trabalho e conseqüentemente a estimativa de esforço, as *User Stories* foram divididas em tarefas menores, ou em *Stories* menores. Além disso, foi implantada a prática de classificar as *User Stories* por nível de complexidade. A manutenção dessas práticas, associada com a experiência adquirida pela equipe com a metodologia Scrum, permitiu melhorar a precisão das estimativas, validadas nas reuniões de retrospectiva da *sprint*. Porém é consenso da equipe ser a atividade Scrum mais difícil para ser executada.

A transição para a **segunda fase do MALS**, referente à implantação do LSD, ocorreu a partir do 6º ciclo, quando o questionário de adequação foi totalmente satisfeito.

A etapa 9 do MALS, referente ao treinamento dos princípios LSD ocorreu em 12/08/2016. Nessa oportunidade foi solicitado para todos os participantes do treinamento preparar uma ação de melhoria, baseada nos princípios LSD, para ser discutido na próxima reunião de retrospectiva da *sprint*. Além disso, foi realizada entrevista com o *Product Owner* e o *Scrum Master* para identificar as oportunidades de melhoria baseados no LSD.

O Quadro 15 apresenta as ações de melhoria aplicadas conforme princípios do LSD, identificadas em cada ciclo. Os ciclos apresentados no Quadro 15 estão relacionados aos ciclos do Scrum (*sprints*), detalhados no Quadro-Resumo da Pesquisa-Ação Empresa A.

Ciclo	Princípio LSD	Ações de Melhoria
5	Princípio 1: Otimizar o todo	a) Nomeado o <i>Scrum Master</i> e <i>Product Owner</i> como equipe de liderança do fluxo de valor; b) Mapeado o Fluxo de valor para procurar interrupções no fluxo.
	Princípio 2: Respeitar as Pessoas	c) Solicitado que a equipe identifique melhorias no processo com foco na abordagem LSD;
6	Princípio 3: Entregar Rápido	d) Elaborar lista de pendências (solicitações de melhoria e falhas identificadas), para controlar a redução das filas de pendências;
	Princípio 6: Integrar Qualidade	e) Realizar testes unitários utilizando o plano de testes desenvolvido, para testar e corrigir as falhas mais cedo; f) Treinamento com equipe com foco na filosofia de “falha zero”, durante o processo de desenvolvimento e principalmente na entrega aos clientes. Evitando assim retrabalho e melhorando a imagem perante o cliente; g) Orientação para equipe refatorar o código sempre que possível, com objetivo de tornar o código mais simples; h) Aprovado retirar atribuição da equipe de controle de qualidade realizar demonstração de novos produtos, com objetivo de aumentar a disponibilidade da equipe e sincronizar as ações de desenvolvimento e para evitar o trabalho inacabado dentro do ciclo; i) Atribuição de responsável para estudar a técnica TDD para viabilizar a automação dos testes;
	Princípio 7: Eliminar Desperdícios	j) Orientação para equipe reduzir o número de troca de tarefas, finalizando totalmente uma tarefa antes de iniciar a próxima;
7	Princípio 6: Integrar Qualidade	k) Aprovado retirar atribuição da equipe de controle de qualidade realizar os testes de integração dos novos clientes, com objetivo de aumentar a disponibilidade da equipe e sincronizar as ações de desenvolvimento e evitar o trabalho inacabado dentro do ciclo; l) Medir o fluxo de falhas na fila para atender o item (d); m) Aprovado a integração de mais um membro na equipe, com objetivo de contribuir com a Engenharia de <i>Software</i> para reduzir a complexidade do código e desenvolver soluções técnicas mais simples.
8	Princípio 6: Integrar Qualidade	n) Automatizar o lançamento automático de falhas (bugs) no TFS a partir do sistema “versões”;
	Princípio 3: Entregar Rápido	o) Planejar as <i>User Stories</i> em tarefas menores de 1 dia para evitar o trabalho inacabado no <i>sprint</i> ;
9	Princípio 7: Eliminar Desperdícios	p) Criar uma lista de pendências priorizando pelo valor agregado ao cliente;
	Princípio 3: Entregar Rápido	q) Priorizar o atendimento e controlar a redução dessa lista de pendências (p).

QUADRO 15 – AÇÕES DE MELHORIA APLICADAS PELA EMPRESA A BASEADAS NA ABORDAGEM LSD. FONTE: ELABORADO PELO AUTOR.

A análise do desenho do fluxo de valor possibilitou identificar que a principal interrupção no fluxo estava relacionada às atividades de controle de qualidade. Para resolver a interrupção e aumentar a disponibilidade da atividade de controle de qualidade, foi autorizada a transferência da responsabilidade por

realizar os treinamentos com clientes externos para equipe de suporte e a atribuição dos testes de integração com novos clientes para equipe de desenvolvimento.

De acordo com LSD, os testes devem ser escritos primeiro e o código testado mais cedo. Para isso, foi solicitada a continuidade da prática de antecipação do plano de testes, desenvolvido pelo de controle de qualidade, para permitir melhorar a qualidade dos testes unitários realizados durante a implementação do *software*. Além disso, foi definido responsável para estudar a técnica de Desenvolvimento Guiado por Testes (TDD), com objetivo de automatizar a tarefa de realização dos testes unitários.

Durante a reunião de retrospectiva do ciclo 6 (seis), foi realizado treinamento com a equipe, realizado pelo *Scrum Master* e a alta direção da empresa abordando os assuntos:

- Filosofia “falha zero”, com foco em eliminar as falhas detectadas no controle de qualidade e no cliente. O objetivo é reduzir o desperdício com o retrabalho gerado por falhas detectadas no controle de qualidade e melhorar a imagem da empresa perante o cliente;
- Refatorar constantemente o código, com objetivo de reduzir a complexidade e facilitar as ações de controle de qualidade e mudanças;
- Evitar troca de tarefas, finalizando totalmente uma tarefa antes de iniciar outra, dessa forma o Quadro Scrum (Figura 4) deve apresentar apenas 1 (uma) tarefa “fazendo” por membro da equipe.

Para avaliar a redução das filas de pendências, a partir do ciclo 8 (oito), o *Scrum Master* deve controlar o número de pendências que entram na fila e que são encerradas. Essa prática tem o objetivo de melhorar a satisfação do cliente por meio de entregas e solução de problemas com maior rapidez. Com esse mesmo objetivo foi definido um padrão para relato e identificação de falhas, permitindo automatizar os sistemas de relacionamento com clientes, nomeado de “Versões”, com o sistema de gerenciamento de tarefas do Scrum, o TFS.

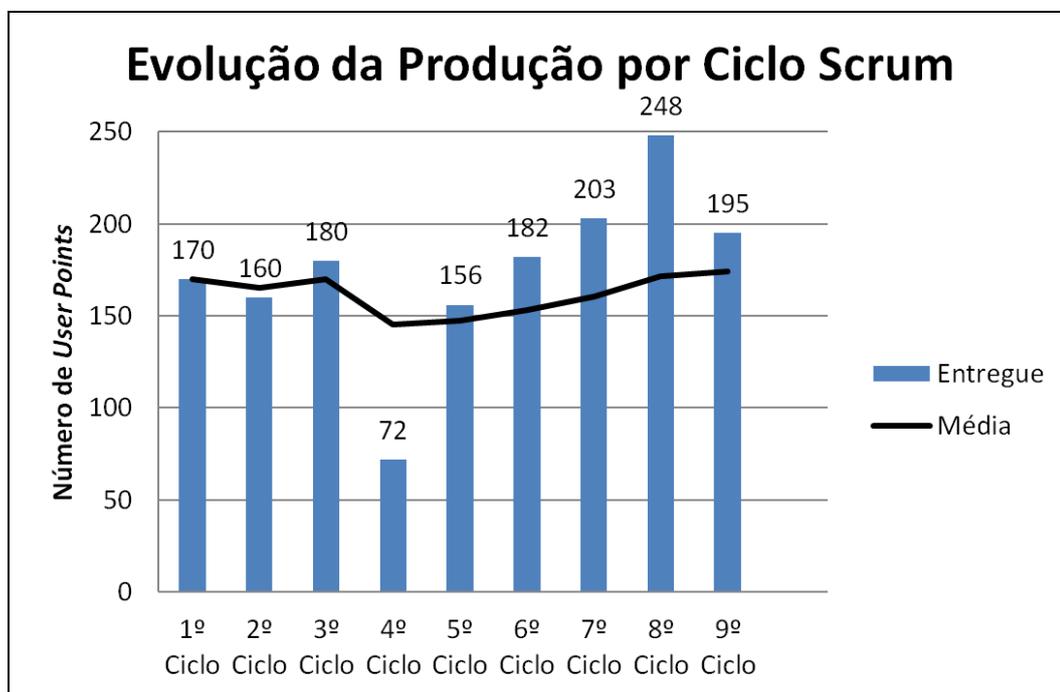
O ciclo 9 (nove) foi o último ciclo de pesquisa e coleta de dados realizado na Empresa A e as ações estavam focadas em reduzir a lista de pendências originadas por solicitação do cliente e definidas como objetivo no ciclo 7 (sete) e viabilizadas pelas automações criadas no ciclo 8.

Por solicitação da equipe da Empresa A, foram realizados 4 (quatro) ciclos de coleta de dados e avaliação dos resultados após a implantação do LSD. Ficou evidenciado que ações foram identificadas com objetivo de melhoria do processo, porém não foi suficiente abordar todos os princípios LSD durante os ciclos de pesquisa realizados.

A **segunda fase de implantação do MALS**, referente a implantação e avaliação do LSD, de acordo como os critérios de avaliação definidos pela empresa A, apresentaram os seguintes resultados:

- a) **Práticas de Gestão de Projetos:** Continuidade das práticas implantadas na primeira fase do MALS;
- b) **Práticas de Engenharia de Software:** Estudar TDD, antecipar o Plano de Testes e planejar as *User Stories* em períodos de 1 (um) dia ou menores, foram práticas aplicadas para apoiar o LSD;
- c) **Integração e Comunicação da Equipe:** Mantidas as ações e os resultados implantados na primeira fase do MALS;
- d) **Qualidade do Produto:** A partir da 4^a (quarta) *Sprint* os *releases* entregues não apresentaram falhas identificadas pelo cliente, portanto as ações de melhoria se concentraram em criar códigos estáveis para reduzir as falhas durante o controle de qualidade, evitando desperdícios com retrabalho;
- e) **Prazo de entrega de releases estáveis:** A partir da 4^a (quarta) *sprint*, iniciou a entrega de *releases* estáveis dentro da *sprint*, portando o prazo de entrega foi fixado em períodos de 2 (duas) semanas. No *sprint* 7 (sete) ocorreu atraso na entrega devido a problemas discutidos em reunião de retrospectiva e resolvidos no próximo *sprint* (detalhes vide Apêndice A);

- f) **Produtividade:** A Figura 9 apresenta a evolução de produtividade nos 9 (nove) ciclos de pesquisa. O cálculo da média corresponde a média de *User Points* dos ciclos anteriores.



*FIGURA 9 – EVOLUÇÃO DA PRODUÇÃO POR CICLO SCRUM: EMPRESA A.
FONTE: ELABORADO PELO AUTOR.*

Com implantação do LSD, a partir do 6º ciclo, a equipe apresentou a produtividade média de **207** *User Points* entregues, representando um aumento de **39,9%** em relação à média de produtividade entre o 1º e 5º ciclos, correspondendo à fase de implantação do Scrum.

Além disso, a partir do 6º ciclo a produtividade apresentou maior estabilidade, demonstrando a maturidade da equipe e o resultado das ações de melhoria implantadas.

5.3.2 EMPRESA B: ESCOPO E RESULTADOS

Empresa B: Escopo

Na Empresa B, a aplicação do MALS ocorreu no período de 08/08/2016 a 07/11/2016. Nesse período foram avaliados 6 (seis) ciclos de pesquisa,

correspondendo a 3 (três) ciclos na primeira fase do MALS (Implantação Scrum) e 3 (três) ciclos na segunda fase do MALS (Implantação LSD).

A Empresa B, utilizava a metodologia Scrum antes da implantação do MALS, porém para avaliação da adequação identificou que muitos processos não estavam implementados, portanto foi necessário iniciar a implantação com a primeira fase do MALS, que corresponde à implantação do Scrum, conforme as etapas 3 (três) a 8 (oito) descritas no Quadro 11.

A participação do pesquisador na primeira fase do MALS teve o objetivo de treinar a equipe, orientar sobre a utilização da metodologia Scrum (conforme Roteiro Scrum) e participar da implementação das ações de melhoria identificadas.

Empresa B: Resultados

Os critérios de avaliação da aplicação do MALS definidos pela Empresa B nos ciclos de pesquisa foram: a) Práticas de gestão de projetos; b) Práticas de Engenharia de *Software*; c) Integração e Comunicação da Equipe; d) Qualidade do Produto; e) Produtividade; f) Fila de Pendências.

A **primeira fase de implantação do MALS**, referente à implantação e avaliação do Scrum, permitiu uma reavaliação da aplicação da metodologia e a organização dos eventos, apresentando os resultados:

- a) **Práticas de Gestão de Projetos:** Assim como na Empresa A, a Empresa B apresentou uma sistematização do processo de gestão de projetos e necessidades dos clientes, facilitando a gestão e planejamento das tarefas e *feedback* mais rápido ao cliente;
- b) **Práticas de Engenharia de Software:** Foi definido o uso da ferramenta JIRA *Agile* para visualização e acompanhamento do trabalho (*Taskboard* e Gráfico *BurnDown*). A Empresa B utilizou a técnica de *User Stories* para especificação de requisitos e *User Points* como métrica de produtividade e estimativa de esforço;

- c) **Integração e Comunicação da Equipe:** A equipe evidencia nas entrevistas coletivas que as reuniões diárias facilitaram a comunicação, além de possibilitar o compartilhamento de experiências, portanto o aprendizado coletivo. Além disso, as reuniões de planejamento possibilitaram a definição clara dos objetivos da *sprint*, reduzindo os atrasos por dependências de integrantes da equipe que estão trabalhando em tarefas diferentes;
- d) **Qualidade do Produto:** O *feedback* contínuo ao cliente possibilitou melhor adequação dos produtos às suas reais necessidades;
- e) **Produtividade:** As observações documentais possibilitou identificar a capacidade produtiva da equipe. Média de **85 User Points** entre o 2º e 3º ciclo/*sprint*;
- f) **Fila de Pendências (Backlog):** A fila de pendências no *backlog* do produto apresentava **250 pendências** antes de iniciar o MALS. No 3º ciclo Scrum, a fila de pendências no *backlog* do produto apresentava **251 pendências**. Nesse período 169 pendências foram resolvidas e entregues e 170 novas pendências foram inseridas, mantendo a quantidade de pendências praticamente inalterada em relação ao início do MALS;
- g) **Melhorias identificadas pela equipe:** Reavaliação da lista de pendências e priorização das tarefas do *backlog* do produto; Integração da equipe e conhecimento por todos das tarefas que estão sendo implementadas.

O Quadro 16 apresenta um resumo das práticas sugeridas pela equipe para melhoria do processo.

Ciclo	Práticas Sugeridas pela Equipe	Melhorias do Processo
1	a) Antecipar a análise do <i>backlog</i> do produto pelo Scrum Master e Product Owner;	Reduzir o tempo da reunião de planejamento.
2	b) Padronizar o horário das reuniões diárias; c) Maior objetividade na realização da reunião diária;	Reduzir o tempo da reunião diária (não deve ultrapassar 15 minutos).
3	d) Atribuir o esforço para todas as <i>user stories</i> da <i>sprint</i> .	Definir a capacidade produtiva da equipe.

QUADRO 16 – PRÁTICAS SUGERIDAS PELA EMPRESA B PARA MELHORIA PROCESSO SCRUM. FONTE: ELABORADO PELO AUTOR.

Na empresa B, as dificuldades encontradas na primeira fase de implantação do MALS estavam relacionadas à priorização e organização do *backlog* do produto, devido a grande quantidade de pendências identificadas.

A estimativa de esforço utilizando a técnica de *user points* também foi uma dificuldade da equipe, porém devido a experiência com Scrum, antes da implantação do MALS, essa dificuldade foi superada a partir do 2º ciclo do Scrum. Devido a isso, o primeiro ciclo do Scrum não foi considerado para média de produtividade da equipe na primeira fase do MALS.

Outra dificuldade encontrada foi o tempo longo dedicado às reuniões de planejamento e reunião diária. Essas reuniões necessitam a presença de toda a equipe disponível, reduzindo a disponibilidade e produtividade. Para resolver esse problema, nas reuniões diárias foi definido horário fixo (9:00hs) para iniciar a reunião independentemente da ausência de algum membro da equipe. Para reduzir o tempo das reuniões de planejamento são realizadas reuniões preliminares de análise do *backlog* do produto pelo o *Product Owner* e *Scrum Master*, apenas.

A transição para a **segunda fase do MALS**, referente à implantação do LSD, ocorreu a partir do 4º ciclo, quando o questionário de adequação foi totalmente satisfeito.

A etapa 9 do MALS, referente ao treinamento dos princípios LSD ocorreu em 07/10/2016. Nessa oportunidade foi solicitado para equipe preparar uma ação de melhoria, baseada nos princípios LSD, para ser discutido na próxima reunião de retrospectiva da *sprint*.

O Quadro 17 apresenta as ações de melhoria aplicadas conforme princípios do LSD, identificadas pela equipe em cada ciclo. Os ciclos apresentados no Quadro 17 estão relacionados aos ciclos do Scrum (*sprints*), detalhados no Quadro-Resumo da Pesquisa-Ação Empresa B.

Ciclo	Princípio LSD	Ações de Melhoria
4	Princípio 1: Otimizar o todo	a) Nomeado o Scrum Master e Product Owner como equipe de liderança do fluxo de valor;
	Princípio 2: Respeitar as Pessoas	b) Solicitado que a equipe identifique melhorias no processo com foco na abordagem LSD;
	Princípio 6: Integrar Qualidade	c) Estudo da técnica TDD para viabilizar a automação dos testes; d) Planejar as <i>user stories</i> em tarefas menores, de 1 (um) dia para evitar o trabalho inacabado no <i>sprint</i> ;
	Princípio 7: Eliminar Desperdícios	e) Orientação para equipe reduzir o número de troca de tarefas, finalizando totalmente uma tarefa antes de iniciar a próxima; f) Integrar o cliente na priorização do <i>backlog</i> do produto e refinamento dos requisitos, para melhorar as especificações e focar nos requisitos que agreguem valor;
5	Princípio 1: Otimizar o todo	g) Criar medidas de fluxo de valor: Definir métrica para avaliar o tempo em que uma pendência permanece na fila, por nível de criticidade;
	Princípio 3: Entregar Rápido	h) Esvaziar a fila de pendências: Avaliar com os clientes a real necessidade das pendências da lista e priorizar sua implementação;
	Princípio 6: Integrar Qualidade	i) Orientação para equipe refatorar o código sempre que possível, com objetivo de tornar o código mais simples;
6	Princípio 6: Integrar Qualidade	j) Automatizar e Sincronizar: Integrar o módulo ITIL com o JIRA, gerando automaticamente no <i>backlog</i> do produto as falhas identificadas pelo cliente.

QUADRO 17 – AÇÕES DE MELHORIA APLICADAS PELA EMPRESA B BASEADAS NA ABORDAGEM LSD. FONTE: ELABORADO PELO AUTOR.

Com o objetivo de “otimizar o todo”, foi solicitado ao Scrum Master criar uma métrica para avaliar o tempo em que uma pendência demora a ser entregue, ou seja, desde a solicitação até o uso pelo cliente.

Uma dificuldade encontrada pela Empresa B na implantação do LSD foi o trabalho inacabado, ou seja, pendências planejadas e não entregues no *sprint*, para resolver isso, a equipe foi orientada a criar tarefas menores de 1 (um) dia e evitar a troca de tarefas, executando e finalizando uma tarefa de cada vez. Além disso, iniciaram-se os estudos da técnica de Desenvolvimento Guiado por Testes (TDD), com objetivo de automatizar e antecipar a tarefa de realização dos testes.

O foco principal da Empresa B, definido pelo gerente de projetos, foi reduzir a fila de pendências e entregar mais rápido o valor ao cliente. Portanto, as ações foram direcionadas para melhorar a especificação dos requisitos, integrando o cliente ao desenvolvimento, definindo claramente suas necessidades e prioridades e facilitando a comunicação e relato de falhas, com a integração do Sistema ITIL (*Information Technology Infrastructure Library*) com o JIRA Agile.

A verificação dos critérios de avaliação na **segunda fase de implantação do MALS** possibilitou concluir que as práticas de gestão de projetos e engenharia de *software*, assim como integração e comunicação da equipe apresentaram os mesmos resultados da primeira fase, ou seja, durante a implantação do Scrum.

As ações para resolver a **qualidade do produto**, estão diretamente relacionadas à melhoria do código, para isso foi solicitado à equipe adotar a prática de refatorar continuamente o código, com objetivo de torná-lo mais simples.

A evolução da **produtividade** da equipe, na Empresa B pode ser observada na Figura 10. O cálculo da média corresponde a média de *User Points* dos ciclos anteriores e a eficiência indica a quantidade de horas necessárias para entregar 1 (um) *User Point* por ciclo. A eficiência considera a disponibilidade de horas produtivas da equipe descontando os feriados e férias.

A análise não considerou o 1º ciclo Scrum devido a incertezas nas estimativas de esforço utilizando *User Points*.

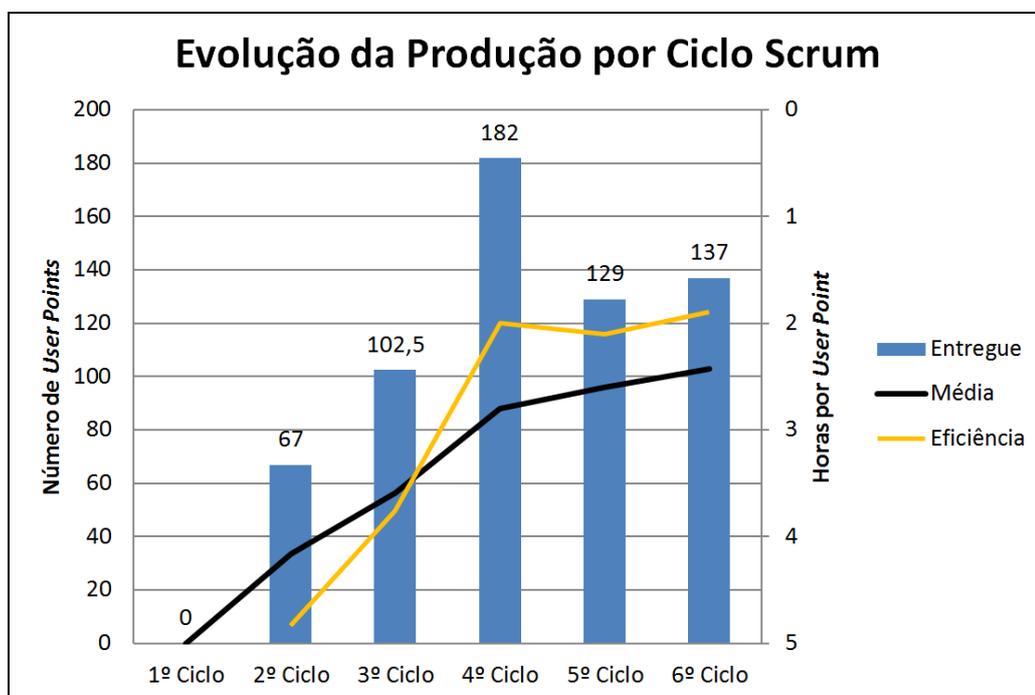


FIGURA 10 – EVOLUÇÃO DA PRODUÇÃO POR CICLO SCRUM: EMPRESA B.
FONTE: ELABORADO PELO AUTOR.

Com implantação do LSD, a partir do 4º ciclo, a equipe apresentou a produtividade média de **149 User Points** entregues, representando um aumento de **76,2%** em relação à média de produtividade entre o 2º e 3º ciclos, correspondendo à fase de implantação do Scrum.

Além disso, a partir do 4º ciclo, a quantidade de horas produtivas da equipe necessárias para entregar cada *User Point* reduziu significativamente, de **4,8 horas** no 2º ciclo para **1,9 horas** no 6º ciclo, evidenciando assim que as ações LSD contribuíram para aumento da produtividade e eficiência da equipe.

A Figura 11 apresenta a evolução da **fila de pendências** por ciclo. A partir do 4º ciclo, pode-se observar uma tendência de queda da fila de pendências do *backlog* do produto evidenciando que as ações LSD também impactaram na redução do fluxo de pendências que foram resolvidas e entregues e as novas pendências inseridas. O *backlog* do produto apresentou uma redução total de **42,0%** em relação a fase inicial, antes da aplicação do MALS.

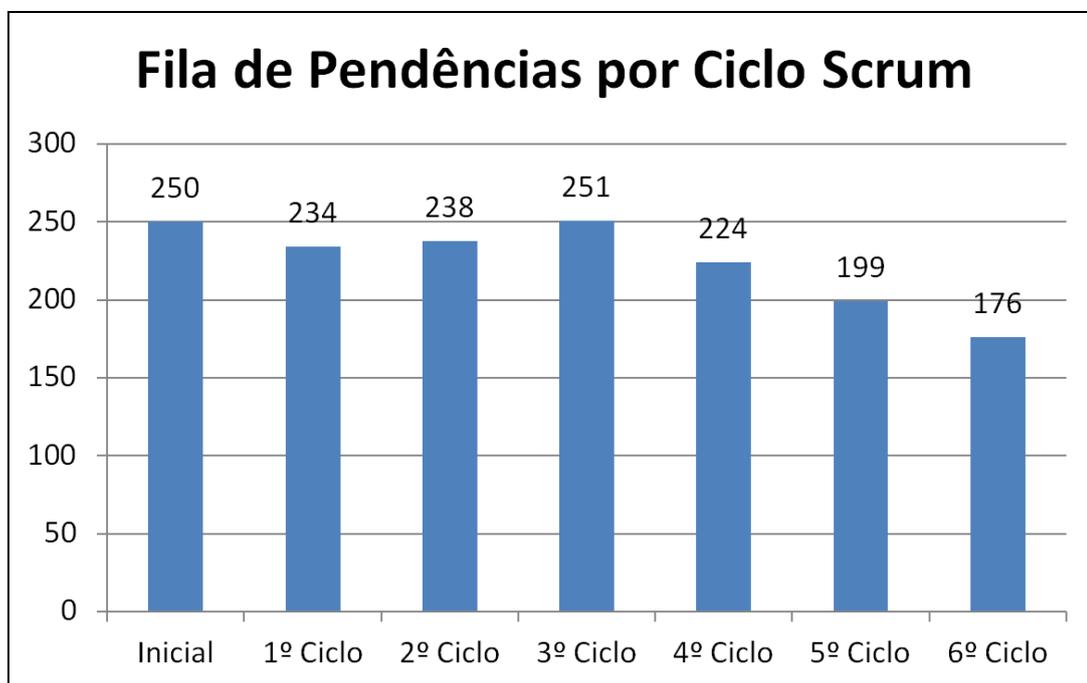


FIGURA 11 – EVOLUÇÃO DA FILA DE PENDÊNCIAS POR CICLO SCRUM: EMPRESA B.
FONTE: ELABORADO PELO AUTOR.

5.4 RESULTADOS IDENTIFICADOS COM A IMPLANTAÇÃO DO MALS

O Quadro 18 apresenta um resumo dos resultados identificados com a implantação do MALS nas unidades de análise.

Resultados	Empresa A	Empresa B
Produtividade	207 UP (Aumento de 39,9% em relação a 1ª Fase do MALS)	149 UP (Aumento de 76,2% em relação a 1ª Fase do MALS)
Fila de Pendências	-	Redução de 250 para 176 Pendências (42%)
Eficiência da Equipe	-	Redução de 4,8 Horas para 1,9 Horas por UP
Qualidade Produto	Releases Estáveis (eliminação de falhas no cliente)	Foco nas necessidades do cliente
Ciclo de Entrega do Produto	Redução de 3 meses para 2 semanas	Redução de 2 a 4 semanas para 2 semanas
Práticas de Gestão de Projetos	Implantadas	Implantadas
Práticas de Engenharia de Software	Implantadas	Implantadas
Integração e Comunicação da Equipe	Evolução	Evolução

QUADRO 18 – RESUMO DOS RESULTADOS COM A IMPLANTAÇÃO DO MALS.
 FONTE: ELABORADO PELO AUTOR.

Na empresa A, a produtividade apresentou crescimento de 39,9% em relação à 1ª fase do MALS e a empresa B apresentou crescimento de 76,2%.

A empresa B, apresentou uma redução de 42% das pendências da fila, além disso, apresentou um aumento da eficiência da equipe, evidenciado pela redução da quantidade de horas necessárias para entregar um *User Point* (UP).

A qualidade do produto, na empresa A, foi evidenciado pela eliminação de falhas identificadas no cliente. Na empresa B a qualidade do produto foi identificada pelo foco nas necessidades que agregam valor.

O ciclo de entrega do produto, na empresa A, reduziu de uma média de 3 meses para 2 semanas, assim como na empresa B que padronizou os ciclos de entrega também em 2 semanas.

Implantação de práticas de gestão de projetos, práticas de engenharia de *software* e a evolução da integração colaborativa da equipe foram realizadas em ambas empresas.

Além disso, a segunda fase do MALS revelou que algumas práticas LSD foram implantadas e consolidadas na primeira fase, ou seja, durante a implantação do Scrum. O Quadro 19 apresenta as práticas LSD implantadas na primeira fase do MALS, identificadas nas unidades de análise.

Etapa	Práticas LSD	Evento/Princípio Scrum
Princípio 2: Respeitar as Pessoas		
5. Transferir a responsabilidade e tomada de decisão para os mais baixos níveis possíveis	Oferecer condições para as equipes de trabalho criar valor; Fazer com que as equipes de trabalho projetem seus próprios processos.	Princípio Scrum, equipes motivadas e auto organizadas.
Princípio 3: Entregar Rápido		
8. Limitar o tamanho conforme a capacidade	Trabalhar em uma cadência repetível para estabelecer a capacidade; Esperar que as equipes puxem as filas com base em sua velocidade comprovada para terminar completamente o trabalho antes de começarem com mais trabalho; Limitar o tamanho das filas e não aceitar trabalho a menos que exista espaço vazio na fila.	Ciclo Scrum, por meio das <i>Sprints</i> permite uma cadência repetível estabelecendo assim a capacidade produtiva da equipe; Com base em técnicas de planejamento e estimativas como o <i>Planning Poker</i> e métricas como <i>User Points</i> é possível determinar a velocidade da equipe e limitar o trabalho de acordo com a capacidade.
Princípio 5: Criar Conhecimento		
13. Criar equipes construtoras de design	Fornecer a equipe a liderança e os incentivos adequados para manter o engajamento, a transparência e o <i>feedback</i> intensivo; Encorajar a equipe para compartilhar mais cedo e mais frequentemente, a falhar rápido e a aprender constantemente.	O engajamento, transparência, compartilhamento e <i>feedback</i> intensivo são garantidos nas reuniões diárias e reuniões de planejamento.
14. Manter uma cultura de melhoria constante	Criar o tempo e a expectativa de que cada equipe deve melhorar seus processos; Promover eventos multi-equipes e multifuncionais para identificar acomodações e restrições no fluxo de valor.	A cultura de melhoria constante é garantida nas reuniões de retrospectiva da <i>sprint</i> .
Princípio 7: Eliminar Desperdícios		
19. Fornecer liderança de mercado e liderança técnica	Garantir que exista uma responsabilidade e o entendimento do que o cliente valoriza associada ao entendimento do que a tecnologia pode entregar.	O princípio, cliente sempre presente, representado pelo <i>Product Owner</i> , e as reuniões de revisão da <i>sprint</i> tem o objetivo de entregar o que o cliente valoriza.

QUADRO 19 – EQUIVALÊNCIA SCRUM E PRÁTICAS LSD.

FONTE: ELABORADO PELO AUTOR.

A equivalência das práticas LSD e resultados apresentados com a implantação Scrum revelou um alinhamento das duas abordagens principalmente em relação a organização do trabalho, atribuindo à equipe de desenvolvimento maior responsabilidade e autonomia. Com isso possibilitou a integração da equipe, aprendizado coletivo e consciência para melhoria contínua do processo. Além disso, aproximou o cliente do processo produtivo facilitando a comunicação e o entendimento claro das necessidades.

Dessa forma, o processo de implantação LSD foi facilitado pela implantação do Scrum, pois muitas práticas foram previamente implantadas. Porém, não ficou evidenciado que todas as práticas LSD foram satisfeitas na implantação do Scrum. Práticas técnicas de engenharia de *software* como antecipação e automação dos testes, refatoração, integração contínua do código são definidas apenas no LSD.

Portanto, conclui-se que as abordagens possuem pontos comuns, mas são complementares.

6 CONCLUSÃO

Esse estudo avaliou, por meio da proposta de um método, como a integração dos princípios *Lean Thinking* pode contribuir para melhoria dos processos de produção de *software* em equipes que utilizam Scrum.

Os resultados evidenciaram que a utilização do Método de Aplicação *Lean Thinking* em projetos Scrum (MALS) é de fácil entendimento e serviu como facilitador para implantação e integração das abordagens. A evolução do MALS contou com a participação e contribuição das unidades de análise, possibilitando apoio e engajamento das equipes envolvidas.

A aplicação dos princípios de *Lean Thinking* para área de desenvolvimento de *software*, conhecido como *Lean Software Development* (LSD), mostrou-se possível e revelou que as abordagens LSD e Scrum são complementares. Esse alinhamento, também justifica o sucesso da utilização do MALS.

O MALS se divide em duas fases: a implantação da metodologia Scrum e a implantação dos princípios do LSD.

A implantação da metodologia Scrum nas unidades de análise facilitou o gerenciamento e a organização, sistematizando o planejamento do trabalho, facilitando o aprendizado coletivo e a rápida resposta aos problemas identificados. Porém, a metodologia Scrum não determina práticas técnicas para melhoria de todo o ciclo de vida de desenvolvimento de *software*.

Entretanto, o LSD indica práticas técnicas para melhoria de todo o fluxo de valor, ou seja, todo o ciclo de vida de desenvolvimento de *software*. Além disso, o aumento de produtividade das equipes envolvidas apenas foi possível com a integração de práticas LSD.

O MALS aplicado na unidade de análise A evidenciou a eliminação de falhas identificadas após a entrega do produto ao cliente e com a integração de práticas LSD houve um aumento de 39,9% na produtividade. Além disso, o

prazo de entrega das *releases* estáveis para o cliente reduziu para 1 (um) ciclo do Scrum, ou seja, 2 (duas) semanas. As *releases* estáveis representam as melhorias do produto desenvolvidas para atender as necessidades e solicitações de clientes, que não apresentam pendências de controle de qualidade e pode ser distribuída para todos os clientes. A unidade de análise A relatou que antes da implantação do MALS, o prazo médio para entrega de um *release* estável era em torno de 3 (três) meses.

A implantação das práticas LSD contribuiu para melhorar o entendimento de todo o fluxo de valor, identificando interrupções no fluxo que não ficaram evidentes durante a implantação do Scrum. Dessa forma, foi possível identificar na unidade de análise A melhorias externas ao processo de desenvolvimento do produto, ou seja, melhorias organizacionais, como a realocação de tarefas e mudança da cultura organizacional, com foco para melhoria contínua e em agregar valor ao cliente.

A mudança da cultura organizacional possibilitou o apoio da alta administração, para implantar melhorias no processo de produção, como por exemplo, a antecipação de planos de testes, intensificação dos testes unitários, disponibilidade da equipe para refatorar o código e automatizar rotinas de trabalho.

O apoio da alta administração foi fundamental para efetividade da implantação do MALS, pois os resultados começam a ficar evidentes após alguns ciclos, com o domínio do método pela equipe e a sincronização do fluxo, dentro do fluxo de valor.

Na unidade de análise B, os resultados com a implantação do MALS também evidenciou aumento na produtividade da equipe após a implantação do LSD. O aumento foi de 76,2% em relação à fase de implantação do Scrum. Além disso, possibilitou a redução da fila de pendências com clientes em 42,0%, evidenciado principalmente após a aplicação das práticas LSD.

A fila de pendências da unidade de análise B impossibilitava o gerenciamento e controle das entregas (*releases*). Portanto, as ações de melhoria tiveram foco em reduzir as filas de pendências, para possibilitar a sincronização do trabalho solicitado e executado dentro de um mesmo ciclo Scrum e assim determinar a capacidade produtiva da equipe. Para isso, a aproximação do cliente na avaliação e especificação das necessidades foi fundamental, possibilitando o direcionamento das atividades de produção para atividades que agreguem valor.

A implantação do MALS nas unidades de análise evidenciou uma melhoria nos indicadores de produtividade, qualidade e rapidez na entrega das necessidades de clientes, principalmente após a implantação do LSD, analisando 3 (três) ciclos de implantação. Porém, revelou também que, os 3 (três) ciclos de implantação do LSD, proposto nesse projeto para coleta dos resultados, são insuficientes para implantação de todos os princípios.

Práticas LSD, como substituição das medidas locais, por medidas globais que envolvem todo o fluxo de valor não puderam ser avaliadas, sendo considerada uma limitação desse projeto de pesquisa.

A pesquisa identificou similaridades entre as abordagens possibilitando, com a implantação do Scrum, implantar alguns princípios e práticas previstas no LSD, tais como: a valorização e integração da equipe, o aprendizado coletivo, a cultura de melhoria constante, a cadência de produção e o foco no cliente. Devido a isso, considera-se que a implantação do Scrum prepara e facilita a implantação do LSD.

O procedimento de pesquisa utilizado nesse estudo, pesquisa-ação, foi fundamental para alcançar os resultados apresentados, pois permitiu a inserção participativa do pesquisador no processo de implantação e integração das abordagens. Essa inserção contribuiu de forma efetiva, por meio da pesquisa de práticas para melhoria do processo, aplicação e avaliação dos resultados em ambiente de real de produção. Com isso, as sugestões de melhoria propostas pela equipe nas entrevistas coletivas e observação

participante, refinaram o modelo conceitual teórico resultando em um método genérico que pode ser utilizado para facilitar os trabalhos de implantação e integração do Scrum e LSD.

6.1 PROPOSTA PARA TRABALHOS FUTUROS

A partir do que foi pesquisado e desenvolvido no MALS somado às observações e interações realizadas no processo de pesquisa-ação, podem-se propor alguns trabalhos futuros de pesquisa:

- Ficou evidenciado que a implantação do Scrum pode agregar valor para a melhoria do processo de desenvolvimento de *software* independente da integração com LSD, porém a pesquisa não possibilitou avaliar se a aplicação LSD sem o Scrum é eficiente;
- Os princípios LSD alteram as práticas operacionais de produção de *software*, mas principalmente a cultura da empresa, envolvendo todo o fluxo de valor. Entende-se que os resultados dessas mudanças apenas poderão ser avaliados em período de longo prazo, após o amadurecimento do processo. Portanto, avaliar o nível de adequação do LSD em período de longo prazo (1 ano ou mais) pode revelar novos resultados e contribuições;
- Ampliar essa pesquisa, com a aplicação em outras unidades de análise, pode evoluir o MALS e revelar novas oportunidades de melhoria do processo de desenvolvimento de *software*, até então não identificadas.

REFERÊNCIAS BIBLIOGRÁFICAS

AGILE ALLIANCE. **Manifesto for agile software development**. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em 08 de julho de 2016.

BELL, S.C.; ORZEN, M. A. **Lean IT: enabling and sustaining your lean transformation**. New York, NY: CRC Press, 2011.

CANTANHEDE, M.A.D. Lean Thinking em Desenvolvimento de Software Estudo e Aplicação de Ferramenta para Avaliação do Lean em Software. Faculdade de Tecnologia da Universidade Estadual de Campinas, 2014. **Dissertação** Mestrado em Tecnologia.

CARDOSO, R. Análise da Aplicabilidade dos Princípios e Ferramentas da Produção Enxuta para Melhoria da Gestão de Processos Operacionais de Educação a Distância em Instituições de Ensino Superior. Santa Bárbara d'Oeste: Universidade Metodista de Piracicaba, 2010. 103 p. **Dissertação** Mestrado em Engenharia de Produção.

CONFORTO, E.C. Modelo e ferramenta para avaliação da agilidade no Gerenciamento de Projetos. 2013. **Tese** (Doutorado em Engenharia de Produção)-Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2013.

COGHLAN, D.; BRANNICK, T. **Doing action research in your own organization**. 2.ed. London: Sage, 2008.

COUGHLAN, P.; COGHLAN, D. **Action research for operations management**. International Journal of Operations & Production Management. v. 22, n. 2, p.220-240, 2002.

DEMO, P. **Metodologia Científica em Ciências Sociais**. 3. ed. rev. e ampl. São Paulo: Atlas, 1995.

DINGSOYR, T. *et al.* A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, 2012.

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

ETTINGER, D. **A engrenagem do SCRUM**. Disponível em: <<http://danielettinger.com/2011/04/06/a-engrenagem-do-scrum/>>. Acesso em 04 de julho de 2016.

FERRANCE, E. **Themes in education: Action research**. Brown University: Educational Alliance, 2000.

FONSECA, I; CAMPOS, A. Por que SCRUM ?. **Engenharia de Software Magazine**, DevMedia, p. 30-35, 2008.

GANNON, M. An agile implementation of SCRUM. **IEEE Aerospace Conference Proceedings**, 2013. p. 1-7.

GIL, A.C. **Como elaborar projeto de pesquisa**. São Paulo: Editora Atlas, 2009

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

GIL, A.C. **Métodos e técnicas de pesquisa social**. São Paulo: Editora Atlas, 2011.

HINES, P.; HOLWEG, M.; RICH, N. Learning to evolve: A review of contemporary Lean Thinking. **International Journal of Operations & Production Management**, 2004. v.24. Issue 10. p.994-1011.

HODA, R.; NOBLE, J.; MARSHALL, S. Self-organizing roles on agile software development teams. **IEEE Transactions on Software Engineering**, 2013.

HOLDEN, R.J. Lean Thinking in Emergency Departments: A Critical Review. **Annals of Emergency Medicine**, 2011. v.57. p.265-278.

HOLWEG, M. The genealogy of lean production. **Journal of Operations Management**, 2007. p. 420-437.

HOSSAIN, E.; BABAR, M. A.; AIK, H. P. H. Using Scrum in Global Software Development: A Systematic Literature Review. **IEEE International Conference on Global Software Engineering (ICGSE)**, 2009. p. 175-184.

HIGHSMITH, J.; COCKBURN, A. Agile software development: The business of innovation. **Computer**, 2001. v.34. Issue 9. p.120-127.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. EBSE Technical Report, version 2.3. School of Computer Science and Mathematics, Keele University, UK, 2007.

LAGERBERG, L.; SKUDE, T.; EMANUELSSON, P.; SANDAHL, K.; STAHL, D. **The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson**. Paper presented at the International Symposium on Empirical Software Engineering and Measurement, 2013.

MANN, C.; MAURER, F. **A case study on the impact of Scrum on overtime and customer satisfaction**. Agile Development Conference, 2005.

MARTINS, R. A. **Abordagens quantitativa e qualitativa**. In: Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações. Cap. 3. Paulo A. C. Miguel (org.). Rio de Janeiro: Elsevier, 2010.

MEHTA, M.; ANDERSON, D.; RAFFO, D. **Providing value to customers in software development through lean principles**. Software Process Improvement and Practice, 2008.

MELLO, C.H.P *et al.* Pesquisa-ação na engenharia de produção: proposta de estruturação para sua condução. **Revista Produção**, v.22. n.1. p.1-13. jan/fev 2012.

MIGUEL, P.A.C. Estudo de caso na engenharia de produção: estruturação e recomendações para sua condução. **Revista Produção**. v.17. n.1. p.216-229, jan/abr 2007.

MIDDLETON, P.; FLAXEL, A.; COOKSON, A. **Lean software management case study: Timberline inc**. Computer Science, 2005.

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

MISRA, S. C.; KUMAR, V.; KUMAR, U. Identifying some critical changes required in adopting agile practices in traditional software development projects. **International Journal of Quality and Reliability Management**, 2010.

NAKANO, D. N. **Métodos de pesquisa adotados na Engenharia de Produção e Gestão de Operações**. In: Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações. Cap. 4. Paulo A. C. Miguel (org.). Rio de Janeiro: Elsevier, 2010.

NONAKA, I.; TAKEUCHI, H. **Criação e conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação**. Rio de Janeiro: Campus, 1997.

NOOR, T.B. et al. **A Novel Approach to Implement Burndown Chart in Scrum Methodology**. Dept. of Computer Science & Engineering, Stamford University Bangladesh, 2012. v.2. Issue 10. p. 421-427.

OHNO, T. **O Sistema Toyota de Produção: além da produção em larga escala**. Porto Alegre: Bookman, 1997.

PERERA, G. I. U. S.; FERNANDO, M. S. D. Enhanced agile software development - hybrid paradigm with LEAN practice. **ICIIS 2007 - International Conference on Industrial and Information Systems**, 2007.

PERUCCI, C. C.; CAMPOS, F. C. *Lean Thinking* em Processos de Produção de *Software*: uma Revisão Sistemática de Literatura. V Congresso Brasileiro de Engenharia de Produção. **CONBREPRO**, 2015.

PETERSEN, K.; WOHLIN, C. **Measuring the flow in lean software development**. *Software - Practice and Experience*, 2011.

PETERSEN, K.; WOHLIN, C. Software process improvement through the lean measurement (SPI-LEAM) method. **Journal of Systems and Software**, 2010.

POPPENDIECK, M.: Lean Software Development. **29th International Conference on Software Engineering**. IEEE, 2007.

POPPENDIECK, M.; CUSUMANO, M. A. Lean Software Development: A Tutorial. **IEEE Software**, September/October 2012. p.26-32.

POPPENDIECK, M.; POPPENDIECK, T.; **Implementando o desenvolvimento Lean de software: do conceito ao dinheiro**. Porto Alegre: Bookman, 2011. 280p.

PORTIOLI-STAUDACHER, A. Lean Implementation in Service Companies. **International Federation for Information Processing**, 2010. v.338. p.652-659.

PRESSMAN, R. **Engenharia de Software**. 6.ed. São Paulo: McGrawHill, 2006.

RISING, L.; JANOFF, N. S. (2000). The Scrum Software Development for Small Teams. **IEEE Software**, 2000. p. 26-32.

SCHWAB, D. P. **Research methods for organizational studies**. 2.ed. London: Routledge, 2005.

SCHWABER, K. **Agile Project Management with SCRUM**. Microsoft Press, 2004.

SCRUM ALLIANCE. **The State Of Scrum: Benchmarks and Guidelines**. Jun. 2013. Disponível em: <[https:// www.scrumalliance.org/](https://www.scrumalliance.org/)>. Acesso em 04 de agosto de 2016.

SHINGO, S. **O Sistema Toyota de Produção do ponto de vista da Engenharia de Produção**. 2. ed. Porto Alegre: Bookman, 1996.

SCROCCO, J. H. T. C.; MACEDO, P.C. **Metodologias Ágeis: Engenharia de Software sob Medida**. 1. ed. São Paulo: Érica, 2012.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

STAATS, B. R.; BRUNNER, D. J.; UPTON, D. M. Lean principles, learning, and knowledge work: Evidence from a software services provider. **Journal of Operations Management**, 2011.

STARR, D. **The Lean of Scrum**. MSDN, 2012. Disponível em: <[https://msdn.microsoft.com/en-us/library/jj161049\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/jj161049(v=vs.120).aspx)>. Acesso em 24 de julho de 2016.

SUGIMORI, Y.; KUSUNOKI, K.; CHO F., UCHIKAWA S. Toyota productions system and Kanban system Materialization of just-in-time and respect-for-human system. **International Journal of Production Research**, 1977. p. 553-564.

SUTHERLAND, J. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**. Tradução Natalie Gerhardt. São Paulo: LeYa, 2014.

SUTHERLAND, J.; SCHWABER, K. **SCRUM Guide**. Scrum.org, 2013. Disponível em: <<http://www.scrumguides.org>>. Acesso em 05 de julho de 2016.

TAKEUCHI, H.; NONAKA I. The New New Product Development Game. **Harvard Business Review**, 1986. p. 285-305.

THIOLLENT, M. **Metodologia da Pesquisa-Ação**. 10. ed. São Paulo: Cortez, 2000.

TRANFIELD, D.; DENYER, D.; SMART, P. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. **British Journal of Management**, v.14, n.3, p.207-222, 2003.

TURRIONI, J.B.; MELLO, C.H.P. **Metodologia de Pesquisa em Engenharia de Produção**. Universidade Federal de Itajubá, 2012.

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**, 2012.

WOMACK, J.; JONES, D. T. **A Mentalidade Enxuta nas Empresas – Lean Thinking: Elimine o Desperdício e Crie Riqueza**. 11. ed. Rio de Janeiro: Elsevier, 2004.

WOMACK, J.; JONES, D. T.; ROOS, D. **A máquina que mudou o mundo: baseado no estudo do Massachusetts Institute of Technology sobre o futuro do automóvel**. Tradução Ivo Korytowski. Rio de Janeiro: Elsevier, 2004.

WOMACK, J.; JONES, D. T.; ROOS, D. **The machine that changed the world: based on the Massachusetts Institute of Technology 5-million dollar 5-year study on the future of the automobile**. Rawson Associates, 1990.

APÊNDICES

APÊNDICE A – QUADRO RESUMO DA PESQUISA-AÇÃO EMPRESA A

Ciclo	Período	Fase do MALS	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
-	12/05/2016	1. Abertura do Projeto	Aceitação da Unidade de análise; Questionário de Avaliação MALS.	Apresentação da proposta MALS para os envolvidos; Avaliação Inicial do MALS; Apresentação da equipe e das instalações da empresa.	Aprovado implantação MALS na Empresa A pelo responsável-legal da empresa; Técnicas utilizadas para especificação de requisitos. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Sim.	Estudar as ferramentas Taiga.IO; JIRA e TFS para definir as ferramentas que suportam o Scrum; Estudar as técnicas de especificação de requisitos: <i>Epics, Features e User Story.</i> Pontos Melhoria do MALS: Equipe de projeto solicitou que a definição das técnicas para visualização e acompanhamento do trabalho; e as práticas para especificação de requisitos e estimativa de esforço fossem definidas no Treinamento Scrum.
-	12/05/2016	2. Avaliação Inicial Scrum	Questionário de Adequação Scrum	Avaliação do Nível de Adequação Scrum.	Empresa A não conhece ou utiliza Scrum.	Equipe não utiliza Scrum, necessária treinamento conforme fase 3 do Scrum.
-	19/05/2016	3. Treinamento Scrum	Definição das práticas de Engenharia de Software	Treinamento do Scrum (1h); Análise das ferramentas Taiga.IO, JIRA e TFS.	Definido que o TFS é a ferramenta que será utilizada, para monitoramento do Scrum, pois disponibiliza recursos aderentes a metodologia (Gráfico <i>BurnDown e TaskBoard</i>), além disso a equipe tem experiência com a ferramenta; Definido o uso de <i>User Stories</i> para especificação dos requisitos e <i>User Points</i> para avaliação do esforço; Definido o uso da técnica de <i>Planning Poker</i> para estimativas dos <i>User Points</i> .	Avaliar a capacidade produtiva da equipe destinada ao projeto. Analisar o escopo do projeto.

(CONTINUA)

Ciclo	Período	Fase do MALS	Crerios de avaliaão	Atividades executadas	Avaliaão	Melhoria e aprendizagem
-	27/05/2016	4. Definir a Equipe Scrum	Definiao da equipe e responsabilidades; Definiao Ciclo Scrum	Planejamento do Ciclo Scrum; Definir a estrategia de implantacao Scrum.	Definido a equipe de projeto Scrum Master (SM), Product Owner (PO) e a Equipe. Definido que o Ciclo Scrum (Sprint) sera de 2 semanas: <ul style="list-style-type: none"> Sprint Planning (Planejamento da Sprint): Primeira segunda-feira (manha) da Sprint; Envolvidos: SM+PO+TP. Revisao Sprint: Ultima sexta-feira (manha). Envolvidos: SM+PO+TP. Retrospectiva da Sprint: Ultima sexta-feira (manha). Envolvidos: SM+TP Primeiro Sprint deve iniciar dia 30/05/2016.	Definir o BackLog do Produto; Definir a dedicacao do time de projeto.
-	27/05/2016 a 30/05/2016	5. Definir o BackLog do Produto	Maturidade do Backlog do produto	Definiao do BackLog do Produto	BackLog do Produto Aprovado em 29/05/2016 pelo PO.	Nao identificado aoes de melhoria.

Ciclo	Período	Fase do MALS	Crerios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
1	30/05/2016 a 10/06/2016	6. Executar o <i>Sprint</i> #1; 7. Avaliar o Nível de Maturidade Scrum	Adaptação da Equipe de Projeto; Resultado das cerimônias <i>Sprint</i> ; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	Pontos Positivos abordados pela equipe: Melhoria na comunicação e integração da equipe ocasionada pela Reunião Diária; Rápido tempo de reação perante um problema; Ferramenta TFS facilitou o trabalho da equipe de acompanhamento das tarefas; Abordagens da Equipe em relação à metodologia SCRUM: "Melhorou o conceito de equipe", "Coragem para resolver problemas, como requisitos que estavam pendente a quase dez anos", "Direcionou o trabalho para projeto e não apenas para pendências de cliente", "Foco no cliente, retorno diretamente para o cliente" Nível de Maturidade Scrum: 1. O Dono do Produto (Product Owner – PO) está definido ? Sim. 2. A Equipe do projeto está definida ? Sim. 3. O Scrum Master está definido ? Sim. 4. O BackLog do Produto está criado e priorizado ? Parcialmente. 5. O esforço dos requisitos do BackLog do Produto estão estimados ? Parcialmente. 6. O planejamento da <i>Sprint</i> está sendo realizado em períodos definidos ? Sim. 7. Existem técnicas para visualização e acompanhamento do trabalho que está sendo realizado ? Sim. 8. Reunião de acompanhamento diário estão sendo realizadas ? Sim. 9. O resultado dos trabalhos realizados estão sendo demonstrados para equipe e o PO ? Parcialmente. 10. O <i>Sprint</i> está sendo avaliado sistematicamente, promovendo ciclos de melhoria contínua ? Parcialmente. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Sim.	Pontos Melhoria apontados pela equipe: a) Análise preliminar antes do planejamento; b) Detalhamento da <i>User Story</i> ; c) Treinamento detalhado no TFS; d) Maior precisão na estimativa de esforço para garantir a entrega no <i>Sprint</i> (User Points Planejado <i>Sprint</i> #1: 240, realizado: 170), todos concordaram que algumas <i>stories</i> foram super dimensionadas enquanto outras foram subdimensionadas. Pontos Melhoria do MALS: Equipe julgou conveniente não fixar o número de ciclos Scrum antes de iniciar a implantação da segunda etapa do MALS.

Ciclo	Período	Fase do MALS	Crerios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
2	13/06/2016 a 24/06/2016	6. Executar o <i>Sprint</i> #2; 7. Avaliar o Nível de Maturidade Scrum	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	Melhorias Implementadas da última reunião: a) Análise preliminar antes do planejamento; b) Detalhamento da User Story; c) Treinamento detalhado no TFS. Pontos Positivos abordados pela equipe: a) Reuniões diárias com resultados positivos, permitindo a visão do processo, algumas reuniões com menos 15 minutos; b) Todos os requisitos planejados foram apresentados na revisão da <i>Sprint</i> ; c) Mantido a motivação e maior integração da equipe. Discussões sobre problemas da <i>Sprint</i>: Não entrega de todas as tarefas, principalmente os testes: a) Sugeriu evitar alterações em requisitos depois de liberado para testes, para evitar retrabalho; PO comentou sobre existir pendências de testes e a <i>release</i> ainda não liberado; Nível de Maturidade Scrum (Questões 1-10): 1.Sim.; 2.Sim.;3.Sim.; 6.Sim; 7.Sim; 8.Sim; 4. O BackLog do Produto está criado e priorizado ? Sim. 5. O esforço dos requisitos do BackLog do Produto estão estimados ? Parcialmente. 9. O resultado dos trabalhos realizados estão sendo demonstrados para equipe e o PO ? Sim. 10. O <i>Sprint</i> está sendo avaliado sistematicamente, promovendo ciclos de melhoria contínua ? Sim. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.	Pontos Melhoria apontados pela equipe: a) Sugeriu evitar alterações em requisitos depois de liberado para testes, para evitar retrabalho; b) Uso ferramenta Sublime para controle dos casos de teste; Próxima Reunião de Planejamento de <i>Sprint</i> deve considerar as atividades de testes pendentes e minimizar ações de desenvolvimento de novos requisitos para gerar um alinhamento das atividades de desenvolvimento e testes com entregas de releases no <i>Sprint</i> . Sugestões de Melhoria de processo: a) Entregar as Stories dentro da <i>sprint</i> , utilizar o próximo <i>sprint</i> para propiciar esse alinhamento; b) Se as <i>stories</i> forem cumpridas no <i>sprint</i> , liberar <i>release</i> por <i>sprint</i> ; c) Manter o levantamento da capacidade produtiva da equipe, gerar uma média de produtividade (projeto) e buffer;

Ciclo	Período	Fase do MALS	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
3	27/06/2016 a 08/07/2016	6. Executar o <i>Sprint</i> #3; 7. Avaliar o Nível de Maturidade Scrum	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	Melhorias identificadas na <i>Sprint</i> anterior: a) Não realizado alterações em requisitos depois de liberado para testes; b) Uso da Ferramenta sublime facilitou o controle de casos de testes. Pontos Positivos abordados pela equipe: a) Manutenção da efetividade das reuniões diárias; b) Todos os requisitos planejados foram entregues na revisão da <i>Sprint</i> ; Observações Documentais: Linha Realizada do Gráfico <i>BurnDown</i> , muito próxima com Linha Ideal; Análise do <i>TaskBoard</i> : Todos as estórias foram entregues. Histórico de User Points: <i>Sprint</i> #1. Planejado: 240 / Entregue: 170 <i>Sprint</i> #2. Planejado: 200 / Entregue: 160 <i>Sprint</i> #3. Planejado: 165 / Entregue: 180 Nível de Maturidade Scrum (Questões 1-10): 1.Sim.; 2.Sim.;3.Sim.; 4.Sim; 6.Sim; 7.Sim; 8.Sim; 9.Sim; 10.Sim. 5. O esforço dos requisitos do BackLog do Produto estão estimados ? Parcialmente. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.	Pontos Melhoria apontados pela equipe: a) Scrum Master sugeriu um adaptação no método do Planning Poker, utilizando a classificação por complexidade antes da votação das estórias (Alta; Média; Baixa).

Ciclo	Período	Fase do MALS	Crítérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
4	11/07/2016 a 22/07/2016	6. Executar o <i>Sprint</i> #4; 7. Avaliar o Nível de Maturidade Scrum	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico BurnDown e TaskBoard; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	<p>Melhorias identificadas na <i>Sprint</i> anterior: Funcionou e manter a classificação por complexidade; Pontos Positivos abordados pela equipe: a) Comunicação da Equipe (sempre melhorou); Observações Documentais: Linha Realizada do Gráfico BurnDown, desalinhada com Linha Ideal; Análise do TaskBoard: Pendências em relação aos testes</p> <p>Problemas Identificados na <i>Sprint</i>: Forte queda de entrega de <i>User Points</i> (182 planejados e 72 entregues);</p> <p>Discussões sobre problemas da <i>Sprint</i>: Ausência de um integrante da equipe de testes por motivo de férias; Atualização <i>release</i> muito antigo (102 → 174) provocou muitas interrupções na equipe; Ocorreram alguns problemas de ambiente; Sobrecarga de trabalho de testes;</p> <p>Nível de Maturidade Scrum: Em virtude do desequilíbrio observado na produtividade da equipe (<i>User Points</i>), planejado para próxima reunião de retrospectiva o treinamento Princípios <i>Lean</i>.</p> <p>Histórico de User Points: <i>Sprint</i> #4. Planejado: 182 / Entregue: 72</p> <p>Nível de Maturidade Scrum (Questões 1-10): 1.Sim.; 2.Sim.;3.Sim.; 4.Sim; 6.Sim; 7.Sim; 8.Sim; 9.Sim; 10.Sim.</p> <p>5. O esforço dos requisitos do BackLog do Produto estão estimados ? Parcialmente. Reavaliar o processo de estimativa de esforço.</p> <p>Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.</p>	<p>Ações Sugeridas pela Equipe: Planejar o próximo <i>Sprint</i> para sincronizar as atividades de desenvolvimento e controle de qualidade.</p> <p>Pontos Melhoria apontados pela equipe: a) Antecipar o Plano de Testes (Testar mais cedo, encontrar a falha mais cedo) para apoiar os desenvolvedores no auxílio dos testes; b) Sincronização das tarefas de testes e desenvolvimento; c) Entregar dentro da <i>Sprint</i>, todas as tarefas concluídas; d) Automatizar testes; e) Reduzir bugs produzidos na produção, avaliar a quantidade de falhas detectadas após desenvolvimento; Obs.: Falhas detectadas no controle de qualidade são retrabalho; f) Entregar <i>releases</i> mais rápido (dentro da <i>sprint</i>), isso evita o problema da atualização de um <i>release</i> muito antigo; g) Valor para o cliente, evitar criar funcionalidades excedentes (caso de apresentar a foto).</p>

Ciclo	Período	Fase do MALS	Crerios de avaliaço	Atividades executadas	Avaliaço	Melhoria e aprendizagem
5	25/07/2016 a 05/08/2016	6. Executar o <i>Sprint</i> #5; 7. Avaliar o Nvel de Maturidade Scrum	Resultado das cerimnias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionrio de Adequao Scrum; Questionrio de Avaliao MALS.	Cerimnias da <i>Sprint</i> ; Avaliar o nvel de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Anlise do Grfico BurnDown e TaskBoard; Avaliao do Nvel de Adequao Scrum; Avaliao da Proposta do MALS; Treinamento Lean.	Melhorias identificadas na <i>Sprint</i> anterior: a) Realizado aes de antecipao do plano de testes; b) Foco com a sincronizao das atividades de desenvolvimento e controle de qualidade. Pontos Positivos abordados pela equipe: a) Comunicao/Integrao da Equipe (sempre melhorou); b) Estabilidade do release 176 – no identificado falhas no cliente (ltimo release estvel 102 de fev/2016); c) Disponvel para entrega o release 177 (referente ao <i>Sprint</i> #5). Observaes Documentais: Linha Realizada do Grfico <i>BurnDown</i> alinhada com a planejada evidenciando a entrega da maioria das tarefas; Anlise do TaskBoard: As tarefas executadas no <i>Sprint</i> tiveram foco para estabilizar as atividades de controle de qualidade. Discussões sobre problemas da <i>Sprint</i>: No ocorreram problemas significativos no <i>Sprint</i> . Histrico de User Points: (<i>vide reunies anteriores</i>) <i>Sprint</i> #5. Planejado: 270 / Entregue: 156 Nvel de Maturidade Scrum (Questões 1-10): 1.Sim.; 2.Sim.;3.Sim.; 4.Sim; 5.Sim; 6.Sim; 7.Sim; 8.Sim; 9.Sim; 10.Sim. Avaliao MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantao do MALS ? No. A equipe avalia oportunidade de melhorias no MALS ? No. Treinamento Preliminar Lean (Presente na Reunião Product Owner - PO e Scrum Master - SM). O foco da reunio foi discutir o processo de implantao Scrum at o momento e treinamento preliminar Lean, PO afirmou que ainda no observou uma melhoria da produtividade da equipe. Scrum Master apresentou alguns indicadores de melhoria como a disponibilidade de releases a partir a <i>Sprint</i> #4 (prncpio entregar rpido) e a melhoria da qualidade observado com a ausncia de falhas observadas no cliente no ltimo release (prncpio desperdicio); PO sugeriu aguardar prximos ciclos e avaliar a implantao do Lean para observar a evoluo da produtividade da equipe.	Aes Sugeridas pela Equipe: a) Para sincronizar as atividades de controle de qualidade e desenvolvimento, sugerido que as demonstraes sejam realizadas pela equipe de suporte; b) Para facilitar a comunicao das atividades que esto sendo realizadas no <i>Sprint</i> , liberar o TFS para o PO; c) Estudo e prticas com TFS; Integrao do TFS com o Sistema Verses; Definir claramente para que utilizar o sistema Verses e o TFS. Implantao MALS: Nvel de da primeira fase atingido, planejado treinamento <i>Lean Software Development</i> (LSD) para 12/08/2016 com toda equipe.

Ciclo	Período	Fase do MALS	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
-	12/08/2016	9. Treinamento dos Princípios LSD; Identificar as Oportunidades de Melhoria	Oportunidades de melhoria	Treinamento Lean (1h); Identificar as oportunidades de melhoria.	Treinamento Lean realizado: Princípios Lean; e práticas <i>Lean Software Development</i> (LSD). Identificar as oportunidades de melhoria: a) Definido que a equipe deveria estudar os princípios <i>Lean</i> e trazer para próxima reunião de retrospectiva no mínimo uma sugestão de melhoria relacionado aos princípios e práticas LSD.; b) Discussão das oportunidades de melhoria com SM e PO.	Melhorias de Processo Identificado: a) Desenhar e Estudar o fluxo de valor (PO) e procurar interrupções no fluxo. Foco identificar oportunidades de melhoria para estabilizar as atividades de desenvolvimento e controle de qualidade; b) Nomeado o <i>Scrum Master</i> e <i>Product Owner</i> como equipe de liderança do fluxo de valor; Obs.: Foram identificados algumas práticas LSD implantadas no processo Scrum.
6	08/08/2016 a 19/08/2016	6. Executar o <i>Sprint</i> #6 10. Identificar as Oportunidades de Melhoria LSD 12. Avaliar os resultados das ações de melhoria LSD	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias identificadas na <i>Sprint</i> anterior: a) Aprovado pela alta direção a liberação das atividades de demonstração da equipe de controle de qualidade; b) Decidido não liberar o TFS para equipe de suporte, pois identificado necessidade de treinamento da equipe para evitar comprometimento equivocado com cliente (prazo de entrega). Pontos Positivos abordados pela equipe: a) Estabilidade do release 177 – não identificado falhas no cliente (referente ao <i>Sprint</i> #5); b) A expressão utilizada nas primeiras reuniões “bug volta”, decorrente de problemas de comunicação e encadeamento de releases não ocorreu nos últimos 2 sprints (#5 e #6); Observações Documentais: Linha Realizada do Gráfico <i>BurnDown</i> um desalinhamento no 3o dia da <i>sprint</i> devido a inclusão do requisito “Importador”; Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues. Histórico de User Points: <i>Sprint</i> #6. Planejado: 294 / Entregue: 182. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.	Melhorias de Processo Identificado pela equipe (baseado nos princípios LSD): a) Reduzir a troca de tarefas, para evitar o desperdício. b) Foco em sincronizar as atividades de desenvolvimento e controle de qualidade; c) Filosofia de “erro zero”, orientação efetuada pelo <i>Scrum Master</i> para eliminar os erros de desenvolvimento entregues ao controle de qualidade, ocasionando retrabalho; d) Filosofia de “refatorar sempre”, melhorando e simplificando o código sempre que possível; e) Definido como meta redução das filas de pendências/bugs e Sugestões de Melhoria; f) Estudo do TDD para automação dos testes e futuramente implantar no novo sistema.

Ciclo	Período	Fase do MALS	Crerios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
7	22/08/2016 a 02/09/2016	6. Executar o <i>Sprint #7</i> 10. Identificar as Oportunidades de Melhoria LSD 12. Avaliar os resultados das ações de melhoria LSD	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias identificadas na <i>Sprint</i> anterior: Mantidas práticas de redução de troca de tarefas e foco em sincronizar as atividades de controle de qualidade; mantidas as filosofia “erro zero” e “refatorar sempre”; Definido responsável para fazer estudo do TDD para automação dos testes; Não identificado na <i>sprint</i> o levantamento das filas de pendências/bugs e Sugestões de Melhoria. Pontos Positivos abordados pela equipe: a) Utilização do plano de testes pela equipe de desenvolvimento identificou falhas em codificação, reduzindo entrega de falhas para controle de qualidade; b) Release 179 – não identificado falhas no cliente (referente a <i>Sprint #6</i>); Pontos Negativos Identificados na <i>Sprint</i>: a) Release estável não liberado durante a <i>Sprint</i> ; b) Aumento da fila de pendências por falhas de produtos. Discussões sobre problemas na <i>Sprint</i>: Ausência de um membro da equipe de controle de qualidade por motivo de férias e feriado municipal afetaram a entrega da <i>release</i> estável. Observações Documentais: Gráfico <i>BurnDown</i> revelou a maior parte do tempo a linha ideal não foi atendida; Análise do <i>TaskBoard</i> : Tarefa entregues parcialmente. Histórico de User Points: (<i>vide reuniões anteriores</i>) <i>Sprint #7</i> . Planejado: 331 / Entregue: 203. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim . A equipe compreende as fases do MALS (<i>Scrum/LSD</i>) ? Sim . A equipe identifica algum impedimento para implantação do MALS ? Não . A equipe avalia oportunidade de melhorias no MALS ? Não .	Pontos Melhoria apontados pela equipe (baseado nos princípios LSD): a) Sincronizar: a1) Para reduzir o trabalho inacabado e sincronizar as atividades de controle de qualidade a equipe decidiu atribuir à equipe de desenvolvimento as atividades de levantamento de informações com cliente, verificação e entrega para os módulos “importador”; b) Entregar rápido: b1) Para reduzir as filas de pendências, foi definido que será avaliado o fluxo de falhas que entram e saem da fila; c) Refatorar: Integrado a equipe <i>Scrum</i> líder técnico, com objetivo de reduzir a complexidade do código e criar soluções técnicas mais simples.

Ciclo	Período	Fase do MALS	CrITÉrios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
8	05/09/2016 a 16/09/2016	6. Executar o <i>Sprint</i> #8 10. Identificar as Oportunidades de Melhoria LSD 12. Avaliar os resultados das ações de melhoria LSD	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias identificadas na <i>Sprint</i> anterior: Atribuir o módulo "importador" para equipe de desenvolvimento permitiu entrega <i>release</i> estável na <i>sprint</i> ; Líder técnico integrado e atuante na equipe; Histórico de falhas não identificado no <i>sprint</i> ; Estudo do TDD em andamento. Pontos Positivos abordados pela equipe: a) <i>release</i> estável liberado no <i>sprint</i> ; Ações de melhoria aumentaram a produtividade. Pontos Negativos Identificados na <i>Sprint</i>: a) Queda acentuada do Gráfico <i>BurnDown</i> identificou que os requisitos estão muito grandes (>1 semana); b) "Bug" (falha) identificada pelo setor de qualidade demora para ser resolvida. Discussões sobre problemas na <i>Sprint</i>: Avaliado que o problema da demora em resolver uma falha decorre de não identificar no TFS o responsável por resolvê-la. Observações Documentais: Gráfico <i>BurnDown</i> identificou queda brusca da linha ideal durante a segunda semana; Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues. Histórico de User Points: (<i>vide reuniões anteriores</i>) <i>Sprint</i> #8. Planejado: 329 / Entregue: 248. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.	Pontos Melhoria apontados pela equipe (baseado nos princípios LSD): a) Automatizar: a1) Sugerido automatizar o lançamento automático de falhas (bugs) no TFS a partir do sistema "versões"; b) Entregar rápido: b1) Para evitar não entregar <i>sprint</i> com requisito inacabado, solicitado o planejamento com requisitos menores (1dia); Pontos Melhoria apontados pela equipe: a) Não estimar na reunião de planejamento os "bugs", utilizar média histórica; b) Equipe de qualidade atribuir um responsável a resolução do "bug", caso não identificado responsável atribuir ao Scrum Master; Avaliação MALS: A equipe não identificou melhorias no MALS, porém solicitou mais uma visita do pesquisador para avaliar os resultados das ações de melhoria.

Ciclo	Período	Fase do MALS	CrITÉRIOS de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
9	18/09/2016 a 30/09/2016	6. Executar o <i>Sprint</i> #9 10. Identificar as oportunidades de Melhoria 12. Avaliar os resultados das ações de melhoria LSD 13. Gerar Relatório	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias identificadas na <i>Sprint</i> anterior: Tempo de resolução de problemas reduziu com a atribuição de responsável; Reunião de planejamento reduziu o tempo de realização devido a utilizar a média histórica de estimativa de esforço para falhas; O lançamento automático de falhas (bugs) não implementado (em estudo); Redução do tamanho dos requisitos permitiu não deixar requisitos inacabados e o gráfico de acompanhamento <i>burndown</i> alinhado (previsto x realizado). Pontos Positivos abordados pela equipe: a) <i>release</i> estável liberado no <i>sprint</i> ; Não ocorreram falhas identificadas no cliente; Pontos Negativos Identificados na <i>Sprint</i>: Não ocorreram problemas significativos no <i>Sprint</i> . Discussões sobre problemas na <i>Sprint</i>: Requisitos de maior complexidade afetam a produtividade, devido a curva de aprendizado de nova técnica; A atribuição da média histórica da estimativa de esforço para falhas foi satisfatório para as novas falhas identificadas; Solicitado criar um padrão para suporte realizar o lançamento de falhas no sistema “versões” e TFS. Observações Documentais: Gráfico <i>burndown</i> e <i>TaskBoard</i> : Tarefas finalizadas e entregues. Histórico de User Points: (<i>vide reuniões anteriores</i>) <i>Sprint</i> #9. Planejado: 291 / Entregue: 195. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.	Pontos Melhoria apontados pela equipe (baseado nos princípios LSD): a) Eliminar Desperdícios: a1) Criar uma lista de pendências que agreguem valor ao cliente; b) Entregar rápido: Controlar a redução do tamanho dessa lista. Pontos Melhoria apontados pela equipe: a) Padronizar o lançamento de falhas no sistema “versões” e TFS. Avaliação MALS: A equipe não identificou melhorias no MALS. A avaliação final do método pela equipe demonstrou que ocorreu grande evolução no processo de produção de <i>software</i> , evidenciado pelos indicadores como a produtividade, porém não foi possível implantar todos os princípios sugeridos pelo LSD. Para isso seriam necessários vários outros ciclos de melhoria.

APÊNDICE B – QUADRO RESUMO DA PESQUISA-AÇÃO EMPRESA B

Ciclo	Período	Fase da pesquisa	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
-	08/08/2016	1. Abertura do Projeto	Aceitação da Unidade de análise; Questionário de Avaliação MALS.	Apresentação da proposta MALS para os envolvidos; Avaliação Inicial do MALS.	Aprovado implantação MALS na Empresa B pelo Gerente de Projetos da empresa; Técnicas utilizadas para especificação de requisitos. Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Não. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Sim.	Avaliar as práticas de Engenharia de Software e nível de adequação Scrum. Pontos Melhoria do MALS: Equipe solicitou melhorar a estrutura visual do MALS para evidenciar que na fase de implantação do LSD os ciclos do Scrum continuam sendo executados.
		2. Avaliação Inicial Scrum	Questionário de Adequação Scrum	Questionário adequação Scrum aplicado ao Gerente de Projetos.	1. O Dono do Produto (Product Owner – PO) está definido ? Não. 2. A Equipe do projeto está definida ? Sim. 3. O Scrum Master está definido ? Não. 4. O <i>BackLog</i> do Produto está criado e priorizado ? Não está priorizado. 5. O esforço dos requisitos do <i>BackLog</i> do Produto estão estimados ? Parcialmente 6. O planejamento da Sprint está sendo realizado em períodos definidos ? Não, os períodos não estão definidos. 7. Existem técnicas para visualização e acompanhamento do trabalho que está sendo realizado ? Sim, utilizado ferramenta JIRA que oferece recursos para visualização e acompanhamento (Gráfico BurnDown e TaskBoard). 8. Reunião de acompanhamento diário estão sendo realizadas ? Não. 9. O resultado dos trabalhos realizados estão sendo demonstrados para equipe e o PO ? Não. 10. O Sprint está sendo avaliado sistematicamente, promovendo ciclos de melhoria contínua ? Não.	Informações: Na avaliação do Scrum a equipe está rodando o Sprint#23, algumas práticas e ferramentas Scrum eram utilizadas. Melhorias e Definições: A fase Scrum do MALS deve começar como se não a empresa não utilizasse o Scrum.

(CONTINUA)

Ciclo	Período	Fase da pesquisa	Crítérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
-	10/08/2016	3. Treinamento Scrum	Práticas de Engenharia de Software.	Treinamento do Scrum (1:15h).	Aprovado a ferramenta JIRA para suporte ao Scrum, pois disponibiliza recursos aderentes a metodologia (Gráfico <i>BurnDown</i> e <i>TaskBoard</i>), além disso a equipe tem experiência com a ferramenta; Definido o uso de <i>User Stories</i> para especificação dos requisitos e <i>User Points</i> para avaliação do esforço.	Definir a Equipe do Projeto e o Backlog do Produto.
		4. Definir a Equipe Scrum	Definição da equipe e responsabilidades; Definição Ciclo Scrum.	Planejamento do Ciclo Scrum e Ciclo Scrum.	Definido a equipe de projeto Scrum <i>Master</i> (SM), <i>Product Owner</i> (PO) e a Equipe Scrum (ES). SM: Analista de Sistemas; PO: Gerente de Projetos ou cliente (quando projeto solicitado pelo cliente); Equipe incluindo SM e PO com 9 integrantes. Definido que o Ciclo Scrum (<i>Sprint</i>) será de 2 semanas: <ul style="list-style-type: none"> <i>Sprint</i> Planning (Planejamento da <i>Sprint</i>): Primeira segunda-feira (manhã) da <i>Sprint</i>; Envolvidos: SM+PO+ES. Revisão <i>Sprint</i>: Última sexta-feira (manhã). Envolvidos: SM+PO+ES. Retrospectiva da <i>Sprint</i>: Última sexta-feira (manhã). Envolvidos: SM+ES Primeiro <i>Sprint</i> deve iniciar dia 17/08/2016.	Refinar e priorizar o Backlog do Produto.
-	17/08/2016	5. Definir o <i>BackLog</i> do Produto; Planejamento da <i>Sprint</i>	Planejamento da <i>Sprint</i> e <i>BackLog</i> do Produto.	Planejamento do Ciclo Scrum e <i>Backlog</i> do Produto; Definido os critérios de avaliação da aplicação do MALS.	Avaliação do Backlog do Produto (<i>Scrum Master</i> e <i>Product Owner</i>): Avaliado o Backlog existente e priorizado os trabalhos a serem realizados. Planejamento da <i>Sprint</i> , alocando o trabalho para equipe e estimando a duração das atividades. Crítérios de avaliação da aplicação do MALS: a) Práticas de gestão de projetos; b) Práticas de Engenharia de Software; c) Integração e Comunicação da Equipe; d) Qualidade do Produto; e) Produtividade; f) Fila de Pendências.	Ações Sugeridas pela Equipe: a) Avaliar previamente o Backlog do produto, para que a reunião de Backlog seja mais objetiva. b) Identificado 250 tarefas pendentes no <i>backlog</i> (fila), relacionados a solicitação de clientes e sugestões de melhoria. Solicitado (pelo gerente de projetos) que essa fila seja reduzida.

Ciclo	Período	Fase da pesquisa	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
1	17/08/2016 A 26/08/2016	6. Executar o <i>Sprint</i> #1 (JIRA025); 7. Avaliar o Nível de Maturidade Scrum	Adaptação da Equipe de Projeto; Resultado das cerimônias <i>Sprint</i> ; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	<p>O que deu certo na última Sprint: a) Organização da lista de pendências (BackLog); b) Entendimento claro de todas as tarefas realizadas pela equipe; c) Realizado.</p> <p>Observações Documentais: a) Análise do <i>TaskBoard</i>: Tarefas finalizadas e entregues; b) Análise do <i>BackLog</i> do Produto para identificar a lista de pendências.</p> <p>Nível de Maturidade Scrum: 1. O Dono do Produto (<i>Product Owner</i> – PO) está definido ? Sim. 2. A Equipe do projeto está definida ? Sim. 3. O Scrum <i>Master</i> está definido ? Sim. 4. O BackLog do Produto está criado e priorizado ? Parcialmente. 5. O esforço dos requisitos do BackLog do Produto estão estimados ? Parcialmente. 6. O planejamento da <i>Sprint</i> está sendo realizado em períodos definidos ? Sim. 7. Existem técnicas para visualização e acompanhamento do trabalho que está sendo realizado ? Sim. 8. Reunião de acompanhamento diário estão sendo realizadas ? Parcialmente. 9. O resultado dos trabalhos realizados estão sendo demonstrados para equipe e o PO ? Parcialmente. 10. O <i>Sprint</i> está sendo avaliado sistematicamente, promovendo ciclos de melhoria contínua ? Sim.</p> <p>Avaliação MALS: A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.</p>	<p>Ações Sugeridas pela Equipe: a) Solicitação presença ativa do <i>Product Owner</i> para viabilizar as reuniões de Revisão e Retrospectiva do Scrum; b) Definido que as reuniões de Retrospectiva e Planejamento sejam realizadas no mesmo dia (segunda-feira), ou seja, Revisão/Retrospectiva período da manhã e Planejamento à tarde; c) Análise preliminar do <i>BackLog</i> deve ser realizada pelo Scrum <i>Master</i> e <i>Product Owner</i>; c) Integração frequente do cliente no processo de validação do produto.</p> <p>Histórico de Pendências na Fila: Planejamento <i>Sprint</i> (17/08/2016): 250 Pendências. #1: (26/08/2016): 234 Pendências. Pendências Entregues: 43 Pendências não entregues: 26</p> <p>Histórico de User Points: #1: Planejado: 0 / Entregue: 0</p> <p>Capacidade Produtiva (Horas): #1. 263</p>

Ciclo	Período	Fase da pesquisa	Crerios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
2	26/08/2016 A 12/09/2016	6. Executar o <i>Sprint</i> #2 (JIRA026); 7. Avaliar o Nível de Maturidade Scrum	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	Melhorias Implementadas da última reunião: a) Alterado o período de revisão/retrospectiva da <i>sprint</i> permitindo a participação ativa do PO; b) Análise preliminar do planejamento realizado com SM+PO na sexta-feira que antecede o planejamento. O que deu certo na última Sprint: Mantido as melhorias Sprint#1. Problemas identificados na última Sprint: Reuniões diárias muito longas, indefinição do horário das reuniões ocorrendo "desencontros". Discussões sobre problemas/melhoria da Sprint: Para funcionar o Scrum as cerimônias devem que ser respeitadas por todos; Gráfico BurnDown apresentou queda brusca na última quinzena, porém o ciclo foi encerrado com a maioria dos requisitos planejados e entregues. Observações Documentais: a) Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues; b) Análise do Gráfico BurnDown: Queda brusca na última quinzena. Nível de Maturidade Scrum (Questões 1-10): (Respostas: 1.Sim.; 2.Sim.;3.Sim.; 6.Sim; 7.Sim.) 4. O <i>BackLog</i> do Produto está criado e priorizado ? Parcialmente. 5. O esforço dos requisitos do <i>BackLog</i> do Produto estão estimados ? Parcialmente. 8. Reunião de acompanhamento diário estão sendo realizadas ? Parcialmente. 9. O resultado dos trabalhos realizados estão sendo demonstrados para equipe e o PO ? Sim. 10. O <i>Sprint</i> está sendo avaliado sistematicamente, promovendo ciclos de melhoria contínua ? Sim. Avaliação MALS: (Repostas: Sim; Sim; Não; Não). A equipe compreende as atividades do MALS ? Sim. A equipe compreende as fases do MALS (Scrum/LSD) ? Sim. A equipe identifica algum impedimento para implantação do MALS ? Não. A equipe avalia oportunidade de melhorias no MALS ? Não.	Ações Sugeridas pela Equipe: a) Definido para 9:00 o horário para a realização das reuniões diárias. Caso necessário o horário pode ser alterado, porém toda equipe deve ser avisada; b) Solicitado objetividade na abordagem das pendências. Histórico de Pendências na Fila: #2: (12/09/2016): 238 Pendências. Pendências Entregues: 53 Pendências não entregues: 27 Histórico de User Points: #2: Planejado: 94 / Entregue: 67 Capacidade Produtiva (Horas): #2. 324

Ciclo	Período	Fase da pesquisa	Critérios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
3	12/09/2016 A 27/09/2016	6. Executar o <i>Sprint</i> #3 (JIRA027); 7. Avaliar o Nível de Maturidade Scrum	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Adequação Scrum; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Avaliar o nível de maturidade Scrum; Entrevista Coletiva com a Equipe de Projeto; Avaliação do Nível de Adequação Scrum; Avaliação da Proposta do MALS.	Melhorias Implementadas da última reunião: a/b) Reuniões diárias objetivas e executadas no horário definido. O que deu certo na última Sprint: Reuniões Diárias permitiu maior integração da equipe. Problemas identificados na última Sprint: (-). Discussões sobre problemas/melhoria da Sprint: Gráfico BurnDown apresentou queda brusca na última quinzena, porém o ciclo foi encerrado com a maioria dos requisitos planejados e entregues. Observações Documentais: a) Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues; b) Análise do Gráfico BurnDown: Queda brusca na última quinzena. Nível de Maturidade Scrum (Questões 1-10): (Respostas: 1.Sim.; 2.Sim.;3.Sim.; 6.Sim; 7.Sim; 9.Sim; 10.Sim) 4. O <i>BackLog</i> do Produto está criado e priorizado ? Sim. 5. O esforço dos requisitos do <i>BackLog</i> do Produto estão estimados ? Sim. 8. Reunião de acompanhamento diário estão sendo realizadas ? Sim. Avaliação MALS: (Repostas: Sim; Sim; Não; Não).	Ações Sugeridas pela Equipe: a) Atribuir o esforço para todas as <i>User Stories</i> para identificar a capacidade produtiva da equipe; b) Definido a data de 07/10/2016 para iniciar a segunda fase do MALS: Implantação do LSD. Histórico de Pendências na Fila: #3: 251 Pendências. Pendências Entregues: 73 Pendências não entregues: 17 Histórico de User Points: #3: Planejado: 244,5 / Entregue: 102,5 Capacidade Produtiva (Horas): #3. 385
-	07/10/2016	9. Treinamento dos Princípios LSD; Identificar as Oportunidades de Melhoria	Oportunidades de melhoria	Treinamento LSD (1h); Identificar as oportunidades de melhoria.	Treinamento Lean realizado: Histórico <i>Lean Thinking</i> ; e práticas <i>Lean Software Development</i> (LSD). Identificar as oportunidades de melhoria: a) Definido que a equipe deveria estudar os princípios <i>Lean</i> e trazer para próxima reunião de retrospectiva no mínimo uma sugestão de melhoria relacionado aos princípios e práticas LSD.	Melhorias de Processo Identificado: a) Nomeado o <i>Scrum Master</i> e <i>Product Owner</i> como equipe de liderança do fluxo de valor.

Ciclo	Período	Fase da pesquisa	Crerios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
4	27/09/2016 A 10/10/2016	6. Executar o <i>Sprint</i> #4 (JIRA028); 10. Identificar as oportunidades de Melhoria 12. Avaliar os resultados das ações de melhoria LSD	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias Implementadas da última reunião: a) Esforço atribuído para todas as novas <i>User Stories</i> e reavaliado o esforço do Backlog; b) Treinamento LSD realizado conforme planejado. O que deu certo na última Sprint: Atribuição de <i>User Stories</i> permitiu maior proximidade da capacidade produtiva da equipe.. Problemas identificados na última Sprint: (-). Discussões sobre problemas/melhoria da Sprint: Sugestões de melhoria baseada no Lean. Observações Documentais: a) Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues; b) Análise do Gráfico <i>BurnDown</i> : Alinhado Previsto x Realizado. Avaliação MALS: (Repostas: Sim; Sim; Não; Não).	Melhorias de Processo Identificado pela equipe (baseado nos princípios LSD): a) Melhorar a especificações dos requisitos para facilitar o entendimento e implementação, entender melhor antes de iniciar a implementação; b) Reduzir a troca de tarefas, realizando uma tarefa por vez, para evitar o desperdício; c) Dividir as <i>User Stories</i> em períodos menores, esforço para 1 (um) dia ou menos; d) Especificar os requisitos por camadas MVC; e) Estudar o TDD para antecipar o plano de testes; f) Separar a avaliação da capacidade produtiva da equipe por projeto (WEB/ERP). Histórico de Pendências na Fila: #4: 224 Pendências. Pendências Entregues: 66 Pendências não entregues: 26 Histórico de User Points: #4: Planejado: 309 / Entregue: 182 Capacidade Produtiva (Horas): #4. 364

Ciclo	Período	Fase da pesquisa	Crerios de avaliação	Atividades executadas	Avaliação	Melhoria e aprendizagem
5	10/10/2016 A 24/10/2016	6. Executar o <i>Sprint</i> #5 (JIRA029); 10. Identificar as oportunidades de Melhoria 12. Avaliar os resultados das ações de melhoria LSD	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias Implementadas da última reunião: a) Realizado; b) Realizado; c) Realizado; d) Em estudo; e) Em Estudo; f) Permitiu maior precisão na atribuição de esforço. O que deu certo na última Sprint: Maturidade do processo, integração da equipe e melhoria contínua. Problemas identificados na última Sprint: (-). Discussões sobre problemas/melhoria da Sprint: Sugestões de melhoria baseada no Lean. Observações Documentais: a) Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues; b) Análise do Gráfico <i>BurnDown</i> : Alinhado Previsto x Realizado. Avaliação MALS: (Repostas: Sim; Sim; Não; Não).	Melhorias de Processo Identificado pela equipe (LSD): a) Definir uma métrica para avaliar o tempo de todo o ciclo de entrega, da data que recebeu a OS até a entrega agrupando por nível de criticidade; b) Reunir com usuários chave para identificar a real necessidade dos requisitos da fila de pendências; c) Orientação para equipe refatorar código continuamente. Histórico de Pendências na Fila: #5: 199 Pendências. Pendências Entregues: 43 Pendências não entregues: 18 Histórico de User Points: #5: Planejado: 194 / Entregue: 129 Capacidade Produtiva (Hs): #5. 271
6	24/10/2016 A 07/11/2016	6. Executar o <i>Sprint</i> #6 (JIRA030); 10. Identificar as oportunidades de Melhoria 12. Avaliar os resultados das ações de melhoria LSD 13. Gerar Relatório	Resultado das cerimônias <i>Sprint</i> ; Melhorias apresentadas na Reunião de Retrospectiva; Questionário de Avaliação MALS.	Cerimônias da <i>Sprint</i> ; Entrevista Coletiva com a Equipe de Projeto; Análise do Gráfico <i>BurnDown</i> e <i>TaskBoard</i> ; Avaliação da Proposta do MALS.	Melhorias Implementadas da última reunião: a) Em estudo; b) Em fase de realização, observada satisfação dos clientes com a atenção às suas necessidades. O que deu certo na última Sprint: Satisfação dos clientes com a atenção dada aos . Problemas identificados na última Sprint: (-). Discussões sobre problemas/melhoria da Sprint: Sugestões de melhoria baseada no Lean. Observações Documentais: a) Análise do <i>TaskBoard</i> : Tarefas finalizadas e entregues; b) Análise do Gráfico <i>BurnDown</i> : Alinhado Previsto x Realizado. Avaliação MALS: (Repostas: Sim; Sim; Não; Não).	Melhorias de Processo Identificado pela equipe (LSD): a) Automação: Integração do Sistema ITIL com JIRA. Avaliação Final MALS: A avaliação final do método pela equipe demonstrou que ocorreu evolução no processo de produção de <i>software</i> , evidenciado pelos indicadores como a redução de filas de pendências, porém não foi possível implantar todos os princípios sugeridos pelo LSD. Para isso seriam necessários vários outros ciclos de melhoria. Histórico de Pendências na Fila: #6: 176 Pendências. Pendências Entregues: 63 Pendências não entregues: 26 Histórico de User Points: #6: Planejado: 295 / Entregue: 137 Capacidade Produtiva (Hs): #6. 265