

Everton Michels

**UM MODELO DE REFERÊNCIA PARA IMPLANTAÇÃO DAS
ROTINAS *TOYOTA KATA* NO PROCESSO ÁGIL DE
DESENVOLVIMENTO DE *SOFTWARE***

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia de Produção. Orientador: Prof. Dr. Fernando Antônio Forcellini.

Florianópolis
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Michels, Everton
Um modelo de referência para implantação das
rotinas Toyota Kata no processo ágil de
desenvolvimento de software / Everton Michels ;
orientador, Fernando Antônio Forcellini.
Forcellini, 2019.
195 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Engenharia de Produção, Florianópolis,
2019.

Inclui referências.

1. Engenharia de Produção. 2. Modelo Toyota Kata
. 3. Oportunidades e Barreiras. 4. Processo Ágil de
Desenvolvimento de Software. 5. Scrum. I.
Forcellini, Fernando Antônio Forcellini. II.
Universidade Federal de Santa Catarina. Programa de
Pós-Graduação em Engenharia de Produção. III. Título.

Everton Michels

**UM MODELO DE REFERÊNCIA PARA IMPLANTAÇÃO DAS
ROTINAS *TOYOTA KATA* NO PROCESSO ÁGIL DE
DESENVOLVIMENTO DE *SOFTWARE***

Esta Dissertação foi julgada adequada para obtenção do Título de “Mestre”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção.

Florianópolis, 20 de fevereiro de 2019.

Prof^a. Lucila Maria de Souza Campos, Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Fernando Antônio Forcellini, Dr.
Orientador
Universidade Federal de Santa Catarina

Prof. Marcelo Gitirana Gomes Ferreira, Dr.
Universidade do Estado de Santa Catarina

Prof. Helio Aisenberg Ferenhof, Dr.
Universidade Federal de Santa Catarina

Este trabalho é dedicado a minha mãe e minha esposa que sempre me apoiaram e incentivaram na aquisição e compartilhamento de novos conhecimentos.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe, Ilse Meri Michels, que de forma aguerrida e solitária, sempre incentivou seus três filhos no estudo mesmo em condições difíceis como era o caso quando atuava como professora do estado. Da mesma forma, agradeço também a minha esposa Fernanda Barcelos Medeiros, que sempre esteve ao meu lado nos momentos mais complicados e sempre me apoiou quando o assunto era conhecimento.

Um agradecimento especial ao meu Orientador, Professor Dr. Fernando Antônio Forcellini, por se mostrar sempre disposto a auxiliar da melhor forma para que este trabalho ficasse adequado a obtenção do título que merece, mas principalmente, por ter compartilhado muito de seu rico conhecimento e de sua vasta experiência a fim de proporcionar a melhoria contínua pessoal que busco diretamente com os estudos aqui realizados.

A empresa onde realizei o estudo, a qual abriu as portas e compartilhou as informações necessárias, em especial, ao time de desenvolvimento de *software* na área jurídica, que não só comprou a ideia de implantação das rotinas, como fez com que ela acontecesse de fato, por mais que as vezes isso parecesse impossível.

Aos meus professores e colegas do PPGEP – Programa de Pós-Graduação em Engenharia de Produção, por todo conhecimento compartilhado, por todas as parcerias nas disciplinas, artigos e atividades externas, bem como, pelo incentivo e pelo apoio.

Ao PPGEP, na figura das pessoas da secretaria, as quais estavam sempre dispostas a ajudar da melhor forma em questões administrativas e burocráticas.

Por fim, a todos aqueles que de alguma forma me apoiaram ou auxiliaram na jornada para esta conquista, mas que por algum motivo não nomeei aqui.

MUITO OBRIGADO!!!

“Oshite-mo dame-nara hiite mina.”

Tradução: Se empurrar não funcionar, tente puxar. (PROVÉRPIO JAPONÊS, 2018).

RESUMO

Alguns estudos na literatura atual já explicitam o conhecimento em torno das rotinas *Toyota Kata*, bem como, os benefícios da utilização das mesmas. Da mesma forma, a literatura sobre processo de desenvolvimento de *software* já está bem consolidada e contribui de forma bem ampla em diversas áreas de conhecimento. No entanto, existe ainda uma lacuna no que tange principalmente ao método, ou forma com que as rotinas *Toyota Kata*, - *Kata* de Melhoria (KM) e *Kata* de *Coaching* (KC) -, podem ser implantadas em contextos distintos, como por exemplo, no processo ágil de desenvolvimento de *software*. Além disso, verificar se a integração das rotinas *Toyota Kata* no *Scrum*, são capazes de reduzir alguns dos problemas encontrados neste método. Com base nesse gap, o objetivo deste estudo é desenvolver um modelo de implantação das rotinas *Toyota Kata* no processo ágil de desenvolvimento de *software*, em especial o *Scrum*, a fim de verificar se as lacunas do método podem ser atendidas pelas rotinas *Toyota Kata*, bem como, se essa integração entre as rotinas e método facilita a implantação de *Toyota Kata*. Para tal, foi realizado uma Revisão Bibliográfica Sistemática (RBS), na qual foram identificadas oportunidades e barreiras na implantação das rotinas *Toyota Kata*. Posteriormente, foi desenvolvido o modelo com base no resultado da RBS e de alguns problemas encontrados no *Scrum*. Por fim, foi realizada uma pesquisa-ação, na qual o modelo de implantação das rotinas *Toyota Kata* integradas ao *Scrum* foi verificado. Como resultados, foi possível observar que o modelo proposto foi capaz de proporcionar melhorias no contexto estudado, tanto minimizando as lacunas encontradas no *Scrum*, especialmente no que tange a uma visão mais sistêmica, ao aprendizado de grupo, o compartilhamento de conhecimento, a explicitação de conhecimento e a utilização dos ciclos PDCA, bem como, facilitando a implantação das rotinas *Toyota Kata* no segmento de *software*. Como conclusões, pode-se verificar que as principais oportunidades e barreiras levantadas pela literatura se mostraram válidas para o contexto estudado na pesquisa-ação e puderam ser atendidas pelo modelo. Constatou-se também, que o modelo desenvolvido facilita a implantação das rotinas *Toyota Kata* integradas ao *Scrum*, e assim, conseguem minimizar as lacunas encontradas no método.

Palavras-chave: Modelo. *Toyota Kata*. Oportunidades e Barreiras. Processo Ágil de Desenvolvimento de *Software*. *Scrum*.

ABSTRACT

Some studies in the current literature already explain the knowledge about the Toyota Kata routines, as well as the benefits of using them. Likewise, the literature on the software development process is already well established and contributes in a very broad way in several areas of knowledge. However, there is still a gap in what concerns mainly the method, or way with which the routines Toyota Kata, - Improvement Kata (IK) and Coaching Kata (CK) -, can be implanted in different contexts, in the agile process of software development. In addition, check if the integration of Toyota Kata routines in Scrum, are able to reduce some of the problems encountered in this method. Based on this gap, the objective of this study is to develop a model of implementation of the Toyota Kata routines in the agile process of software development, especially Scrum, in order to verify if the gaps of the method can be satisfied by the routines Toyota Kata, as well like, whether this integration between routines and method facilitates the deployment of Toyota Kata. For this, a Systematic Bibliographic Review (SBR) was carried out, in which opportunities and barriers were identified in the implementation of the Toyota Kata routines. Subsequently, the model was developed based on the SBR result and some problems encountered in Scrum. Finally, an action research was carried out, in which the model of implementation of the Toyota Kata routines integrated to the Scrum was verified. As results, it was possible to observe that the proposed model was able to provide improvements in the studied context, both minimizing the gaps encountered in Scrum, especially with regard to a more systemic view, to group learning, knowledge sharing, knowledge and the use of PDCA cycles, as well as facilitating the implementation of Toyota Kata routines in the software segment. As conclusions, it can be verified that the main opportunities and barriers raised in the literature were valid for the context studied in action research and could be met by the model. It was also verified that the developed model facilitates the implementation of Toyota Kata routines integrated with Scrum, and thus, they can minimize the gaps found in the method.

Keywords: Model. Toyota Kata. Opportunities and Barriers. Agile Software Development Process. Scrum.

LISTA DE FIGURAS

Figura 1 – Procedimento metodológico	26
Figura 2 – Revisão bibliográfica sistemática (RBS)	28
Figura 3 – Ciclo básico da investigação-ação	30
Figura 4 – Processo da pesquisa-ação	31
Figura 5 – Estrutura da dissertação	34
Figura 6 – <i>Kata</i> de melhoria.....	41
Figura 7 – Modelo ou padrão de pensamento científico de trabalho.....	44
Figura 8 – <i>Kata</i> de <i>coaching</i>	46
Figura 9 – <i>Kata</i> de melhoria e <i>Kata</i> de <i>coaching</i>	47
Figura 10 – As 5 Questões	48
Figura 11 – <i>Storyboard</i>	49
Figura 12 – Interação mentor-aprendiz no <i>Kata</i> de <i>coaching</i>	50
Figura 13 – Processo de implantação das rotinas de <i>Kata</i>	51
Figura 14 – Plano de desenvolvimento	53
Figura 15 – Níveis de capacidade.....	54
Figura 16 – Níveis de treinamento	54
Figura 17 – Trânsito de treinamento em uma organização	55
Figura 18 – Modelo cascata	56
Figura 19 – Métodos ágeis e práticas	60
Figura 20 – <i>Framework Scrum</i>	62
Figura 21 – Burndown.....	77
Figura 22 – <i>Framework Scrum</i>	82
Figura 23 – Conexões <i>Scrum</i> e <i>Toyota Kata</i>	88
Figura 24 – Proposta do modelo.....	96
Figura 25 – Entradas e saídas da fase preparação	97
Figura 26 – Entradas e saídas da fase planejamento	98
Figura 27 – Obstáculos e a fronteira do conhecimento	99
Figura 28 – Entradas e saídas da fase execução	100
Figura 29 – Entradas e saídas da fase reflexão.....	101
Figura 30 – Processo da pesquisa-ação	103
Figura 31 – Macroprocesso de demandas do cliente.....	105
Figura 32 – Sensibilização	108
Figura 33 – Integração <i>Scrum</i> e <i>Toyota Kata</i> no tempo	110
Figura 34 – <i>Storyboard</i> da pesquisa-ação	111
Figura 35 – <i>Storyboard</i> do time	112
Figura 36 – <i>Storyboard</i> da <i>sprint</i> 1	114
Figura 37 – <i>Hoshin Kanri</i>	119
Figura 38 – Macroprocesso de finalização de documentos	120
Figura 39 – Mapa do fluxo de valor de finalização de documentos....	121

Figura 40 – A3 para o processo finalização de documentos	123
Figura 41 – <i>Backlog</i> do produto.....	125
Figura 42 – <i>Storyboard</i> de pesquisa-ação da <i>sprint 2</i>	127
Figura 43 – <i>Storyboard</i> do time da <i>sprint 2</i>	132
Figura 44 – <i>Storyboard</i> de pesquisa-ação da <i>sprint 3</i>	140
Figura 45– <i>Storyboard</i> do time da <i>sprint 3</i>	142
Figura 46 – <i>Storyboard</i> da pesquisa-ação da <i>sprint 4</i>	144
Figura 47 – <i>Storyboard</i> do time da <i>sprint 4</i>	146
Figura 48 – <i>Storyboard</i> da pesquisa-ação da <i>sprint 5</i>	148
Figura 49 – Registro dos ciclos PDCA da <i>sprint 5</i>	150
Figura 50 – <i>Storyboard</i> do time da <i>sprint 5</i>	151
Figura 51 – <i>Storyboard</i> da pesquisa-ação da <i>sprint 6</i>	153
Figura 52 – <i>Storyboard</i> do time da <i>sprint 6</i>	155
Figura 53 – Registro dos ciclos pdca da <i>sprint 6</i>	156
Figura 54 – Estrutura do Mapeamento Sistemático	178
Figura 55 – Etapas de Seleção dos Estudos	182
Figura 56 – RBS <i>Roadmap</i>	181
Figura 57 - RBS	186

LISTA DE QUADROS

Quadro 1 – Etapa das rotinas <i>Toyota Kata</i> no <i>Scrum</i>	83
Quadro 2 – Oportunidades e barreiras da implantação de <i>Toyota Kata</i> na literatura	92
Quadro 3 – Participantes da pesquisa-ação	107
Quadro 4 – <i>Sprint 1</i> da pesquisa-ação e do time	115
Quadro 5 – <i>Storyboard</i> da pesquisa-ação da <i>sprint 2</i>	128
Quadro 6 – <i>Storyboard</i> do time da <i>sprint 2</i>	133
Quadro 7 – Lacunas e Oportunidades de Melhoria do <i>Scrum</i>	178
Quadro 8 – Síntese da RBS	190

LISTA DE ABREVIATURAS E SIGLAS

ASD - Adaptive Software Development
CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CK - Coaching Kata
DSDM - Dynamic Systems Development Method
FDD - Feature Driven Development
GC - Gestão do Conhecimento
GEPPS - Grupo de Engenharia de Produto, Processo e Serviço da UFSC
GTA - Grupo de Transformação Ágil
JCR – Journal Citation Reports
IK - Improvement Kata
KM – Kata de Melhoria
KC – Kata de Coaching
MFV - Mapeamento do Fluxo de Valor
PDCA - Plan, Do, Check e Act
PF - Pontos de Função
PO - Product Owner
PLAN – Planejamento
RAD - Rapid Application Development
RBS - Revisão Bibliográfica Sistemática
SBR - Systematic Bibliographic Review
RTC - Rational Team Concert
SM - Scrum Master
TIC - Tecnologia da Informação e Comunicação
TQM - Total Quality Management
WOS – Web of Science
XP - Extreme Programming

SUMÁRIO

1	INTRODUÇÃO	23
1.2	PROBLEMÁTICA	24
1.3	OBJETIVOS	25
1.3.1	Objetivo Geral	25
1.3.2	Objetivos Específicos	25
1.4	PROCEDIMENTOS METODOLÓGICOS.....	25
1.4.1	Pesquisa bibliográfica	27
1.4.2	Proposição do modelo	29
1.4.3	Pesquisa-ação	30
1.5	JUSTIFICATIVA	31
1.6	ESTRUTURA DA DISSERTAÇÃO.....	33
2	PESQUISA BIBLIOGRÁFICA	37
2.1	<i>LEAN</i>	37
2.2	<i>TOYOTA KATA</i>	38
2.2.1	<i>Kata</i> de melhoria	40
2.2.2	<i>Kata</i> de <i>Coaching</i>	44
2.2.3	Processo de implantação das rotinas <i>Toyota Kata</i>	50
2.3	MÉTODOS ÁGEIS	55
2.4	<i>SCRUM</i>	60
2.4.1	Papéis do <i>Scrum</i>	63
2.4.2	Cerimônias do <i>Scrum</i>	69
2.4.3	Artefatos do <i>Scrum</i>	75
2.5	CONSIDERAÇÕES FINAIS DA REVISÃO BIBLIOGRÁFICA.....	79
3	PROPOSIÇÃO DO MODELO DE REFERÊNCIA PARA IMPLANTAÇÃO DAS ROTINAS <i>TOYOTA KATA</i> NO <i>SCRUM</i>	81
3.1	REVISÃO	81
3.2	CONEXÕES ENTRE <i>SCRUM</i> E <i>TOYOTA KATA</i>	87
3.3	PROPOSIÇÃO DO MODELO	90
3.3.1	Preparação	97
3.3.2	Planejamento	97
3.3.3	Execução	98
3.3.4	Reflexão	100
3.4	CONSIDERAÇÕES FINAIS SOBRE O MODELO PROPOSTO	101
4	PESQUISA-AÇÃO	103
4.1	PLANEJAMENTO.....	104

4.1.1	Definição do problema	104
4.1.2	Equipe.....	106
4.1.3	Sensibilização	108
4.2	DIAGNÓSTICO.....	108
4.2.1	Plano de ação.....	109
4.3	IMPLANTAÇÃO	113
4.3.1	Implantação do plano.....	113
4.3.1.1	<i>Sprint 1</i>	113
4.3.1.2	<i>Sprint 2</i>	126
4.3.1.3	<i>Sprint 3</i>	141
4.3.1.4	<i>Sprint 4</i>	143
4.3.1.5	<i>Sprint 5</i>	147
4.3.1.6	<i>Sprint 6</i>	154
4.3.2	Coleta de dados	157
4.3.3	Avaliação e resultados.....	157
4.4	CONSIDERAÇÕES FINAIS DA PESQUISA-AÇÃO.....	159
5	CONCLUSÕES.....	163
	REFERÊNCIAS	167
	APENDICE A – Mapeamento Sistemático.....	171
	APENDICE B – Revisão Bibliográfica Sistemática.....	181

1 INTRODUÇÃO

Este capítulo visa contextualizar o tema e a problemática que motivaram a pesquisa, bem como, os objetivos e procedimentos metodológicos. Ao final da mesma, é apresentada a estrutura a qual a dissertação seguirá de forma genérica, com o objetivo de introduzir cada capítulo.

1.1 CONTEXTUALIZAÇÃO

Nas últimas décadas empresas ao redor do mundo vêm buscando constantemente utilizar da abordagem *Lean*, a fim de melhorar seus resultados (INDUSTRY WEEK, 2008). Grande parte destas empresas, entretanto, não consegue obter de fato resultados sustentáveis como mostra a pesquisa da Industry Week, a qual aponta que ao redor do mundo apenas 2% das empresas conseguiram alcançar seus objetivos com *Lean*, bem como, em torno de apenas 24% das mesmas relataram um resultado significativo com esta abordagem (INDUSTRY WEEK, 2008).

Os motivos pelos quais tão poucas empresas conseguem obter resultados significativos com o *Lean* se concentram principalmente na maneira como as pessoas são lideradas e nos comportamentos e rotinas que estas utilizam (LIKER; MEIER, 2007; LIKER; HOUSES, 2009; ROTHER, 2010; SONY; NAIK; THERISA, 2019).

Rother (2010) mostra em seu estudo duas rotinas que a empresa Toyota utiliza no seu dia-a-dia, as quais conseguem contemplar de forma sistêmica alguns aspectos dos comportamentos, rotinas e liderança.

Verifica-se, porém, que não só o trabalho de Rother (2010), como outros mais (Reverol, 2012; Soltero, 2012; Iberle, 2015; Toivonen, 2015), estão utilizando estas rotinas a fim de melhorarem seus resultados, porém, em contextos específicos como por exemplo a manufatura e a área de saúde, sendo que nenhum deles é abordado na área de *software*.

Da mesma forma como a abordagem *Lean* é muito utilizada na manufatura (Liker; Meier, 2007; Industry Week, 2008; Liker; Houses, 2009), o *framework Scrum* é o método mais utilizado no mundo para o desenvolvimento ágil de *software* (VERSIONONE AGILE MADE EASIER, 2017).

No entanto, como explicitado anteriormente, da mesma forma que a utilização da abordagem *Lean* possui problemas, isso ocorre também com o *Scrum*, o qual possui também seus problemas e suas lacunas

(HAMED; ABUSHAMA, 2013; RAUF; ALGHAFEEES, 2015; SHARMA; HASTEER, 2016).

Desta forma, este estudo visa abordar e explicitar como esta lacuna de implantação das rotinas *Toyota Kata* no processo ágil de desenvolvimento de *software*, especialmente no *Scrum*, pode ser reduzida, a fim de tornar a implantação destas rotinas, mais naturais possíveis. Além disso, possibilitar uma integração entre as rotinas *Toyota Kata* e o *Scrum*, com intuito de minimizar ou mesmo eliminar as lacunas encontradas no método.

1.2 PROBLEMÁTICA

É notório na literatura que os temas relacionados a este estudo em maior ou menor grau, já são bem difundidos. A área de gerenciamento de projetos ágil, bem como, de *Lean*, já estão bem-conceituadas e compartilhadas na literatura (LIKER; MEIER, 2007; LIKER; HOUSES, 2009; ROTHER, 2010; HAMED; ABUSHAMA, 2013; RAUF; ALGHAFEEES, 2015; SHARMA; HASTEER, 2016; SONY; NAIK; THERISA, 2019). No entanto, existe na literatura, conforme será verificado neste estudo, uma lacuna no que tange o tema principal deste trabalho que são as rotinas *Toyota Kata*, especialmente quanto a forma como as mesmas são implantadas nos diferentes contextos, e se estas podem ser implantadas com outros métodos de gestão, especialmente o *Scrum*, no processo ágil de desenvolvimento de *software* (REVEROL, 2012; SOLTERO, 2012; IBERLE, 2015; TOIVONEN, 2015).

Embora alguns trabalhos explicitem o conhecimento e as experiências relacionadas a este tema (Reverol, 2012; Soltero, 2012; Iberle, 2015; Toivonen, 2015, etc), em nenhum deles foi encontrado o conhecimento relativo a forma detalhada de como estas rotinas, especialmente utilizadas na Toyota, são implantadas, principalmente quando o contexto é a indústria de *software*.

Além disso, verifica-se na literatura que alguns estudos relatam lacunas e problemas encontrados na utilização do *Scrum*, e que estes não estão sendo abordados e eliminados conforme as necessidades relatadas (HAMED; ABUSHAMA, 2013; RAUF; ALGHAFEEES, 2015; SHARMA; HASTEER, 2016).

Com base no exposto, a implantação destas rotinas, a fim de melhorar o ambiente onde estas atuam, pode se tornar difícil, já que os contextos encontrados até o momento na literatura não são de *softwares*.

Da mesma forma, conforme relatado até o momento, os problemas encontrados no *Scrum*, ainda não obtiveram em sua totalidade uma

solução adequada para o contexto de *software*, além de não se encontrar literatura que explicita um aprendizado referente a integração das rotinas *Toyota Kata* e *Scrum* a fim de minimizar ou eliminar estes problemas, como por exemplo, a falta de visão sistêmica, a falta de compartilhamento e explicitação do conhecimento, falta de métodos com bases científicas para utilização no dia a dia, baixo nível de aprendizado, entre outros.

1.3 OBJETIVOS

O objetivo geral deste estudo, bem como seus objetivos específicos, que são utilizados para alcançar o objetivo geral, são demonstrados nas sessões a seguir.

1.3.1 Objetivo Geral

Desenvolver um modelo para a implantação das rotinas *Toyota Kata* integradas ao *Scrum*, a fim de facilitar a implantação das rotinas, possibilitar uma visão sistêmica, fomentar o compartilhamento do conhecimento, o engajamento e o aprendizado.

1.3.2 Objetivos Específicos

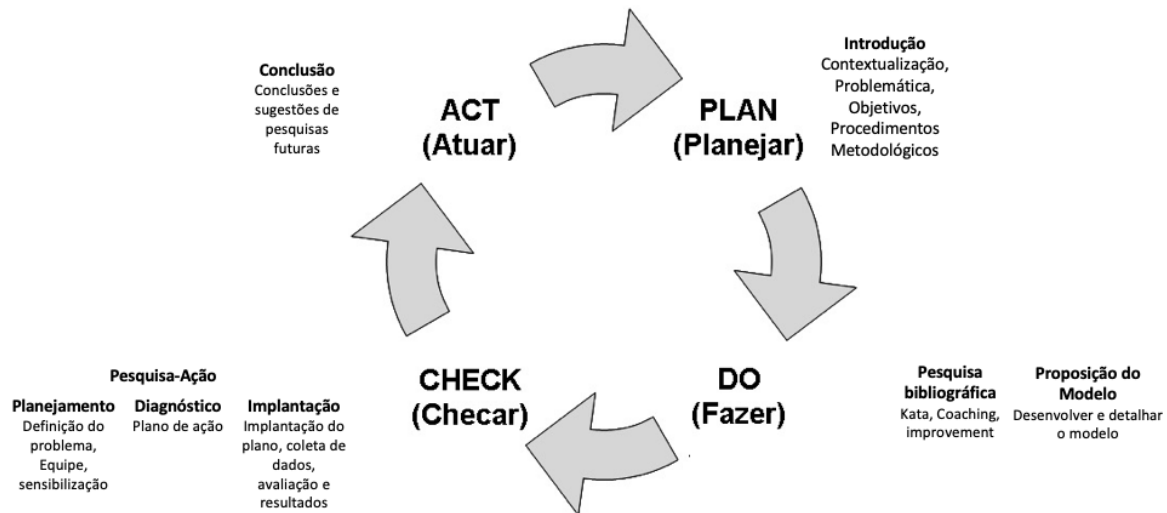
A fim de alcançar o objetivo geral, foram identificados os seguintes objetivos específicos:

- a) Identificar na literatura existente, oportunidades e barreiras na implantação de *Toyota Kata*;
- b) Identificar as lacunas na utilização do *Scrum* na indústria do *software*;
- c) Verificar o comportamento do modelo no contexto deste estudo.

1.4 PROCEDIMENTOS METODOLÓGICOS

A fim de proporcionar uma visão sistêmica e o melhor entendimento deste estudo, fez-se necessário a estruturação de um procedimento metodológico. Desta forma, para a elaboração do mesmo utilizou-se do princípio do PDCA de Deming (1986), o qual se mostra o mais aderente ao estudo, onde o mesmo foi definido com os seguintes passos: introdução; pesquisa bibliográfica; proposição do modelo, pesquisa-ação e conclusão. Esta estrutura do procedimento metodológico é apresentada conforme mostra a Figura 1.

Figura 1 – Procedimento metodológico



Fonte: Elaborada pelo autor.

Sendo assim, a pesquisa bibliográfica visa fornecer o embasamento teórico quanto aos assuntos relacionados a este estudo, sendo eles: *Toyota Kata*, processo ágil de desenvolvimento de *software*. A proposição do modelo busca estabelecer o processo pelo qual a implantação das rotinas *Toyota Kata* melhor se adapta ao processo ágil de desenvolvimento de *software*, especialmente o *Scrum*. Por fim, a pesquisa-ação é utilizada com intuito de verificar o comportamento do modelo, e com isso, identificar possíveis melhorias e o aprendizado gerado.

1.4.1 Pesquisa bibliográfica

A pesquisa bibliográfica neste estudo se dará por meio de uma revisão bibliográfica sistemática. A revisão bibliográfica sistemática (RBS) tem como característica ser um processo bem estruturado, ser replicável e transparente (COOPER, 1998).

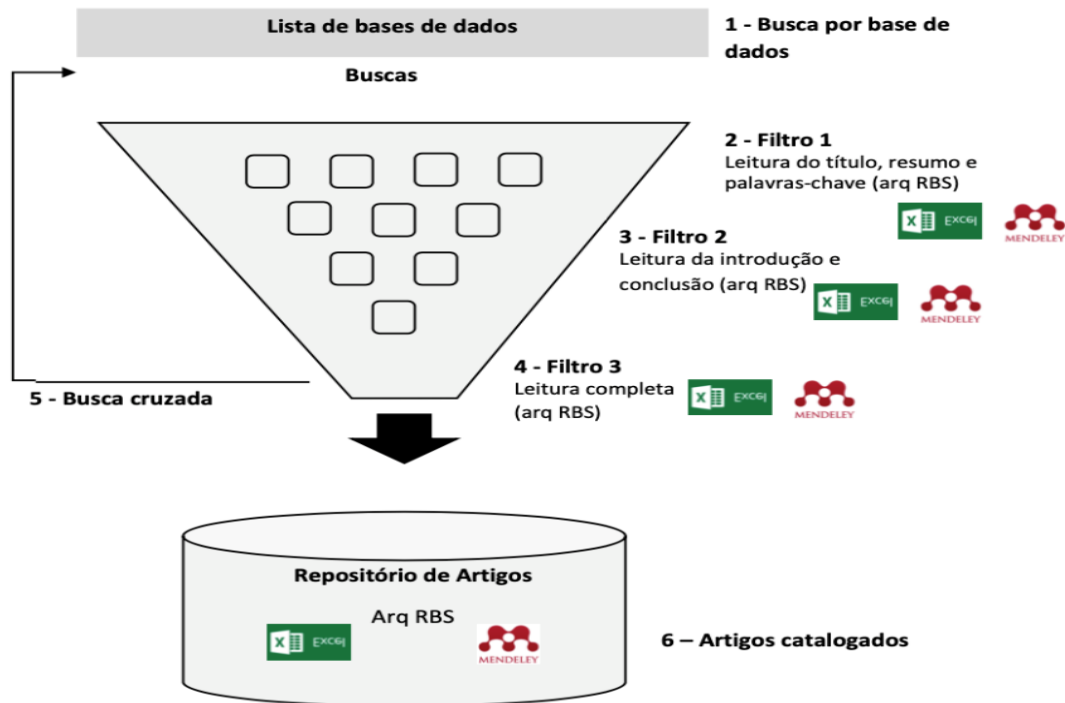
Cook, Mulrow e Haynes (1997), por sua vez, explicita que a revisão bibliográfica sistemática possui uma característica narrativa, e utiliza-se de métodos com maior cunho científico, e desta forma acaba por alcançar resultados mais significativos, reduzindo assim possíveis erros provenientes do responsável pela pesquisa.

Segundo Cook, Mulrow e Haynes (1997 apud CONFORTO; AMARAL; SILVA, 2011, p. 2) esse processo possui também as seguintes características: permite ao pesquisador compilar dados, refinar hipóteses, estimar tamanho de amostras; definir melhor o método de pesquisa a ser adotado para aquele problema; definir direções para futuras pesquisas.

Neste trabalho a revisão bibliográfica sistemática busca verificar estudos na literatura que abordem sobre os temas deste trabalho, bem como, fundamentar os conceitos de *Toyota Kata*. Além disso faz-se necessário também, elucidar os conceitos de processo de desenvolvimento de *software*, a fim de se embasar o contexto estudado.

Para a revisão bibliográfica sistemática, utilizou-se uma adaptação do método de Conforto, Amaral e Silva (2011), sendo este composto por 6 etapas, conforme mostra a Figura 2.

Figura 2 – Revisão Bibliográfica Sistemática (RBS)



Fonte: Adaptado de Conforto, Amaral e Silva (2011).

Conforme apresentado na Figura 2, a primeira etapa consiste na busca por periódicos nas fontes primárias já identificadas. Para cada fonte primária é realizada uma busca, utilizando a *String* de busca. Os resultados são registrados utilizando o arquivo RBS, sendo que no mesmo são preenchidos os seguintes campos: ID; Nome da fonte primária; Citações; Ano do estudo; Autor(es) do estudo; Título do estudo; Periódico/Evento de publicação. Na segunda etapa, realiza-se a leitura do título, resumo e palavras-chave. Desta forma os trabalhos que não passarem por esse filtro por não se adequarem ao objetivo do estudo são marcados como “inaptos” na coluna “Filtro” do arquivo RBS. A terceira etapa consiste na leitura da introdução e conclusão dos trabalhos.

Da mesma forma que no passo anterior, se os mesmos não se adequarem ao objetivo do estudo, estes também são marcados como “inaptos”. Na etapa quatro, é realizada a leitura completa do artigo, e desta forma os trabalhos que se mostrarem adequados ao objetivo do estudo são marcados como aptos, já os demais são designados como inaptos. Na etapa cinco busca-se identificar estudos relevantes que não foram encontrados durante a busca nas bases de dados, sendo, porém, importantes referências para o embasamento deste trabalho.

Por fim, a etapa seis constitui a documentação dos estudos e resultados das buscas. Os estudos que passarem pela etapa 4 são catalogados e armazenados em um *software* para gestão de referências bibliográficas. Para este estudo foi utilizado o Mendeley (2018).

1.4.2 Proposição do modelo

Para a proposição do modelo, este estudo identificou primeiramente as oportunidades e barreiras de trabalhos anteriores e buscou analisar as mesmas a fim de não passar pelos mesmos problemas, e desta forma tornar mais natural os testes e validações do modelo proposto.

Para isso, o primeiro passo tomado foi a elaboração do mapa de fluxo de valor atual, que visa entender o estado atual da unidade estudada antes da proposição do modelo, a fim de garantir a representação sistêmica do contexto estudado.

Com base nisso, o próximo passo foi a elaboração do modelo a fim de explicitar o mesmo da forma mais adequada possível ao contexto estudado, e assim, disponibilizar o mesmo para que empresas que se enquadrem no mesmo contexto de projetos, possam replicar o modelo sem maiores problemas.

Por fim, foi realizada a implantação das rotinas de *Toyota Kata* no processo de desenvolvimento de *software*, neste caso o *Scrum*, para desta forma verificar o comportamento do modelo proposto.

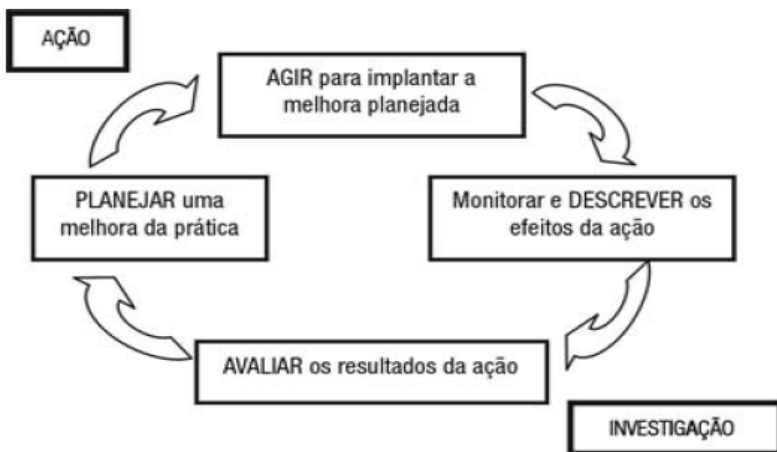
1.4.3 Pesquisa-ação

Como o próprio nome indica, a pesquisa-ação une a teoria com a prática no intuito de intervir no processo de aprendizagem, de forma a melhorar o contexto específico que está sendo pesquisado. Diferente da pesquisa tradicional, ela é participativa e busca resolver problemas específicos do cotidiano prático das pessoas (ENGEL, 2000). Engel (2000, p. 2), aborda a pesquisa-ação da seguinte forma:

A pesquisa-ação surgiu da necessidade de superar a lacuna entre teoria e prática. Uma das características deste tipo de pesquisa é que através dela se procura intervir na prática de modo inovador já no decorrer do próprio processo de pesquisa e não apenas como possível consequência de uma recomendação na etapa final do projeto.

Da mesma forma que este trabalho e as práticas aqui utilizadas, a pesquisa-ação possui dois objetivos genéricos. Um deles é a busca pela resolução de problemas. O outro é científico, e visa principalmente a criação de conhecimento (TRIPP, 2005). A estrutura da pesquisa-ação está diretamente ligada a forma como se apresenta a investigação-ação, a qual pode ser visualizada na Figura 3.

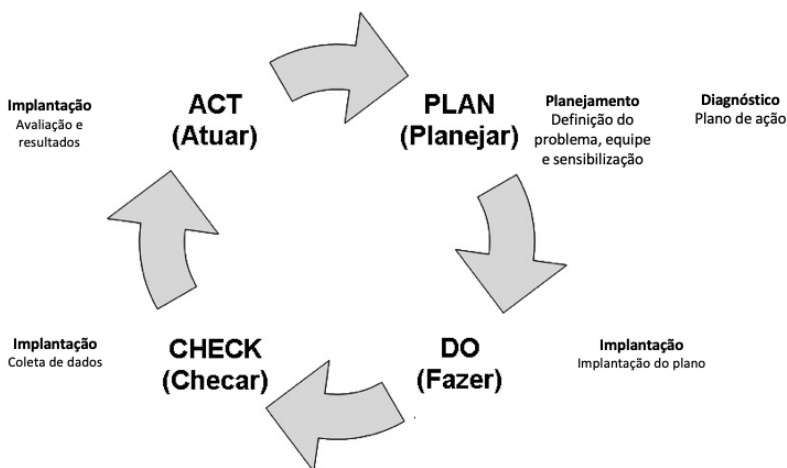
Figura 3 – Ciclo básico da investigação-ação



Fonte: Tripp (2005).

Neste estudo a pesquisa-ação foi conduzida em ciclos alinhados com as rotinas *Toyota Kata*, bem como, com o método de gestão ágil de projetos utilizado, no caso o *Scrum*. Conforme mostra a Figura 4, os ciclos usaram o conceito de PDCA difundido por Deming (1986), no qual se baseiam tanto as rotinas *Toyota Kata*, quanto o *Scrum*. Na etapa de planejamento (*Plan*) buscou-se estabelecer o problema em questão, a equipe designada para o trabalho, e uma sensibilização a fim de contextualizar as práticas que seriam utilizadas, além de realizar um diagnóstico contendo o plano de ação.

Figura 4 – Processo da pesquisa-ação



Fonte: Elaborada pelo autor.

Na etapa de execução (*Do*) foi contemplada a implantação do que foi planejado.

Na etapa de checagem (*Chek*) foi realizada a coleta dos dados oriundos da implantação.

Por fim, na etapa de atuação (*Act*) foram avaliados estes dados e identificados os resultados finais da pesquisa-ação.

1.5 JUSTIFICATIVA

Toyota Kata quando é aplicado, serve de mecanismo de melhoria contínua dentro das organizações (REVEROL, 2012). Ele envolve as pessoas em uma rotina de resolução de problemas, o que aumenta o

número de soluções sugeridas e consequentemente a probabilidade de sucesso da solução priorizada (TOIVONEN, 2015). Algumas características que permitem o sucesso das iniciativas de implantação das rotinas *Toyota Kata* são o foco em processos, a utilização de métricas e a definição da direção e desafio corretos alinhados a estratégia (IBERLE, 2015).

Por sua vez, com a mudança no contexto do desenvolvimento de *software*, migrando de um desenvolvimento *waterfall* para o ágil, diversos métodos surgiram para reduzir e até eliminar alguns problemas no processo de desenvolvimento. Dentre estes métodos, o mais utilizado no mundo é o *Scrum* (Versionone Agile Made Easier, 2017), que acabou trazendo diversos benefícios para o desenvolvimento de *software*, bem como a entrega de melhores resultados (CHAOS REPORT, 2015). Entretanto, por mais que os resultados tenham melhorados de forma significativa, os mesmos se encontram estáveis, e além disso, lacunas e problemas são encontrados com o passar do tempo nestes métodos ágeis, inclusive no *Scrum* (HAMED; ABUSHAMA, 2013; CHAOS REPORT, 2015; RAUF; ALGHAFEEES, 2015; SHARMA; HASTEER, 2016).

Com base nesse contexto, este estudo tem uma motivação profissional, a qual enfrenta problemas e lacunas encontradas na utilização do *Scrum*, como por exemplo, a falta de visão e direção, o baixo compartilhamento de conhecimento e aprendizado, a falta de alinhamento entre estratégia e operação, entre outras (HAMED; ABUSHAMA, 2013; RAUF; ALGHAFEEES, 2015; SHARMA; HASTEER, 2016). Além disso, uma motivação teórica e prática do pouco material encontrado na literatura no que tange a implantação das rotinas *Toyota Kata* no segmento do *software*, em especial, que utilizem o *Scrum* no seu processo de desenvolvimento, verificando assim sua originalidade.

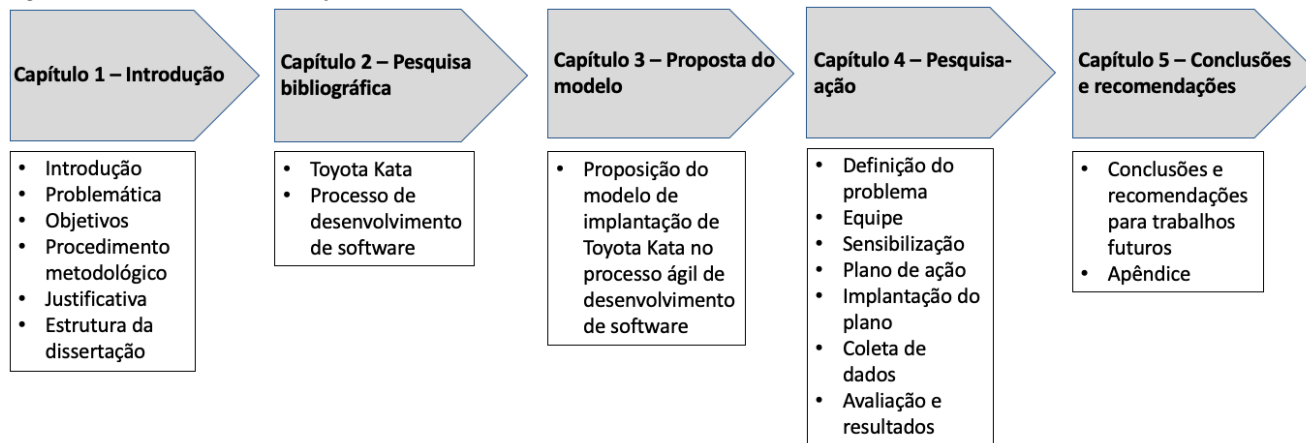
Por fim, busca-se com o trabalho contribuir para academia, especialmente para o programa de Pós-Graduação em Engenharia de Produção, o qual possui uma abordagem multidisciplinar, de forma não só identificar as lacunas encontradas no *Scrum*, bem como as oportunidades e barreiras na implantação das rotinas *Toyota Kata*, como também na proposição de um modelo que contemple tudo isso, e que assim, fomente o compartilhamento de conhecimento entre as pessoas envolvidas, promova o engajamento e o aprendizado, utilize métodos estruturados e científicos para solução de problemas, possibilite uma visão sistêmica, direcione o desenvolvimento do *software* de forma alinhada, entre outros.

1.6 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está estruturado em 5 capítulos, conforme mostrado na Figura 5 e detalhado posteriormente.

Na sessão um é apresentada a introdução, seguida de uma breve contextualização, a problemática do tema aqui estudado, os objetivos deste estudo, os procedimentos metodológicos utilizados, as contribuições esperadas, as limitações e por fim a estrutura do trabalho.

Figura 5 – Estrutura da dissertação



Fonte: Elaborada pelo autor.

O Capítulo dois aborda a fundamentação teórica, a qual explicita o conhecimento relevante aos temas principais desse estudo como por exemplo: *Toyota Kata* e *Scrum*.

No Capítulo três a proposição de modelo é feita, a fim de diminuir a lacuna de conhecimento, bem como, propor uma maneira mais natural e facilitada na implantação das rotinas *Toyota Kata* integradas ao *Scrum* visando promover a melhoria contínua no processo de desenvolvimento de *software*.

No Capítulo quatro, tem-se a fase de pesquisa-ação, contemplando desde o problema em questão até a análise e discussão dos resultados encontrados, visando assim verificar o comportamento do modelo.

Por fim, o Capítulo cinco aborda as conclusões e recomendações para trabalhos futuros com base nos levantamentos teóricos e práticos deste trabalho, bem como, os apêndices utilizados para apoio ao estudo.

2 PESQUISA BIBLIOGRÁFICA

Neste Capítulo o intuito é detalhar de forma clara e objetiva os temas deste estudo, os quais resultaram do Mapeamento Sistemático (MS), que pode ser encontrado no Apêndice A, este realizado para auxiliar o trabalho em linhas gerais, bem como, da Revisão Bibliográfica Sistemática (RBS), a qual pode ser encontrada no Apêndice B.

Desta forma, as sessões a seguir, tem o objetivo de explicitar os temas abordados por este estudo de forma mais detalhada.

2.1 LEAN

O termo *Lean*, foi cunhado por Womack e Jones (1990), no *best seller* chamado ‘A máquina que mudou o mundo’. Para a Toyota, *Lean* nada mais é do que a rotina de trabalho ou o sistema de trabalho que a empresa empregou para ultrapassar os obstáculos impostos principalmente pela segunda guerra mundial.

Para isso, utilizou-se talvez a principal transformação que Ohno conseguiu gerar na época de produção em massa, que foi a inserção dos colaboradores como fonte de ativos intangíveis e participantes ativos do processo produtivo e principalmente decisório (DENNIS, 2008).

Desta forma, a Toyota se “apossava” de métodos, técnicas e ferramentas que já eram muito conhecidas da chamada *Total Quality Management* (TQM) (Gestão da Qualidade Total), ou de outras áreas e segmentos. O diferencial maior foi que a empresa colocou isso em prática de fato, principalmente engajando os trabalhadores em todos os processos da empresa, já que estes, como abordado, eram patrimônios vitalícios da organização e com isso seriam as peças fundamentais para retirar a empresa da situação complicada que se encontrava na época (DENNIS, 2008).

Lean preconiza um mapa mental completamente distinto da época que surgiu, talvez, ainda hoje. Na Toyota, ao longo dos anos foram incentivados a aprender e trabalhar de uma forma a qual se tornou a quebra de um paradigma com o surgimento do *Lean* (DENNIS, 2008; ROTHER, 2010).

Diante disso, o Pensamento *Lean* está baseado em cinco princípios fundamentais, sendo eles:

1 – Especifique o valor;

- 2 – Identifique a cadeia de valor;
- 3 – Fluxo;
- 4 – Produção puxada;
- 5 – Perfeição (JONES; WOMACK, 2004).

Corroborando com Jones e Womack (2004) e Dennis (2008), Rother (2010) explicita talvez o principal benefício do Pensamento *Lean*, que está diretamente ligado ao quinto princípio, que é a busca pela melhoria contínua, benefício este que é o foco deste estudo. O autor elucida em sua obra duas rotinas que a Toyota tem incutidas na cultura da empresa, que a torna um diferencial de todas as outras empresas que procuraram implementar *Lean*, que são: *Kata* de Melhoria e *Kata* de *Coaching* (ROTHER, 2010).

Prova disto é uma frase utilizada por Eiji Toyoda, na qual o mesmo dizia que ‘na Toyota estamos constantemente buscando um jeito melhor de fazer as coisas’.

2.2 TOYOTA KATA

Antes de descrever e elucidar os conceitos relacionados as rotinas *Toyota Kata*, é importante dar um passo atrás, e explicitar primeiramente o conceito de *Kata*.

No Japão milenar o conceito de *Kata* vem principalmente das formas básicas, repetitivas e padronizadas de movimentos da arte marcial. No entanto algumas derivações desse conceito podem ser encontradas em Rother (2010, p. 32):

- a) Um modo de fazer alguma coisa; um método ou rotina;
- b) Um padrão;
- c) Uma forma padrão de movimento;
- d) Uma sequência de movimentos pré-definida ou coreografada;
- e) O procedimento habitual;
- f) Um método de treinamento ou exercício;
- g) Uma maneira de fazer alguma coisa;
- h) Um padrão geral;
- i) Uma rotina de prática pequena e estruturada. Aprender e, em seguida, combinar essas rotinas de práticas individuais são uma maneira de desenvolver competência na forma geral ou padrão de fazer algo.

Embora estas definições sejam congruentes e similares, Rother (2010, p. 32-33) explicita uma abordagem de *Kata* mais adequada a visão da Toyota, e que vem ao encontro também deste estudo, sendo ele “um modo de manter duas coisas em alinhamento ou sincronismo com uma terceira. [...] um *Kata* significa um meio de manter os seus pensamentos e ações em sincronismo com as condições dinâmicas e imprevisíveis.”.

Outro ponto importante a ser analisado é que as rotinas *Toyota Katas* são distintas de princípios. Enquanto um princípio auxilia na escolha a rotina mostra como fazer alguma coisa. Desta forma, Rother (2010, p. 33- 34) aponta alguns princípios de *Kata*, os quais são utilizados para a melhoria contínua e adaptação, sendo eles:

O método funciona, em particular, no nível do processo. Seja na natureza ou numa organização humana, a melhoria e a adaptação parecem ocorrer no detalhe ou no nível do processo. Pode-se e precisa-se pensar e planejar em níveis mais altos, como em eliminar a fome ou desenvolver um carro pequeno e lucrativo, mas as mudanças que levam no fim das contas à melhoria e à adaptação são, com frequência, mudanças nos detalhes, tendo como base lições aprendidas nos processos.

Se o objetivo for melhorar cada processo diariamente, então as rotinas *Toyota Katas* estão implícitas e são inseparáveis do trabalho diário nesses processos. As rotinas *Toyota Katas* se tornam o modo como trabalhamos no decorrer do nosso dia (ROTHER, 2010; ROTHER; AULINGER, 2017).

Como os seres humanos não possuem a capacidade de prever o que está por vir, o método que gera a melhoria e a adaptação tem conteúdo neutro; ou seja, é aplicável a qualquer situação. O método, o procedimento, é descrito, mas o conteúdo não (ROTHER, 2010; ROTHER; AULINGER, 2017).

Uma vez que o julgamento não é exato ou imparcial, o método se baseia, sempre que possível, em fatos e não em opiniões ou julgamentos. Em outras palavras, é despersonalizado (ROTHER, 2010; ROTHER; AULINGER, 2017).

O método de melhoria fica além da posse de qualquer líder. Todos na organização trabalham de acordo com o método, independente de quem esteja no comando no momento (ROTHER, 2010; ROTHER; AULINGER, 2017).

Com base na definição de *Kata* abordada anteriormente, e na diferenciação dos princípios das rotinas, o próximo passo é entender melhor o que são estas rotinas.

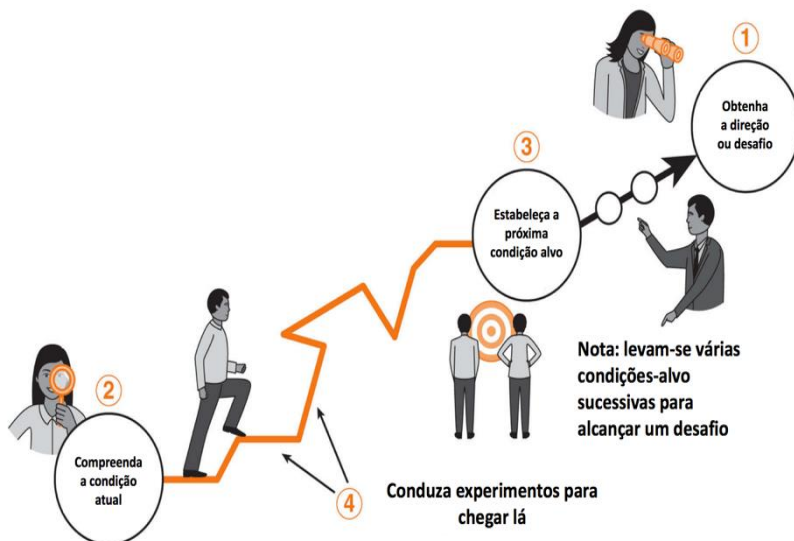
2.2.1 *Kata* de melhoria

Conhecendo a definição de *Kata*, em especial as que fundamentam este estudo, e sabendo que elas indicam resumidamente uma forma ou padrão, de antemão pode-se vislumbrar que a rotina de *Kata* de melhoria nada mais é do que o padrão de melhoria contínua utilizada pela Toyota.

Rother (2010, p. 34) explicita o *Kata* de melhoria como sendo:

[...] a rotina repetida pela qual a Toyota se aprimora, se adapta e evolui. Este *Kata* de melhoria se ajusta com precisão aos atributos enunciados acima e proporciona um modelo altamente eficaz de como as pessoas podem trabalhar em conjunto; isto é, como gerenciar uma organização.

Kata de melhoria, sugere como mostra a Figura 6, a busca contínua e “obscura” pela perfeição, de modo a se atingir a direção ou desafio da empresa. No entanto, na Toyota tem-se em mente que a perfeição é uma eterna busca, e por isso ela utiliza *Kata* de Melhoria como uma rotina (forma) para melhorar e evoluir constantemente toda a organização (JONES; WOMAK, 2004; OSONO; SHIMIZU; TAKEUCHI, 2008; ROTHER, 2010).

Figura 6 – *Kata* de Melhoria

Fonte: Adaptado de Rother e Aulinger (2017).

Sendo *Kata* de Melhoria uma rotina estruturada e constante, a mesma tem seu ponto de início, conforme mostra a figura 6, a partir da obtenção da direção ou do desafio de longo prazo. Essa direção de longo prazo pode variar bastante em termos de tempo, e caso seja atingida ou seja necessária sua adequação, a mesma é substituída por uma nova. Desta forma, sendo ela o norte verdadeiro ao qual está diretamente ligada as necessidades dos clientes, é ela que faz com que a Toyota se mantenha competitiva ao longo do tempo (OSONO; SHIMIZU; TAKEUCHI, 2008; ROTHER, 2010).

A obtenção dessa direção não é algo trivial que não precise de um esforço, bem como, de um consenso por parte de toda empresa. Além disso, sendo essa direção algo distante e vago, o caminho a se percorrer até a mesma é algo obscuro e imprevisível (ROTHER, 2010).

Esta direção está diretamente ligada a filosofia dos fundadores que tinham como princípio colocar uma sensação constante de crise nos colaboradores, a fim dos mesmos nunca se acomodarem (OSONO; SHIMIZU; TAKEUCHI, 2008; EHNI; KERSTEN, 2015).

Após obtida a direção de longo prazo o próximo passo do *Kata* de Melhoria é a compreensão da Condição Atual. Nela os envolvidos irão se aprofundar no entendimento dos problemas (lacunas) que estão sendo

verificados na atualidade a fim de identificar padrões, comportamentos e principalmente fatos que fazem com que os mesmos estejam ocorrendo, compartilhando o conhecimento adquirido com os demais e consequentemente com a organização, com intuito de disseminar o que foi aprendido, para que este problema não precise ser minuciosamente analisado posteriormente, já que o conhecimento principal sobre o mesmo já foi adquirido por alguém (ROTHER, 2010).

Nesse ponto da rotina podem e devem ser utilizados outros tipos de métodos, técnicas e ferramentas a fim de traduzir em fatos e dados aquilo que pode ser apenas um sentimento da equipe. Métodos, técnicas e ferramentas como A3, solução de problemas, gráficos, histogramas, entre outros, servem como fatos e dados que a Toyota se apropria para analisar a condição atual a fim de utilizar a mesma como base para estabelecer a condição-alvo (ROTHER, 2017).

A Condição-Alvo, por sua vez, nada mais é do que o estado a se alcançar no futuro, ou em que situação a empresa se imagina após resolvidos os obstáculos do caminho, sempre na busca da direção idealizada. Esta condição é determinada de forma a ser o mais tangível possível, na intenção de ter claro para todos na empresa qual condição, principalmente de processo, deve ser alcançada, em que dependendo dos aspectos da mesma, esta em alguns casos deve possuir dados mensuráveis a fim de verificar se a condição foi ou não alcançada (ROTHER, 2010; EHNI; KERSTEN, 2015).

A Condição-Alvo na Toyota é estabelecida justamente a partir da condição atual, principalmente por meio de observação direta (*gemba*), bem como, fazendo a ponte da mesma com a direção da empresa.

Sendo a Condição-Alvo vinculada sempre que possível a um processo, esta é visualizada pela Toyota também como um padrão. Porém, diferente de muitas organizações, a Toyota está preocupada continuamente em como irá fazer para atender o padrão, e não especificamente se está atendendo a este padrão dia após dia. Desta forma, a luta da empresa é continuamente se perguntar se a lacuna entre Condição Atual e Condição-Alvo (padrão) está sendo eliminada, e caso não esteja, como a empresa fará para atingir (ROTHER, 2010).

Um ponto importante a ser observado sobre a Condição-Alvo é quanto a sua dificuldade. Conforme apontado por Rother (2010), o CEO da Toyota Engineering Company Ltd., Toshio Horikiri, explicita que uma Condição-Alvo não pode ser estabelecida em seus extremos, ou seja, nem tão fácil que se possa ver como será resolvida, nem tão desafiadora que pareça impossível de ser alcançada. Rother (2010) sugere que uma Condição-Alvo esteja em um horizonte máximo próximo a três meses,

mas que o mesmo obteve sucesso num horizonte de até um mês, e que pode ser que haja Condições-Alvo de até um ano. Além disso, o autor explicita que uma Condição-Alvo adequada deve possuir 4 itens importantes, sendo eles: As etapas do processo com seu passo a passo e seus tempos; as características do processo; o indicador o processo e por fim o indicador de resultado.

Por fim, a última parte da rotina é, para cada um dos obstáculos identificados, a aplicação de ciclos PDCA de forma contínua na busca diária para superar os obstáculos e assim conduzir o comportamento do sistema da Condição Atual para a Condição-Alvo.

O primeiro ponto a ser esclarecido a respeito desta questão é que geralmente o caminho a ser percorrido entre a Condição Atual até a Condição-Alvo é obscuro. Invariavelmente, a realidade, seja na Toyota ou em outra organização qualquer, dificilmente será linear ou previsível.

Desta forma, é necessário um método capaz de atender basicamente todo tipo de problema, bem como, de uma forma estruturada, que mostre a maneira como se deve fazer, e não o que se deve fazer, já que cada problema e cada empresa está inserida dentro de um contexto específico, sendo assim o conteúdo do método é variável (ROTHER, 2010).

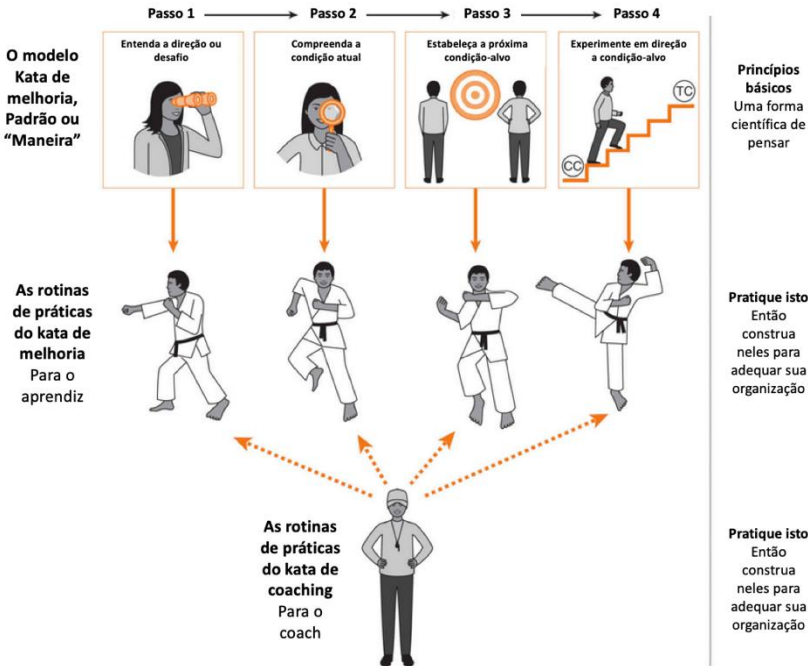
Sendo o PDCA estruturado com base no método científico, o mesmo é utilizado na Toyota principalmente para a aquisição e compartilhamento de conhecimento. É a aplicação contínua do PDCA que leva a empresa sair da Condição Atual e chegar até a Condição-Alvo (ROTHER, 2010; EHNI; KERSTEN, 2015).

Por fim, Rother (2010, p. 77) elucida algo importante a respeito da rotina de *Kata* de Melhoria:

[...] este *Kata* também faz parte do estilo Toyota de gerenciar pessoas diariamente. A psicologia do *Kata* de melhoria é universal e na Toyota todos são ensinados a trabalhar com as linhas dessa abordagem sistemática.

Esta preocupação da Toyota para com o aprendizado e a maneira psicológica de gerir pessoas de forma sistemática pode ser melhor vislumbrada por meio da Figura 7.

Figura 7 – Modelo ou padrão de pensamento científico de trabalho



Fonte: Adaptado de Rother e Aulinger (2017).

2.2.2 Kata de Coaching

Tendo a rotina de *Kata* de Melhoria sido estabelecida, o próximo passo é entender e detalhar como funciona a rotina de *Kata* de *Coaching*, uma outra rotina que a Toyota utiliza principalmente para ensinar todos os funcionários aprender a aprender (ROTHER, 2010).

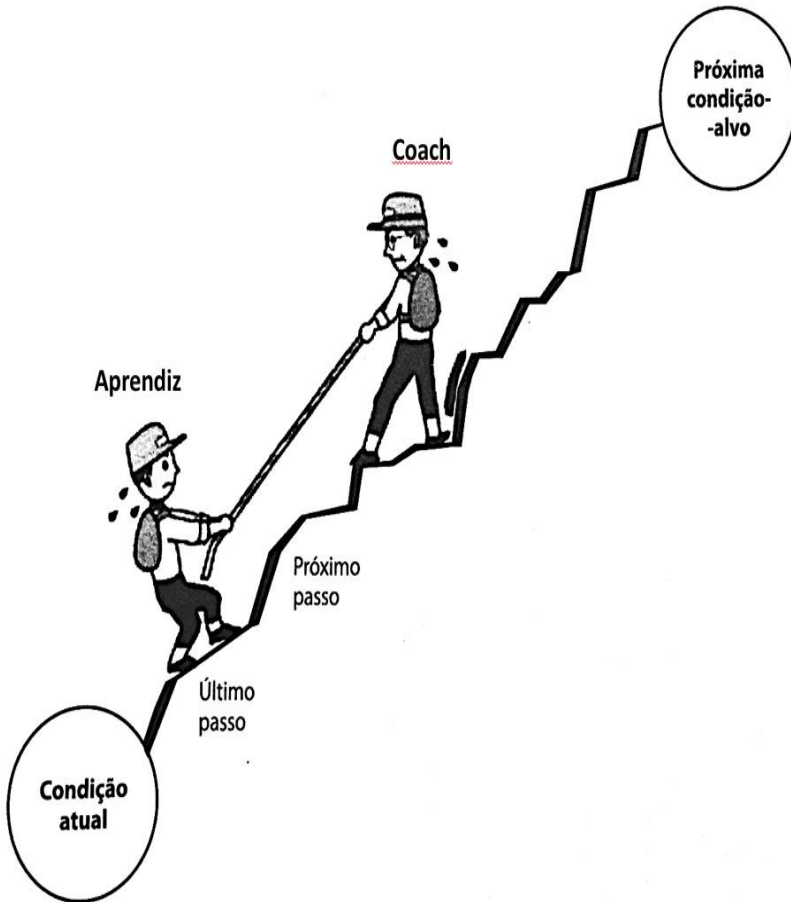
Enquanto *Kata* de Melhoria, é a rotina capaz de fomentar a melhoria contínua por toda a organização, em contrapartida, *Kata* de *Coaching* é a responsável maior por ensinar em especial o que é *Kata* de Melhoria para todas as pessoas da organização, além de compartilhar o conhecimento adquirido na *Kata* de Melhoria e que na Toyota é chamado de *yokoten* – uma abreviatura para *yokoni tenkai-suru*, que significa desdobrar-se ou abrir-se – a qual estimula as pessoas a compartilharem seus conhecimentos (*know-how*), sem restrições, de forma transversal por toda organização (OSONO; SHIMIZU; TAKEUCHI, 2008; ROTHER, 2010).

Como pode-se visualizar na Figura 8, esta traduz de forma simples e objetiva o que é a rotina de *Kata* de *Coaching* para a Toyota. Nela, se pode verificar que a rotina é a responsável por fazer com que o caminho entre a condição atual e a Condição-Alvo seja compartilhada entre *Coach* e aprendiz, de forma constante e diária a fim de fazer com que o entendimento do aprendiz sobre a rotina de *Kata* de Melhoria seja consolidado, para que assim sejam adquiridos e compartilhados novos conhecimentos, e que estes sejam propagados por toda empresa, já que na Toyota todos possuem um *Coach*, e essa união entre mentor e aprendiz segue o *yokoten*, ou seja, compartilhar o *know-how* de forma transversal e abrangente (OSONO; SHIMIZU; TAKEUCHI, 2008; ROTHER, 2010).

Como estas rotinas não caminham sozinhas, faz-se necessário discorrer um pouco sobre sua integração. Desta forma, a Figura 9 ilustra de maneira clara como a rotina *Kata* de Melhoria e a de *Coaching* estão integradas ao longo do tempo.

O objetivo final de ambas as rotinas é o aprendizado, seja por meio do PDCA, como é o caso da rotina *Kata* de Melhoria, seja por meio das perguntas *Coach/aprendiz*, em especial o que o aprendiz aprendeu e precisa aprender, como é o foco da *Kata* de *Coaching*, pois a Toyota acredita e usa como pilar o respeito as pessoas, e este respeito se traduz em tornar estas pessoas seres pensantes, que sejam os protagonistas da melhoria e da vantagem competitiva da empresa e que tornem esse aprendizado algo instituído de forma abrangente (OSONO; SHIMIZU; TAKEUCHI, 2008; ROTHER, 2010).

Figura 8 – Kata de Coaching



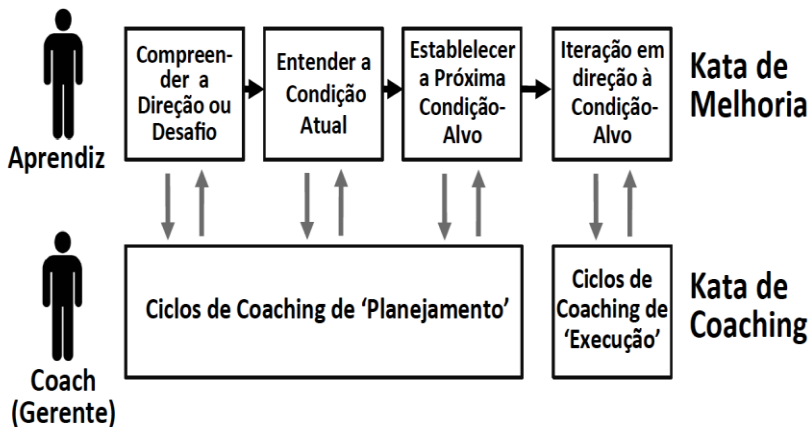
Fonte: Adaptado de Rother (2010).

Para aplicar este procedimento como uma rotina é necessária uma internalização. A Toyota consegue isso usando justamente a figura do *Coach*, e como falado, todos na empresa possuem um. Esta abordagem é didaticamente valiosa, porque suporta os seguintes pressupostos fundamentais de aprendizagem (LIKER; MEIER, 2007 apud EHNI; KERSTEN, 2015, p. 6):

1. As pessoas aprendem a dar pequenos passos por um período de longo prazo,

2. O conhecimento deve ser transferido por um *coach*,
3. O processo deve ser realizado como aprendizagem *on-the-job*, as etapas de aprendizagem pequenas devem ser integradas em uma grande figura e ser padronizadas.

Figura 9 – *Kata* de Melhoria e *Kata* de Coaching



Fonte: Adaptado de Rother (2010).

Embora a rotina *Kata* de *Coaching* percorra toda a rotina de melhoria, pode-se verificar conforme mostra a Figura 10, uma intensa integração entre as duas rotinas no momento dos experimentos dos ciclos PDCA. É nesta hora que o contato e o aprendizado entre mestre e aprendiz são mais significativos, já que os passos do PDCA são diariamente analisados e medidos, a fim de se observar como anda o aprendizado do aprendiz, e como este se depara com os obstáculos, a fim de contorna-los até a Condição-Alvo.

Figura 10 – As 5 questões

COACHING KATA

As cinco questões

- 1 - Qual é a condição-alvo?
- 2 - Qual é a condição atual agora?
----- vire o cartão -----
- 3 - Quais obstáculos você acredita que estão impedindo você de chegar a condição-alvo?
- 4 - Qual é seu próximo passo? (Próximo experimento). O que você espera?
- 5 - O quão rápido nós podemos ir e ver o que nós aprendemos com este passo?

Muitas vezes você trabalhará no mesmo obstáculo com vários experimentos

Refleta sobre o último passo dado

Porque você realmente não sabe qual será o resultado de um passo!

- 1 – O que você planejou no seu último passo?
- 2 – O que você esperava?
- 3 – O que aconteceu realmente?
- 4 – O que você aprendeu?

----->
Retorne para a questão 3

Fonte: Adaptado de Rother e Aulinger (2017).

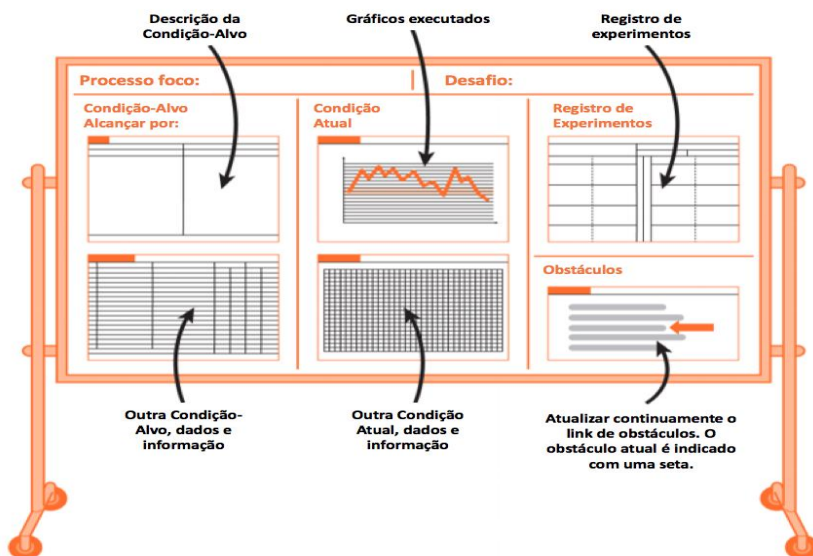
Estas perguntas ajudam *Coach* e aprendiz na contextualização da Condição Atual e Condição-Alvo, bem como, dos experimentos e aprendizados que se adquirem conforme o aprendiz avança na busca pelas soluções planejadas para cada obstáculo nestes experimentos, e dessa forma, *Coach* e aprendiz conseguem juntos analisar de forma holística se os experimentos estão tendo sucesso na busca da Condição-Alvo (ou

superação dos obstáculos), ou se necessitam mudar a trajetória a fim de atingi-la.

Além destas perguntas, *Coach* e aprendiz contam com outra ferramenta visual a fim de auxiliar a construção diária da aprendizagem, sendo ela o “*Storyboard*” de acompanhamento das rotinas *Toyota Kata*, conforme mostra a Figura 11. Como visto nas Figuras 11 e 12, cada aprendiz possui seu próprio *Storyboard* a fim de acompanhar o andamento das rotinas *Toyota Kata*. Neste *Storyboard* pode-se encontrar espaços para a descrição do processo foco da melhoria, o desafio que está sendo abordado, a condição atual do sistema, a condição desejada do sistema, os registros dos experimentos até a condição desejada, e por fim, os obstáculos encontrados rumo a condição desejada.

É por meio destes obstáculos encontrados no percurso da Condição Atual para a Condição-Alvo que o aprendiz trabalha seus experimentos por meio dos ciclos PDCA’s. A cada obstáculo encontrado, um novo ciclo PDCA é realizado a fim de contornar o mesmo na direção da Condição-Alvo. Em suma, são estes obstáculos que proporcionam ao aprendiz a aquisição de conhecimento e o aprendizado, seja contornando ou não estes problemas (ROTHER, 2010).

Figura 11 – *Storyboard*



Fonte: Adaptado de Rother e Aulinger (2017).

A Figura 12 traduz mais fielmente um acompanhamento entre *Coach* e aprendiz utilizando um *Storyboard*, bem como, o auxílio de um segundo mentor a fim de tornar o processo mais colaborativo e compartilhado possível.

Figura 12 – Interação mentor-aprendiz no *Kata* de *coaching*



Fonte: Rother (2017).

Tendo explicitado os conceitos das rotinas *Toyota Kata*, suas integrações, seus objetivos e princípios, o próximo passo é tratar como é sugerido o processo de implantação das rotinas com base no estudo feito por Rother (2010).

2.2.3 Processo de implantação das rotinas *Toyota Kata*

É de suma importância explicitar o conhecimento no que tange o processo de implantação sugerido por Rother (2010), pois conforme verificado na literatura, os estudos do autor são os mais citados no que tange as rotinas *Toyota Kata*, bem como, para verificar e comparar se o

modelo que será proposto se assemelha ou não em algum ponto deste processo, e mesmo não se assemelhando, quais conhecimentos essa diferença pode gerar.

Desta forma, estes conhecimentos podem servir para futuras pesquisas, pois conforme visto no modelo proposto por Rother (2010), o autor se baseou nos estudos de 5 anos focados na Toyota, já o modelo deste estudo é dirigido ao processo de desenvolvimento de *software*, especialmente para quem utiliza *Scrum* como método de gestão, e não na manufatura.

Para melhor explicitar o conhecimento no que tange ao processo de implantação proposto pelo autor, faz-se necessário a exposição do modelo por meio da Figura 13, para que fique mais claro e visível como o mesmo sugere o processo de implantação das rotinas *Toyota Kata*.

Figura 13 – Processo de implantação das rotinas *Toyota kata*



Fonte: Adaptado de Rother (2010).

O primeiro passo no modelo é a criação ou estabelecimento de um grupo de desenvolvimento que será o responsável por promover as rotinas *Toyota Kata* futuramente na empresa. Este grupo teria o auxílio de uma consultoria externa, ou mesmo de um especialista sênior que conhecesse na prática o funcionamento das rotinas *Toyota Kata*. O ideal nesse instante é que se designe um gerente sênior para o grupo, bem como, outros membros da média gerência, a fim de possibilitar com que a implantação futura das rotinas *Toyota Kata* tenha maior aderência e sucesso (ROTHER, 2010).

Como segundo passo, este grupo deve praticar as rotinas no processo natural do ambiente onde se está aplicando *Toyota Kata*, para que o aprendizado seja o mais natural possível. Este passo se dá quase que simultaneamente com o terceiro passo, que é a realização de ciclos de treinamento. Este treinamento se dá obrigatoriamente pela passagem das 5 perguntas de *Kata* de *Coaching*, conforme já explicitado anteriormente, assim que uma Condição-Alvo é estabelecida (ROTHER, 2010).

Sugere-se inicialmente que as rodadas dos ciclos de PDCA sejam mais curtas, levando uma semana para o atingimento da Condição-Alvo. Essa necessidade se dá pelo fato de ser mais de um aprendiz para o *Coach*, bem como, para mostrar resultados de forma mais rápida, para assim os aprendizes ganharem confiança, experiência e ritmo nas rotinas *Toyota Kata*. Aos poucos essas Condições-Alvo podem ser escaladas a uma duração de até um mês (ROTHER, 2010).

Desta forma, o *Coach* é capaz de analisar de forma rápida, passando pelas 5 perguntas, se os respectivos aprendizes precisam de apoio durante o próximo passo ou não. Caso isso não seja necessário, o *Coach* poderá retornar em outro momento num novo ciclo rápido de treinamento (ROTHER, 2010).

Rother (2010, p. 209) explicita que o propósito deste tipo de treinamento é:

- a) Permitir ao treinador (*Coach*) entender rapidamente a condição atual tanto no processo que está sendo melhorado quanto no pupilo, de modo que esse treinador possa julgar qual é o próximo passo mais apropriado.
- b) Fornecer uma rotina para o treinamento de condicionamento.
- c) Reconhecer o esforço do pupilo (Aprendiz).

Tendo realizado os ciclos de treinamento e o aprendizado executando as rotinas *Toyota Kata*, o próximo passo sugerido por Rother (2010) é a elaboração de um plano de desenvolvimento de longo prazo. Neste ponto o autor sugere um período de até 12 meses. Este plano deve ser acompanhado pelo *Coach* (externo ou interno), e o mesmo é elaborado da mesma forma que o A3 das rotinas *Toyota Kata*, mudando apenas o horizonte de tempo (ROTHER, 2010).

Para ilustrar melhor esse plano, e verificar como se dá a sua execução, como é mostrado no último passo do modelo de Rother (2010), a Figura 14 exemplifica de forma sucinta a abordagem do plano de desenvolvimento.

Figura 14 – Plano de desenvolvimento



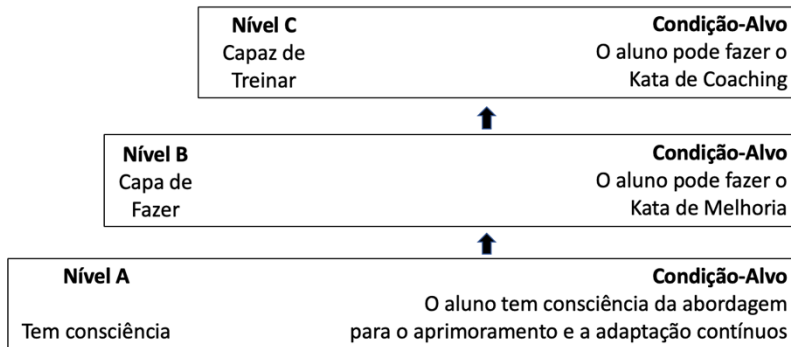
Conexão com os objetivos e as necessidades da organização	Trabalhar junto	Ligação com as metas do departamento	Com autoridade para fazer mudanças	Reconhecimento pelos processos e resultados
Sensação de pertencer	Enxergar potencial para o aprimoramento	Estado final claramente definido	Senso de responsabilidade	Sensação de pertencer
Trabalho desafiador	Senso de responsabilidade	Nem difícil nem fácil demais	Aprendizagem	
		Não muito distante	Desafios frequentes	
			Respeito	

Fonte: Adaptado de Rother (2010).

No plano de desenvolvimento, sendo ele um fiel retrato das rotinas *Toyota Kata*, como visto na Figura 14, o mesmo busca diante de uma Condição-Atual, chegar até a Condição-Alvo. No entanto, entra em ação durante este percurso a inclusão de aprendizes vinculados ao grupo de desenvolvimento, que serão os maiores responsáveis pela execução do plano, e desta forma, como aconteceu com o grupo de desenvolvimento, estes aprendizes também aprenderão sobre *Toyota Kata*, para que de uma forma gradual e escalável, este grupo se amplie vertical e transversalmente a fim de abranger toda a organização.

Para que o conhecimento e o aprendizado a respeito das rotinas sejam realizados de forma gradual e responsável, Rother (2010) sugere 3 níveis de capacidade para os aprendizes, conforme mostra a Figura 15.

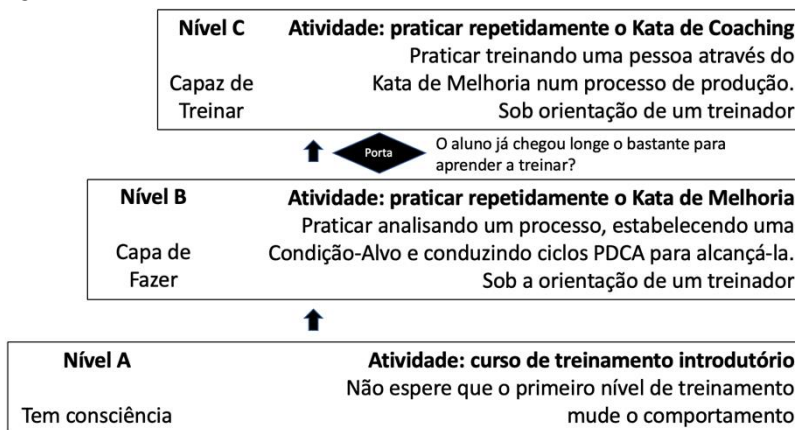
Figura 15 – Níveis de capacidade



Fonte: Adaptado de Rother (2010).

O autor explicita ainda, que ao caminhar da Condição Atual para a Condição-Alvo, existe uma associação dos níveis de capacidade, com os níveis de atividade de treinamento que estes aprendizes estarão passando ao longo do caminho, a fim de saírem da condição de aprendizes, e passarem a ser *Coach's*, como mostra a Figura 16.

Figura 16 – Níveis de treinamento



Fonte: Adaptado de Rother (2010).

Por fim, Rother (2010) sugere que a escalada das rotinas ao passar do tempo se dê, conforme os níveis hierárquicos da empresa, levando em consideração é claro, os níveis que cada empresa possui. Para tal, o autor

mostra como se dá isso, onde e quem fará, de forma simples, por meio da Figura 17, como segue.

Figura 17 – Trânsito de treinamento em uma organização

1. Grupo de desenvolvimento	2. Gestores sênior	3. Gestores de área	4. Supervisores	5. Líderes de equipe
				Comece a treinar regularmente
			Comece a treinar regularmente	Exercite o Kata de Coaching
		Comece a treinar regularmente	Exercite o Kata de Coaching	Exercite o Kata de Melhoria
	Comece a treinar regularmente	Exercite o Kata de Coaching	Exercite o Kata de Melhoria	Aula de conscientização
Comece a treinar regularmente	Exercite o Kata de Coaching	Exercite o Kata de Melhoria	Aula de conscientização	
Exercite o Kata de Coaching	Exercite o Kata de Melhoria	Aula de conscientização		
Exercite o Kata de Melhoria	Aula de conscientização			
Aula de conscientização				
As setas horizontais indicam relações Coach ↔ Aprendiz				

← Envolvimento crescente com o passar do tempo →

Fonte: Adaptado de Rother (2010).

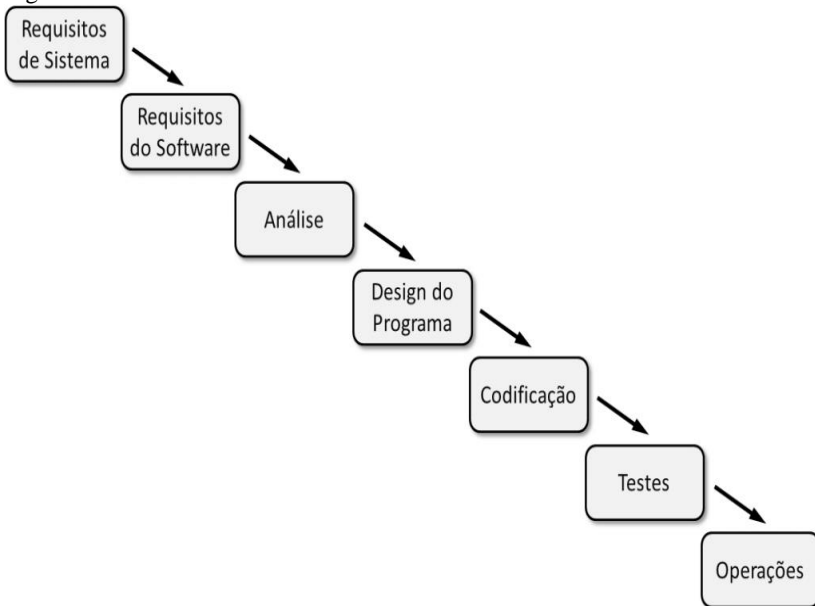
2.3 MÉTODOS ÁGEIS

Os métodos ágeis surgiram da necessidade de se desenvolver *software* de forma ágil, interativa e iterativa, proporcionando assim mais flexibilidade e entregas de valor em um curto espaço de tempo, se comparado com o processo de desenvolvimento de *software* mais tradicional, conhecido também por *waterfall* ou cascata (HIGHSMITH, 2000; COCKBURN, 2002; BOEHM; TURNER, 2003; DECARLO, 2004; HIGHSMITH, 2004; CHIN, 2004).

No modelo *waterfall*, conforme visto na Figura 18, o processo de desenvolvimento de *software* é oneroso e mais lento, sem interações e iterações constantes. As fases do processo são realizadas do início ao fim, e desta forma uma fase só inicia após a outra ter terminado em sua grande parte.

Devido a esta, entre outras, características, este tipo de modelo possui um problema no que tange a mudança e os custos com a mesma. Quanto mais o tempo passa e mais se avançam nas fases do processo de desenvolvimento, maior se torna o custo da mudança, já que o valor entregue ao cliente se dá de fato somente, em sua grande maioria, no final de cada fase, especialmente na de operações.

Figura 18 – Modelo cascata



Fonte: Adaptado de Royce (1970).

Em contrapartida ao modelo cascata de desenvolvimento de *software*, os métodos ágeis tiveram seu início em 1986 com a abordagem do *Scrum* por Takeuchi e Nonaka (1986) e desde então vem se modificando e ganhando cada vez mais espaço, principalmente na área de Tecnologia da Informação e Comunicação (TIC).

Os idealizadores destes métodos se organizaram em 2001, e de uma reunião com 17 pessoas, apresentaram ao mundo o que eles chamaram de “Manifesto ágil”. Esse manifesto nada mais é do que um documento que possui quatro valores fundamentais, conforme pode-se ver a seguir (MANIFESTO AGIL, 2001):

- Indivíduos e interações mais que processos e ferramentas
- *Software* em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

Da mesma forma que os valores, quem desenvolve *softwares* de forma ágil também busca seguir os doze princípios do manifesto, os quais podem ser vislumbrados a seguir:

- A maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de *software* de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar *software* funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- *Software* funcional é a medida primária de progresso.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e designs emergem de times auto organizáveis.

- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo. (MANIFESTO ÁGIL, 2018).

Dentre estes métodos que formaram o manifesto ágil, podemos citar os principais e que deram sua origem, como: *Scrum*, *Crystal Clear*, *Extreme Programming* (XP), *Adaptive Software Development* (ASD), *Feature Driven Development* (FDD) e *Dynamic Systems Development Method* (DSDM).

Os mesmos possuem suas respectivas funcionalidades e usabilidades, como apresentadas de forma sucinta posteriormente, porém, todos possuem o mesmo objetivo comum, entregar valor de forma rápida e flexível (HIGHSMITH, 2000; COCKBURN, 2002; BOEHM; TURNER, 2003; DECARLO, 2004; HIGHSMITH, 2004; CHIN, 2004).

O XP apareceu por volta de 1996, porém sua primeira publicação data de 1999, na qual Beck (1999) aborda as práticas do método necessárias para aplicar os valores e princípios do XP no desenvolvimento de *software*, como: *Planning Game*, *Small Releases*, *Metaphor*, *Simple Design*, *Whole Team*, *Sustainable Pace*, *Stand-up Meeting*, *Collective Ownership*, *Pair Programming*, *Coding Standards*, *Test Driven Development*, *Refactoring* e *por fim Continuous Integration*.

Outro método que deu origem ao manifesto ágil foi o DSDM, criado pelo DSDM Consortium em 1995, o mesmo é um *framework* para desenvolvimento RAD (*Rapid Application Development*). O ponto focal do DSDM é ajustar o tempo e os recursos, para depois ajustar a quantidade de funções de um produto (ABRAHAMSSON et al., 2002). O *framework* contém cinco fases: *Feasibility Study*, *Business Study*, *Functional Model Iteration*, *Design and Build Iteration* e *Implementation*.

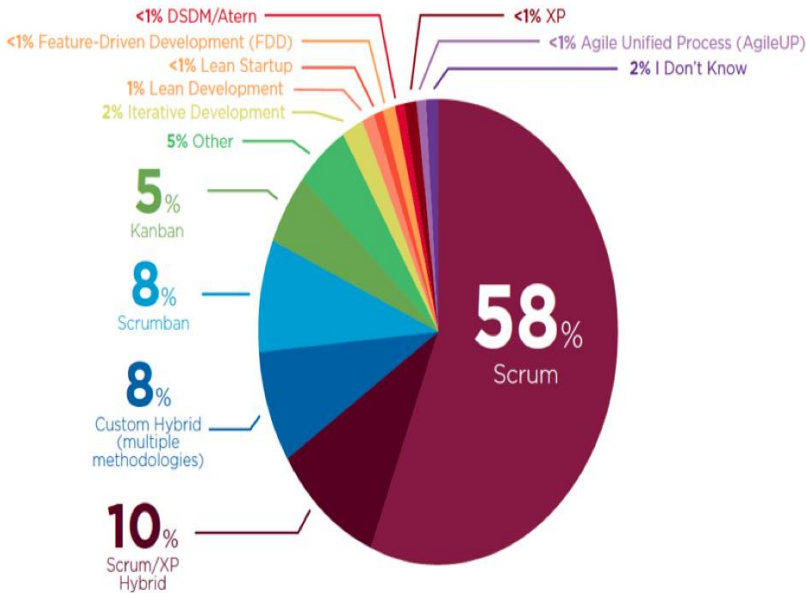
O Crystal por sua vez é uma família de métodos criada por Cockburn (2002) – um dos autores do manifesto ágil – a qual teve seu início por volta de 1998. Segundo o autor, sendo os projetos diferentes, estes devem ser também executados de forma diferente. Esta abordagem possui princípios para a adaptação dos métodos em diversas circunstâncias de projetos. Cada método da família Crystal é nomeado com uma cor: branca, amarela, laranja e vermelha, que será apropriada para seus respectivos projetos, conforme seu tamanho e aspecto crítico. Mesmo distintos, existem regras, valores e características comuns nas metodologias Crystal (COCKBURN, 2002).

Desenvolvido por Highsmith (2000), o ASD aborda o desenvolvimento incremental e iterativo com constante prototipação. O ciclo ASD possui três fases: *Speculate*, *Collaborate* e *Learn*. Estas são assim denominadas para ressaltar o papel da mudança no processo (ABRAHAMSSON et al., 2002). O objetivo do ASD é promover as partes complexas do desenvolvimento adaptativo, especialmente a colaboração e o aprendizado em um projeto (FOWLER, 2005).

Dentre as “práticas” que deram origem ao manifesto ágil destaca-se também o FDD, que como os outros, também é uma abordagem ágil e adaptativa para o desenvolvimento de *software*, nascida de um projeto em Singapura entre 1997 e 1999, a qual destaca pontos de qualidade durante todo o processo e abarca entregas frequentes, com monitoramento minucioso ao longo do progresso do projeto (COAD et al., 1999; PALMER; FELSING, 2002). O FDD aborda cinco processos, durante os quais o projeto e a construção do *software* são realizados: *Develop an Overall Model*, *Build a Features List*, *Plan by Feature*, *Design by Feature* e *Buid by Feature* (COAD et al., 1999; PALMER; FELSING, 2002).

Por fim, encontramos o *Scrum*, o método mais utilizado no mundo para gestão ágil de projetos, dentre todos os outros originais do manifesto, bem como, dos novos que surgiram após o mesmo, conforme visto na Figura 19.

Figura 19 – Métodos ágeis e práticas



Fonte: Versionone Agile Made Easier (2017).

Sendo este o principal método de uso para projetos de TIC, especialmente os de *software*, o mesmo será devidamente abordado na sequência, pois este é o método também utilizado no estudo em questão.

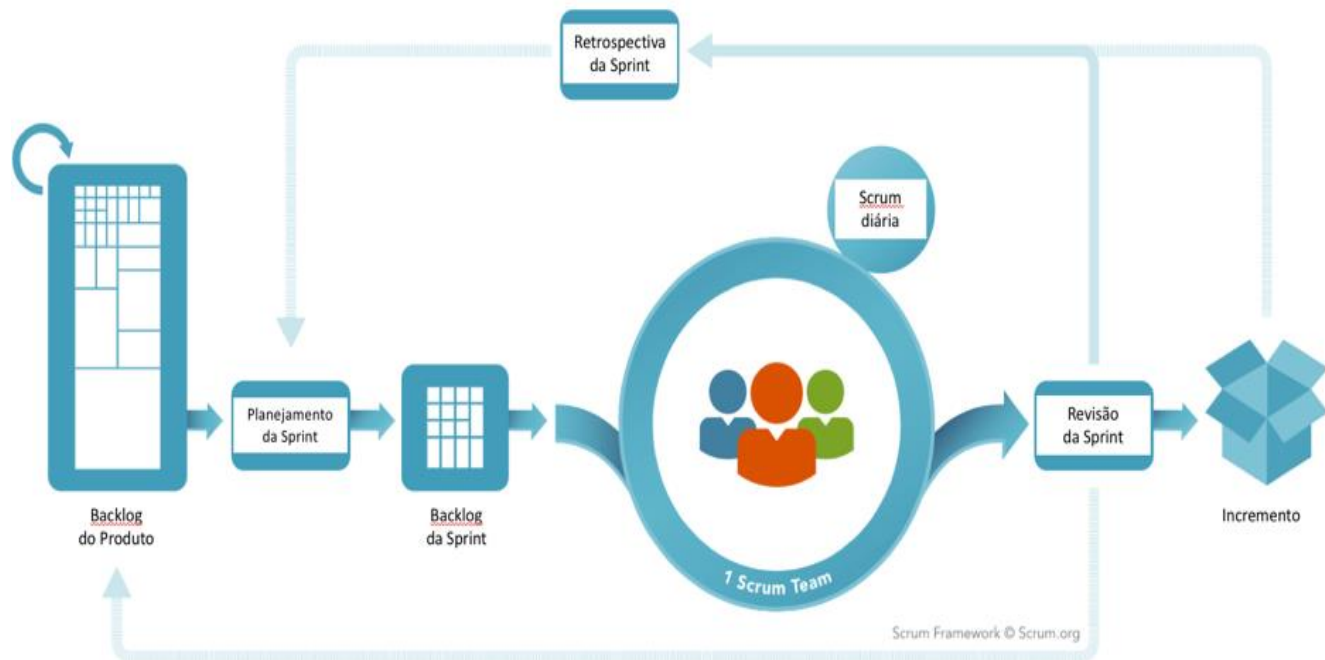
2.4 SCRUM

O nome *Scrum* foi utilizado pela primeira vez por Takeuchi e Nonaka (1986). Nesse artigo os autores discorrem sobre as características de equipes de trabalho em empresas como Fujitsu, Xerox e Honda. Os autores comparam duas abordagens para o desenvolvimento de produtos as quais chamam de velha abordagem e a abordagem “*rugby*”, daí o nome *Scrum*. Nessa abordagem “*rugby*” as equipes de trabalho demonstraram ser mais inovadoras com o processo de desenvolvimento do produto emergindo a partir da interação constante a partir de equipes de trabalho multidisciplinares. Através de equipes auto organizáveis e com a autonomia necessária para tomar decisões ao longo do projeto, as empresas lograram alto níveis de inovação sustentável.

Em 1993, Jeff Sutherland e Ken Schwaber formalizaram um método chamado *Scrum*, utilizando o nome a partir de Takeuchi e Nonaka

(1986), inicialmente focado para o processo de desenvolvimento de *software*. O método foi desenhado para possibilitar equipes a transformar sua forma de trabalho e atualmente é utilizado por diversas organizações pelo mundo. Ambos autores formalizaram o *Scrum* na conferência OOPSLA'95. O método *Scrum* é composto por um *framework* de 5 cerimônias (reuniões), 3 papéis e 3 artefatos, como visualizado na Figura 20, bem como em seu guia oficial. Os papéis interagem nas cerimônias frequentemente com deadlines claros e com o objetivo de melhorar continuamente o processo de trabalho e entregar valor para o cliente ao final de cada iteração de trabalho, chamada *Sprint* (SUTHERLAND; SCHWABER, 2017).

Figura 20 – Framework Scrum



Fonte: Adaptado de Scrum.Org (2018).

2.4.1 Papéis do *Scrum*

O *Scrum Master* (SM) é responsável por garantir que a equipe entenda o *framework* e o aplique, além de desenvolver a equipe do projeto visando com que a mesma seja auto organizada, removendo os impedimentos que bloqueiam o *empowerment* e autonomia dos membros da mesma, já que uma equipe auto organizada é o ponto central do *Scrum* (SUTHERLAND; SCHWABER, 2017).

Ele auxilia as pessoas que não estão no time a entender quais as interações que eles terão com o time que são úteis, e quais não são. Assim o SM facilita a mudança dessas interações, para apoiar as úteis e minimizar as inúteis, favorecendo assim o valor criado pelo time (SUTHERLAND; SCHWABER, 2017).

Mesmo o *Scrum Master* sendo um papel chave no *Scrum*, no qual o mesmo é visto como um líder, existe uma preocupação e um mito ao redor dessa liderança quando se pensa na possível contradição com o trabalho de servir para a equipe, porém não ter autoridade sobre a mesma (COHN, 2011). Isso não se aplica na prática, pois mesmo não tendo autoridade sobre a equipe, o *Scrum Master* se torna a referência máxima do processo o que acaba gerando de forma natural uma liderança especialista e colaborativa, o que é mais importante para o processo.

O papel do SM pode ser dividido em três grandes grupos: o trabalho com a organização, o trabalho com o *Product Owner* (PO) e o trabalho com o time. Vamos ver quais as responsabilidades dele com cada um destes três grupos, conforme aborda Sutherland e Schwaber (2017).

O *Scrum Master* trabalhando para o *Product Owner* serve o mesmo de várias maneiras, incluindo (SUTHERLAND; SCHWABER, 2017, p. 8):

1. Garantindo que objetivos, escopo e domínio do produto sejam entendidos o melhor possível por todos do Time *Scrum*;
2. Encontrando técnicas para o gerenciamento efetivo do *Backlog* do Produto;
3. Ajudando o Time *Scrum* a entender as necessidades para ter itens de *Backlog* do Produto claros e concisos.
4. Compreendendo o planejamento do Produto em um ambiente empírico;
5. Garantindo que o *Product Owner* saiba como organizar o *Backlog* do Produto para maximizar valor;

6. Compreender e praticar a agilidade; e,
7. Facilitar os eventos *Scrum* conforme exigidos ou necessários.

O *Scrum Master* trabalhando para o Time de Desenvolvimento serve o mesmo das seguintes maneiras (SUTHERLAND; SCHWABER, 2017, p. 8):

- Treinando o Time de Desenvolvimento em autogerenciamento e interdisciplinaridade;
- Ajudando o Time de Desenvolvimento na criação de produtos de alto valor;
- Removendo impedimentos para o progresso do Time de Desenvolvimento;
- Facilitando os eventos *Scrum* conforme exigidos ou necessários; e,
- Treinando o Time de Desenvolvimento em ambientes organizacionais nos quais o *Scrum* não é totalmente adotado e compreendido.

O *Scrum Master* trabalhando para a organização de várias maneiras, incluindo (SUTHERLAND; SCHWABER, 2017, p. 8):

- Liderando e treinando a organização na adoção do *Scrum*;
- Planejando implementações *Scrum* dentro da organização;
- Ajudando funcionários e partes interessadas a compreender e tornar aplicável o *Scrum* e o desenvolvimento de produto empírico;
- Causando mudanças que aumentam a produtividade do Time *Scrum*; e,
- Trabalhando com outros *Scrum Masters* para aumentar a eficácia da aplicação do *Scrum* nas organizações.

Além destas responsabilidades e formas de trabalho com cada grupo desses, o SM possui atributos com os quais pode desempenhar melhor seu trabalho, além de facilitar a alta gestão na nomeação de alguém para este tipo de papel, sendo eles (COHN, 2011):

Responsabilidade: sugere ao fato de assumir a responsabilidade de maximizar o rendimento da equipe, além de fazer com que o método seja utilizado pelo time. Esta responsabilidade é assumida como já visto sem

nenhuma autoridade. Pode-se fazer uma alusão nesse ponto com uma orquestra e um maestro, em que este fornece orientação e liderança para um time no intuito de retirar deles de forma conjunta, o que de forma individual seria improvável.

Humildade: nesse ponto o SM faz o que for preciso para auxiliar a equipe a conquistar seu objetivo, ao invés de priorizar por suas necessidades. Um SM que tem humildade sabe que este papel não traz embutido uma vaga privativa no estacionamento, nem um carro da empresa. Ao invés de se vangloriar por coisas que ele fez, um SM humilde faz propaganda das conquistas de seu time.

Colaboração: um *Scrum Master* de valor fomenta uma cultura de colaboração e compartilhamento de conhecimento entre a equipe. Ele facilita um ambiente onde os problemas, as alternativas e as soluções são bem-vindas, e faz com que os integrantes da equipe saibam que podem contar com ele, e com esta cultura para levantar, compartilhar e discutir sobre coisas que sejam um consenso para a maioria, senão todo o time.

Comprometimento: em muitos casos, mesmo um SM não utilizando seu tempo de forma integral para exercer este papel, isso não significa que ele não deva ter comprometimento com o projeto e com a equipe. Sendo um papel de facilitação, ele terá que atuar constantemente com a alta gestão e com a equipe, onde os pontos de vista na maioria das vezes são distintos. Sendo assim, um bom SM mostra comprometimento atuando de forma a solucionar os impeditivos do time, facilitando o processo e a *Sprint*, entre outros.

Influência: usando o termo “política corporativa” um *Scrum Master* sabe das coalisões na empresa e sabe que a utilização de forma correta e adequada da mesma, esta pode ser um trunfo para a equipe. Além disso um *Scrum Master* que consegue influenciar tanto a equipe, quanto fora dela, seja pela persuasão, pelo convencimento ou pela cooperação, terá maiores chances de aplicar o método da forma mais coesa e consistente aos interesses da organização.

Informação: neste ponto o *Scrum Master* deve possuir uma expertise tanto técnica, quanto de negócio a fim de auxiliar a equipe no melhor desempenho do projeto e do próprio time. Embora o SM não precise ser um especialista nem na parte técnica, nem na área de negócio, ele deve conhecer de forma adequada ambas as áreas para melhor desempenhar seu papel.

O *Product Owner* (PO), também conhecido como dono do produto, é o papel que tem a responsabilidade por agregar valor ao produto, além de facilitar o trabalho do Time (SUTHERLAND; SCHWABER, 2017).

De forma conceitual e mais adequada, o PO é o único papel responsável por gerenciar o *Backlog* do Produto, e isto inclui (SUTHERLAND; SCHWABER, 2017, p. 6):

- Expressar claramente os itens do *Backlog* do Produto;
- Ordenar os itens do *Backlog* do Produto para alcançar melhor as metas e missões;
- Otimizar o valor do trabalho que o Time de Desenvolvimento realiza;
- Garantir que o *Backlog* do Produto seja visível, transparente, claro para todos, e mostrar o que o Time *Scrum* vai trabalhar a seguir; e,
- Garantir que o Time de Desenvolvimento entenda os itens do *Backlog* do Produto no nível necessário.

O PO tanto tem a possibilidade de fazer o trabalho explicitado, quanto delegar o mesmo para o Time fazer, entretanto, ele continua sendo o responsável pelos trabalhos (SUTHERLAND; SCHWABER, 2017).

É fundamental para um PO ter sucesso, que toda a organização respeite as decisões que este assumir. Estas decisões do PO são traduzidas e visíveis tanto no conteúdo quanto na priorização do *Backlog* do Produto. Desta forma somente o PO tem permissão para falar com o Time a respeito de mudanças nas configurações de prioridade, e com isso o Time não tem autonomia para tomar ações sobre o que outras pessoas solicitarem (SUTHERLAND; SCHWABER, 2017).

Enquanto o SM auxilia o Time a ser mais eficiente e entregar um incremento de forma eficaz, o PO é o responsável por mostrar de forma clara qual é o objetivo ou a meta a ser alcançada (COHN, 2011).

Para ser um bom PO, o mesmo deve assumir algumas responsabilidades importantes, como por exemplo fornecer a visão do produto. É importante para um Time que este seja envolvido e estimulado com base na visão clara de um produto, e que o mesmo seja visto por esse Time como algo de longo prazo, que possa ser exclusivo, que tenha um diferencial, que o mesmo evoluirá conforme o passar do tempo, entre outros (COHN, 2011).

Além de fornecer uma visão de produto, um bom PO fornece também os limites deste produto. Embora sejam assuntos antagônicos, os limites são importantes para que o Time não perca o foco. Para Cohn (2011, p. 148), os limites vêm na forma de restrição da seguinte forma:

- Preciso disso em junho.
- Temos que reduzir o custo por unidade pela metade.
- É preciso executar com o dobro da velocidade.
- Só pode usar metade da memória da versão atual.

Mesmo tendo um objetivo desafiador, o que é extremamente comum para projetos ágeis, o PO é o responsável por criar o ambiente no qual o Time se encoraja a utilizar suas habilidades de forma a pensar, abordar e aproveitar ao máximo os espaços das suas “caixas” e assim fornecer soluções criativas e inovadoras a fim de entregar o melhor produto possível.

Além disso um bom PO pode ter os seguintes atributos (COHN, 2011):

Disponibilidade: ou seja, ao contrário do que encontramos nos dia-a-dia das organizações, um PO com disponibilidade é aquele que mostra não só uma abertura para com a equipe, mas também um comprometimento e uma credibilidade maior ao projeto.

Especialidade: é de fundamental importância para o *Scrum* e para os projetos que o PO seja um exímio especialista no negócio, além do conhecimento do mercado, do cliente, dos usuários, entre outros, já que é ele que muitas vezes irá tomar decisões baseados nos fatos de cada segmento desses.

Comunicação: devido a essa farta interação com as partes interessadas dos projetos, desde os usuários ao Time de projeto, um PO precisa mostrar uma boa comunicação para interagir com esses *stakeholders* de forma a tornar claras as informações, além de identificar e relacionar o melhor tipo de comunicação para cada parte interessada envolvida na comunicação.

Resoluto: tão importante e comentado quanto a disponibilidade, um bom PO, deve se mostrar decidido e firme em suas decisões e respostas. Um Time saberá identificar um PO que está sempre com dúvidas e se esquivando de tomar decisões difíceis. Um bom PO só mudará sua decisão ou resposta, caso o convençam do contrário com boas razões para isso.

Autoridade: para quem um PO possa exercer sua função de forma adequada, é importante e necessário que o mesmo possua a autoridade cabível para isso. Esta autoridade pode se dar de diversas formas, no entanto um PO que é constantemente desautorizado passa uma imagem

para o Time na qual a função dele acaba transmitindo falta de credibilidade e necessidade.

Por fim, o último papel do *Scrum* é o Time. O Time é o papel responsável pelas entregas que geram um incremento do produto “pronto” de forma com que este possa ser usado de forma a se aproveitar suas funcionalidades e esses incrementos são criados apenas por esse Time (COHN, 2011. SUTHERLANDE; SCHWABER, 2016).

Os Times seguem uma estrutura e organização que difere de muitas outras, já que estes segundo o conceito do *Scrum* são auto organizáveis. Sutherland e Schwaber (2017, p. 6) complementam ainda quando elucidam que “a sinergia resultante aperfeiçoa a eficiência e a eficácia do Time de Desenvolvimento como um todo.”.

Para Sutherland e Schwaber (2017, p. 7) os Times tem as seguintes características:

- Eles são auto organizados. Ninguém (nem mesmo o *Scrum Master*) diz ao Time de Desenvolvimento como transformar o *Backlog* do Produto em incrementos de funcionalidades potencialmente liberável;
- Times de Desenvolvimento são multifuncionais, possuindo todas as habilidades necessárias, enquanto equipe, para criar o incremento do Produto.
- O *Scrum* não reconhece títulos para os integrantes do Time de Desenvolvimento, independente mente do trabalho que está sendo realizado pela pessoa;
- O *Scrum* não reconhece sub-times no Time de Desenvolvimento, independente dos domínios de conhecimento que precisam ser abordados, tais como teste, arquitetura, operação ou análise de negócios; e,
- Individualmente os integrantes do Time de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence ao Time de Desenvolvimento como um todo.

Para Sutherland e Schwaber (2017, p. 7) “O tamanho ideal do Time de Desenvolvimento é pequeno o suficiente para se manter ágil e

grande o suficiente para completar um trabalho significativo dentro da *Sprint*.”.

Quanto ao tamanho dos Times a sugestão que o *Scrum* preconiza é um número mínimo de 3 integrantes, e um máximo de 9, sem considerar o PO e o *Scrum Master*. Caso o PO e o *Scrum Master* executem o trabalho estes poderão então ser considerados nessa contagem (COHN, 2011; SUTHERLANDE; SCHWABER, 2016; PRIKLADNICKI; WILLI; MILANI, 2014).

2.4.2 Cerimônias do *Scrum*

No *Scrum* as reuniões são chamadas cerimônias para dar um contexto informal, ainda que mantendo uma disciplina ritualizada. Isso porque o objetivo dos encontros no *Scrum* é fomentar o compartilhamento do conhecimento, promovendo a interação entre os membros da equipe, responsabilidade e comprometimento que a equipe do projeto tem para com o produto em desenvolvimento.

O core business do *Scrum* é a *Sprint*, um período de tempo que pode variar entre uma e quatro semanas, durante o qual um incremento “Pronto”, ou seja, uma versão passível de utilização do produto, é criada (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

A composição de uma *Sprint* se dá pela reunião de planejamento, as *daily meetings*, o trabalho, a revisão da *Sprint* e a retrospectiva da *Sprint* (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

Durante a *Sprint* não é recomendado fazer alterações que ponham em risco o objetivo da *Sprint*, e sempre que isto acontece deve-se haver uma negociação entre o PO e o Time, para melhor entender e negociar o objeto de trabalho em questão (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

Quando a duração de uma *Sprint* é muito longa, esta acaba por gerar alguns riscos para o processo como por exemplo as mudanças de prioridade, a complexidade pode aumentar e o risco crescer (SUTHERLANDE; SCHWABER, 2017). Para Sutherland e Schwaber (2017, p. 9) “*sprints* permitem previsibilidade que garante a inspeção e adaptação do progresso em direção à meta pelo menos a cada mês corrido. *Sprints* também limitam o risco ao custo de um mês corrido.”.

Durante a execução da *Sprint* a mesma pode ser cancelada caso esta não atenda mais o objetivo proposto, porém somente o PO tem essa autoridade, embora o mesmo possa sofrer alguma influência por parte dos envolvidos no processo (COHN, 2011; SUTHERLANDE; SCHWABER,

2017). Mesmo sendo possível o cancelamento da *Sprint*, isto não é muito comum de acontecer.

Caso a *Sprint* seja cancelada, os itens da mesma que estejam completados e “Pronto” são devidamente revisados e sendo o trabalho potencialmente utilizável, o *Product Owner* o aceita. Já os demais itens da *Sprint* que não foram completos são então estimados novamente e colocados de volta no *Backlog* do Produto (SUTHERLANDE; SCHWABER, 2017).

Diante do que foi visto no tema *Sprint*, podemos verificar que a mesma não só é o coração do *Scrum*, mas também a cerimônia onde as possibilidades mais improváveis e incertas podem acontecer, desde a simples intervenção de um superior com alguma prioridade, até a entrega de incrementos potencialmente utilizáveis. O que deve ficar claro para todos é que a mesma tem seu início na reunião de planejamento, e seu fim na reunião de retrospectiva, além de uma duração que varia geralmente de uma a quatro semanas.

A reunião de planejamento da *Sprint* por sua vez, é o evento que marca o início da *Sprint*, onde o Time junto com o *Scrum Master* e o PO determinam qual será o trabalho a ser realizado no intervalo de tempo definido para entregar um incremento de “Pronto” (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Esta reunião pode durar até 8 horas, para *sprints* de 4 semanas, e gradativamente menos, para *sprints* menores. O *Scrum Master* é o responsável para que esta reunião aconteça, bem como, pelo entendimento de todos do valor da mesma para a entrega da *Sprint* (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Para Sutherland e Schwaber (2017, p. 10) a reunião de planejamento da *Sprint* responde as seguintes questões:

- O que pode ser entregue como resultado do incremento da próxima *Sprint*?
- Como o trabalho necessário para entregar o incremento será realizado?

Neste evento o Time está concentrado em definir quais atividades, ou *features* eles se comprometerão em entregar até o momento final da *Sprint*. Além disso, outra preocupação do mesmo é como se dará a execução da *Sprint*, qual o caminho eles deverão seguir, e quais serão as ações que eles terão de tomar para entregar um incremento potencialmente utilizável.

Por fim, verificou-se o que a literatura explícita de forma intensa, que é a preocupação com a fase de planejamento dos projetos. Sendo esta fase bem-feita, a probabilidade de maiores complicações depois é minimizada e os riscos conseqüentemente também diminuem. É justamente para isso que serve uma reunião de planejamento, para esclarecer as partes envolvidas as dúvidas que por ventura essas tenham, além de identificar e estabelecer qual o planejamento para o próximo período.

A reunião diária é outra cerimônia do *Scrum*. Este evento destina-se ao monitoramento constante das atividades da *Sprint*, servindo para deixar a todos integrantes do Time devidamente informados a respeito dos acontecimentos e impeditivos que acercam a *Sprint*. Esta reunião coloca todos a par dos acontecimentos da última reunião diária, bem como, deixa claro o que deverá ser executado até a próxima reunião (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

Para que não haja problemas de comunicação, bem como, dúvidas e dificuldades, tem-se por boa prática manter este encontro sempre no mesmo horário e local (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Sutherland e Schwaber (2017, p. 12) explicitam que durante a reunião os membros do Time de Desenvolvimento devem esclarecer:

- O que eu fiz ontem que ajudou o Time de Desenvolvimento a atender a meta da *Sprint*?
- O que eu farei hoje para ajudar o Time de Desenvolvimento atender a meta da *Sprint*?
- Eu vejo algum obstáculo que impeça a mim ou o Time de Desenvolvimento no atendimento da meta da *Sprint*?

A reunião diária serve especialmente para verificar se o incremento de “Pronto” que deve ser entregue até o final da *Sprint* está sendo desenvolvido como o esperado. O Time utiliza este encontro como forma de autogerenciamento das atividades a serem desempenhadas na *Sprint* e assim, inspecionar o trabalho realizado dentro dela, para que no final aquilo que foi prometido tenha sido entregue. É comum também logo após esta reunião as pessoas do Time tirarem dúvidas a respeito de itens do *Backlog* da *Sprint*, já que esta reunião não é para entrar em detalhes a respeito dos mesmos (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

O *Scrum Master* é quem fornece os conhecimentos necessários para a melhor execução dessa reunião pelo Time, já que estes são os responsáveis de fato por ela. Além disso o *Scrum Master* deixa claro que apenas o Time é quem deve participar desta reunião (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

As reuniões diárias são capazes e responsáveis por melhorar as comunicações, eliminar outras reuniões, identificar e remover impedimentos, promover rápidas tomadas de decisão, e melhorar o nível de conhecimento do Time, sendo esta uma reunião indispensável pela inspeção e adaptação (SUTHERLANDE; SCHWABER, 2017).

Como pôde ser observado então, a importância da reunião diária é vital para um melhor monitoramento e controle do que está acontecendo na *Sprint*. Ela ajuda a identificar entraves e riscos que não foram possíveis de serem previstos na reunião de planejamento. Além disso ela torna capaz a possibilidade do amplo conhecimento do que se está fazendo durante a *Sprint* e porque não do compartilhamento de conhecimento entre o Time de Desenvolvimento.

A penúltima cerimônia do *Scrum* é a revisão da *Sprint*. Este evento tem a finalidade de inspecionar o incremento da *Sprint* e verificar se o mesmo entregou o que havia sido prometido pelo Time na reunião de planejamento. Nesta reunião os principais papéis são o PO, o Time e outras partes interessadas chave para a *Sprint*, caso necessite. A intenção aqui é fazer com que o PO analise o que foi feito durante a *Sprint* e aprove ou não este desenvolvimento, a fim de garantir que o que está pronto realmente cumpre com a funcionalidade prometida (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Da mesma maneira que nas outras reuniões, o *Scrum Master* é o responsável por fazer com que este encontro aconteça. É ele que fornece o conhecimento e as diretrizes da reunião, e esta pode durar até 4 horas para *Sprints* de 4 semanas, e consequentemente durações menores para *Sprints* menores (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

A Reunião de Revisão inclui os seguintes elementos (SUTHERLANDE; SCHWABER, 2017, p. 13):

- Os participantes incluem o Time *Scrum* e os *Stakeholders* chaves convidados pelo *Product Owner*;
- O *Product Owner* esclarece quais itens do *Backlog* do Produto foram “Prontos” e quais não foram “Prontos”;

- O Time de Desenvolvimento discute o que foi bem durante a *Sprint*, quais problemas ocorreram dentro da *Sprint*, e como estes problemas foram resolvidos;
- O Time de Desenvolvimento demonstra o trabalho que está “Pronto” e responde as questões sobre o incremento;
- O *Product Owner* discute o *Backlog* do Produto tal como está. Ele (ou ela) projeta os prováveis alvos e datas de entrega baseado no progresso até a data (se necessário);
- O grupo todo colabora sobre o que fazer a seguir, e é assim que a Reunião de Revisão da *Sprint* fornece valiosas entradas para o Planejamento da *Sprint* subsequente;
- Revisão de como o mercado ou o uso potencial do produto pode ter mudado e o que é a coisa mais importante a se fazer a seguir; e,
- Revisão da linha do tempo, orçamento, potenciais capacidades, e mercado para a próxima versão esperada de funcionalidade ou de capacidade do produto.

Como resultado tem-se um *Backlog* do Produto revisado, bem como, algumas ideias de possível *Backlog* para a próxima *Sprint*. Dependendo da situação o *Backlog* pode ser devidamente modificado, conforme a necessidade (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Esta é a cerimônia capaz de fornecer ao PO e as partes interessadas o amplo entendimento do que foi realizado durante a *Sprint*, além dos mesmos poderem comparar o que foi prometido, com que foi entregue de fato. Resumidamente, essa reunião serve para colocar em prova se o que eu planejei entregar realmente estava adequado, ou potencialmente utilizável pelo cliente. Se o conceito de “Pronto” não for atendido, o PO tem todo o direito de não aceitar alguma entrega da *Sprint*, para que assim, um item sem funcionalidade não seja colocado em produção no cliente, e isso acabe gerando ainda mais transtorno.

Resumidamente esta reunião é uma barreira de qualidade, onde principalmente o PO faz o serviço de inspetor de qualidade atestando ou não se o “produto” está ou não conforme, e assim aprova ou não a entrega, a fim de evitar maiores problemas futuros.

Por fim, a última cerimônia do método é a retrospectiva da *Sprint*. Nesta cerimônia o Time *Scrum* tem a oportunidade de busca pela melhoria contínua, tanto se autoconhecendo e criticando, quanto procurando melhorar o método (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

É comum para *Sprints* de 4 semanas que esta cerimônia dura aproximadamente 3 horas, e para *Sprints* menores este número vai diminuindo gradativamente. Esta reunião se dá logo após a reunião de revisão, e antes da próxima reunião de planejamento da *Sprint*. Como nas outras cerimônias o *Scrum Master* é o responsável pelo acontecimento e instruções deste evento (SUTHERLANDE; SCHWABER, 2017).

O propósito da Retrospectiva da *Sprint* segundo Sutherland e Schwaber (2017, p. 14) é:

- Inspecionar como a última *Sprint* foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas;
- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias; e,
- Criar um plano para implementar melhorias no modo que o Time *Scrum* faz seu trabalho.

Sendo o *Scrum Master* o responsável pelo acontecimento e instruções dessa reunião o mesmo encoraja o Time a sempre buscar uma forma de melhorar tanto o processo quanto a qualidade do produto, caminhando assim para a melhoria contínua, além de possibilitar com que estas melhorias sejam aplicadas já na próxima *Sprint*, caso isso seja necessário e possível (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

Nesta reunião tem-se claro a ligação entre os princípios de outros métodos e técnicas que foram utilizados em benefício do *Scrum*. Exemplos claros disso são o conceito de perfeição (melhoria contínua) que o *Lean* utiliza e o *Scrum* se apossa do mesmo, bem como, exemplos de empresas como Honda, Natura, Toyota, Microsoft, entre outras, que utilizam técnicas de gestão do conhecimento, como por exemplo o *After Action Review*, a qual o *Scrum* também incorpora nesta cerimônia (NONAKA; TAKEUCHI, 1997; TAKEUCHI; NONAKA, 2008).

Fica claro que neste evento a intenção maior é resumidamente tornar a aplicação do método cada vez melhor, e fazer com que isso proporcione a melhoria contínua do todo, entregando assim produtos com cada vez mais valor aos clientes.

2.4.3 Artefatos do Scrum

O *Backlog* do Produto é uma lista priorizada de tudo que deve acontecer de desenvolvimento no projeto, sendo ela a única fonte de referência em termos de requisitos do mesmo, bem como, de qualquer mudança que possa ser necessária no projeto. O responsável por todo o gerenciamento deste artefato é o PO, e tão somente ele (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Um *Backlog* do Produto inicialmente não possui uma alta definição, ele evolui conforme o andamento do projeto pois o *Scrum* sabe que mudanças irão ocorrer neste caminho. No início o PO tem ainda pouco conhecimento do produto, sendo assim, a quantidade de itens, a priorização e a granularidade deles não é ainda a ideal. Conforme o projeto avança e o conhecimento é compartilhado este cenário começa a mudar, e isto modifica a visão e a gestão do *Backlog* do Produto, fazendo com que a mesma seja cada vez mais assídua e efetiva (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

Sendo assim, em resumo, o *Backlog* do Produto é uma lista que contempla tudo o que precisa ser feito para entregar o produto final, e isso pode abranger desde simples características até as correções que geram as mudanças que devem ser feitas no produto para serem implantadas nas futuras *releases*.

Como falado anteriormente o do *Backlog* do Produto no início ainda não possui a granularidade e o detalhamento necessário para o melhor desenvolvimento possível, porém com o passar do tempo o PO é o responsável por fornecer os detalhes necessários aos itens do *Backlog* do Produto. Isso deve ser um processo contínuo, tanto pelas mudanças esperadas, quanto pela necessidade de detalhamento. PO e Time contribuem para que este processo seja o mais adequado possível, tornando o *Backlog* do Produto mais real e factível para posterior desenvolvimento (COHN, 2011; SUTHERLANDE; SCHWABER, 2017; PRIKLADNICKI; WILLI; MILANI, 2014).

Como o *Backlog* do Produto está priorizado, o topo da lista possui prioridade máxima, além de já possuir um detalhamento suficiente para o Time de Desenvolvimento se apossar do mesmo para desenvolver. Quanto mais abaixo estiver o item no *Backlog*, menos ele é prioritário e menos detalhes o mesmo possui. Estando o *Backlog* priorizado, este é capaz de ser estimado de alguma forma. É aqui onde entra o papel tanto do PO como influenciador do trabalho e auxiliador na tomada de decisões,

quanto do Time, que é responsável pela estimativa dos itens do *Backlog* (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

O *Backlog* do Produto além de ser uma das principais fontes de problemas posteriores, é também o principal artefato que pode tornar o trabalho do Time mais eficiente e eficaz. Isso porque também aqui podem ser utilizados outros métodos e técnicas para melhor compreender o que o cliente deseja com base nos problemas que o mesmo em frente. Técnicas utilizadas no *Lean*, ou mesmo o *Design Thinking* podem auxiliar o PO a extrair do cliente aquilo que ele tem dificuldade de transmitir, e até mesmo proporcionar ao mesmo a solução mais adequada ao problema que ele enfrenta (MICHELS, 2015; SANTOS et al., 2018; PEREIRA, 2018).

Outro artefato do *Scrum* é conhecido como *Backlog* da *Sprint*. No *Backlog* da *Sprint* temos claramente um recorte do *Backlog* do Produto. Neste ponto o Time seleciona do *Backlog* do Produto os itens que estes verificam a possibilidade de entregar uma funcionalidade ou um incremento de “Pronto” no final do período da *Sprint*. Com esse recorte, fica claro para o Time de Desenvolvimento o que será necessário fazer, para poder cumprir com o planejado, e assim contemplar o objetivo da *Sprint* (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Durante a *Sprint* o Time por sua vez não está imune a interferências das partes interessadas, as quais podem invariavelmente solicitar mudanças no escopo da *Sprint*. Isso é normal e até corriqueiro. Porém, sempre que um novo item, não planejado neste caso, tem a necessidade de ser desenvolvido em uma *Sprint* em andamento o Time em conjunto com o PO verificam as questões de impacto e prioridade para o melhor entendimento da situação, bem como da inclusão ou não deste item na *Sprint*, sendo que somente o Time de Desenvolvimento tem autonomia para alterar o *Backlog* da *Sprint* durante a mesma (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

No *Backlog* da *Sprint*, da mesma forma que no *Backlog* do Produto também é possível utilizar métodos e técnicas que facilitem a compreensão do Time, e façam com que o mesmo gere alternativas para uma melhor escolha da solução mais apropriada para cada caso, e assim entregar ao final de cada *Sprint* um incremento potencialmente utilizável (MICHELS, 2015).

Como visto, este artefato é de fundamental importância para alinhar a necessidade e expectativa do cliente, que já foi trabalhada com o PO, com o ponto de vista do Time de Desenvolvimento. Esse entendimento do Time para com essa expectativa do cliente é o ponto inicial primordial que possibilitará o Time desenvolver aquilo que o cliente tanto almeja. Sendo assim, o que é trazido para este artefato já

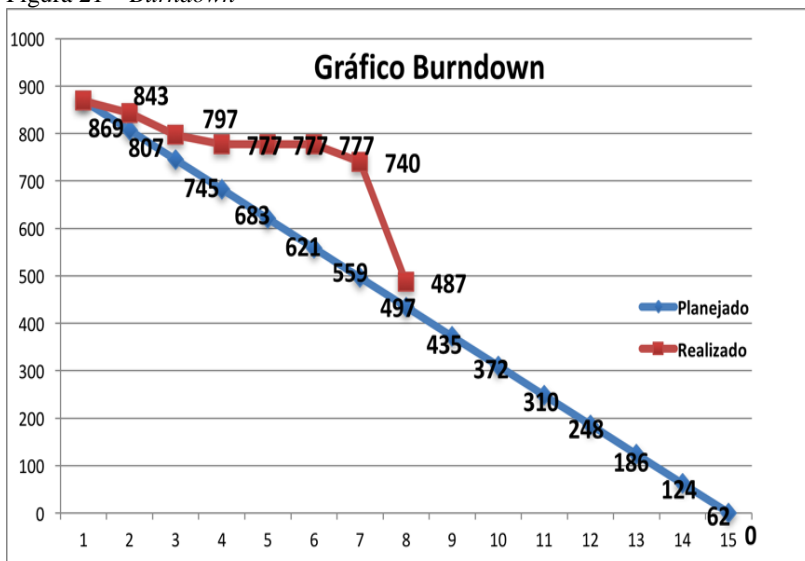
deve estar claro o suficiente para que o Time possa desempenhar e desenvolver o planejado sem maiores problemas.

Por fim, outro artefato de extrema importância para as partes interessadas do projeto, seja ela o Time, o PO, o SM ou mesmo o cliente, são os gráficos de acompanhamento do projeto. O *Scrum* geralmente utiliza dois gráficos para o monitoramento e controle do progresso do projeto, sendo eles o *Burndown* e o *Burnup*. Conforme mostra a Figura 21, um gráfico *Burndown* pode ser melhor visualizado.

Estes gráficos podem ser utilizados em qualquer ponto do tempo, e os mesmos fornecem informações como por exemplo: o total do trabalho gasto até o momento na *Sprint*; o total do trabalho restante para alcançar o objetivo da *Sprint*, entre outras (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Cada envolvido pode ter uma percepção e interesse diferente quanto a utilização do gráfico. Enquanto o PO e o cliente monitoram o trabalho gasto e restante do projeto, o SM e o Time podem estar mais preocupados com o trabalho realizado e restante da *Sprint* (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Figura 21 – *Burndown*



Fonte: Elaborada pelo autor.

Estas práticas como *Burndown*, *Burnup*, entre outras, são constantemente utilizadas para o monitoramento e controle do progresso

do projeto, e estão se mostrando úteis para o processo. No entanto, não são capazes de substituírem a importância da prática, já que em ambientes complexos e inovadores o futuro é desconhecido, sendo assim, somente o que foi realizado pode servir de base para uma tomada de decisão a respeito do que está por vir (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Diante do exposto, estes gráficos são as ferramentas principais da gestão visual que fornecem as partes interessadas insumos tanto para um melhor monitoramento e controle, quanto para a tomada de decisão. É por meio destes e de outros gráficos que SM, PO e Cliente podem se basear para melhor ajustar suas decisões com o sucesso das entregas e do projeto como um todo.

Por fim, o último artefato que o *Scrum* aborda é o incremento. Para Sutherland e Schwaber (2017, p. 16) o incremento é:

[...] a soma de todos os itens do *Backlog* do Produto completados durante a *Sprint* e o valor dos incrementos de todas as *Sprints* anteriores. Ao final da *Sprint* um novo incremento deve estar “Pronto”, o que significa que deve estar na condição de ser utilizado e atender a definição de “Pronto” do Time *Scrum*. Um incremento é um a parte principal inspecionável de trabalho pronto que suporte empirismo no final da *Sprint*. O incremento é um passo na direção de uma visão ou de um objetivo. O incremento deve estar na condição de ser utilizado independente do *Product Owner* decidir por liberá-lo ou não.

O conceito de “Pronto” do incremento pode variar de Time para Time, projeto para projeto ou mesmo dentro das organizações. No entanto o conceito deve estar claro para todos o que este significa. A partir do momento que o conceito é definido, todas os Times e integrantes do mesmo irão trabalhar para que este seja atendido conforme o esperado (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Quando as organizações já possuem este conceito definido, todas as equipes já sabem o que esperar e buscar para contemplar o mesmo. Caso isto não exista na organização, fica a critério do Time *Scrum* definir qual será este conceito e a partir daí trabalhar para que o mesmo contemple o incremento do produto a ser desenvolvido (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Quando diversas equipes estão envolvidas no desenvolvimento de um projeto, todos destas equipes devem ter em mente qual o significado deste conceito, e o que esperar dele. Todos devem trabalhar de forma síncrona, para atender o que o conceito de “Pronto” significa, seja ele definido pela organização ou pelo Time (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Esta definição é quem orienta o Time na seleção de itens do *Backlog* do Produto durante a Reunião de Planejamento da *Sprint* e consequentemente possibilita que o mesmo desenvolva estes itens conforme o esperado. O propósito maior de cada *Sprint* é justamente a entrega de incrementos potencialmente utilizáveis que estão adequados à definição de “Pronto” que o Time *Scrum* ou a organização estabeleceu (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Tendo um incremento contemplado a definição de “Pronto” o PO tem a possibilidade de escolher por liberá-lo ou não. Para isso, todos os incrementos entregues até o momento são integrados, a fim de verificar e validar se os mesmos funcionam juntos (COHN, 2011; SUTHERLANDE; SCHWABER, 2017).

Desta forma, diante do exposto neste tópico em geral, observa-se que por seus aspectos intrínsecos, a Gestão do Conhecimento (GC) está inerente dentro desse *framework* e com isso, o mesmo pode ser integrado com técnicas e métodos da GC sem maiores surpresas, como é o caso das rotinas *Toyota Kata* que busca como resultado final o aprendizado, conforme já visto.

2.5 CONSIDERAÇÕES FINAIS DA REVISÃO BIBLIOGRÁFICA

O primeiro passo neste capítulo foi explicitar de forma objetiva alguns conceitos primordiais do *Lean*, a fim de contextualizar posteriormente o principal tema deste estudo que são as rotinas *Toyota Kata*, oriundas justamente dessa abordagem.

Após esclarecidos alguns conceitos do *Lean*, foram detalhadas as rotinas de *Toyota Kata* para uma melhor compreensão e entendimento, e desta forma, servir para integração posterior com o *Scrum* na proposição do modelo que será o resultado final deste estudo.

Tendo esclarecido as rotinas de *Toyota Kata*, fez-se necessário também explicitar o modelo de implantação das rotinas *Toyota Kata* sugerido por Rother (2010) no qual o mesmo foi elucidado para sua melhor compreensão, bem como sua comparação com o modelo que será proposto posteriormente.

Por fim, também foram explicitados os conhecimentos a respeito do processo de desenvolvimento de *software* por meio dos métodos ágeis, em especial o *Scrum*, que além de ser o método de gestão ágil de projetos mais utilizado no mundo conforme já explicitado anteriormente, também é o método utilizado na empresa em que a pesquisa-ação foi realizada.

Como resultados essenciais deste capítulo, pode-se identificar as oportunidades e barreiras encontradas na literatura que mais implicaram nos estudos que buscaram implantar as rotinas *Toyota Kata*. Desta forma, com base nessas oportunidades e barreiras, bem como, apoiado pelas rotinas *Toyota Kata*, o modelo proposto no capítulo a seguir, buscará não só integrar as rotinas ao *Scrum* e minimizar as lacunas do método, como também contemplar as principais oportunidades e barreiras encontradas na literatura.

3 PROPOSIÇÃO DO MODELO DE REFERÊNCIA PARA IMPLANTAÇÃO DAS ROTINAS *TOYOTA KATA* NO *SCRUM*

Neste Capítulo será descrita a construção da proposição teórica de um modelo para implantação das rotinas de *Toyota Kata* tendo como base o *framework Scrum*. Conforme visto no capítulo anterior, a utilização do *Scrum* como base do modelo se deve a dois fatos principais, sendo eles: o método ágil mais utilizado no mundo para o desenvolvimento de *software* e ser o método em uso no local em que a pesquisa-ação será realizada, a fim de verificar o modelo proposto.

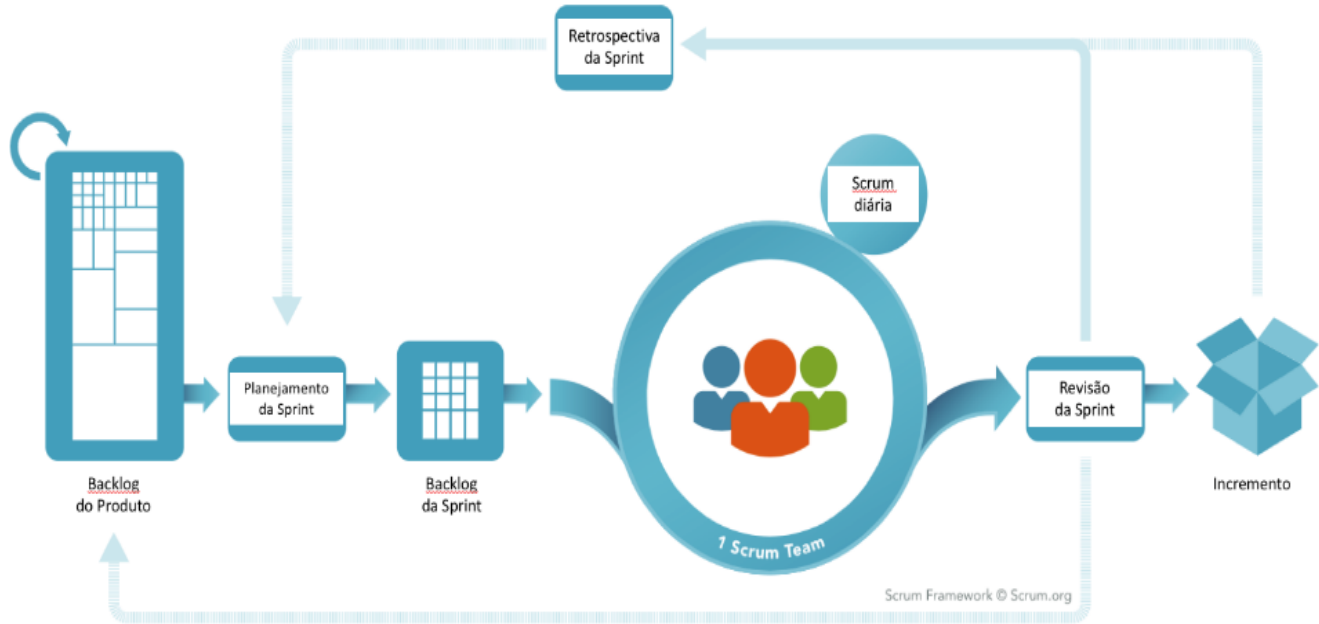
O modelo será construído por meio de três etapas, sendo elas: Revisão, na qual as rotinas de *Kata* e o *framework* que são a base para o modelo são expostos e as lacunas e oportunidades de melhoria do *Scrum* são identificadas; Conexões entre *Scrum* e *Toyota Kata*, na qual ambos são analisados, levando-se em consideração as possíveis relações entre os mesmos com base na análise das lacunas e oportunidades de melhoria identificadas no *Scrum* que podem ser atendidas pelas rotinas *Toyota Kata*; Proposição do Modelo, na qual o modelo é proposto considerando tanto as conexões entre *Scrum* e *Toyota Kata*, explicitadas no item 3.2, quanto os resultados obtidos na revisão da literatura referente as oportunidades e barreiras encontradas.

3.1 REVISÃO

Esta etapa busca revisar o *Scrum*, de forma a identificar principalmente as lacunas e oportunidades de melhoria no *framework*, e com isso relacionar junto a estas oportunidades de melhoria os pontos nos quais as rotinas *Toyota Kata* podem atuar, a fim de melhorar as mesmas.

A Figura 22 mostra as etapas do *framework*, iniciando no *Backlog* do Produto até a última etapa, que é a cerimônia de Retrospectiva da *Sprint*, conforme já explicitado no capítulo anterior. Com base nessas etapas da Figura 22, serão identificadas as lacunas e oportunidades de melhoria do *framework Scrum*.

Figura 22 – Framework Scrum



Fonte: Adaptada de Scrum.org (2018).

Para identificar as lacunas e oportunidades encontradas no *Scrum*, foi realizado um Mapeamento Sistemico (MS), o qual pode ser encontrado no Apêndice A. Com base nisso, foi construído o Quadro 1 para relacionar em cada etapa do *Scrum*, qual etapa das rotinas *Toyota Kata* pode ser utilizada como forma de atender a oportunidade de melhoria, para assim diminuir a lacuna encontrada no método.

Quadro 1 – Etapa das rotinas *Toyota Kata* no *Scrum*

(Continua)

Passo do <i>Scrum</i>	Autor	Lacunas	Oportunidades	Etapa das rotinas <i>Toyota Kata</i>
<i>Backlog</i> do Produto	Akhtar, Ahsan e Sadiq (2010); Hamed e Abushama (2013); Hummel (2014); Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018).	Visão de produto não considera alinhamento entre a estratégia e a operação	Alinhar a estratégia com a operação	Compreender a Direção ou Desafio

Quadro 1 – Etapa das rotinas *Toyota Kata* no *Scrum*

(Continuação)

Passo do <i>Scrum</i>	Autor	Lacunhas	Oportunidades	Etapa das rotinas <i>Toyota Kata</i>
Planejamento da <i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018).	Confusão na visão devido a diversos pontos de vista	Compreender o estado atual de forma sistêmica por meio de dados, fatos e processos e não por opiniões e pontos de vista	Condição Atual <i>Kata de Coaching</i> (Planejamento)
<i>Backlog</i> da <i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Não leva em conta dados, fatos e processo para o estabelecimento de objetivos, tomada de decisão e posterior comprovação dos resultados.	Possuir um objetivo sistêmico comum, baseado em dados, fatos e em cima de processos, e não em pontos de vista	Condição-Alvo
<i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018).	Não utiliza método científico para desenvolvimento do incremento, não aborda as <i>User Stories</i> como sendo “obstáculos”	Utilização do PDCA	Experimentação em direção a Condição-Alvo (PDCA)

Quadro 1 – Etapa das rotinas *Toyota Kata* no *Scrum*

(Continuação)

Passo do <i>Scrum</i>	Autor	Lacunas	Oportunidades	Etapa das rotinas <i>Toyota Kata</i>
<i>Scrum</i> Diária	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018).	Não compartilha conhecimento e aprendizado de forma estruturada	Utilização de PDCA e uma abordagem rotineira para compartilhar conhecimento e aprendizado	<i>Kata</i> de <i>Coaching</i> (Execução)
Revisão da <i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Falta de utilização de dados, fatos e processos para comprovar os resultados e garantir a entrega de valor	Utilização de dados e fatos baseados em processos para comprovar se o que foi entregue está de acordo com o objetivo esperado	Condição-Alvo
Incremento	Hamed e Abushama (2013); Rauf e AlGhafees (2015); Sharma e Hasteer (2016)	Falta de entregas alinhadas a estratégia	Alinhar a direção da estratégia com as funcionalidades entregues	Condições-Alvo e Desafio

Quadro 1 – Etapa das rotinas *Toyota Kata* no *Scrum*

(Conclusão)

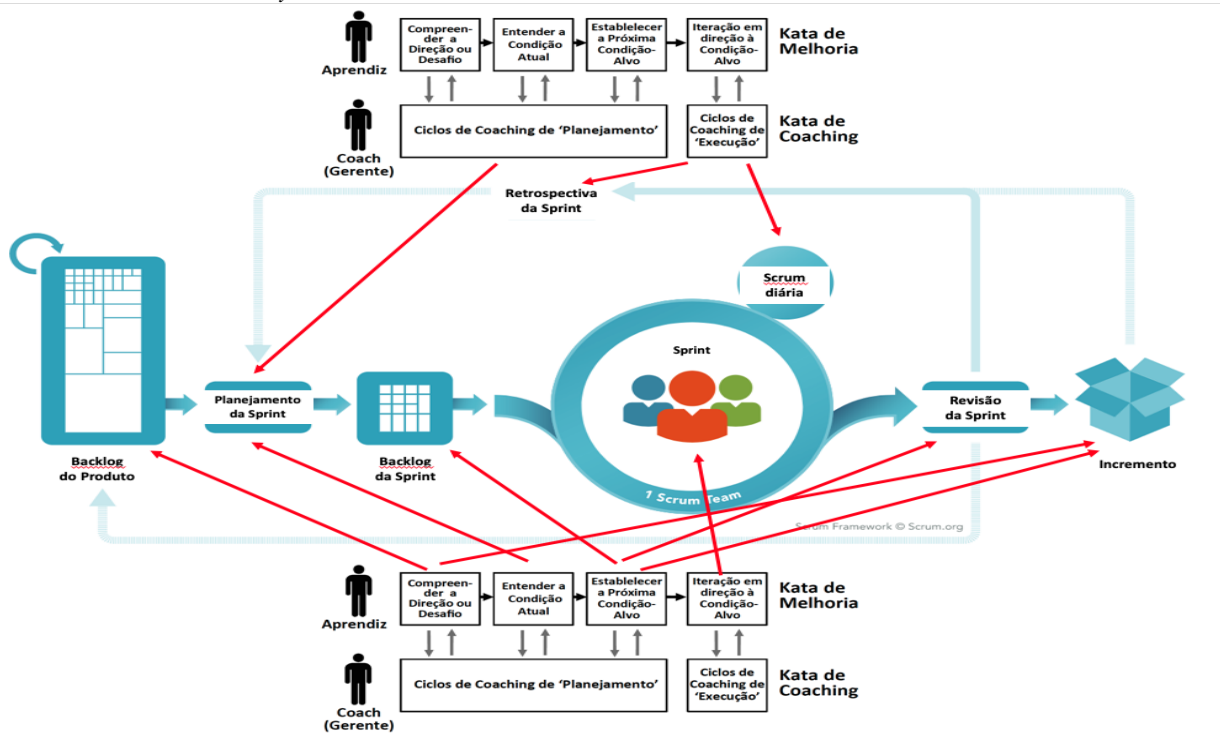
Passo do <i>Scrum</i>	Autor	Lacunas	Oportunidades	Etapa das rotinas <i>Toyota Kata</i>
Retrospectiva da <i>Sprint</i>	Akhtar, Ahsan e Sadiq (2010); Hamed e Abushama (2013); Hummel (2014); Rauf e AlGhafees (2015); Sharma e Hasteer (2016).	Falta de método estruturado para compartilhamento de conhecimento e aprendizado	Utilização de PDCA e uma abordagem rotineira para compartilhar conhecimento e aprendizado	<i>Kata de Coaching</i> (Execução)

Fonte: Elaborado pelo autor.

Com base nas conexões identificadas no Quadro 1, o próximo passo é detalhar as mesmas e mostrar de forma mais clara onde estas são utilizadas inicialmente no modelo proposto. Isto será feito no próximo tópico.

3.2 CONEXÕES ENTRE *SCRUM* E *TOYOTA KATA*

Tendo identificado onde as etapas *Toyota Kata* podem ser utilizadas nas etapas do *Scrum*, o próximo passo em direção a construção do modelo é detalhar onde, por que e como isso será feito, conforme mostra a Figura 23 e seu detalhamento em seguida.

Figura 23 – Conexões *Scrum* e *Toyota Kata*

Fonte: Elaborada pelo autor.

Conforme é ilustrado na Figura 23, as conexões das rotinas *Toyota Kata* que podem auxiliar no *Scrum*, são identificadas, especialmente porque é neste contexto, conforme já informado, que se pretende implantar o modelo.

A primeira conexão entre o *Scrum* com *Toyota Kata* se dá no *Backlog* do Produto. Como existe uma lacuna no que tange o alinhamento entre a estratégia e a operação, a etapa Compreender a Direção ou Desafio, da rotina *Kata* de Melhoria, pode auxiliar nesse alinhamento.

Já na cerimônia “Planejamento da *Sprint*” acontece a utilização de dois passos das rotinas *Toyota Kata* no *Scrum*, sendo elas a compreensão da Condição Atual a qual faz parte da rotina de *Kata* de Melhoria, bem como, o Ciclo de *Coaching* de Planejamento, o qual faz parte da rotina de *Kata* de *Coaching*. Estes passos das rotinas de *Kata* são utilizados nesse momento para tratar a lacuna do *Scrum* quanto a compreensão da situação atual do sistema por meio de dados e fatos baseados em um ou mais processos.

A próxima conexão se dá no *Backlog* da *Sprint*, artefato este resultante da cerimônia de Planejamento da *Sprint*. Neste momento é utilizado o estabelecimento da Condição-Alvo da rotina de *Kata* de Melhoria. A utilização deste passo da rotina tem o intuito tanto de tornar mais claro e explícito o objetivo desejado, quanto o de verificar por meio de dados no momento da Revisão da *Sprint*, e da entrega do Incremento, se o estabelecido na Condição-Alvo foi ou não alcançado, e se não foi, o que faltou para isso. Isso atende justamente a lacuna de não levar em conta dados, fatos e processo para o estabelecimento de objetivos, tomada de decisão e posterior comprovação dos resultados.

No passo seguinte, o qual seria a própria *Sprint* do *Scrum*, são desenvolvidas principalmente suas *User Stories*, a conexão se dá na experimentação em direção a Condição-Alvo, principalmente por meio do PDCA, da rotina de *Kata* de Melhoria. Isto é feito para contemplar a lacuna da falta de utilização de um método estruturado que proporcione aprendizado diário, bem como, uma melhor qualidade na entrega de valor.

Durante a *Sprint* é realizada diariamente no *Scrum*, a cerimônia “*Scrum* Diária”, a fim de se identificar o que foi e o que será feito, bem como os obstáculos que impedem isso. Neste ponto, ao invés de uma conexão, pode se dar uma substituição da “*Scrum* Diária” pela rotina *Kata* de *Coaching*, pois está além de conter os mesmos aspectos da “*Scrum* Diária”, é ainda mais completa pois pode proporcionar maior compartilhamento de conhecimento e aprendizado, que é justamente uma lacuna encontrada no *Scrum*, além de também poder ser utilizada diariamente.

Na cerimônia de Revisão da *Sprint* se dá a próxima conexão. Neste ponto é utilizada a Condição-Alvo a fim de verificar se o resultado alcançado ao final da *Sprint* está de acordo com o que foi estabelecido na mesma, ou não. Isso é feito para comprovar por meio de dados se o estado futuro desejado foi alcançado conforme o esperado, bem como, para fortalecer o compartilhamento de conhecimento e aprendizado entre a equipe, principalmente quando o objetivo estabelecido não foi alcançado, conforme lacuna levantada anteriormente.

Ao término da Revisão da *Sprint* tem-se o incremento potencialmente utilizável, que pode ser uma ou mais funcionalidades que alcançaram a Condição-Alvo estabelecida, e que são passíveis de uso pelo cliente. A conexão neste ponto se dá com o uso da Direção/Desafio da rotina *Kata* de Melhoria, verificando se a Condição-Alvo alcançada pelo incremento está de fato alinhada com a estratégia, conforme planejado no estabelecimento da mesma. Como o *Scrum* não considera de fato um alinhamento entre a estratégia da empresa e sua operação – pois o *framework* por si só não é escalável – além de não possuir o estabelecimento de um desafio sistêmico que crie na equipe um senso de urgência, faz-se necessário algo que contemple isso, como é o caso deste passo da rotina de *Kata* de Melhoria, que também foi identificado nas lacunas do *framework Scrum*.

Por fim, o último passo, e talvez um dos mais importantes, é a Retrospectiva da *Sprint*. Neste momento a lacuna a ser reduzida é o compartilhamento de conhecimento e o aprendizado por meio de um método estruturado. É neste ponto que a conexão entre o PDCA e a rotina de *Kata* de *Coaching* são utilizados na retrospectiva, a fim de fomentar o compartilhamento de conhecimento e o aprendizado da equipe de uma forma estruturada e contínua.

Diante do exposto, foram detalhados conforme o previsto, todas as conexões entre as rotinas de *Toyota Kata* que poderiam ser utilizadas no *Scrum*, a fim de diminuir as lacunas encontradas no mesmo. Com base nisso, o próximo passo será avaliar os constructos encontrados na literatura e analisar onde os mesmos podem contribuir, para assim, combinado com as conexões relatadas aqui, propor o modelo, conforme será visto no próximo tópico.

3.3 PROPOSIÇÃO DO MODELO

Tendo identificado e analisado as conexões entre o *Scrum* e as rotinas de *Toyota Kata*, o próximo passo então é identificar dentre estas

barreiras e oportunidades encontradas na revisão bibliográfica da literatura, quais podem contribuir na proposição do modelo, como e em que momento, conforme constam no Quadro 2.

Quadro 2 – Oportunidades e barreiras da implantação de *Toyota Kata* na literatura

(Continua)

ID	Oportunidades	Barreiras
1	Ambiente colaborativo; Desenvolvimento de liderança; Criação de um ambiente com segurança psicológica	Falta de consenso da equipe na utilização de rotinas e métodos; Conflito de interesses entre os que apoiam e os que não apoiam a iniciativa
2	Compartilhamento de conhecimento; Rotina de aprendizado e melhoria constantes; Desenvolvimento de lideranças; Adaptabilidade	Preparação do ambiente na indústria da construção é diferente e até desprezada, comparada ao ambiente da manufatura; Necessidade de maior experimentação das rotinas de <i>Kata</i>
3	Contrapõe os preconceitos cognitivos; Cria alinhamento na organização do nível estratégico ao nível operacional; Envolve os funcionários através de desafios comuns e sucessos frequentes em relação a ele; A abordagem comum para a melhoria e gestão facilita a colaboração eficaz em toda a organização; Implementação Fractal permite a utilização da inteligência coletiva das organizações para atividades de melhoria contínua; Um número maior de ideias mais criativas para resolver problemas; Uma abordagem sistemática para estender os esforços de melhoria aos aspectos intangíveis, que são cada vez mais importantes em muitas indústrias hoje; Uma abordagem sistemática para muitas das partes difíceis das rotinas <i>Toyota Kata</i> , como compreender a situação atual e escolher o desafio certo	Cria pressão sobre os treinadores que deveriam ser muito competentes no ensino do método para torná-lo acessível para todos na organização; A quantidade de trabalho para criar um nível sustentável de competência e cultura de apoio

Quadro 2 – Oportunidades e barreiras da implantação de *Toyota Kata* na literatura

(Continuação)

ID	Oportunidades	Barreiras
4	A implantação de <i>Kata</i> de <i>Coaching</i> pode facilitar a implantação de <i>Kata</i> de Melhoria; O pré-trabalho pode significar o sucesso da implantação das rotinas <i>Toyota Kata</i> ; desenvolver colaboradores mais adaptáveis e valiosos para a organização	Preparar a estratégia da empresa e identificar a cadeia de valor antes de aplicar <i>Toyota Kata</i> ; Tempo adequado para <i>Coach</i> e aprendiz trabalharem <i>Kata</i> de <i>Coaching</i> ; Não seguir as etapas de implantação das rotinas <i>Toyota Kata</i> , como estabelecido na literatura
5	Aplicar <i>Toyota Kata</i> primeiramente nos processos que agregam valor; automatizar a coleta e os processos para melhorar os índices expostos	Coleta de dados manual; Processos manuais; Nem todos colaboradores foram incluídos para a estimativa de custos
6	Fomentar uma cultura de inovação por toda organização; abordar a melhoria contínua com viés na inovação para o cliente externo.	Falta de experimentação das rotinas <i>Toyota Kata</i> ; Falta de um mapeamento de fluxo de valor atual
7	Adaptabilidade; Integração entre TWI e <i>Toyota Kata</i>	Falta de um mapeamento de fluxo de valor atual
8	Inspiração criativa; Processo padronizado de resolução de problemas; Melhoria nos padrões comportamentais; Impactos das melhorias obtidas; Ciclos curtos no uso de <i>Kata</i> de Melhoria	Falta de um mapeamento de fluxo de valor atual

Quadro 2 – Oportunidades e barreiras da implantação de *Toyota Kata* na literatura

(Conclusão)

ID	Oportunidades	Barreiras
9	Passos curtos nos ciclos PDCA se encaixam melhor na carga de trabalho das pessoas; Reuniões de <i>coaching</i> de 2 a 3 vezes por semana no mínimo; as perguntas de <i>Kata</i> de <i>Coaching</i> empurram o <i>Coach</i> e o aprendiz em uma direção mais assertiva.	Baixa qualidade na formulação de hipóteses; Falta de consciência em relatar o aprendizado; Reuniões não serem curtas e em horário regular; Falta de reuniões regulares entre <i>Coach</i> e aprendiz; Falta de observação do <i>gemba</i> ; Reunião de <i>Kata</i> de <i>Coaching</i> a apenas uma vez na semana;
10	Definição da condição alvo; Utilização do <i>Hoshin Kanri</i> ; Utilização do mapa de fluxo de valor	Falta de utilização do PDCA de forma sistêmica

Fonte: Elaborado pelo autor.

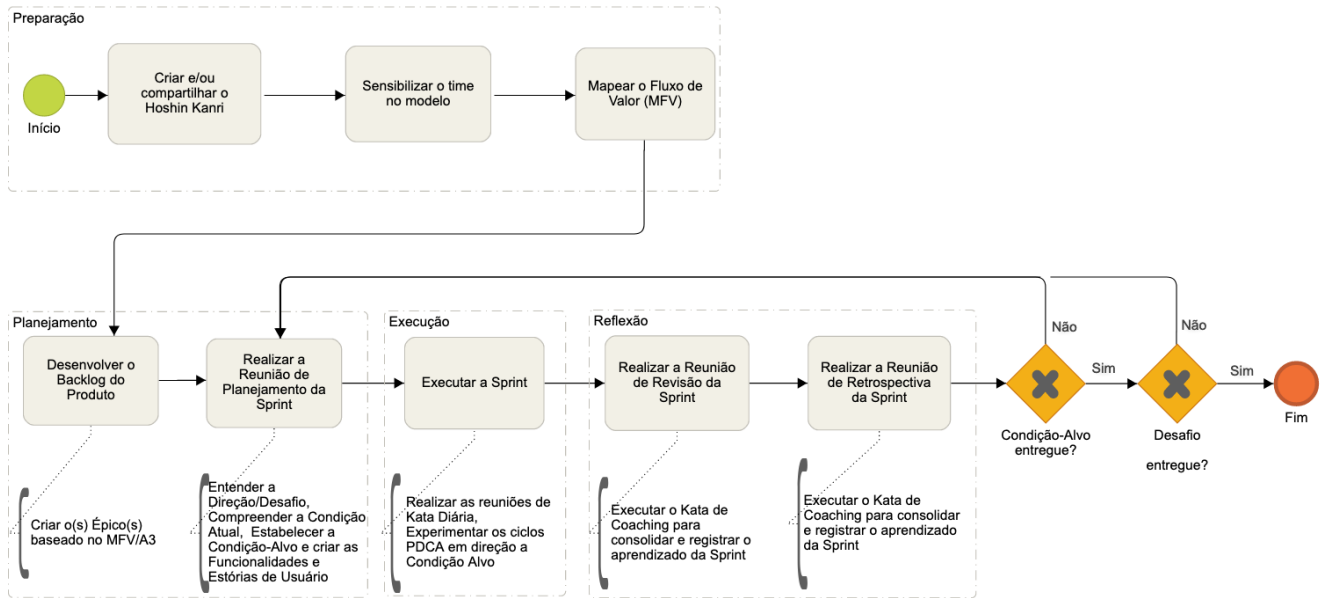
Desta forma, identificou-se com base na frequência dos relatos na RBS que as três principais barreiras relatadas na literatura conforme constam no Quadro 2, foram: 1 - falta de direção e preparação do ambiente (sensibilização), 2 - falta de mapa de fluxo de valor, e 3 - a falta de reuniões de *Kata* de *Coaching* constantes.

Por sua vez, utilizando do mesmo critério das barreiras, as três principais oportunidades encontradas foram: 1 - ciclos curtos, 2 - compartilhamento de conhecimento, e 3 - adaptabilidade.

Diante disso, o modelo é proposto conforme mostra a Figura 24 a fim de contemplar tanto as conexões identificadas no tópico anterior, quanto as principais oportunidades e barreiras encontradas na RBS.

Além disso, algumas oportunidades e barreiras do Quadro 2, que são identificados por seus ID's já possuíam relação direta com o *framework Scrum* ou com as rotinas *Toyota Kata* de forma intrínseca, sendo assim, continuaram a manter esta relação, sendo eles: 2, 3, 4, 6, 7, 8, 9, 10. Os demais, que poderiam ser tratados pelo modelo foram abordados conforme a possibilidade.

Figura 24 – Proposta do Modelo



Fonte: Elaborada pelo autor.

Diante disso, faz-se necessário explicar como se dá a implantação do modelo, e para tal, como mostra a Figura 24, o mesmo foi dividido em 4 fases, a fim de melhor organizar e compreender a sua implantação e utilização, conforme segue.

3.3.1 Preparação

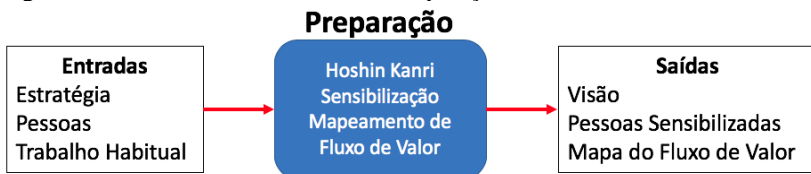
Na primeira fase chamada de Preparação, a mesma inicia com a criação e/ou compartilhamento do *Hoshin Kanri* da empresa, a qual pode ser relacionada a Visão das rotinas *Toyota Kata*. Dessa forma, o *Hoshin Kanri* atende diretamente a barreira de falta de visão/direção encontrada na RBS.

Após esta definição, a próxima etapa da fase é a sensibilização do time no modelo proposto, a qual é responsável pela preparação do ambiente para a implantação do modelo. Esta etapa por sua vez, também era uma das principais barreiras encontradas na RBS.

Por fim, na fase de Preparação, a fim de contemplar a última barreira da RBS que era a falta de mapeamento de fluxo de valor atual, após a sensibilização, o intuito é justamente mapear o fluxo de valor atual com as partes interessadas mais adequadas, para desta forma compreender não só a cadeia de valor, como também identificar os possíveis problemas nela.

Com isso, conforme mostra a Figura 25, a fase de Preparação possui as seguintes entradas e saídas.

Figura 25 – Entradas e Saídas da Fase Preparação



Fonte: Elaborada pelo autor.

3.3.2 Planejamento

Na fase de Planejamento, a primeira etapa é a de desenvolver o *Backlog* do Produto, a qual, conforme já visto, é apoiada pela rotina de *Kata* de Melhoria, ao fazer uso do entendimento da Direção/Desafio para alinhar a estratégia a operação, e desta forma estabelecer e priorizar o *Backlog* da melhor forma com os subsídios resultantes do MFV ou do A3.

Como neste estudo foi utilizado o A3, o mesmo forneceu a causa raiz, a qual gerou o respectivo épico na ferramenta utilizada pelo time. Independente da ferramenta que se deseja utilizar, o conceito de épico é o mesmo para todas.

O *Backlog* do Produto é atualizado constantemente, conforme é realizada a etapa de Planejamento da *Sprint*, em que o modelo conta com o apoio da compreensão da Condição Atual, a fim de entender como é o estado atual do sistema, antes de iniciar o caminho em direção a Condição-Alvo.

Nesse momento são criadas a funcionalidade, utilizando o Desafio da rotina *Kata* de Melhoria, bem como as Estórias de Usuário, por meio da Condição-Alvo, com intuito de estabelecer por meio de dados, fatos e baseados em processos, qual será a condição do sistema em um determinado ponto futuro quando acabar a *Sprint*, marcando assim o final da fase, e da mesma forma, conforme mostra a Figura 26, pode-se verificar quais são as entradas e saídas desta fase.

Figura 26 – Entradas e Saídas da Fase Planejamento



Fonte: Elaborada pelo autor.

3.3.3 Execução

A fase de Execução, por sua vez, é iniciada pela etapa da *Sprint*, que é chamada de coração do *Scrum*, por ser o período de tempo onde o trabalho é realizado. Neste ponto a rotina *Kata* de Melhoria auxilia com a experimentação em direção a Condição-Alvo, a qual foi estabelecida na etapa anterior.

Seguindo o modelo, a etapa que anda em paralelo com a *Sprint* chamada *Kata* Diária, faz uso da rotina de ciclos curtos de *Coaching* de execução, para fortalecer o compartilhamento de conhecimento e aprendizado, para a assim atender uma das três principais oportunidades encontradas na literatura.

Vale salientar nesse momento algo que a literatura não deixa explícito, mas que pode ser importante para quem quiser tanto aplicar o modelo, quanto para quem se interessa nas rotinas *Toyota Kata*.

A *Kata* Diária busca acompanhar a evolução da equipe em direção a Condição-Alvo estabelecida, utilizando para isso os ciclos curtos de *Coaching* de execução, e isso não é segredo na literatura. No entanto, conforme mostra a Figura 27, entre a Condição Atual e a Condição-Alvo existe um caminho obscuro, e nele encontra-se a fronteira do conhecimento, seja em qual for o ponto do caminho.

Além disso tem-se que lidar também com os obstáculos, que podem ou não estar dentro dessa fronteira do conhecimento. A principal diferença desses obstáculos que estão dentro ou fora dessa fronteira é a maneira como se atua neles.

Nos obstáculos que se encontram dentro da fronteira do conhecimento, os mesmos invariavelmente são tratados de forma a se desviar deles, e é aqui onde pode se ter novos benefícios. Conforme será visto posteriormente de forma empírica ao implantar o modelo, quando os obstáculos estão dentro da fronteira, as pessoas tendem a atuar de forma conhecida, e assim não procuram criar novas maneiras de se resolver um problema que dominam.

O lado bom deste tipo de atuação é que mais de um obstáculo pode ser abordado durante o ciclo, levando-se em consideração o sistema. No entanto, o lado ruim é que as pessoas acabam por não criar novos conhecimentos e aprendizados por não abordarem a resolução do problema com outro ponto de vista, para desta forma aprender novas formas de resolver um problema conhecido, bem como, melhorar o sistema.

Figura 27 – Obstáculos e a Fronteira do Conhecimento

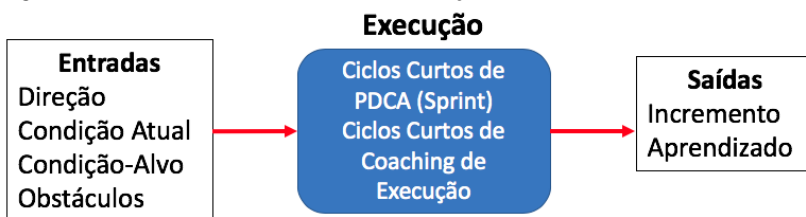


Fonte: Adaptada de Rother e Aulinger (2017).

Em contrapartida, quando os obstáculos não se encontram dentro da fronteira, soluções inovadoras, novos conhecimentos e novos aprendizados surgem naturalmente, pois a forma como se tratam os problemas neste ponto, não podem ser a mesma como são tratados dentro da fronteira.

Da mesma forma que nas fases anteriores, a Figura 28 também mostra as entradas e saídas desta fase.

Figura 28 – Entradas e Saídas da Fase Execução



Fonte: Elaborada pelo autor.

3.3.4 Reflexão

Por fim, na última fase chamada Reflexão, que nada mais é que uma reflexão do ocorrido em toda Sprint, a próxima etapa é a Revisão da Sprint, a qual é apoiada novamente pela Condição-Alvo, com intuito de verificar se o estado desejado do sistema, foi de fato alcançado ou não, e caso não tenha sido, se adaptar para a próxima Condição-Alvo.

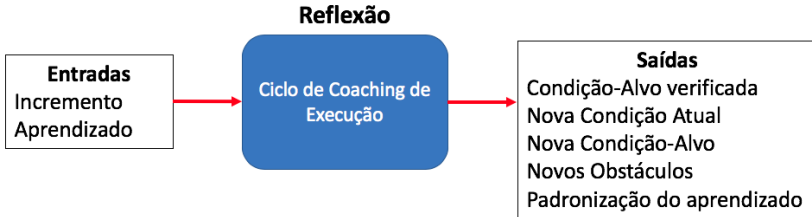
Com isso, o modelo finaliza na etapa de Retrospectiva da *Sprint*, que utiliza da rotina de ciclos de *Coaching* de execução mais uma vez, além da Condição-Alvo novamente, para identificar o que deu ou não certo na caminhada comparando o esperado com o realizado, e assim compartilhar conhecimento e aprendizado de forma sistêmica e contínua a fim de servir como lições aprendidas para a próxima Condição-Alvo e para ciclos futuros.

É neste ponto do modelo que o mesmo reforça a importância da adaptabilidade e faz uso da mesma, com o intuito de analisar o que deu certo e errado durante o caminho da Condição Atual até a Condição-Alvo, para desta forma aprender e adaptar o caminho da próxima Condição-Alvo.

Por fim, verifica-se se tanto o Incremento quanto o Desafio foram entregues ao cliente conforme o esperado e se estão de fato alinhados com a estratégia e com o planejado.

Desta forma, o modelo finaliza com as seguintes entradas e saídas, conforme mostra a Figura 29.

Figura 29 – Entradas e Saídas da Fase Reflexão



Fonte: Elaborada pelo autor.

3.4 CONSIDERAÇÕES FINAIS SOBRE O MODELO PROPOSTO

Conforme visto no capítulo, o *Scrum*, por mais que seja o método ágil mais utilizado no mundo para o desenvolvimento de *software*, ainda assim, possui suas fragilidades e lacunas.

Diante dessas fragilidades e lacunas que foram explicitadas em cada etapa do método, procurou-se então verificar nas rotinas *Toyota Kata*, quais e como estas poderiam apoiar o *Scrum* para o seu melhor funcionamento, minimizando assim suas fragilidades.

Desta forma, o modelo procurou identificar, relacionar e atuar com as rotinas de *Toyota Kata* nas fragilidades do *Scrum*, sendo elas: na visão de produto não estar alinhada entre estratégia e operação, foi utilizado a compreensão da direção ou desafio. Para a confusão com diversos pontos de vista, a atuação se deu utilizando a Condição Atual. Para a falta da utilização de dados e fatos para estabelecimento de objetivos e tomada de decisão, o modelo utilizou a Condição-Alvo da rotina de *Kata* de Melhoria. Já na falta de compartilhamento e aprendizado esta lacuna foi contemplada pelos ciclos de execução de *Kata* de *Coaching*. Por fim, para a lacuna de falta de entregas alinhadas a estratégia, foi utilizado o Desafio das rotinas *Toyota Kata*.

Além de contemplar e atuar nas fragilidades encontradas no *Scrum*, o modelo proposto procurou contemplar também as três principais oportunidades e barreiras encontradas na literatura, bem como, verificar dentre todas as que foram encontradas, se as mesmas já eram abordadas de forma intrínseca tanto pelo *Scrum*, quanto pelas rotinas de *Kata*.

Para isso, o modelo contemplou a barreira de falta de direção e preparação do ambiente utilizando o *Hoshin Kanri*. Já para a lacuna da falta de mapa de fluxo de valor, o modelo fez utilização da etapa de

mapeamento do fluxo de valor, e para a lacuna de falta de reuniões constantes foi utilizado os ciclos de execução de *Kata* de *Coaching* diariamente.

Já para as três oportunidades encontradas na literatura, sendo elas, ciclos curtos, compartilhamento de conhecimento, e adaptabilidade, o modelo contemplou respectivamente com o seguinte: realização de *Sprints* (ciclos curtos) de quinze dias; realização de ciclos de execução de *Kata* de *Coaching* diários, bem como, quinzenais na reunião de Retrospectiva da *Sprint*, para compartilhar conhecimento e aprendizado, e por fim, com a análise da Condição-Alvo real comparado ao esperado, para verificação de desvios, obstáculos e aprendizados a fim de poder se adaptar e melhorar o caminho entre a próxima Condição Atual e Alvo.

Desta forma, a proposição do modelo contemplou de forma abrangente, tanto as fragilidades encontradas no *Scrum*, quanto as barreiras e oportunidades encontradas na literatura a respeito da utilização das rotinas *Toyota Kata*, fazendo assim com que o mesmo possa ser largamente aplicado na indústria do *software*, a fim de verificar seus benefícios e resultados empíricos.

4 PESQUISA-AÇÃO

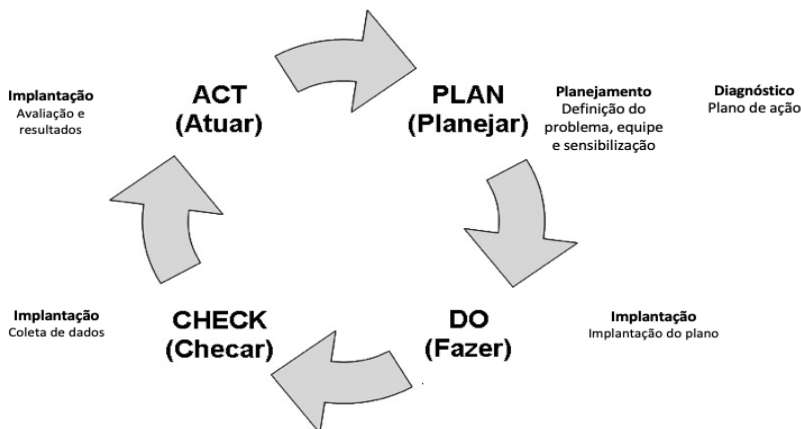
Tendo sido concluída a revisão dos conceitos teóricos, bem como, da proposição do modelo, o próximo passo da pesquisa é o teste e verificação do mesmo, a fim de averiguar seu comportamento, bem como, identificar novos aprendizados decorrentes da utilização do mesmo.

Sendo assim, a Figura 30 mostra de forma objetiva como se dará o processo de pesquisa-ação a qual foi adaptada a partir do estudo de Engel (2000), pois alguns itens da mesma já foram contemplados anteriormente nesse estudo, bem como, do ciclo PDCA de Deming (1986).

O processo de pesquisa-ação foi dividido em três grandes etapas, sendo elas: Planejamento, Diagnóstico e Implantação. O Planejamento e o Diagnóstico estão relacionados com a fase Planejar do ciclo PDCA de Deming (1986). Já a Implantação percorre o ciclo PDCA da fase, fazer, até a Atuar. A etapa de Planejamento contempla as atividades de definição do problema, equipe e sensibilização. Na etapa de Diagnóstico tem-se o plano de ação. Por fim, na etapa de Implantação a mesma possui as atividades de implantação do plano, a coleta dos dados, e a avaliação e resultados obtidos.

Desta forma, a pesquisa-ação segue o ciclo PDCA, sendo atendidos também os princípios e conceitos do método científico, a fim de estruturar melhor a forma como o estudo foi conduzido.

Figura 30 – Processo da pesquisa-ação



Fonte: Elaborada pelo autor.

4.1 PLANEJAMENTO

Esta etapa da pesquisa-ação é constituída pelas seguintes atividades: definição do problema a ser abordado, definição da equipe que será responsável pela implantação do modelo, e a sensibilização da mesma sobre o modelo, conforme serão vistos de forma mais detalhada a seguir.

4.1.1 Definição do problema

O primeiro passo deste tópico, conforme já explicitado e seguindo os princípios da pesquisa-ação sugerido por Engel (2000), é definir o problema de pesquisa. Desta forma, neste ponto da pesquisa-ação o problema que será tratado é a falta de conhecimento do comportamento do modelo proposto. Assim, se fará a verificação do comportamento do modelo no momento que este é implantado e utilizado.

Como a pesquisa-ação visa testar e verificar o comportamento do modelo, a primeira coisa a se fazer neste momento é justamente caracterizar e explicitar este ambiente, a fim de tornar mais claro o contexto de aplicação do modelo.

Desta forma, o ambiente de pesquisa é caracterizado como uma empresa de desenvolvimento de *software* privada que tem sede na cidade de Florianópolis-SC e possui aproximadamente 1500 funcionários, a qual será chamada nesta pesquisa de “Renew”. Esta empresa está estruturada em 3 grandes unidades de gestão, as quais focam nos seguintes mercados respectivamente: Justiça; Indústria Civil e Gestão Pública. A empresa foi fundada em 1990 e sua área de atuação na época era apenas a indústria de construção civil. Em 1992 a empresa iniciou na área de gestão pública, e por fim em 1993 começou sua trajetória com um produto para o poder judiciário.

Estas três unidades possuem seus respectivos diretores, e cada uma tem autonomia para gerenciar seus negócios de forma independente dos demais. A pesquisa-ação se dará dentro da unidade de Justiça. Esta unidade de negócio é dividida em quatro grandes áreas, sendo elas: Tribunais; Ministério Público, Procuradorias e Advocacia.

Desde 2010, essa unidade vem passando por uma transição com relação ao seu processo de desenvolvimento de *software*. Até então, a mesma utilizava um processo de desenvolvimento de *software* do tipo *Waterfall* ou modelo cascata. A partir desta época começou-se uma transição para modelos mais ágeis, principalmente o *Scrum*.

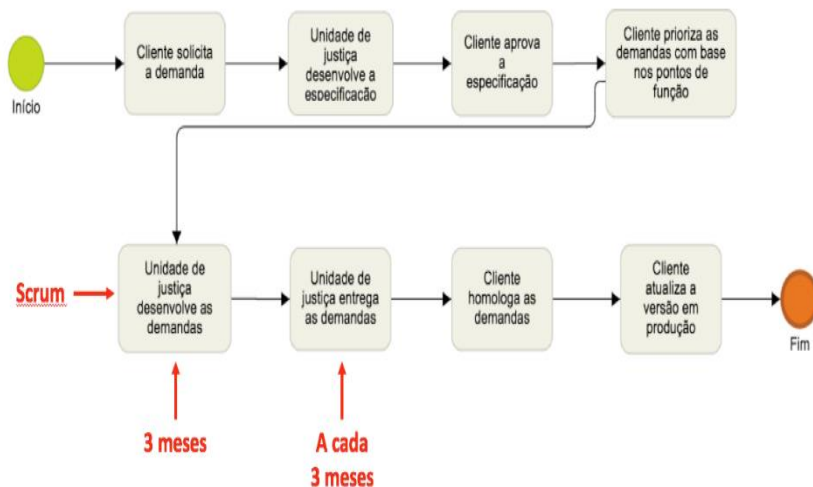
Por volta de 2016 a empresa começou a escalar o *Scrum*, já que nessa época havia em torno de 13 equipes na área de tribunais, todas utilizando este método de gestão de projetos.

Em 2017 foi criado o Grupo de Transformação Ágil (GTA), a fim de tornar as práticas da unidade mais aderentes a abordagem *Lean*, principalmente com um melhor nível de padronização. Desde então muitas práticas, métodos, e ferramentas têm sido utilizadas a fim de auxiliar no processo de desenvolvimento de *software*.

A unidade de Justiça possui atualmente sete grandes clientes, os quais recebem novas atualizações no sistema a cada 3 meses. Para isso, estes clientes possuem contratos por meio de pontos de função, uma métrica muito utilizada no segmento de *software* para balizar valores e funcionalidades. Desta forma, cada cliente possui uma quantidade “X” de Pontos de Função (PF), e com base nessa quantidade contratual, o cliente prioriza as demandas solicitadas a fim de serem desenvolvidas no próximo ciclo (próximos 3 meses).

Para ilustrar melhor de forma macro como se dá todo o processo, desde a criação da demanda, até sua entrada em produção, foi elaborada a Figura 31, conforme segue.

Figura 31 – Macroprocesso de Demandas do Cliente



Fonte: Elaborada pelo autor.

Como visto na Figura 31, a cada 3 meses o cliente prioriza suas demandas para o ciclo de desenvolvimento, e a unidade de Justiça desenvolve as mesmas neste período, sendo que ao final deste tempo, as demandas priorizadas são entregues em uma nova versão do sistema para o cliente.

Este período de 3 meses, por sua vez, é dividido em 6 *Sprints* de 15 dias, conforme orienta o *Scrum*, método utilizado na unidade. Durante os dias finais desse ciclo de 3 meses, e antes do início do próximo ciclo, assim que o cliente prioriza suas próximas demandas, os *Product Owners* (PO's) dos Times de Desenvolvimento atualizam as demandas a fim de seguir o *Scrum* conforme estabelecido na unidade.

Para isso, as demandas são decompostas – as quais estão em nível de especificação – em 3 níveis menores, respectivamente, seguindo as práticas do *Scrum*. Estes níveis são: Épicos (Evoluções), *Features* (Funcionalidades) e *User Stories* (Estórias de Usuário).

A Evolução é um agrupador de funcionalidades as quais, ao final do ciclo, serão agrupadas para gerar então uma nova versão do sistema aos clientes. As Funcionalidades por sua vez são agrupadores de Estórias de Usuário, e é por meio destas que o Time de Desenvolvimento entrega valor ao cliente. Já as Estórias de Usuário são de fato as atividades nas quais os membros do Time trabalham para que sempre que possível ao final de uma *Sprint* haja a entrega de uma ou mais funcionalidades.

É, portanto, neste ambiente de constantes mudanças e limitado por contratos com o poder judiciário que possuem pouca ou nenhuma flexibilidade que o modelo será aplicado.

4.1.2 Equipe

Na etapa de pesquisa-ação, uma equipe de 27 pessoas foi formada, sendo destas, 2 participantes do GEPPS – Grupo de Engenharia de Produto, Processo e Serviço da UFSC, e outros 25 participantes correspondem aos colaboradores da unidade de justiça que integram o time que aplicará o modelo proposto, onde algumas características são descritas conforme mostrado no Quadro 3.

Quadro 3 – Participantes da Pesquisa-ação

Papel	Equipe	Quantidade	Tempo na empresa	Experiência em Lean	Experiência em Toyota Kata	Experiência em MFV
Orientador	UFSC	1	Não se aplica	Mais de 10 anos	Mais de 5 anos	Mais de 10 anos
Mestrando de Engenharia de Produção	UFSC / “Renew”	1	1,5 anos	5 anos	4 anos	5 anos
Business Owner	“Renew”	1	Mais de 15 anos	Nenhuma	Nenhuma	Nenhuma
<i>Product Owners</i>	“Renew”	3	Mais de 2 anos	Nenhuma	Nenhuma	Nenhuma
Líder	“Renew”	1	Mais de 7 anos	Nenhuma	Nenhuma	Nenhuma
Analistas de Sistemas	“Renew”	3	Mais de 3 anos	Nenhuma	Nenhuma	Nenhuma
<i>Scrum Master</i>	UFSC / “Renew”	1	1,5 anos	5 anos	4 anos	5 anos
Desenvolvedores	“Renew”	12	Em média 1 ano	Nenhuma	Nenhuma	Nenhuma
Testadores	“Renew”	4	Em média 1 ano	Nenhuma	Nenhuma	Nenhuma
Arquiteto de <i>Software</i>	“Renew”	1	Mais de 5 anos	Nenhuma	Nenhuma	Nenhuma
	Total de participantes	27		Nenhuma	Nenhuma	Nenhuma

Fonte: Elaborado pelo autor.

Estes colaboradores executaram diretamente o desenvolvimento da pesquisa-ação, sendo que um deles (*Scrum Master*) é a mesma pessoa referenciada como mestrando e pesquisador deste estudo.

4.1.3 Sensibilização

A sensibilização foi realizada com a participação da equipe citada anteriormente, conforme mostra a Figura 32. A equipe será a responsável pela implantação e utilização do modelo com o objetivo de testá-lo e verificá-lo. Nesta etapa, foram apresentados os conhecimentos no que tange as rotinas *Toyota Kata*, já que o *framework Scrum* já era de domínio da equipe, bem como o *Hoshin Kanri* da empresa.

Figura 32 –Sensibilização



Fonte: Acervo do autor.

Após realizado o compartilhamento de conhecimento das rotinas *Toyota Kata* e *Hoshin Kanri*, foi apresentado também o modelo proposto, bem como, as diretrizes de utilização do mesmo, para que no momento de sua aplicação, não aparecessem dúvidas, além de possibilitar que todos do time entendessem o contexto sistêmico de atuação do modelo.

4.2 DIAGNÓSTICO

No diagnóstico a intenção, como próprio nome já explicita, é identificar, explicitar, e planejar as ações necessárias a fim de implantar o modelo para seu posterior teste e verificação do comportamento.

Desta forma o diagnóstico contempla a seguinte atividade: plano de ação. Esta, por sua vez, será devidamente detalhada a seguir.

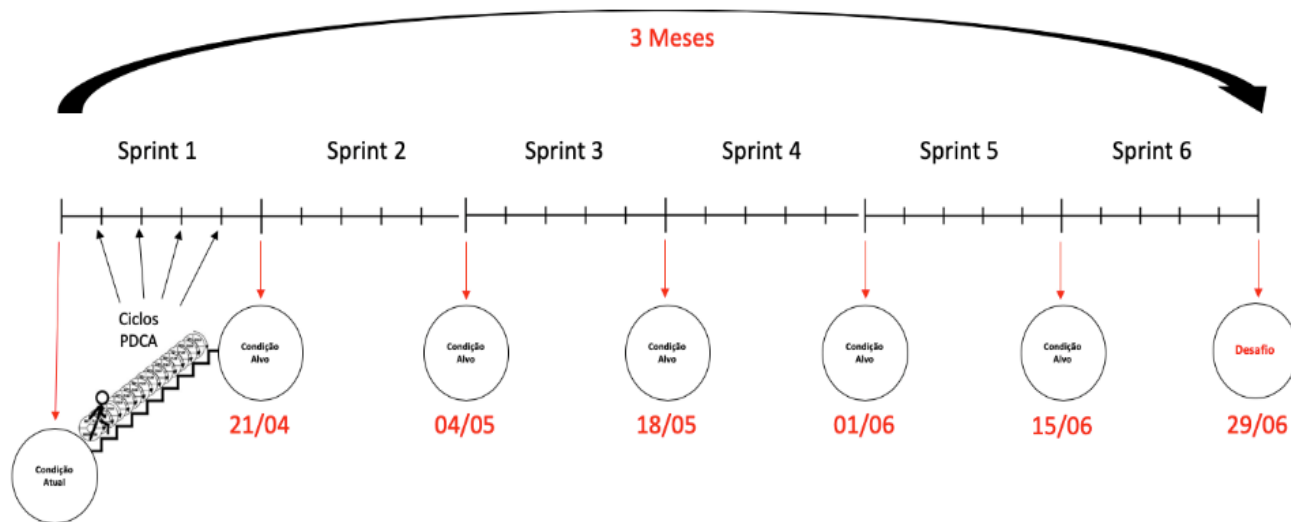
4.2.1 Plano de ação

O plano de ação para a implantação do modelo visa estruturar e auxiliar a pesquisa-ação na aplicação do modelo proposto. Desta forma, o plano de ação foi dividido em duas partes. A primeira parte contempla as ações necessárias para a implantação do modelo proposto, as quais serão identificadas, executadas e, relatados os seus devidos comportamentos utilizando os conceitos *Toyota Kata*.

Estas ações constam nos *Storyboards* intitulados Pesquisa-ação (*Kata* – Pesquisa-ação 1 ao 6), que contemplam seis *Sprints* de quinze dias cada, número esse que visa abranger o ciclo de 3 meses para a entrega ao cliente, conforme já visto anteriormente. Esses *Storyboards* visam registrar a implantação do modelo, bem como, os comportamentos e as experiências relacionadas a ela.

Já na segunda parte, a ideia é similar a parte da implantação do modelo, no entanto a intenção é registrar o andamento das seis *Sprints* das demandas do time, com a integração de fato dos conceitos do *Scrum* e *Toyota Kata*. Cada *Sprint* do time possui seu respectivo *Storyboard* (*Kata* – Time 1 ao 6) nos quais serão registrados o andamento dos ciclos PDCA em direção as suas respectivas Condições-Alvo, as quais são quinzenais, e o Desafio, que por sua vez é trimestral, estando alinhado assim ao ciclo de entrega para o cliente final.

A fim de mostrar essa integração das rotinas *Toyota Kata* e *Scrum* ao longo do tempo, junto com a entrega final ao cliente, foi desenvolvida a Figura 33.

Figura 33 – Integração *Scrum* e *Toyota Kata* no Tempo

Fonte: Elaborada pelo autor.

Assim sendo, optou-se por se registrar o andamento da implantação e de cada *Sprint* do time por meio de uma planilha digital, separada por abas que representam cada uma das seis *Sprints*, as quais possuíam suas respectivas Condições Atual e Alvo, os respectivos desafios e os processos nos quais atuariam, conforme mostra a Figura 34.

Figura 34 – *Storyboard* da Pesquisa-ação

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O Hoshin Kanri não foi compartilhado com a equipe que implantará o modelo	1 - Apresentar o Hoshin Kanri para o time que irá implantar o modelo	Desconhecimento do time sobre Hoshin Kanri	Compartilhar o Hoshin Kanri com o time	Que a equipe compreendesse os objetivos estratégicos da empresa, e como a mesma pretende alcançá-los	A equipe teve conhecimento do Hoshin Kanri da empresa, e entendeu onde eles podem auxiliar nos mesmos	Qual a visão da empresa e qual a direção que devemos caminhar a fim atingi-la
2 - Não foi realizado a sensibilização da equipe quanto as rotinas Toyota Kata, bem como, no modelo de implantação	2 - Realizar a sensibilização das rotinas Toyota Kata e do modelo para o time que irá implantá-lo	Desconhecimento do time sobre Toyota Kata e o modelo proposto	Ministrar a sensibilização	Que o time conhecesse as rotinas Toyota Kata e o modelo de implantação	O time teve conhecimento das rotinas Toyota Kata, bem como, do modelo proposto que será implantado	Que o tempo de 4 horas de workshop influencia no aprendizado, pois não se mostrou adequado a todo conhecimento envolvido, e que mesmo sendo um workshop prático, o time teve diversas dúvidas que só puderam ser esclarecidas posteriormente com a prática
3 - Não temos o mapa de fluxo de valor (MFV) de um problema crítico e recorrente no time	3 - Mapear o fluxo de valor do processo de finalização de documentos	Falta de conhecimento do time sobre o MFV	Mapear o fluxo de valor de finalização de documentos	Compreender a cadeia de valor do processo de finalização de documentos, quais os desperdícios do processo e quais os pontos críticos	O time conheceu o processo de finalização de documentos, e onde estão suas maiores lacunas, as quais, podem ser reduzidas ou	Que a falta de apoio do time no mapeamento pode gerar distorções do processo, e até não representá-lo como é de fato. Além disso, foi aprendido o que é feito no

Fonte: Dados primários do autor.

O formato e os campos de ambos *Storyboards* são iguais, mudando apenas o conteúdo relatado, já que a finalidade dos dois tipos é distinta. Este arquivo é mostrado na televisão utilizada pelo time para relatos e acompanhamentos diários feitos na reunião *Kata* Diária, conforme consta no modelo e visto na Figura 35.

Figura 35 – *Storyboard* do Time

Condição Atual		Condição-Alvo	Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
			PDCA				
			Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
As movimentações automáticas no fluxo não são paralelizadas		Verificar se existe uma maneira de paralelizar as movimentações automáticas do fluxo, sem causar uma mudança no comportamento do sistema atual	Desconhecimento sistêmico das movimentações	Analisar o código das movimentações internas	Entender como funcionam as movimentações	Foi compreendido como e quando a primeira movimentação interna é lançada	Que a primeira movimentação interna na verdade é a movimentação de "documento criado", quando se está criando um documento, pois antes de finalizar, se é um novo documento, o mesmo é salvo antes de iniciar a finalização
			Desconhecimento do SP4 e do SIT	Iniciar uma implementação simples para encontrar os possíveis primeiros problemas	Descobrir como fazer uma conexão. Se será criando uma conexão secundária no servidor com uma thread que irá ficar verificando uma fila de movimentações,	Foi feito o primeiro teste com as movimentações em paralelo utilizando thread ao invés do SIT	Como conectar no servidor utilizando conexão secundária (classe TspConexao). Delphi extremamente lento, teve-se que reiniciar o mesmo diversas vezes.

Fonte: Dados primários do autor.

Por fim, somado a utilização das rotinas de *Kata* de Melhoria integradas no *Scrum*, também foram aplicados os ciclos de planejamento e execução de *Kata* de *Coaching*. Ambos os ciclos foram utilizados conforme proposto pelo modelo, concentrando-se principalmente nas reuniões de *Kata* Diárias com a equipe, para acompanhar o andamento das *Sprints*, bem como, na Retrospectiva da *Sprint*, a fim de consolidar os conhecimentos adquiridos com o time, e quando necessário, se adaptar para a próxima *Sprint*, conforme será visto no decorrer do tópico a seguir.

4.3 IMPLANTAÇÃO

Na fase de implantação tem-se a verificação de fato do comportamento do modelo, a fim de identificar seus problemas, suas oportunidades de melhoria e seus resultados finais. Para tal, esta fase foi composta de três atividades, sendo elas: implantação do plano; coleta de dados e avaliação e resultados. As mesmas serão detalhadas a seguir.

4.3.1 Implantação do plano

Na implantação do plano, como a própria atividade sugere, se dará então a execução do que foi planejado anteriormente. Neste momento serão executadas as seis *Sprints* conforme sugerido, para que desta forma seja verificado o comportamento do modelo, e assim identificar suas respectivas contribuições.

Além das informações sobre o comportamento do modelo, este tópico descreve sobre a rotina diária do time, suas dificuldades, obstáculos e suas evoluções em direção ao objetivo a ser alcançado.

4.3.1.1 *Sprint* 1

Desta forma, na primeira *Sprint* tem-se o início da implantação do modelo com base no que foi planejado para a mesma, conforme mostra a Figura 36. A primeira *Sprint* é comum aos dois tipos de *Storyboards*, sendo assim, optou-se para a mesma relatar a evolução e o aprendizado apenas em um *Storyboard* (*Kata1* – Pesquisa-ação e Time), não sendo necessário neste momento específico a divisão dos *Storyboards*.

Figura 36 – *Storyboard da Sprint 1*

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O Hoshin Kanri não foi compartilhado com a equipe que implantará o modelo	1 - Apresentar o Hoshin Kanri para o time que irá implantar o modelo	Desconhecimento do time sobre Hoshin Kanri	Compartilhar o Hoshin Kanri com o time	Que a equipe compreendesse os objetivos estratégicos da empresa, e como a mesma pretende alcançá-los	A equipe teve conhecimento do Hoshin Kanri da empresa, e entendeu onde eles podem auxiliar nos mesmos	Qual a visão da empresa e qual a direção que devemos caminhar a fim atingi-la
2 - Não foi realizado a sensibilização da equipe quanto as rotinas Toyota Kata, bem como, no modelo de implantação	2 - Realizar a sensibilização das rotinas Toyota Kata e do modelo para o time que irá implantá-lo	Desconhecimento do time sobre Toyota Kata e o modelo proposto	Ministrar a sensibilização	Que o time conhecesse as rotinas Toyota Kata e o modelo de implantação	O time teve conhecimento das rotinas Toyota Kata, bem como, do modelo proposto que será implantado	workshop influencia no aprendizado, pois não se mostrou adequado a todo conhecimento envolvido, e que mesmo sendo um workshop prático, o time teve diversas dúvidas que só puderam ser esclarecidas posteriormente com a prática
3 - Não temos o mapa de fluxo de valor (MFV) de um problema crítico e recorrente no time	3 - Mapear o fluxo de valor do processo de finalização de documentos	Falta de conhecimento do time sobre o MFV	Mapear o fluxo de valor de finalização de documentos	Compreender a cadeia de valor do processo de finalização de documentos, quais os desperdícios do processo e quais os pontos críticos	O time conheceu o processo de finalização de documentos, e onde estão suas maiores lacunas, as quais, podem ser reduzidas ou	Que a falta de apoio do time no mapeamento pode gerar distorções do processo, e até não representá-lo como é de fato. Além disso, foi aprendendo o que é feito no

Fonte: Dados primários do autor.

Com base na Figura 36, optou-se nesse momento por transformá-la no Quadro 4, a fim de mostrar de uma forma clara tudo o que ocorreu na *Sprint 1*, e como tudo foi registrado.

Quadro 4 – *Sprint 1* da Pesquisa-ação e do Time

(Continua)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
<p>1 - O <i>Hoshin Kanri</i> não foi compartilhado com a equipe que implantará o modelo</p> <p>2 - Não foi realizado a sensibilização da equipe quanto as rotinas <i>Toyota Kata</i>, bem como, no modelo de implantação</p> <p>3 - Não temos o mapa de fluxo de valor (MFV) de um problema crítico e recorrente no time</p>	<p>1 - Apresentar o <i>Hoshin Kanri</i> para o time que irá implantar o modelo</p> <p>2 - Realizar a sensibilização das rotinas <i>Toyota Kata</i> e do modelo para o time que irá implantá-lo</p> <p>3 - Mapear o fluxo de valor do processo de finalização de documentos</p>	Desconhecimento do time sobre <i>Hoshin Kanri</i>	Compartilhar o <i>Hoshin Kanri</i> com o time	Que a equipe compreendesse os objetivos estratégicos da empresa, e como a mesma pretende alcançá-los	A equipe teve conhecimento do <i>Hoshin Kanri</i> da empresa, e entendeu de que forma eles podem auxiliar nos mesmos	A equipe entendeu qual é a visão da empresa e qual a direção que se deve caminhar a fim atingi-la

Quadro 4 – *Sprint 1* da Pesquisa-ação e do Time

(Continuação)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Desconhecimento do time sobre <i>Toyota Kata</i> e o modelo proposto	Ministrar a sensibilização	Que o time conhecesse as rotinas <i>Toyota Kata</i> e o modelo de implantação	O time teve conhecimento das rotinas <i>Toyota Kata</i> , bem como, do modelo proposto que será implantado	Que o tempo de 4 horas de workshop influencia no aprendizado, pois não se mostrou adequado a todo conhecimento envolvido, e que mesmo sendo um workshop prático, o time teve diversas dúvidas que só puderam ser esclarecidas posteriormente com a prática

Quadro 4 – *Sprint 1* da Pesquisa-ação e do Time

(Continuação)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Falta de conhecimento do time sobre o MFV	Mapear o fluxo de valor de finalização de documentos	Compreender a cadeia de valor do processo de finalização de documentos, quais os desperdícios do processo e quais os pontos críticos	O time conheceu o processo de finalização de documentos, e onde estão suas maiores lacunas, as quais, podem ser reduzidas ou eliminadas com a utilização do modelo	Que a falta de apoio do time no mapeamento pode gerar distorções do processo, e até não o representar como é de fato. Além disso, foi aprendido o que é feito no processo de finalização de documentos, bem como, quais são seus desperdícios

Quadro 4 – *Sprint 1* da Pesquisa-ação e do Time

(Conclusão)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Desconhecimento sobre a causa raiz do problema relacionado ao desperdício do MFV	Elaborar o A3 de finalização de documentos	Descobrir a causa raiz do problema de lentidão na finalização de documentos	O time encontrou a causa raiz do problema de lentidão, e definiu as ações necessárias para experimentar a fim de resolver o problema	Quais as operações e modelos de documento mais afetam o processo de finalização de documentos, e qual a causa raiz que mais prejudica o processo e pode ser atacada

Fonte: Elaborado pelo autor.

Assim sendo, a primeira e segunda etapa do modelo foram realizadas no momento da sensibilização, para aproveitar que o time estava reunido. Desta forma, na mesma reunião aconteceram a sensibilização do time (já visto no tópico 1.1.3), bem como, a apresentação do *Hoshin Kanri*, conforme mostra a Figura 37, a fim de ficar claro quais são os *Hoshins* da empresa para o time.

Figura 37 – *Hoshin Kanri*

HOSHINS

Hoshin1: Novas Ofertas

Indicador: Market Share + Cliente Share

Meta 2018: R\$ 20 mi

Hoshin2: Internacionalização

Indicador: Número de países prospectados

Meta 2018: 3 países

Hoshin 3: Satisfação dos Clientes

Indicador: NPS

Meta 2018: 40 pontos

Hoshin 4: Eficiência Operacional

Indicador: Modelo de Excelência em Gestão (MEG)

Meta 2018: 420 pontos

Indicador: Coeficiente de mercado

Meta: Coeficiente de eficiência

Hoshin 5: Pessoas Incríveis

Indicador: Felicitômetro

Meta 2018: Acima de 3

Indicador: GPTW

Meta 2018: Superior a 2017

Indicador: Avaliação de Desempenho

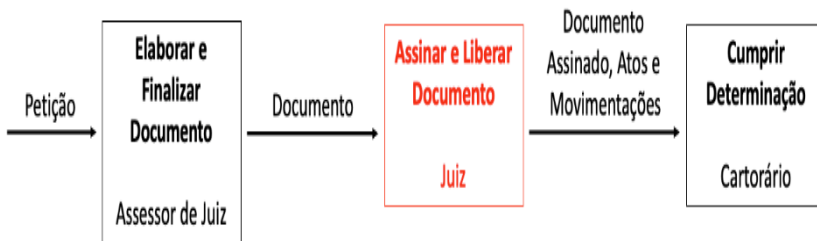
Meta: Acima de 3

Fonte: Elaborada pelo autor.

Conhecendo os *Hoshins* da empresa, bem como, o time sensibilizado, o próximo passo da primeira *Sprint*, foi a realização do Mapeamento do Fluxo de Valor (MFV) a ser trabalhado pela equipe posteriormente nas demais *Sprints*, com base nos desperdícios que forem encontrados pelo MFV.

Antes, porém, do mapeamento do fluxo de valor, se faz necessário explicitar o macro processo de finalização de documentos, a fim de identificar e situar em que lugar está o gargalo do mesmo, bem como, qual a entrada e saída do processo problemático, para que então possa ser elaborado o mapa do fluxo de valor do mesmo, conforme mostra a Figura 38.

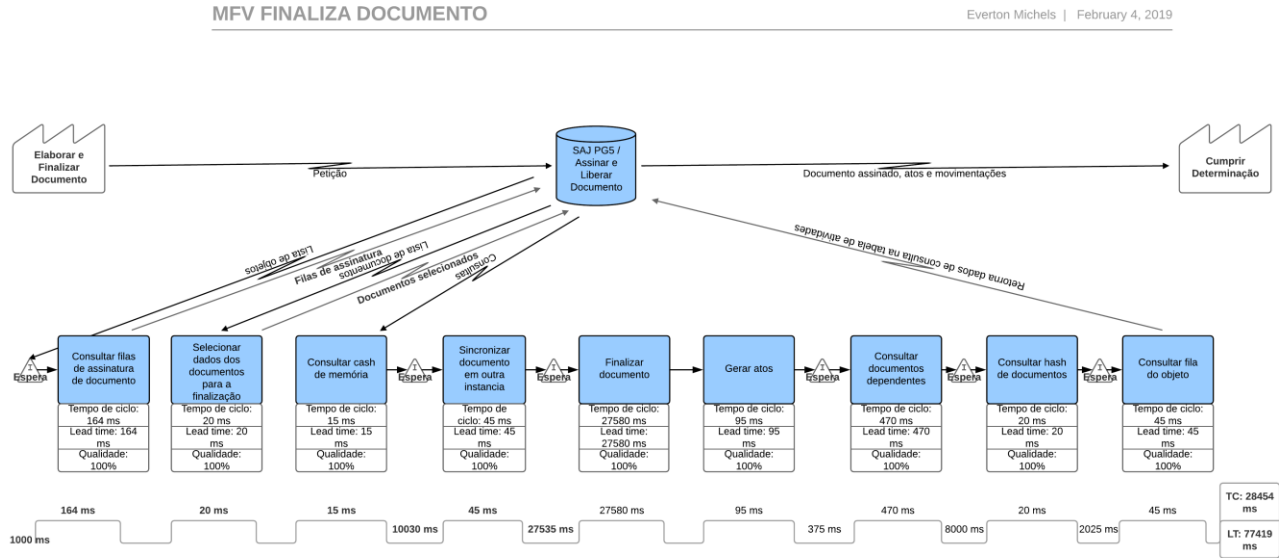
Figura 38 – Macroprocesso de Finalização de Documentos



Fonte: Elaborada pelo autor.

Desta forma, a equipe responsável pela implantação do modelo se reuniu para mapear o fluxo de valor do processo problemático (Assinar e Liberar Documento), gerando então o mapa ao qual será atuado posteriormente, conforme mostra a Figura 39.

Figura 39 – Mapa do Fluxo de Valor de Finalização de Documentos



Fonte: Elaborada pelo autor.

Finalizando o MFV, fica explícito para o time quais são os principais desperdícios que serão abordados nas próximas 5 *Sprints*, além de se ter identificado o processo ao qual será atuado (Assinar e Liberar Documento), bem como o desafio esperado (Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários).

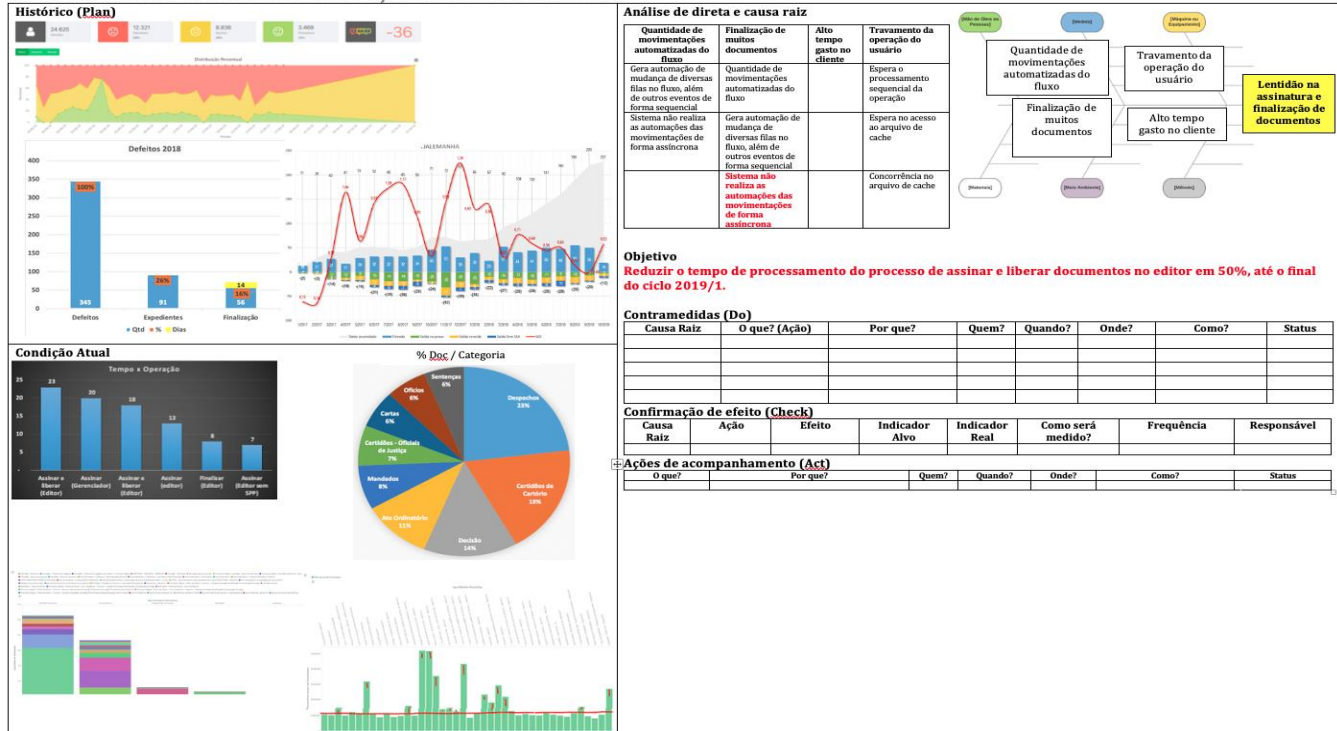
Esse desafio por sua vez, está alinhado a três dos *Hoshins Kanri* da empresa, sendo eles: *Hoshin* 3, 4 e 5, pois a melhoria no processo tem ligação direta com a satisfação dos clientes, além de ocasionar uma melhoria operacional, e por fim, aumentar o status quo das pessoas envolvidas na solução do problema, o que resulta na contribuição com o *Hoshin* de pessoas incríveis. Esse alinhamento entre visão e direção é justamente um dos pontos passíveis de melhoria que o modelo se propõe.

No entanto, neste momento foi descoberta uma nova possibilidade que pode ser acrescentada ao modelo de uma forma just in time, ou seja, quando oportuno, da forma oportuna e no local oportuno. Como o MFV não visa explicitar os problemas relacionados aos desperdícios, foi necessário utilizar uma outra ferramenta *Lean* para tornar isso claro e detalhado.

Desta forma, a equipe que elaborou o MFV do Estado Atual, utilizou o A3 para solução de problemas, como uma ferramenta para esclarecer melhor o problema relacionado ao desperdício de tempo encontrado no processo de assinatura e liberação de documentos, o qual é parte do processo de finalização de documentos.

Esta ferramenta, conforme mostra a Figura 40, foi capaz de deixar claro como foi realizado o processo para encontrar as causas raiz do problema, bem como, quais ações seriam trabalhadas pelo modelo, a fim de solucionar estes problemas.

Figura 40 – A3 para o processo Finalização de Documentos
 A3 – Tema: Lentidão na assinatura e finalização de documentos no editor



Fonte: Adaptado de Sobek and Smalley (2010).

Por fim, como o modelo aqui proposto trabalha com a integração das rotinas *Toyota Kata* junto ao *Scrum*, a última parte da Condição-Alvo dessa *Sprint*, que era a criação do *Backlog* do Produto foi realizada conforme esperado. Esse *Backlog*, embora com itens já em andamento e encerrados, pode ser visto na Figura 41, a qual é um *print* de tela da ferramenta Rational Team Concert (RTC), ferramenta esta utilizada pela empresa para criar *Backlog* de Produto e sincronizar o código do *software*.

Figura 41 – Backlog do Produto

Todos os Painéis Tribunais > Tribunais >

Agile Team - Alemanha [?]

Salvamento Automático Salvar

[Geral](#)
[Indicadores](#)
[PAINEL EPD/ERS](#)
[Backlog Evoluções](#)
[Defeitos Internos](#)
[Causa Raiz - FCA](#)
[Sustentação Geral](#)
[Tokyo Alemanha](#)
[Evoluções](#)
[Evoluções DB1](#)
[Pasta](#)
[Expedientes](#)
[Fluxo](#)

Kata - A3
[Sinergia/TJSP+](#)
[Intimação eletrônica/SPP](#)
[Pasta Digital - Sustentação](#)
[Fluxo de Trabalho](#)
[Itens Performance](#)
[Próxima Entrega TJSP/TJSC](#)
[Painel do DEV](#)
[Incluir Widget](#)

Evolução (1)

160367: Realizar as automações das movimentações na finalização de documentos de forma assíncrona

Alemanha - Stories A3 Em desenvolvimento (1)

163350: Enviar os campos do log como tags ao invés de um JSON, para que seja pesquisável na ferramenta Elastic

Funcionalidades (1)

163441: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários

Alemanha - Stories A3 Em teste (1)

161544: Testar as movimentações paralelas no sistema

Backlog - Estórias de Usuário (2)

162716: Erro de propagação de erro no lançamento de movimentação interna.

163217: Problema ao emitir expedientes quando usuário não tem a fila destino

Alemanha - Stories A3 Encerradas (5)

159999: Testar uma solução que assine documentos digitais no cliente baseados em hash, tanto no delphi tokyo quanto no delphi 5

160000: Validar se existe uma maneira de paralelizar as movimentações automáticas do fluxo

161545: Alterar a solução que foi criada na POC para utilizar o barramento para envio das filas de movimentações

162325: Implementar o envio de mensagens das threads pelo barramento

162326: Alterar o inicio e fim de movimentação paralela para contemplar o FinalizaDocumentoDigital

Fonte: Dados primários do autor.

Desta forma, a causa raiz relatada no A3 virou a respectiva evolução no *Scrum*, a qual consta na Figura 41. Descendo um nível na utilização do *Scrum*, conforme ocorre na empresa, e já informado anteriormente, foi criada a devida funcionalidade (*Feature*), a qual saiu respectivamente do estabelecimento do Desafio do time, que também consta na Figura 41, e assim, conseqüentemente resultaram nas respectivas Estórias de Usuário (*Users Stories*), criadas sob demanda e originadas a partir da Condição-Alvo de cada *Sprint*, também explicitadas na Figura 41. Esse processo de criação das Estórias de Usuário se repetirá durante as cinco próximas *Sprints*, em cada reunião de Planejamento da *Sprint*, como será visto.

4.3.1.2 *Sprint 2*

A *Sprint 2* é aquela em que começa a “mão na massa” de fato. Neste momento a fase de preparação do modelo já foi realizada por completo, e foi feito o primeiro contato com a fase de planejamento, ao executar o Planejamento da *Sprint*, conforme o modelo proposto, já que esta cerimônia ocorrerá de forma constante durante as cinco *Sprints* restantes.

Deste ponto em diante, conforme já informado, as *Sprints* relatarão o andamento dos dois tipos de *Storyboards*, o da pesquisa-ação em si, bem como, o do trabalho realizado na pesquisa pelo time que implantou o modelo no seu dia a dia.

Para a elaboração dos *Storyboards*, o time se reúne na cerimônia de Planejamento da *Sprint*, na qual os mesmos validaram, conforme mostra a Figura 42, o desafio da pesquisa-ação para a *Sprint*.

Desafio este que já veio estabelecido da *Sprint 1* e que será o mesmo para todas as *Sprints* seguintes. Por fim, foram criados os itens do *Backlog* do Produto (Funcionalidade e Estórias de Usuário), conforme a necessidade do *Storyboard* do time.

Figura 42 – *Storyboard* de Pesquisa-ação da *Sprint 2*

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O time não conhece o processo, o desafio e as condições atual e alvo que trabalharão na Sprint	1 - Definir o desafio do time na sprint, identificar o processo que será melhorado, entender a condição atual e estabelecer a condição-alvo da sprint	Desconhecimento dos aspectos do storyboard relacionados a Sprint	Preencher as informações do Storyboard	Que o Storyboard estivesse com seus dados iniciais completos ao final da reunião de planejamento da Sprint, e o time compreendesse qual a funcionalidade do mesmo	Os dois Storyboards foram preenchidos com seus respectivos campos necessários, e o time entendeu para que serve o Storyboard	Qual é o processo, desafio, e condições atual e alvo que serão trabalhadas na Sprint, tanto na pesquisa-ação quanto pelo time
2 - Falta dos itens relativos ao Scrum como, funcionalidade e estória de usuário que contemplarão o desafio	2 - Criar a funcionalidade e as estórias de usuário que contemplem o desafio e a condição-alvo	Falta da funcionalidade e das estórias usuário para trabalhar na Sprint	Criar a funcionalidade e as estórias de usuário da Sprint	Que o time compreendesse qual o valor que será entregue pro cliente ao alcançar o desafio estabelecido, já que o mesmo se traduz na funcionalidade	O Backlog do Produto inicial foi criado, e o time teve a visão macro da entrega de valor para o cliente	Usar do conceito de sistema puxado faz toda a diferença neste ponto, a fim de evitar o desperdício. Desta forma só é criado o que é necessário, e no momento adequado. Vincular o desafio a uma feature, e as condições-alvo as estórias de usuário ficou bem aderente a integração das rotinas de Kata com o Scrum
3 - Não foi realizada a fase de reflexão a fim de verificar se a Condição-Alvo foi atendida, registrar os aprendizados que foram adquiridos, e se é necessária uma adaptação	3 - Realizar a Revisão e a Retrospectiva da Sprint a fim de verificar se a Condição-Alvo foi atendida, bem como, quais aprendizados foram adquiridos se será necessária uma adaptação para a próxima	Falta da revisão, do compartilhamento de conhecimento e adaptabilidade	Realizar a revisão e retrospectiva da Sprint	Que o time refletisse sobre o ocorrido na Sprint, a fim de verificar e comparar o que foi proposto, e o que foi alcançado, e caso necessário, se adaptar a fim de estabelecer uma nova Condição-Alvo aderente ao	Foi realizada a revisão e retrospectiva da Sprint, onde o time refletiu sobre o ocorrido, sugerindo melhorias e compartilhando o que foi aprendido na Sprint	Realizar os ciclos curtos de Kata de Coaching faz com que o aprendiz compartilhe seus conhecimentos e aprendizados diariamente com o time, obtendo apoio sempre que um obstáculo novo é encontrado no caminho. Além disso, a condição-alvo é verificada a fim de identificar se é

Fonte: Dados primários do autor.

Como poderá ser visto nas demais *Sprints*, esse processo de construção, tanto dos *Storyboards*, quanto do *Backlog* do Produto, se repetirá constantemente, mudando claro seu conteúdo, pois conforme explicitado pelo próprio Rother (2010), as rotinas *Toyota Kata* são um padrão ou uma maneira de realizar experimentos em busca de um objetivo, desta maneira, o método em si não muda, mas sim o conteúdo dele. A fim de deixar mais claro o *Storyboard* da Pesquisa-ação, optou-se por explicitar o mesmo por meio do Quadro 5, conforme segue.

Quadro 5 – *Storyboard* da Pesquisa-ação da *Sprint 2*

(Continua)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O time não conhece o processo, o desafio e as condições atual e alvo que trabalharão na <i>Sprint</i>	1 - Definir o desafio do time na <i>Sprint</i> , identificar o processo que será melhorado, entender a condição atual e estabelecer a condição-alvo da <i>Sprint</i>	Desconhecimento dos aspectos do Story-board relacionados a <i>Sprint</i>	Preencher as informações do <i>Storyboard</i>	Que o <i>Storyboard</i> estivesse com seus dados iniciais completos ao final da reunião de planejamento da <i>Sprint</i> , e o time compreendesse qual a funcionalidade do mesmo	Os dois <i>Storyboards</i> foram preenchidos com seus respectivos campos necessários, e o time entendeu para que serve o <i>Storyboard</i>	O time compreendeu qual é o processo, desafio, e as condições atual e alvo que serão trabalhadas na <i>Sprint</i> , tanto no <i>Storyboard</i> de pesquisa-ação quanto no do time

Quadro 5 – *Storyboard* da Pesquisa-ação da *Sprint 2*

(Continuação)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
2 - Falta dos itens relativos ao <i>Scrum</i> como, funcionalidade e estória de usuário que contemplarão o desafio	2 - Criar a funcionalidade e as estórias de usuário que contemplem o desafio e a condição-alvo	Falta da funcionalidade e das estórias de usuário para trabalhar na <i>Sprint</i>	Criar a funcionalidade e as estórias de usuário da <i>Sprint</i>	Que o time compreendesse qual o valor que será entregue ao cliente ao alcançar o desafio estabelecido, já que o mesmo se traduz na funcionalidade	O <i>Backlog</i> do Produto inicial foi criado, e o time obteve a visão macro da entrega de valor para o cliente	O uso do conceito de sistema puxado faz toda a diferença neste ponto, a fim de evitar o desperdício. Desta forma só é criado o que é necessário, e no momento adequado. Incluir o desafio a uma funcionalidade, e as condições-alvo as estórias de usuário ficou bem aderente a integração das rotinas de <i>Kata</i> com o <i>Scrum</i>

Quadro 5 – *Storyboard* da Pesquisa-ação da *Sprint 2*

(Conclusão)

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
3 - Não foi realizada a fase de reflexão a fim de verificar se a Condição-Alvo foi atendida, registrar os aprendizados que foram adquiridos, e se é necessária uma adaptação	3 - Realizar a Revisão e a Retrospectiva da <i>Sprint</i> a fim de verificar se a Condição-Alvo foi atendida, bem como, quais aprendizados foram adquiridos se será necessária uma adaptação para a próxima Condição-Alvo	Falta da revisão, do compartilhamento de conhecimento e adaptabilidade	Realizar a revisão e retrospectiva da <i>Sprint</i>	Que o time refletisse sobre o ocorrido na <i>Sprint</i> , a fim de verificar e comparar o que foi proposto, e o que foi alcançado, se necessário, se adaptar a fim de estabelecer nova Condição-Alvo aderente ao Desafio proposto. aprendizados adquiridos durante a <i>Sprint</i> fossem compartilhados com todos.	Foi realizada a revisão e retrospectiva da <i>Sprint</i> , onde o time refletiu sobre o ocorrido, sugerindo melhorias e compartilhando o que foi aprendido na <i>Sprint</i> .	Realizar os ciclos curtos de <i>Kata</i> de <i>Coaching</i> faz com que o aprendiz compartilhe seus conhecimentos diariamente com o time, obtendo apoio sempre que um obstáculo novo é encontrado. Além disso, a condição-alvo é verificada a fim de identificar se é necessária uma adaptação para a próxima <i>Sprint</i> , bem como, se o que foi estipulado foi atendido

Fonte: Elaborado pelo autor.

Relatado o *Storyboard* da Pesquisa-ação, o passo seguinte foi realizar o mesmo para o *Storyboard* do time. Desta forma, tanto para a *Sprint 2*, quanto para as demais *Sprints*, ficou estabelecido que o processo a ser escolhido foi o de Assinatura e Liberação de Documentos, pois neste, foram identificados desperdícios no MFV, bem como, as devidas causas raiz do problema, no detalhamento do A3. Da mesma forma, foi estabelecido o desafio pelo time, o qual se buscará realizar durante as *Sprints*, sendo este: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários.

Além do processo e do desafio estabelecido para a *Sprint* atual, os quais serão os mesmos para as demais *Sprints*, foi também identificada a Condição Atual, bem como, estabelecida a Condição-Alvo. Desta forma, o time se empenhou em executar os ciclos PDCA a fim de alcançarem o objetivo proposto. Com isso, foram feitas as reuniões de *Kata* Diárias, a fim de executar os ciclos curtos de Coaching para acompanhar os experimentos do time em direção a Condição-Alvo, isso tudo relatado na Figura 43.

Figura 43 – Storyboard do Time da Sprint 2

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários					
Condição Atual		Condição-Alvo	PDCA				
Condição Atual		Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
As movimentações automáticas no fluxo não são paralelizadas		Verificar se existe uma maneira de paralelizar as movimentações automáticas do fluxo, sem causar uma mudança no comportamento do sistema atual	Desconhecimento sistêmico das movimentações	Analisar o código das movimentações internas	Entender como funcionam as movimentações	Foi compreendido como e quando a primeira movimentação interna é lançada	Que a primeira movimentação interna na verdade é a movimentação de "documento criado", quando se está criando um documento, pois antes de finalizar, se é um novo documento, o mesmo é salvo antes de iniciar a finalização
			Desconhecimento do SP4 e do SIT	Iniciar uma implementação simples para encontrar os possíveis primeiros problemas	Descobrir como fazer uma conexão. Se será criando uma conexão secundária no servidor com uma thread que irá ficar verificando uma fila de movimentações, ou pelo SIT	Foi feito o primeiro teste com as movimentações em paralelo utilizando thread ao invés do SIT	Como conectar no servidor utilizando conexão secundária (classe TspConexao). Delphi extremamente lento, teve-se que reiniciar o mesmo diversas vezes.
			Desconhecimento	Alterar para	Isolar a alteração para	A operação foi isolada	Como isolar a operação apenas

Fonte: Dados primários do autor.

Para deixar claro todos os ciclos PDCA que foram realizados na *Sprint 2* em busca da Condição-Alvo, optou-se por explicitar o *Storyboard* do time por meio do Quadro 6.

Quadro 6 – *Storyboard* do Time da *Sprint 2*

(Continua)

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
Ciclos PDCA						
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
As movimentações automáticas no fluxo não são paralelizadas	Verificar se existe uma maneira de paralelizar as movimentações automáticas do fluxo, sem causar uma mudança no comportamento do sistema atual	Desconhecimento sistêmico das movimentações	Analisar o código das movimentações internas	Entender como funcionam as movimentações	Foi compreendido como e quando a primeira movimentação interna é lançada	Que a primeira movimentação interna na verdade é a movimentação de "documento criado", quando se está criando um documento, pois antes de finalizar, se é um novo documento, o mesmo é salvo antes de iniciar a finalização

Quadro 6 – *Storyboard* do Time da *Sprint 2*

(Continuação)

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Desconhecimento do SP4 e do SIT	Iniciar uma implementação simples para encontrar os possíveis primeiros problemas	Criar uma conexão secundária no servidor com uma <i>thread</i> ou pelo SIT, que ficará verificando uma fila de movimentações	Foi feito o primeiro teste com as movimentações em paralelo utilizando thread ao invés do SIT	Como conectar no servidor utilizando conexão secundária (classe Tsp Conexão). Delphi extremamente lento, teve-se que reiniciar o mesmo diversas vezes.

Quadro 6 – *Storyboard* do Time da *Sprint 2*

(Continuação)

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Desconhecimento sobre o parâmetro, variável global ou análise da operação que está sendo finalizada	Alterar para paralelizar apenas as movimentações da operação de finalização	Isolar a alteração para apenas uma operação, que no caso seria a finalização, e deixar em aberto para caso necessário, estender para outras operações também	A operação foi isolada colocando uma flag global do servidor que será acionada pelo cliente. Quando ativada, toda vez que passar no método “Lança Movimentação Fluxo”, não irá executá-lo. Ao invés disso, irá adicionar em uma fila de movimentações para ser executada quando a operação for finalizada	A maneira de como isolar a operação apenas para a finalização, e deixá-la de tal forma que pudesse ser utilizada em outros pontos futuramente

Quadro 6 – *Storyboard* do Time da *Sprint 2*

(Continuação)

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Movimentações não são agrupadas em uma operação	Agrupar movimentações de uma mesma operação	Conseguir agrupar todas as movimentações de uma operação em uma mesma execução	Foi alterado a forma de agrupar as filas para todas as movimentações de uma operação ficarem agrupadas, usando para isso o ID da operação de usuário	Como agrupar as movimentações em uma mesma operação utilizando para isso o ID da operação do usuário
		Os logs não estão em arquivo	Adicionar log das movimentações, ou mostrar de alguma forma o que está sendo executado	Registrar por meio de logs todas as operações da execução em paralelo	Todos os pontos necessários para registrar problemas na execução da movimentação paralela foram contemplados	Quais os principais pontos que eram necessários os logs para registro de problemas futuros caso ocorressem, e quais as informações que os mesmos devem registrar

Quadro 6 – *Storyboard* do Time da *Sprint 2*

(Continuação)

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Desconhecimento sobre iniciar/finalizar uma transação a partir de uma instância do TspConexao	Agrupar transações de uma operação	Que as movimentações de uma operação em uma mesma transação sejam agrupadas de tal forma que não se percam no processamento	Foi criado um método no servidor (Lança Movimentação Lote) para executar todas as movimentações da operação. Isso porque não é possível controlar uma transação através do cliente (a thread não deixa de ser um cliente), esse trabalho deve ser feito todo no servidor	Que o ciclo de vida de uma transação é feito apenas no servidor. Que não é possível controlar a transação através da <i>thread</i> que empilha as movimentações, pois toda vez que uma chamada ao servidor retorna ao cliente, e uma transação está aberta, tudo é desfeito automaticamente. Também não é possível chamar o método de início/fim de transação a partir do cliente

Quadro 6 – *Storyboard* do Time da *Sprint 2*

(Conclusão)

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
		Ciclos PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
		Os objetos do cliente para o servidor não são enviados por meio de lista, nem serializados	Agrupar transações de uma operação	Serializar e enviar uma lista de objetos, sem precisar criar datasets ou mandar todos os parâmetros separados	Foi feito a serialização/deserialização do objeto de lista de movimentações por meio da classe REST.JSON.TJson	Como serializar/deserializar um objeto simples em json. O objeto TQueue não pode ser serializado simplesmente utilizando a classe TJson, pois possui fields privados que são necessários na deserialização. A classe apenas serializa fields públicos. Para isso, deve-se utilizar a classe TJsonMarshal e registrar "conversores" próprios. Para não ter esse trabalho no momento, foi transformado a classe TQueue em TList para serializar/deserializar

Fonte: Elaborado pelo autor.

Desta maneira, ao final dos quinze dias da *Sprint 2*, na reunião de Revisão da *Sprint*, o time se reuniu para validar com o *Product Owner* (PO) se a Condição-Alvo estabelecida havia sido alcançada com sucesso, e se o incremento entregue estava de fato alinhado com a busca pelo Desafio proposto, e caso a *Sprint* não a tenha obtido, ser adaptada para a próxima *Sprint*.

Logo na sequência, na reunião de Retrospectiva da *Sprint*, e utilizando a rotina de *Kata* de *Coaching*, foi realizada uma reflexão, conforme mostra a última fase do modelo, a fim de fazer um apanhado geral da *Sprint*, para compartilhar e consolidar conhecimento, bem como, os aprendizados adquiridos durante toda *Sprint* com todo time.

Essa reflexão fica registrada de forma genérica no *Storyboard* de Pesquisa-ação, no campo o que foi aprendido da sua respectiva *Sprint*. Essas cerimônias de finalização da *Sprint* se repetirão em todas as demais, a fim de verificar o comportamento do modelo, principalmente como se comporta a fase de reflexão da mesma.

4.3.1.3 *Sprint 3*

Da mesma forma como ocorrido na *Sprint 2*, e como ocorrerão nas demais, a *Kata* de Pesquisa-ação possui o mesmo processo, desafio e condições atual e alvo, mudando apenas o contexto da *Sprint*, sendo o foco neste momento a *Sprint 3*, conforme mostra a Figura 44.

Figura 44 – Storyboard de Pesquisa-ação da Sprint 3

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O time não conhece as condições atual e alvo que trabalharão na Sprint	1 - Entender a condição atual e estabelecer a condição-alvo da Sprint do Time	Desconhecimento dos aspectos do storyboard relacionados a Sprint	Preencher as informações do Storyboard	Que o Storyboard estivesse com seus dados iniciais completos ao final da reunião de planejamento da Sprint, e o time compreendesse qual a funcionalidade do mesmo	Os dois Storyboards foram preenchidos com seus respectivos campos necessários, e o time entendeu para que serve o Storyboard	Qual é o processo, desafio, e condições atual e alvo que serão trabalhadas na Sprint, tanto na pesquisa-ação quanto pelo time
2 - Falta dos itens relativos ao Scrum como, as estória de usuário que contemplarão a Condição-Alvo	2 - Criar as estória de usuário que contemplem a condição-alvo	Falta das estórias de usuário para trabalhar na Sprint	Criar as estórias de usuário da Sprint	Que o time estivesse ciente e confortável do que precisa ser entregue na Sprint a fim de alcançar a Condição-Alvo	Com as estórias de usuário criadas conforme a demanda, ficou claro para o time o que precisava ser feito, e desta forma o mesmo alcançou o que foi proposto para a Sprint	Que desta Sprint em diante não foi mais necessária a criação de uma Funcionalidade, pois a mesma é relacionada ao Desafio, que neste caso é único para os Storyboards até o final do ciclo de 3 meses. Criar as estórias de usuário sob demanda evita muitos desperdícios, no entanto o time está acostumado com previsibilidade dos trabalhos futuros, pois no uso somente do Scrum, praticamente tudo o que será feito durante os 3 meses, é criado antecipadamente, mesmo que após a criação dos itens, eles não sejam utilizados
3 - Não foi realizada a fase de reflexão a fim de verificar se a	3 - Realizar a Revisão e a Retrospectiva da Sprint a fim de	Falta da revisão, do compartilhamento de	Realizar a revisão e retrospectiva da	Que o time refletisse sobre o ocorrido na Sprint, a fim	Foi realizada a revisão e retrospectiva da Sprint,	A aplicação diária dos ciclos de coaching está se mostrando bem satisfatória pro

Fonte: Dados primários do autor.

Na mesma cerimônia de Planejamento da *Sprint*, no momento que foi elaborado o *Storyboard* da pesquisa-ação, também foi definido o *Storyboard* do time, conforme mostra a Figura 45.

Para o *Storyboard* do time por sua vez, ficou estabelecido o seguinte: Processo – Assinatura e Liberação de Documentos; Desafio - Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários; Condição Atual - Não é utilizada a assinatura de documentos digitais no cliente baseado em hash; Condição-Alvo - Testar uma solução que assine documentos digitais no cliente baseados em hash, tanto no Delphi Tokyo quanto no Delphi 5.

Figura 45 – *Storyboard* do Time da *Sprint 3*

Condição Atual		Condição-Alvo	Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
			PDCA				
			Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
Não é utilizada a assinatura de documentos digitais no cliente baseado em hash		Testar uma solução que assine documentos digitais no cliente baseados em hash, tanto no delphi tokyo quanto no delphi 5	Componentes já instalados do delphi tokyo dando conflito	Limpar o Delphi Tokyo	Que ao abrir o Delphi tokyo nenhum componente estivesse instalado	A limpeza do delphi possibilitou que os componentes fossem desinstalados, e com isso não ocorresse mais conflito	
			Desconhecimento do add-on de hash	Testar a assinatura com add-on do delphi tokyo	Que ao utilizar o add-on no Delphi Tokyo, fosse possível assinar um documento pelo seu hash	Foi possível utilizar o hash para assinatura de documentos, o que ocasionou em menor tráfego de bytes pela rede, melhorando a performance da assinatura de documentos	Como aplicar criptografia em documento, e como o componente do add-on funciona na assinatura do hash
			Falta de assinatura via hash no delphi 5 com o SBB	Verificar se o SBB no delphi 5 assina hash	Que ao utilizar o add-on no Delphi 5, fosse possível assinar um documento pelo seu hash	Não foi possível realizar esta operação com o SBB no Delphi 5, pois o componente não funciona para Delphi 5	Que o SBB não funciona no delphi 5

Fonte: Dados primários do autor.

Da mesma forma como se deu na *Sprint* anterior, e como se dará nas próximas, a Condição-Alvo, gerou as estórias de usuário.

Com o *Storyboard* do time definido, e visando não tornar a Pesquisa-ação repetitiva, optou-se por não registrar os ciclos PDCA desta *Sprint* neste estudo, nem das demais. Por fim, o time realizou diariamente os experimentos em direção a Condição-Alvo estabelecida, relatando diariamente o PDCA dos experimentos. Desta forma, ao final da *Sprint*, a Condição-Alvo estabelecida foi alcançada conforme previsto, e com isso a expectativa de utilizar *hash* na assinatura de documentos digitais se mostrou viável.

4.3.1.4 *Sprint* 4

Para a *Sprint* 4, na reunião de Planejamento da *Sprint*, foi desenvolvido o seguinte *Storyboard* da Pesquisa-ação: Processo – Pesquisa-ação; Desafio – Implantar, testar e verificar o comportamento do modelo proposto; e a Condição Alvo – 1) Entender a condição atual e estabelecer a condição-alvo da *Sprint* do Time, 2) Criar as estória de usuário que contemplem a condição-alvo, 3) Realizar a Revisão e a Retrospectiva da *Sprint* a fim de verificar se a Condição-Alvo foi atendida, bem como, se será necessária uma adaptação para a próxima Condição-Alvo, conforme mostra a Figura 46.

Figura 46 – *Storyboard* da Pesquisa-ação da *Sprint* 4

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto					
Condição Atual		Condição-Alvo	PDCA				
			Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O time não conhece as condições atual e alvo que trabalharão na Sprint		1 - Entender a condição atual e estabelecer a condição-alvo da Sprint do Time	Desconhecimento dos aspectos do storyboard relacionados a Sprint	Preencher as informações do Storyboard	Que o Storyboard estivesse com seus dados iniciais completos ao final da reunião de planejamento da Sprint, e o time compreendesse qual a funcionalidade do mesmo	Os dois Storyboards foram preenchidos com seus respectivos campos necessários, e o time entendeu para que serve o Storyboard	Qual é o processo, desafio, e condições atual e alvo que serão trabalhadas na Sprint, tanto na pesquisa-ação quanto pelo time
2 - Falta dos itens relativos ao Scrum como, as estória de usuário que contemplarão a Condição-Alvo		2 - Criar as estórias de usuário que contemplem a condição-alvo	Falta das estórias de usuário para trabalhar na Sprint	Criar as estórias de usuário da Sprint	Que o time estivesse ciente e confortável do que precisa ser entregue na Sprint a fim de alcançar a Condição-Alvo	Com as estórias de usuário criadas conforme a demanda, ficou claro para o time o que precisava ser feito, e desta forma o mesmo alcançou o que foi proposto para a Sprint	
3 - Não foi realizada a fase de reflexão a fim de verificar se a Condição-Alvo foi atendida, e se é necessária uma adaptação		3 - Realizar a Revisão e a Retrospectiva da Sprint a fim de verificar se a Condição-Alvo foi atendida, bem como, se será necessária uma adaptação para a próxima Condição-Alvo	Falta da revisão, do compartilhamento de conhecimento e adaptabilidade	Realizar a revisão e retrospectiva da sprint	Que o time refletisse sobre o ocorrido na Sprint, a fim de verificar e comparar o que foi proposto, e o que foi alcançado, e caso necessário, se adaptar a fim de estabelecer uma nova Condição-Alvo aderente ao Desafio proposto. Que os	Foi realizada a revisão e retrospectiva da Sprint, onde o time refletiu sobre o ocorrido, sugerindo melhorias e compartilhando registro do Storyboard fosse relatado na Sprint	Que a perda de foco devido a entrada de outras demandas não previstas, porém mais prioritárias, interferiu na realização da Condição-Alvo proposta. Na retrospectiva o time solicitou que o registro do Storyboard fosse relatado na ferramenta comumente utilizada para controlar as demandas, por isso, os Storyboards 5 e 6 do Time serão

Fonte: Dados primários do autor.

Após elaborado o *Storyboard* da pesquisa-ação, foi a vez do *Storyboard* do time. Este, por sua vez, conforme mostra a Figura 47, identificou o seguinte: Processo – Assinatura e Liberação de Documentos; Desafio – Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários; e a Condição Alvo – 1) implementar o envio de mensagens das threads pelo barramento, e 2) Testar as movimentações paralelas no sistema.

Figura 47 – Storyboard do Time da Sprint 4

Processo: Assinatura e Liberação de Documentos		Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
Não possuímos uma solução para as threads paralelas das movimentações que consiga tornar as mesmas assíncronas	Implementar o envio de mensagens das threads pelo barramento	Falta o docker do ambiente	Configurar o ambiente local do barramento	Que o ambiente do barramento estivesse estável e funcionando a ponto de ser utilizado pelas threads	Mesmo não tendo uma documentação sobre como iniciar o barramento em um ambiente local, com a ajuda do time Japão e alguns colegas, consegui-se configurar o ambiente com sucesso. Uma nova seção foi criada no onenote chamada "Integração" onde foi documentado como foi feita toda a configuração	Configurar o barramento de forma básica (elastic, kibana, rabbitmq, mongo)
As implementações de movimentação paralela realizadas no kata anterior não foram testadas no sistema	Testar as movimentações paralelas no sistema	Nunca ter mexido no barramento, tempo de adaptação e desconhecimento das classes	Implementar o comportameto básico: publicação da fila no barramento e consumir essa fila gerando as movimentações	Que ao enviar as movimentações pelo barramento, por meio das filas, as mesmas cheguem no seu destino corretamente, sem se perder no caminho	Devido a grande quantidade de exemplos que tem no código, foi possível publicar as filas das movimentações e o consumo no barramento	Como enviar/consumir mensagens no barramento, como tirar da fila, etc.

Fonte: Dados primários do autor.

Nesta *Sprint* o time acabou por não conseguir alcançar a Condição-Alvo estabelecida. Primeiramente porque na Condição-Alvo que visava implementar o envio das mensagens, a mesma não obteve êxito, já que não foi implementado o envio dos logs pelo Elastic. Além disso, optou-se nesse momento por deixar a Condição-Alvo de teste das movimentações paralelas para a *Sprint* 6, pois faria mais sentido, já que a implementação das filas não havia sido realizada.

Com base nisso, ao realizar a Retrospectiva da *Sprint*, verificou-se que uma adaptação foi necessária em termos de Condição-Alvo, bem como Estórias de Usuário. Com isso, foi criada uma nova Condição-Alvo como será visto na *Sprint* 5, bem como, adequadas as Estórias de Usuário que já haviam sido criadas, mas não foram entregues.

4.3.1.5 *Sprint* 5

Da mesma forma que as *Sprints* anteriores, a *Sprint* 5 possui o mesmo processo, desafio e Condições-Alvo para o *Storyboard* da pesquisa-ação, focando claro na própria *Sprint*, conforme mostra a Figura 48.

Conforme informado, foram necessárias adaptações para essa *Sprint*. A primeira delas foi uma adequação na Condição-Alvo. Como a Condição-Alvo da *Sprint* 4 não foi alcançada, a mesma continuou sendo o objetivo da *Sprint* 5, com uma pequena adequação com a inclusão de outros itens para compor a Condição-Alvo.

Outra adaptação necessária foi a adequação das Estórias de Usuário já existentes e a criação de duas novas, com base na Condição-Alvo que foi adequada, conforme será visto a seguir.

Figura 48 – *Storyboard* da Pesquisa-ação da *Sprint 5*

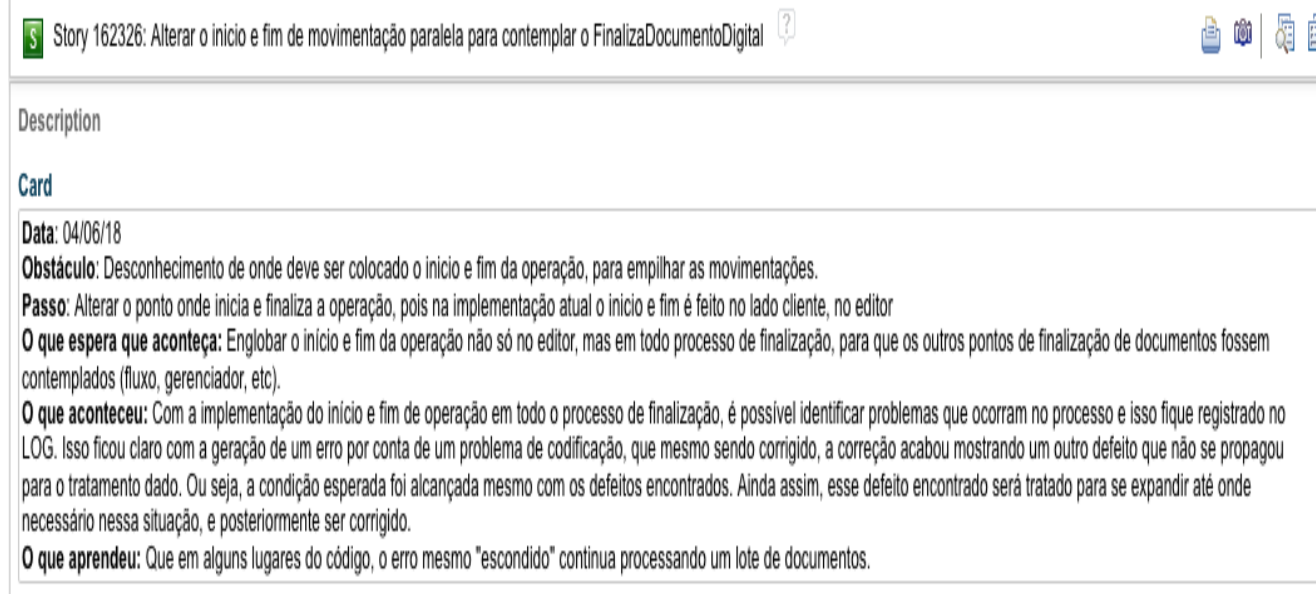
Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O time não conhece as condições atual e alvo que trabalharão na Sprint	1 - Entender a condição atual e estabelecer a condição-alvo da Sprint do Time	Desconhecimento dos aspectos do storyboard relacionados a Sprint	Preencher as informações do Storyboard	Que o Storyboard estivesse com seus dados iniciais completos ao final da reunião de planejamento da Sprint, e o time compreendesse qual a funcionalidade do mesmo	Os dois Storyboards foram preenchidos com seus respectivos campos necessários, e o time entendeu para que serve o Storyboard	Qual é o processo, desafio, e condições atual e alvo que serão trabalhadas na Sprint, tanto na pesquisa-ação quanto pelo time
2 - Falta dos itens relativos ao Scrum como, as estória de usuário que contemplarão a Condição-Alvo	2 - Criar as estória de usuário que contemplem a condição-alvo	Falta das estórias de usuário para trabalhar na Sprint	Criar as estórias de usuário da Sprint	Que o time estivesse ciente e confortável do que precisa ser entregue na Sprint a fim de alcançar a Condição-Alvo	Com as estórias de usuário criadas conforme a demanda, ficou claro para o time o que precisava ser feito, e desta forma o mesmo alcançou o que foi proposto para a Sprint	Que se pode adequar as estórias de usuário, sem a necessidade de criar novas, dependendo do que foi acrescido devido a uma adaptabilidade necessária de uma Sprint para outra
3 - Não foi realizada a fase de reflexão a fim de verificar se a Condição-Alvo foi atendida, e se é necessária uma adaptação	3 - Realizar a Revisão e a Retrospectiva da Sprint a fim de verificar se a Condição-Alvo foi atendida, bem como, se será necessária uma adaptação para a próxima Condição-Alvo	Falta da revisão, do compartilhamento de conhecimento e adaptabilidade	Realizar a revisão e retrospectiva da sprint	Que o time refletisse sobre o ocorrido na Sprint, a fim de verificar e comparar o que foi proposto, e o que foi alcançado, e caso necessário, se adaptar a fim de estabelecer uma nova Condição-Alvo	Foi realizada a revisão e retrospectiva da Sprint, onde o time refletiu sobre o ocorrido, sugerindo melhorias e compartilhando o que foi aprendido na Sprint	Que independente da ferramenta utilizada pela empresa, o processo se mostrou robusto de tal forma que pode ser usado em qualquer área e em qualquer ferramenta, como foi o caso nesta sprint, com a mudança do excel para o Rational Team Concert (RTC), sem quaisquer contratempos

Fonte: Dados primários do autor.

Desta forma, com a Condição-Alvo do time sendo alterada, a mesma acabou por adequar e criar as seguintes Estórias de Usuário para a *Sprint*: 1) Implementar o envio de mensagens das threads pelo barramento; 2) Alterar o início e fim de movimentação paralela para contemplar o `FinalizaDocumentoDigital`; 3) Logar os campos como *tags*, quando logado info com dados da movimentação no Elastic, ao invés de um JSON; 4) Desenvolver os testes unitários para a implementação de movimentação paralela.

Por fim, outra adaptação feita nessa *Sprint* foi a mudança de ferramenta para os registros do Processo, Desafio, Condições Atual e Alvo e dos ciclos PDCA do time. Os registros, conforme visto nas figuras até então, foram feitos num arquivo *Excel*, nos quais os mesmos eram separados por *Kata* de Pesquisa-ação (que continuam sendo registrados nesse arquivo) e *Kata* do Time.

O *Kata* do Time por sua vez, para a *Sprint* 5 e 6 foi feita uma adaptação, pois como o time já utiliza uma ferramenta que integra Evoluções, Funcionalidades e Estórias de Usuário, além de já possibilitar questões de sincronização de código, os registros dos ciclos PDCA para essas *Sprints* se deram na própria Estória de Usuário, conforme mostra a Figura 49.

Figura 49 – Registro dos Ciclos PDCA da *Sprint 5*

Story 162326: Alterar o inicio e fim de movimentação paralela para contemplar o FinalizaDocumentoDigital

Description

Card

Data: 04/06/18

Obstáculo: Desconhecimento de onde deve ser colocado o inicio e fim da operação, para empilhar as movimentações.

Passo: Alterar o ponto onde inicia e finaliza a operação, pois na implementação atual o inicio e fim é feito no lado cliente, no editor

O que espera que aconteça: Englobar o inicio e fim da operação não só no editor, mas em todo processo de finalização, para que os outros pontos de finalização de documentos fossem contemplados (fluxo, gerenciador, etc).



O que aconteceu: Com a implementação do inicio e fim de operação em todo o processo de finalização, é possível identificar problemas que ocorram no processo e isso fique registrado no LOG. Isso ficou claro com a geração de um erro por conta de um problema de codificação, que mesmo sendo corrigido, a correção acabou mostrando um outro defeito que não se propagou para o tratamento dado. Ou seja, a condição esperada foi alcançada mesmo com os defeitos encontrados. Ainda assim, esse defeito encontrado será tratado para se expandir até onde necessário nessa situação, e posteriormente ser corrigido.

O que aprendeu: Que em alguns lugares do código, o erro mesmo "escondido" continua processando um lote de documentos.

Fonte: Dados primários do autor.

Já para explicitar o Processo, o Desafio e as Condições Atual e Alvo da *Sprint 5*, foi feita uma adaptação na Funcionalidade (*Feature*), para que a mesma mostrasse estes itens, no qual os itens já finalizados ficam em negrito, do contrário não, conforme visto na Figura 50.

Figura 50 – *Storyboard* do Time da *Sprint 5*

 Feature 163441: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários 

Description

Conditions

Processo: Assinatura e Liberação de Documentos
Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários

Condição Atual

- 1 - As movimentações automáticas no fluxo não são paralelizadas (Sprint 2)
- 2 - Não utilizamos assinatura de documentos digitais no cliente baseado em hash (Sprint 3)
- 3 - Não possuímos uma solução para as threads paralelas das movimentações que consiga tornar as mesmas assíncronas (Sprint 4 e Sprint 5)
- 4 - A implementação da movimentação paralela foi feita apenas para o editor (Sprint 5)
- 5 - Quando logado info com dados da movimentação no elastic o JSON está ficando muito grande (Sprint 5)
- 6 - Não foi feito nenhum teste unitário para a implementação de movimentação paralela (Sprint 5 e Sprint 6)
- 7 - O usuário não possui um retorno sobre a movimentação realizada pós finalização (Sprint 6)
- 8 - Não foi realizado nenhum teste sistêmico na movimentação paralela utilizando o barramento (Sprint 6)

Condição-Alvo

- 1 - Verificar se existe uma maneira de paralelizar as movimentações automáticas do fluxo, sem causar uma mudança no comportamento do sistema atual (Realizado na Sprint 2)
- 2 - Testar uma solução que assine documentos digitais no cliente baseados em hash, tanto no delphi tokyo quanto no delphi 5 (Realizado na Sprint 3)
- 3 - Implementar o envio de mensagens das threads pelo barramento (Realizado na Sprint 5)
- 4 - Alterar o inicio e fim de movimentação paralela para contemplar o FinalizaDocumentoDigital (Realizado na Sprint 5)
- 5 - Enviar os campos do log como tags ao invés de um JSON, para que seja pesquisável na ferramenta *Elastic* (Realizado na Sprint 5)
- 6 - Desenvolver os testes unitários para a implementação de movimentação paralela (Sprint 5 e Sprint 6)
- 7 - Implementar o feedback de retorno para manter o usuário informado sobre a automação do fluxo pós finalização de documento (Sprint 6)
- 8 - Testar as movimentações paralelas no sistema (Sprint 6)

Fonte: Dados primários do autor.

Com as adaptações feitas, o time experimentou em busca do objetivo da *Sprint*, no entanto, ao final da mesma três das quatro Condições-Alvo foram alcançadas. Assim sendo, ao realizar a Retrospectiva da *Sprint*, o time verificou a necessidade de uma nova adaptação para a próxima *Sprint*, a qual será relatada a seguir.

4.3.1.6 *Sprint* 6

Para a última *Sprint* a ser realizada a fim de verificar e testar o modelo, uma adaptação foi feita a fim de se estabelecer e alcançar uma nova Condição-Alvo, já que a anterior não foi realizada com sucesso.

Desta forma, foi criada uma nova Estória de Usuário, já que as demais que não foram realizadas nas *Sprints* anteriores, acabaram sendo realocadas para esta *Sprint*.

Para finalizar o teste e verificação da pesquisa-ação, o *Storyboard* da *Sprint* 6 continua com o mesmo Processo, Desafio e Condições Atual e Alvo das *Sprints* anteriores, apenas com o foco somente nesta *Sprint*, como visto na Figura 51.



Figura 51 – Storyboard da Pesquisa-ação da Sprint 6

Processo: Pesquisa-ação		Desafio: Implantar, testar e verificar o comportamento do modelo proposto				
		PDCA				
Condição Atual	Condição-Alvo	Obstáculo	Passo	O que espera que aconteça?	O que aconteceu?	O que aprendeu?
1 - O time não conhece as condições atual e alvo que trabalharão na Sprint.	1 - Entender a condição atual e estabelecer a condição-alvo da Sprint do Time	Desconhecimento dos aspectos do storyboard relacionados a Sprint	Preencher as informações do Storyboard	Que o Storyboard estivesse com seus dados iniciais completos ao final da reunião de planejamento da Sprint, e o time compreendesse qual a funcionalidade do mesmo	Os dois Storyboards foram preenchidos com seus respectivos campos necessários, e o time entendeu para que serve o Storyboard	Qual é o processo, desafio, e condições atual e alvo que serão trabalhadas na Sprint, tanto na pesquisa-ação quanto pelo time
2 - Falta dos itens relativos ao Scrum como, as estória de usuário que contemplarão a Condição-Alvo	2 - Criar as estória de usuário que contemplem a condição-alvo	Falta das estórias de usuário para trabalhar na Sprint	Criar as estórias de usuário da Sprint	Que o time estivesse ciente e confortável do que precisa ser entregue na Sprint a fim de alcançar a Condição-Alvo	Com as estórias de usuário criadas conforme a demanda, ficou claro para o time o que precisava ser feito, e desta forma o mesmo alcançou o que foi proposto para a Sprint	
3 - Não foi realizada a fase de reflexão a fim de verificar se a Condição-Alvo foi atendida, e se é necessária uma adaptação	3 - Realizar a Revisão e a Retrospectiva da Sprint a fim de verificar se a Condição-Alvo foi atendida, bem como, se será necessária uma adaptação para a próxima Condição-Alvo	Falta da revisão, do compartilhamento de conhecimento e adaptabilidade	Realizar a revisão e retrospectiva da sprint	Que o time refletisse sobre o ocorrido na Sprint, a fim de verificar e comparar o que foi proposto, e o que foi alcançado, e caso necessário, se adaptar a fim de estabelecer uma nova Condição-Alvo	Foi realizada a revisão e retrospectiva da Sprint, onde o time refletiu sobre o ocorrido, sugerindo melhorias e compartilhando o que foi aprendido na Sprint	Quanto menor a cobertura de código no sistema, maior é a recorrência de defeitos, o que acaba por prejudicar o foco da equipe em melhorias e novas funcionalidades, como é o caso do Desafio do Time, e com isso não alcançar o Desafio estabelecido na data acordada

Fonte: Dados primários do autor.

Para esta *Sprint* além da Estória de Usuário que não foi atendida na *Sprint 5* (Desenvolver os testes unitários para a implementação de movimentação paralela), foi criada uma nova Estória de Usuário, sendo ela: 1) implementar o feedback de retorno para manter o usuário informado sobre a automação do fluxo pós finalização de documento. Outra Estória que volta a fazer parte do contexto, que não foi alcançada na *Sprint 4* é: Testar as movimentações paralelas no sistema. Estes itens constam no *Storyboard* do time, conforme mostra a Figura 52.

Figura 52 – *Storyboard* do Time da *Sprint* 6

 Feature 163441: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários 

Description

Conditions

Processo: Assinatura e Liberação de Documentos
Desafio: Implementar a movimentação paralela pelo barramento, o feedback pós finalização para o usuário e seus testes unitários

Condição Atual

- 1 - As movimentações automáticas no fluxo não são paralelizadas (Sprint 2)
- 2 - Não utilizamos assinatura de documentos digitais no cliente baseado em hash (Sprint 3)
- 3 - Não possuímos uma solução para as threads paralelas das movimentações que consiga tornar as mesmas assíncronas (Sprint 4 e Sprint 5)
- 4 - A implementação da movimentação paralela foi feita apenas para o editor (Sprint 5)
- 5 - Quando logado info com dados da movimentação no elastic o JSON está ficando muito grande (Sprint 5)
- 6 - Não foi feito nenhum teste unitário para a implementação de movimentação paralela (Sprint 5 e Sprint 6)
- 7 - O usuário não possui um retorno sobre a movimentação realizada pós finalização (Sprint 6)
- 8 - Não foi realizado nenhum teste sistêmico na movimentação paralela utilizando o barramento (Sprint 6)


Condição-Alvo

- 1 - Verificar se existe uma maneira de paralelizar as movimentações automáticas do fluxo, sem causar uma mudança no comportamento do sistema atual (Realizado na Sprint 2)
- 2 - Testar uma solução que assine documentos digitais no cliente baseados em hash, tanto no delphi tokyo quanto no delphi 5 (Realizado na Sprint 3)
- 3 - Implementar o envio de mensagens das threads pelo barramento (Realizado na Sprint 5)
- 4 - Alterar o início e fim de movimentação paralela para contemplar o FinalizaDocumentoDigital (Realizado na Sprint 5)
- 5 - Enviar os campos do log como tags ao invés de um JSON, para que seja pesquisável na ferramenta *Elastic* (Realizado na Sprint 5)
- 6 - Desenvolver os testes unitários para a implementação de movimentação paralela (Sprint 5 e Sprint 6)
- 7 - Implementar o feedback de retorno para manter o usuário informado sobre a automação do fluxo pós finalização de documento (Sprint 6)
- 8 - Testar as movimentações paralelas no sistema (Sprint 6)

Fonte: Dados primários do autor.

Desta forma, feita a adaptação necessária para a última *Sprint*, o time foi em busca de alcançar a Condição-Alvo estabelecida, realizando os ciclos PDCA diariamente, conforme mostra o registro de um exemplo do PDCA na Figura 53.

Figura 53 – Registro dos Ciclos PDCA da *Sprint 6*

Story 162333: Desenvolver os testes unitários para a implementação de movimentação paralela* 

Conditions

Data: 18/06

Obstáculo: Falta de testes unitários na movimentação paralela

Passo: Implementar testes unitários nas classes criadas para a movimentação paralela

O que espera que aconteça:

Criação de teste para as classes:

TwflMovimentacaoFluxoParalelaTests - 4 testes

O que aconteceu: Checar se inicia operação com parâmetro habilitado, desabilitado, e se finaliza operação com parâmetro habilitado/desabilitado.

O que aprendeu: O que a classe TwflMovimentacaoFluxoParalelaTests faz, e como o teste unitário utiliza a mesma

Data: 19/06

Obstáculo: Falta de testes unitários na movimentação paralela

Passo: Implementar testes unitários nas classes criadas para a movimentação paralela

O que espera que aconteça:

Criação de teste para as classes:

- TwflMovimentacaoFluxoFilaBarramento - 1 teste para garantir a descrição da fila

- TwflListaMovimentacoesFluxo - 1 teste - garantir método ToJSON que o JSON é válido

O que aconteceu:

- TwflListaMovimentacoesFluxo - não foi feito o teste pois o método ToJSON foi retirado. Ao invés de logar um JSON no Elastic serão criados tags, para melhor visualização.

- TwflMovimentacaoFluxoFilaBarramento - feito teste para garantir descrição da fila.

O que aprendeu: O que a classe TwflListaMovimentacoesFluxo e a classe TwflMovimentacaoFluxoFilaBarramento fazem, e como o teste unitário utiliza a classe TwflMovimentacaoFluxoFilaBarramento. Como utilizar tags no método TwflListaMovimentacoesFluxo para envio ao Elastic.

Fonte: Dados primários do autor.

Findada a última *Sprint*, pode-se observar que o desafio proposto não foi alcançado, já que a Condição-Alvo de Implementar o feedback de retorno para manter o usuário informado sobre a automação do fluxo pós finalização de documento não foi realizada, nem o teste do sistema para verificar se não houve impacto no mesmo. Isso tudo, devido a dificuldade de implementar esta Condição-Alvo, pois a mesma se mostrou mais complexa do que o esperado quando foi estabelecida, além da prioridade ter sido alterada ao longo da *Sprint* com outras demandas.

4.3.2 Coleta de dados

Os dados coletados referentes a pesquisa-ação foram devidamente registrados nos *Storyboards* da própria pesquisa-ação e do time, conforme visto no tópico anterior, a fim de facilitar a análise, a avaliação e os resultados alcançados, conforme será mostrado a seguir.

4.3.3 Avaliação e resultados

Com base nas Condições-Alvo propostas nas seis *Sprints* do time, conforme já visto neste capítulo, pode-se observar que em três delas, especificamente as três primeiras, o objetivo final das mesmas foi alcançado, conforme visto ao longo da implantação do modelo. No entanto, como houveram algumas adequações e alterações ao longo do percurso, bem como, como alguns objetivos esperados não foram alcançados nem relatados detalhadamente, faz-se necessário destacar aqui o que era esperado para cada *Sprint*, bem como, o que foi alcançado em cada uma delas, a fim de comparar os objetivos propostos.

Desta forma, o objetivo da *Sprint* 1 foi: realizar a fase de preparação do modelo. Todos os itens da Condição-Alvo estabelecida foram realizados com sucesso. Nesse momento, como já relatado, o modelo verificou que o uso do A3 para solução de problemas foi uma ferramenta capaz de detalhar de forma clara o desperdício encontrado no MFV, facilitando assim o direcionamento do time e a criação do *Backlog* do Produto de forma incremental e puxada, conforme foi explicitado na pesquisa-ação.

Na *Sprint* 2 o objetivo do time foi: verificar se existe uma maneira de paralelizar as movimentações automáticas do fluxo, sem causar uma mudança no comportamento do sistema atual. Objetivo esse também alcançado com sucesso, sem maiores problemas. Isso foi feito pois o time não sabia se era possível tornar a movimentação automática do fluxo paralela, a fim de melhorar sua performance e torná-la assíncrona. Feita a

verificação, descobriu-se que era possível tornar a movimentação paralela e de forma assíncrona, sem causar prejuízo ao funcionamento normal do fluxo de trabalho. Isso por consequência pode ajudar na visão estabelecida, que é justamente tornar o processo de finalização de documentos assíncrono. Além disso, o usuário fica menos tempo esperando o sistema terminar a operação, liberando-o assim para executar outra operação.

A *Sprint 3* por sua vez teve o seguinte objetivo: Verificar a possibilidade de utilização de alguma tecnologia para assinar *hashs* de uma assinatura digital, e não todo o documento, tanto no Delphi Tokyo quanto no Delphi 5. Da mesma forma que nas *Sprints* anteriores, essa também obteve êxito. Assim, optou-se por utilizar o componente SBB afim de assinar apenas o *hash*, fazendo com que o tráfego diminuísse de forma considerável. Isso tornou o processo de assinatura mais rápido, já que não era mais necessário o envio do documento por completo, e sim, somente seu *hash*, que por sua vez possui um tamanho muito menor que o documento.

Já na *Sprint 4*, pela primeira vez a Condição-Alvo estabelecida não foi alcançada. Conforme visto, o objetivo era o de garantir que as mensagens paralelizadas enviadas pelo barramento chegassem ao seu devido destino. Esse objetivo não foi atingido pois outras demandas se tornaram prioritárias no momento, o que fez com que o time perdesse o foco na implementação do envio das mensagens pelo barramento, além de não realizar os testes para verificar se as movimentações paralelas implementadas não causariam impacto negativo no sistema, bem como, a perda de alguma delas no barramento.

Isso fez com que a *Sprint 5* fosse adaptada de tal forma que contemplasse a condição não alcançada no *Sprint* anterior, bem como, a mesma fosse adequada a fim de torná-la mais aderente ao desafio pretendido, ficando assim com a seguinte Condição-Alvo: 1) Implementar o envio de mensagens das threads pelo barramento; 2) Alterar o início e fim de movimentação paralela para contemplar o Finaliza Documento Digital; 3) Enviar os campos do log como *tags* ao invés de um JSON, para que seja pesquisável na ferramenta Elastic; 4) Desenvolver os testes unitários para a implementação de movimentação paralela.

Ao encontrar dificuldades na experimentação para atingir a Condição-Alvo da *Sprint 5*, e novamente, ao mudar o foco do time por outras demandas mais prioritárias, mais uma vez a mesma não foi alcançada. Foi implementado três dos quatro itens da Condição-Alvo, no entanto, os testes unitários não foram realizados.

Desta forma, a *Sprint* 6 contou também com uma adaptação ao contemplar o que não foi feito na *Sprint* anterior e acrescentando ainda: a) implementação do feedback de retorno para manter o usuário informado sobre a automação do fluxo pós finalização de documento e b) testar as movimentações paralelas no sistema.

Da mesma forma como as duas *Sprints* anteriores, a *Sprint* 6 também não obteve êxito completo, ao não conseguir implementar o feedback de retorno devido a alta complexidade para desenvolver tal feedback, e como o mesmo não foi desenvolvido, os testes no sistema também não foram possíveis de serem completados.

Por fim, pode-se verificar e avaliar a implantação do modelo, mesmo tendo ocorridos alguns contratemplos na busca pela realização de algumas Condições-Alvo pelo time, o que é comum e normal já que o PDCA é justamente um experimento em direção a mesma. Isso de forma alguma invalida a implantação do modelo, pois este se mostrou bem estruturado e detalhado, como visto neste capítulo, a ponto de independente da ferramenta que se usa para estabelecer e registrar as rotinas *Toyota Kata* integradas ao *Scrum*, o modelo é robusto de tal forma que pode ser utilizado sem quaisquer mudanças, ou adaptações significativas.

Conforme colocado no início do capítulo, o objetivo principal do mesmo foi o de implantar, verificar e testar o modelo. Com base nesse objetivo, e descrito no transcorrer de todo o capítulo o mesmo foi alcançado conforme o esperado. Mesmo havendo mudanças no transcorrer do caminho, especialmente na possibilidade de uso de novas ferramentas, como por exemplo, a utilização do A3 ou da mudança do *Excel* para o RTC, o modelo se mostrou aderente de tal forma que independente disso, este obteve os resultados esperados, que era uma integração das rotinas *Toyota Kata* e *Scrum*, contemplando principalmente os pontos fracos do *Scrum*.

4.4 CONSIDERAÇÕES FINAIS DA PESQUISA-AÇÃO

Conforme visto ao longo do capítulo o modelo foi testado numa situação real envolvendo seis *Sprints* a fim de verificar o comportamento do mesmo. Com base nos objetivos principais do estudo, que são a implantação das rotinas *Toyota Kata* no processo de desenvolvimento de *software* que utiliza o *Scrum*, bem como, que as principais lacunas encontradas na literatura na implantação de *Toyota Kata* fossem atendidas, se pode chegar as considerações que seguem.

Na *Sprint 1* a qual se focou principalmente na fase de preparação, apresentando o *Hoshin Kanri* ao time, sensibilizando a equipe e elaborando o MFV, o modelo contemplou então um dos dois objetivos principais deste estudo, que era o de atender as três principais barreiras na implantação das rotinas *Toyota Kata* encontradas na literatura. Desta forma, apresentando a direção da empresa, sensibilizando o time, e elaborando o mapa do estado atual de um processo, foi possível então alinhar a estratégia a operação por meio de um *Backlog* de Produto gerado a partir dos desperdícios e problemas encontrados de fato no *gemba*. Um aprendizado gerado pelo modelo neste momento foi a quantidade de tempo utilizado para a sensibilização, o qual as quatro horas não se mostrou suficiente para um time de aproximadamente 25 pessoas. Pela experiência passada, uma quantidade de oito horas se mostraria mais adequada. Outro conhecimento adquirido neste momento foi a utilização do A3 para solução de problemas. Este foi necessário devido ao pouco esclarecimento do problema real encontrado no *gemba*, o qual o MFV não deixa explícito de forma clara e objetiva.

Já da *Sprint 2* a 6, o modelo executou as fases de planejamento, execução e reflexão de forma constante e consecutiva. Na fase de planejamento, como o próprio nome diz, houve a integração do A3, com a *Kata* de Melhoria e o *Scrum* da seguinte forma: da causa raiz do A3 surgiu a respectiva Evolução (Épico) no *Scrum*; do Desafio da *Kata* de Melhoria, surgiu a respectiva funcionalidade no *Scrum*; e por fim, da Condição-Alvo da rotina *Kata* de Melhoria, deu-se as respectivas Estórias de Usuário (*User Stories*) no *Scrum*. Desta forma se deu então a construção do *Backlog* de Produto de forma puxada, ou seja, a cada *Sprint* se revisava o planejado com o realizado (Revisão da *Sprint*), e com base nisso, se adaptava a próxima *Sprint* ou se criavam os itens da mesma conforme a necessidade. Este tipo de planejamento pode atender os conceitos utilizados pelo *Scrum* e ainda minimizar as lacunas encontradas no método conforme explicitadas no capítulo anterior, como por exemplo, a confusão na visão devido a diversos pontos de vista, bem como, a falta de utilização de fatos, dados e processos para o estabelecimento de objetivos específicos e mensuráveis para a entrega.

Na fase de execução do modelo, o mesmo integrou a experimentação dos ciclos PDCA da rotina *Kata* de Melhoria dentro da *Sprint* do *Scrum*, e da mesma forma, utilizou os ciclos curtos de *Kata* de *Coaching* na reunião de *Kata* Diária para que o time pudesse compartilhar o conhecimento adquirido diariamente com todos, e desta forma relatar o aprendizado no *Storyboard* do time. Com isso, a falta de compartilhamento rotineiro de conhecimento que é uma das lacunas no

Scrum, pode ser contemplada pelo modelo utilizando os ciclos curtos de *Kata* de *Coaching*, além da falta de utilização de método científico para experimentação, o qual o modelo utilizou os ciclos PDCA.

Já na fase de reflexão o modelo começou integrando a rotina de *Kata* de Melhoria no *Scrum* por meio da Condição-Alvo, a qual foi utilizada para comparar se o que foi planejado foi atendido na *Sprint*. Quando esta não era alcançada, se dava uma adaptação para que a próxima *Sprint* contemplasse o que não foi atendido, somada as novas necessidades identificadas, procurando assim, entregar um incremento de valor ao cliente alinhado com a estratégia da empresa. Além disso, talvez um dos pontos mais importantes do modelo, foi a utilização dos ciclos de *Kata* de *Coaching* no momento da Retrospectiva da *Sprint*, a fim de compartilhar e registrar o aprendizado consolidado pelo time no *Storyboard* do mesmo, minimizando assim esta lacuna encontrada no *Scrum*.

Assim, independente das adequações e alterações que ocorreram ao longo do caminho, como por exemplo as adaptações das Condições-Alvo, e a mudança dos registros do *Storyboard* do time, da planilha digital para o RTC, isso pode comprovar que o modelo pode ser uma referência para implantação e utilização das rotinas *Toyota Kata* e *Scrum* de forma integrada, especificamente nesse tipo de contexto pesquisado.

Por fim, o modelo conseguiu atender, conforme visto neste capítulo, todas as três principais barreiras e oportunidades na implantação das rotinas *Toyota Kata* encontradas na literatura, além de contemplar também as lacunas encontradas no *Scrum*.

5 CONCLUSÕES

Este capítulo tem por objetivo apresentar as conclusões deste estudo, bem como, suas dificuldades e recomendações para trabalhos futuros.

Este estudo teve por objetivo desenvolver um modelo para implantação das rotinas *Toyota Kata* integradas ao *Scrum* no processo de desenvolvimento de *software*, para assim facilitar a implantação das rotinas, bem como, reduzir as lacunas encontradas no *Scrum*.

Para tal, o estudo foi dividido em linhas gerais em três grandes etapas, sendo elas: Revisão Bibliográfica, Proposição do Modelo e Pesquisa-ação.

A Revisão Bibliográfica foi separada em dois momentos, sendo um deles o de Mapeamento Sistemático da Literatura, utilizado para sustentar de forma geral o estudo, bem como, identificar as lacunas encontradas no *Scrum*, sendo este um dos objetivos específicos. Além disso, uma Revisão Bibliográfica Sistemática (RBS), a qual estava alinhada diretamente ao objetivo específico de identificar as principais oportunidades e barreiras encontradas na implantação das rotinas *Toyota Kata*.

Com a RBS pode-se constatar as seguintes barreiras na implantação das rotinas *Toyota Kata*: a) falta de direção e preparação do ambiente (sensibilização). Esta barreira foi contemplada utilizando a etapa da criação e/ou compartilhamento do *Hoshin Kanri* e a sensibilização do modelo; b) falta de mapa de fluxo de valor. Esta barreira foi contemplada com a etapa de Mapeamento do Fluxo de Valor, resultando assim no Mapa de Fluxo de Valor do estado atual; e por fim c) a falta de reuniões de *Kata* de *Coaching* constantes. Esta por sua vez foi contemplada se realizando diariamente as reuniões de *Kata* Diária.

Da mesma forma, a RBS auxiliou o modelo a contemplar as três principais oportunidades identificadas na literatura, sendo elas: ciclos curtos; compartilhamento de conhecimento; e adaptabilidade. Estas oportunidades foram atendidas respectivamente pelas seguintes ações: *Sprints* com duração de 15 dias; *Kata* Diária e Retrospectiva da *Sprint* registrando por meio do *Storyboard*, com intuito de compartilhar o conhecimento durante e após terminada a *Sprint*; Revisão e Retrospectiva da *Sprint*, utilizando o *Storyboard* a fim de verificar o planejado com o realizado, e assim, adaptar a Condição-Alvo futura, e seus respectivos itens do *Backlog*, bem como, a posterior execução dos ciclos PDCA.

O Mapeamento Sistemático da Literatura por sua vez, além de sustentar o embasamento em linhas gerais do estudo, também pode

contribuir na busca pelas lacunas encontradas na utilização do *Scrum*, sendo elas: a) Visão de produto não considera alinhamento entre a estratégia e a operação; b) Confusão na visão devido a diversos pontos de vista; c) Não leva em conta dados, fatos e processo para o estabelecimento de objetivos, tomada de decisão e posterior comprovação dos resultados; d) Não utiliza método científico para desenvolvimento do incremento, não aborda as *User Stories* como sendo “obstáculos”; e) Não compartilha conhecimento e aprendizado de forma estruturada; f) Falta de utilização de dados, fatos e processos para comprovar os resultados e garantir a entrega de valor; g) Falta de entregas alinhadas a estratégia; h) Falta de método estruturado para compartilhamento de conhecimento e aprendizado.

Cada uma dessas lacunas foram contempladas por alguma etapa da de pelo menos uma das rotinas *Toyota Kata*, conforme visto na proposição do modelo, resultando nos seguintes benefícios: a) A utilização do *Hoshin Kanri* serviu de base para todo o alinhamento entre a estratégia e a operação, deixando claro a direção a qual todos precisavam seguir; b) A utilização tanto do MFV, quanto do A3, deixou claro a visão sistêmica dos desperdícios e dos problemas encontrados, além de tornar a tomada de decisão mais assertiva por utilizar dados, fatos e processo, tanto para o estabelecimento dos objetivos, quanto para sua comprovação ou não de atendimento; c) A utilização dos ciclos de execução de *Kata* de *Coaching*, bem como do PDCA, proporcionou o uso de um método científico para experimentar os obstáculos em direção a Condição-Alvo. Isso ajudou não só em uma forma estruturada para ultrapassar os obstáculos, como também na constante criação e compartilhamento, e no aprendizado individual dos integrantes do Time; d) A utilização de uma Condição-Alvo ajuda a comprovar se um incremento está tanto alinhado com a estratégia, quanto se o que foi realizado, está de acordo com o planejado, já que a mesma se baseou em fatos e dados.

Por fim, no momento da Pesquisa-ação, a verificação do comportamento do modelo pode explicitar outros benefícios esperados pelo objetivo deste trabalho. O primeiro deles foi o de possibilitar uma visão mais sistêmica e com isso um melhor alinhamento entre estratégia e operação. Sem a utilização dos *Hoshins Kanri*, bem como do MFV e do A3, o Time desenvolve diversas iniciativas desalinhadas com as expectativas dos clientes e da empresa. A partir do uso destas ferramentas, pode-se alinhar melhor estas expectativas, além de visualizar melhor por meio de dados e fatos onde estavam os desperdícios no processo, bem como, quais as principais causas que influenciavam nos problemas. Isso

trouxe não só a visão sistêmica, como também a criação e compartilhamento do conhecimento, além de uma assertividade e produtividade maior no momento de resolver os problemas, pois estes foram analisados de forma estruturada e com profundidade.

O segundo deles, e talvez o mais importante, foi o compartilhamento diário de conhecimento. Isso se deu nas reuniões de *Kata* Diária. Esse é um dos diferenciais da utilização dos ciclos de execução de *Kata* de *Coaching*, ao invés da *Scrum* Diária. Nos ciclos de *Kata* de *Coaching*, o intuito principal é justamente o aprendizado. Como o aprendizado foi compartilhado diariamente com todo time, isso acabou por proporcionar o compartilhamento de conhecimento, benefício esse não encontrado na *Scrum* Diária.

Além disso, o modelo também pode proporcionar o maior engajamento do Time, pois anteriormente as pessoas se mostravam mais isoladas e distantes. Após a utilização da rotina de *Kata* de *Coaching*, tanto nas diárias, quanto nas retrospectivas, todo o Time se mostrou mais interessado em resolver os problemas uns dos outros, bem como, em eliminar conjuntamente os obstáculos e os impedimentos encontrados.

Desta forma, como diferencial e originalidade o modelo proposto apresenta uma visão integrada das rotinas *Toyota Kata* com o *Scrum*. Esta integração se dá devido ao fato do *Scrum* possuir algumas lacunas, das quais as rotinas *Toyota Kata* puderam diminuir ou mesmo erradicar, conforme foi mostrado no decorrer deste estudo. Isso possibilitou com que as rotinas *Toyota Kata* fossem implantadas de uma forma mais natural, pois integradas ao *Scrum*, o método já era de conhecimento do Time. Como essa integração não gerou mudanças radicais no dia a dia de trabalho, as rotinas puderam ser implantadas sem muitas dificuldades.

Embora tenham sido encontradas poucas dificuldades, faz-se necessário relatar quais foram identificadas por este estudo, sendo elas respectivamente: a falta de conhecimento sobre as rotinas *Toyota Kata*, o que acaba gerando dúvidas e confusão na implantação do modelo; como o registro dos ciclos PDCA foram realizados de forma digital, os participantes principalmente no começo, se esqueciam de fazê-lo no momento adequado; a dificuldade de compreender e registrar os campos dos ciclos PDCA da forma correta; o pouco tempo de sensibilização do modelo gerou dúvidas e confusões no momento da implantação.

Com base no exposto, como sugestão para trabalhos futuros pode ser interessante uma pesquisa mais profunda para verificar se a utilização do A3 se faz necessária em qualquer contexto, ou se o nível do MFV já é o suficiente para compreender uma visão sistêmica e auxiliar na criação posteriormente do *Backlog* do Produto. Além disso, a aplicação do

modelo em uma escala maior também é passível de estudo, já que a verificação do comportamento do modelo se deu em apenas um time, com a participação de aproximadamente vinte e cinco pessoas. Por fim, pode ser contemplado também não só times que desenvolvam *software*, como foi o caso, mas também qualquer área da organização que utilize o *Scrum* como método base para gestão de projetos.

Diante de tudo que foi exposto acima, espera-se que o modelo proposto possa ser utilizado como referência a fim de promover a integração entre as rotinas *Toyota Kata* e o *Scrum* de forma natural, e com isso um se beneficiar do potencial do outro, gerando assim melhores resultados tanto para as empresas que o aplicam, quanto para seus clientes.

REFERÊNCIAS

- ABRAHAMSSON, P. et al. **Agile software development methods: Reviews and Analysis**. 2002. Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf>>. Acesso em: 10 set. 2017.
- AKHTAR, M. J.; AHSAN, A.; SADIQ, W. Z. Scrum adoption, acceptance and implementation (a case study of barriers in Pakistan's IT industry and mandatory improvements). In: International Conference on Industrial Engineering and Engineering Management, 17., 2010, Xiamen. **Proceedings...** Xiamen, China: IEEE, 2010. p. 458-461.
- BECK, K. **Extreme programming explained: embrace change**. Boston: Addison-Wesley, 1999.
- BOEHM, B.; TURNER, R. **Balancing agility and discipline: a guide for the perplexed**. Boston: Addison-Wesley, 2003.
- CASTEN, M. H. et al. Construction *Kata*: Adapting *Toyota Kata* to a *lean* construction project production system. In: ANNUAL CONFERENCE OF THE INTERNATIONAL GROUP FOR *LEAN* CONSTRUCTION, 21., 2013, Fortaleza. **Proceedings...** Fortaleza, Brazil: IGLC 2013. Disponível em: <<https://iglcstorage.blob.core.windows.net/papers/attachment-2fb5fff9-ee10-4589-8248-aefa2ae13a5b.pdf>>. Acesso em: 10 set. 2017.
- CHIN, G. **Agile project management: how to succeed in the face of changing project requirements**. New York: Amacom, 2004.
- COAD, P.; LEFEBVRE, E.; DE LUCA, J. **Java modeling in color with UML: enterprise components and process**. New Jersey: Prentice Hall, 1999.
- COCKBURN, A. **Agile software development**. Boston: Addison Wesley, 2002.
- COOK, Deborah J.; MULROW, Cynthia D.; HAYNES, R. Brian. Systematic reviews: synthesis of best evidence for clinical decisions. **Annals of internal medicine**, v. 126, n. 5, p. 376-380, 1997.

COOPER, Harris M. **Synthesizing research**: a guide for literature reviews. Thousand Oaks, CA, US: Sage, 1998.

CONFORTO, E. C.; AMARAL, D. C.; SILVA, S. L. Roteiro para revisão bibliográfica sistemática: aplicação no desenvolvimento de produtos e gerenciamento de projetos. In: CONGRESSO BRASILEIRO DE GESTÃO DE DESENVOLVIMENTO DE PRODUTO, 8., 2011, Porto Alegre. **Anais...** Porto Alegre: CBGDP, 2011. p. 1 - 12.

COHN, M. **Desenvolvimento de software com Scrum**: aplicando métodos ágeis com sucesso. Porto Alegre: Bookman, 2011.

DECARLO, D. **Extreme project management**. California: Jossey Bass, 2004.

DEMING, W. E. Out of the crisis. Massachusetts Institute of Technology. **Center for advanced engineering study, Cambridge, MA**, v. 510, p. 419-425, 1986.

DENNIS, P. **Produção lean simplificada**: um guia para entender o sistema de produção mais poderoso do mundo. 2. ed. Porto Alegre: Bookman, 2008.

EHNI, M.; KERSTEN, W. *Toyota Kata*: empowering employees for target-oriented improvement - a best practice approach. In: INTERNATIONAL CONFERENCE OF LOGISTICS, 20., 2015, Hamburgo, **Proceedings...** Hamburgo: HICL, 2015. p. 175.

ENGEL, G. I. Pesquisa-ação. **Educar em Revista**, n. 16, p. 181-191, 2000.

FOWLER, M. **The new methodology**. 2005. Disponível em <<http://www.martinfowler.com/articles/newMethodology.html>>. Acesso em: 25 ago. 2006.

GHAYYUR, S. A. et al. A Systematic Literature Review of Success Factors and Barriers of Agile *Software* Development. **International Journal of Advanced Computer Science and Applications (IJACSA)**, v. 9, n. 3, 2018.

HAMED, A. M. M.; ABUSHAMA, H. Popular agile approaches in *software* development: review and analysis. In: INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONICS ENGINEERING (ICCEEE), 8., 2013. Khartoum. **Proceedings...** Khartoum: IEEE, 2013. p. 160-166.

HIGHSMITH, J. **Adaptive software development**. New York: Dorset House, 2000.

HIGHSMITH, J. **Agile project management**: creating innovative products. Boston: Addison-Wesley, 2004.

HUMMEL, M. State-of-the-art: A systematic literature review on agile information systems development. In: Hawaii International Conference on System Sciences, 47., 2014, Washington. **Proceedings...** Washington: IEEE, 2014. p. 4712-4721.

INDUSTRY WEEK. **Everybodys jumping on the lean bandwagon but many are being taken for a ride**, 2008. Disponível em: <<https://www.industryweek.com/companies-amp-executives/everybodys-jumping-lean-bandwagon-many-are-being-taken-ride>>. Acesso em: 17 dez. 2017.

JONES, D. T.; WOMACK, J. P. **A mentalidade enxuta nas empresas**: elimine o desperdício e crie riqueza. Rio de Janeiro: Campos Editora, 2004.

LIKER, J. K.; MEIER, D. **Toyota talent**: Developing Your People the Toyota Way. [S.l.]: McGraw-Hill Education, 2007.

LÓPEZ-MARTÍNEZ, Janeth et al. Problems in the adoption of Agile-Scrum methodologies: A systematic literature review. In: International Conference in *Software* Engineering Research and Innovation (CONISOFT), 4., 2016, Puebla. **Proceedings...** Puebla, México: IEEE, 2016. p. 141-148.

MENDELEY. **Reference manager and academic social network**. 2018. Disponível em: <<https://www.mendeley.com/newsfeed/#group:8c3b80e5-6c19-33f4-a057-88279c299459>>. Acesso em: 22 maio 2018.

MICHELS, E. Design thinking e scrum: complementando a geração de ideias e alternativas para os projetos inovadores de TIC no Instituto Senai de Tecnologia em Automação e TIC. In: 12 SEMINARIO DE GERENCIAMENTO DE PROJETOS, 12., 2015, Porto Alegre. **Anais eletrônico...** Porto Alegre: PMI-RS, 2015.

MANIFESTO AGIL. **Manifesto para o desenvolvimento ágil de software**. 2001. Disponível em: <<http://www.manifestoagil.com.br>>. Acesso em: Acesso em: 22 maio 2018.

MANIFESTO AGIL. **Princípios por trás do manifesto ágil**. [2018]. Disponível em: <http://www.manifestoagil.com.br/principios.html>. Acesso em: Acesso em: 22 maio 2018.

NONAKA, I.; TAKEUCHI, H. **Criação de conhecimento na empresa**. Rio de Janeiro: Elsevier Brasil, 1997.

OSONO, E; SHIMIZU, N; TAKEUCHI, H. **Relatório Toyota**. São Paulo: Ediouro, 2008.

PALMER, S. R.; FELSING, J. M. **A practical guide to feature-driven development**. New Jersey: Prentice Hall, 2002.

PEREIRA, J. C. **Aplicação do design thinking integrado com métodos ágeis na gestão de projetos de software**. 2018. 152 f. Dissertação (Mestrado em Gestão de Projetos) – Programa de Mestrado Profissional em Administração: Gestão De Projetos, Universidade Nove de Julho, São Paulo, 2018.

PRIKLADNICKI, R.; WILLI, R.; MILANI, F. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman Editora, 2014.

RAUF, A.; ALGHAFEEES, M. Gap Analysis between State of Practice and State of Art Practices in Agile *Software* Development. In: AGILE CONFERENCE (AGILE), 2015, Washington. **Proceedings...** Washington: IEEE, 2015. p. 102-106.

REVEROL, J. Creating an Adaptable Workforce: Using the *Coaching Kata* for Enhanced Environmental Performance. **Environmental Quality Management**, v. 22, n. 2, p. 19-31, 2012.

ROTHER, M. ***Toyota Kata***: gerenciando pessoas para a melhoria, adaptabilidade e resultados excepcionais. Porto Alegre: Bookman, 2010.

ROTHER, M. **The *Toyota Kata Practice Guide***: Practicing scientific thinking skills for superior results in 20 minutes a day. New York: McGraw-Hill Education, 2018.

ROTHER, M.; AULINGER, G. ***Toyota Kata culture***: building organizational capability and mindset through *kata coaching*. New York: McGraw Hill Professional, 2017.

Royce, W. W. Managing the development of large *software* systems: concepts and techniques”. In: WESTERN ELECTRONIC SHOW AND CONVENTION (WESCON), 1., 1970, Los Angeles. **Proceedings...** Los Angeles: WESCON, 1970. p. 328-338.

SANTOS, N. et al. Modeling in agile *software* development: decomposing use cases towards logical architecture design. In: INTERNATIONAL CONFERENCE, PROFES, 19., 2018, Wolfsburg, Germany. **Proceedings...** Wolfsburg, Germany: Clausthal University of Technology, 2018. p. 396-408.

SCRUN.ORG. **What is Scrum?** 2018. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>. Acesso em: 18 ago. 2018.

SHARMA, S.; HASTEER, N. A comprehensive study on state of Scrum development. In: INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION AND AUTOMATION (ICCCA), 2016, Greater Noida, India. **Proceedings...** Greater Noida, India: IEEE, 2016. p. 867-872.

SUTHERLAND, J.; SCHWABER K. **Scrum guide**. 2017. Disponível em <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 18 ago. 2018.

SOLTERO, C. Creating an adaptable workforce: Lean training and *coaching* for improved environmental performance. **Environmental Quality Management**, v. 21, n. 1, p. 9–22, set. 2011.

SOLTERO, C. Rediscovering the *kata* way. **Industrial Engineer**, v. 44, n. 11, p. 28-33, 2012.

SOLTERO, C. Creating an adaptable workforce: Using the improvement *Kata* for enhanced environmental performance. **Environmental Quality Management**, v. 21, n. 3, p. 47-56, 2012.

TAKEUCHI, H.; NONAKA, I. The new new product development game. **Harvard business review**, v. 64, n. 1, p. 137-146, 1986.

TILLMANN, P.; BALLARD, G.; TOMMELEIN, I. D. A Mentoring approach to implement lean construction. In: INTERNATIONAL GROUP FOR LEAN CONSTRUCTION, 22., 2014, Oslo, Norway. **Proceedings...** Oslo, Norway: Lean Construction, 2014. p. 1283–1293.

TOIVONEN, T. Continuous innovation: combining *Toyota Kata* and triz for sustained innovation. **Procedia Engineering**, v. 131, p. 963-974, 2015.

TRIPP, D. Pesquisa-ação: uma introdução metodológica. **Educação e pesquisa**, v. 31, n. 3, p. 443-466, 2005.

VILLALBA-DIEZ, J.; ORDIERES-MERÉ, J.; RUBIO-VALDEHITA, S. Lean learning patterns. (CPD) n A vs. KATA. **Procedia CIRP**, v. 54, p. 147-151, 2016.

WOMACK, J. P.; JONES, D. T. **Machine that changed the world**. London: Simon & Schuster UK, 1990.

VERSIONONE AGILE MADE EASIER. **Agile methods and practices**. ANNUAL STATE OF AGILE DEVELOPMENT SURVEY, 11th. [2017]. Disponível em: <<http://www.agile247.pl/wp-content/uploads/2017/04/versionone-11th-annual-state-of-agile-report.pdf>>. Acesso em: 20 set. 2017.

APENDICE A – Mapeamento Sistemático

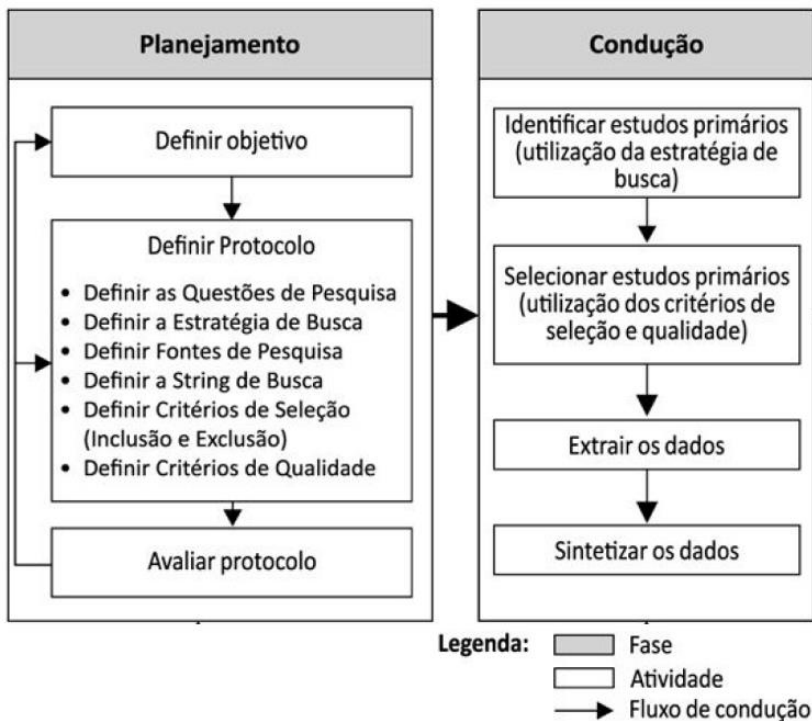
1 MAPEAMENTO SISTEMÁTICO

O Mapeamento Sistemático (MS), tem por objetivo identificar e classificar estudos que se relacionam com um tópico específico de pesquisa (KITCHENHAM; CHARTERS, 2007).

Desta forma, o MS é capaz de gerar resultados que possibilitem a identificação de lacunas na área ou tópico pesquisado, e com isso tornar claro o caminho para possíveis trabalhos futuros (KITCHENHAM; CHARTERS, 2007; PETERSEN et al., 2008; KITCHENHAM et al., 2010).

Para tal, o MS segue uma estrutura conforme é mostrada na Figura 54.

Figura 54 – Estrutura do Mapeamento Sistemático



Fonte: Adaptado de Nakagawa et al. (2017).

1.1 DEFINIR OBJETIVO

O objetivo levantado para o MS foi o seguinte: Identificar estudos que exponham as lacunas e oportunidades do *Scrum* aplicado no desenvolvimento de *software*.

1.2 Definir as Questões de Pesquisa

Como um MS diferencia em partes de uma Revisão Sistemática, o tópico a ser abordado na pesquisa para o MS foi: lacunas e oportunidades do *Scrum*, especialmente no desenvolvimento de *software*.

1.3 Definir a Estratégia de Busca

A estratégia de busca definida para o MS contempla as seguintes atividades: definição das fontes de pesquisa; definição dos métodos de busca e descrição da aplicação dos métodos de busca.

Desta forma, as fontes as quais serão feitas as buscas foram: *Scopus* e *Web of Science* (WOS). Os métodos de busca por sua vez foram a busca automática, e o *snowballing*. Por fim, a *String* de busca definida para o MS conforme o objetivo do mesmo e o tópico de pesquisa foi a seguinte: (*scrum*) AND (*adoption OR implementation OR deployment*) AND (*barriers OR difficulties OR obstacles*) AND ("*software development*").

1.4 Definir Critérios de Seleção

Com base no objetivo do MS, e visando encontrar estudos que contemplem tanto a parte teórica a respeito do *Scrum*, quanto a parte empírica das lacunas na sua aplicação, os critérios para a inclusão de estudos a princípio estão voltados para revisões tradicionais, pesquisação e estudos de caso.

Estes estudos são tanto de periódicos, conferências ou livros conceituados e referenciados nos temas. E devem contemplar os seguintes questionamentos:

1. O estudo utiliza ou implanta o *Scrum* para a gestão ágil de projetos; e
2. O estudo relata dificuldades, problemas ou lacunas na implantação do *Scrum*; e
3. O estudo foi desenvolvido com o foco no desenvolvimento do *software*.

Da mesma forma, os critérios para exclusão dos estudos contemplam as seguintes características:

1. O estudo não relata o *Scrum* como método de gestão ágil de projetos.
2. O estudo não reporta as dificuldades, lacunas ou problemas encontrados na implantação do *Scrum*.
3. O estudo é uma versão anterior de um estudo mais completo sobre a mesma pesquisa.
4. O estudo é a descrição de um curso, editorial, resumo de palestra, workshop ou tutorial;
5. O estudo não é focado no desenvolvimento de *software*.

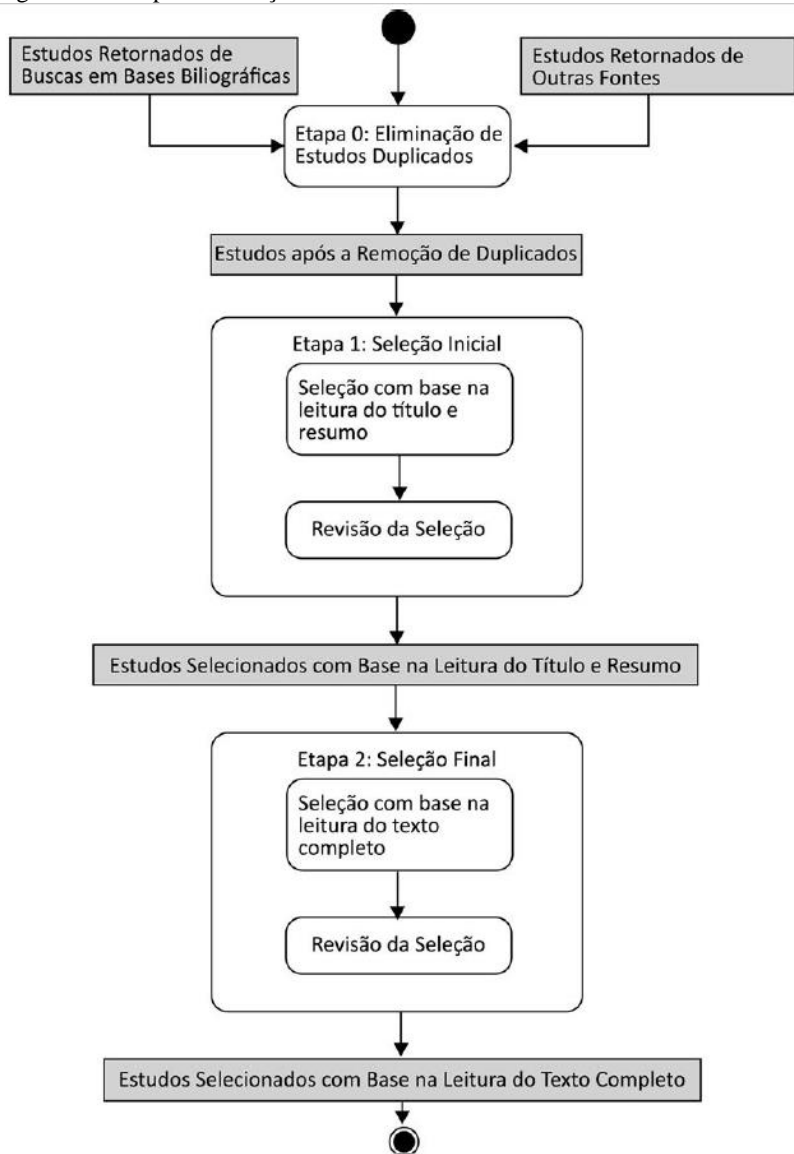
1.5 Definir Critérios de Qualidade

A fim de não correr o risco de introduzir um viés ao MS, optou-se nesse ponto por não incluir nenhum critério de qualidade ao mesmo.

1.6 Condução

Nesta fase do MS optou-se por trazer todas as suas atividades de forma integrada no mesmo tópico. Para a identificação e seleção dos estudos primários foi seguido o processo conforme mostrado na Figura 55.

Figura 55 – Etapas de Seleção dos Estudos



Fonte: Nakagawa et al. (2017).

Desta forma, o primeiro passo foi realizar a busca de estudos nas bases bibliográficas citadas anteriormente, bem como, utilizando snowballing. A busca com a *String (scrum) AND (adoption OR implementation OR deployment) AND (barriers OR difficulties OR obstacles) AND ("software development")* resultou em uma quantidade de 72 estudos na *Scopus* e 12 na *WOS*, totalizando assim 84 estudos.

O próximo passo foi passar pela Etapa 0 (zero), a qual realiza a eliminação dos artigos duplicados, resultando assim em 80 estudos restantes.

O passo seguinte na condução foi a realização da Etapa 1 (um), a qual visa a seleção para a leitura do título e resumo e posteriormente a revisão dessa seleção, a fim de confirmar se os estudos selecionados após a leitura estão de acordo com os critérios definidos. Desta forma, após a realização da Etapa 1 restaram então 18 estudos aderentes aos critérios de exclusão estabelecidos.

Por fim, foi realizada a Etapa 2, a qual foi responsável pela seleção dos estudos para leitura completa dos mesmos e posterior revisão. O resultado final de estudos selecionados para o MS totalizou então 7 trabalhos.

Todos os 84 estudos foram registrados em uma planilha eletrônica, bem como em uma ferramenta para análise (Mendeley, 2018), os quais foram separados conforme a etapa adequada que se encontram, e com as respectivas informações registradas conforme sugerido por Nakagawa et al. (2017), caracterizando assim a etapa de extração de dados.

Com base nos resultados encontrados nos 8 trabalhos restantes do procedimento de seleção dos estudos, chegou-se então à síntese relatada no Quadro 7.

Quadro 7 – Lacunas e Oportunidades de Melhoria do *Scrum*

(Continua)

Passo do <i>Scrum</i>	Autor	Lacunas	Oportunidades
<i>Backlog</i> do Produto	Akhtar, Ahsan e Sadiq (2010); Hamed e Abushama (2013); Hummel (2014); Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Visão de produto não considera alinhamento entre a estratégia e a operação	Alinhar a estratégia com a operação
Planejamento da <i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Confusão na visão devido a diversos pontos de vista	Compreender o estado atual de forma sistêmica por meio de dados, fatos e processos e não por opiniões e pontos de vista
<i>Backlog</i> da <i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Não leva em conta dados, fatos e processo para o estabelecimento de objetivos, tomada de decisão e posterior comprovação dos resultados.	Possuir um objetivo sistêmico comum, baseado em dados, fatos e em cima de processos, e não em pontos de vista
<i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Não utiliza método científico para desenvolvimento do incremento, não aborda as <i>User Stories</i> como sendo “obstáculos”	Utilização do PDCA

Quadro 7 – Lacunas e Oportunidades de Melhoria do *Scrum*

			Conclusão)
Passo do <i>Scrum</i>	Autor	Lacunas	Oportunidades
<i>Scrum</i> Diária	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Não compartilha conhecimento e aprendizado de forma estruturada	Utilização de PDCA e uma abordagem rotineira para compartilhar conhecimento e aprendizado
Revisão da <i>Sprint</i>	Rauf e AlGhafees (2015); Sharma e Hasteer (2016); López-Martínez, et al. (2016); Ghayyur, et al. (2018)	Falta de utilização de dados, fatos e processos para comprovar os resultados e garantir a entrega de valor	Utilização de dados e fatos baseados em processos para comprovar se o que foi entregue está de acordo com o objetivo esperado
Incremento	Hamed e Abushama (2013); Rauf e AlGhafees (2015); Sharma e Hasteer (2016)	Falta de entregas alinhadas a estratégia	Alinhar a direção da estratégia com as funcionalidades entregues
Retrospectiva da <i>Sprint</i>	Akhtar, Ahsan e Sadiq (2010); Hamed e Abushama (2013); Hummel (2014); Rauf e AlGhafees (2015); Sharma e Hasteer (2016)	Falta de método estruturado para compartilhamento de conhecimento e aprendizado	Utilização de PDCA e uma abordagem rotineira para compartilhar conhecimento e aprendizado

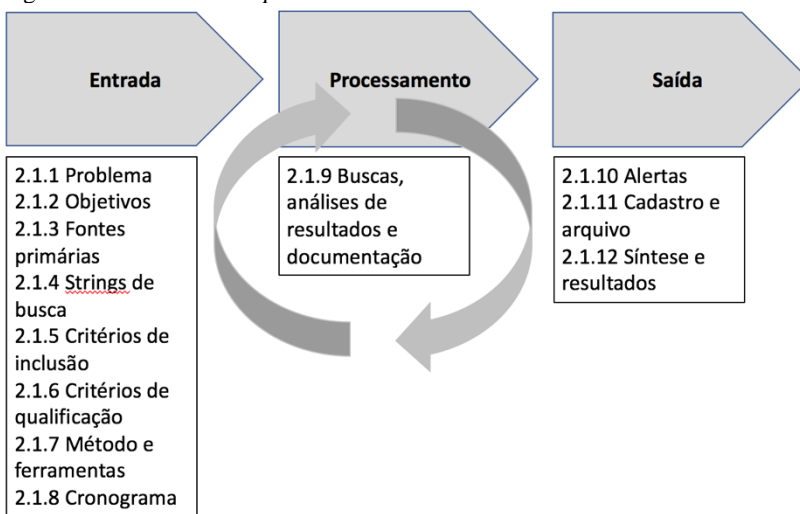
Fonte: Elaborado pelo autor.

APENDICE B – REVISÃO BIBLIOGRÁFICA SISTEMÁTICA

1 REVISÃO BIBLIOGRÁFICA SISTEMÁTICA

Esta sessão tem o intuito de detalhar de forma clara e objetiva como se deu todo o estudo realizado por meio da revisão sistemática bibliográfica (RBS). Conforme mostra a Figura 56, a RBS possui um *Roadmap* para o seu desenvolvimento, no qual será detalhado cada passo a fim de melhor esclarecer como são obtidos os resultados da mesma.

Figura 56 – RBS *Roadmap*



Fonte: Adaptado de Conforto, Amaral e Silva (2011).

1.1 PROBLEMA

Pergunta do estudo: Existe um modelo de implantação das rotinas *Toyota Kata* para o processo ágil de desenvolvimento de *software*?

1.2 OBJETIVOS

Os objetivos levantados para a RBS foram os seguintes:

- Identificar estudos que implantaram as rotinas *Toyota Kata* em casos empíricos.

- b) Identificar quais as principais barreiras e oportunidades na implantação das rotinas *Toyota Kata*.
- c) E por fim, verificar se estes estudos seguiram um modelo padronizado para a implantação e se foi utilizado algum tipo de método ou prática para auxiliar a mesma.

1.3 FONTES PRIMÁRIAS

As fontes primárias para este estudo foram as seguintes bases: *Scopus*, *Web of Science* e Portal de periódicos da CAPES (Capes). Os critérios para a escolha dessas bases foram os seguintes: *Scopus* – maior base de conhecimento indexada do mundo. WOS – uma das bases de conhecimento mais qualificadas do mundo. Portal de periódicos da CAPES – base de conhecimento brasileira, a fim de verificar se o problema a ser estudado já foi abordado anteriormente no Brasil. As pesquisas foram realizadas entre dezembro de 2016 e dezembro de 2017. Além disso, foram feitas buscas em bibliotecas físicas a fim de encontrar os livros citados pelos autores dos resultados preliminares.

1.4 STRINGS DE BUSCA

Após a análise prévia das fontes primárias e visando buscar resultados mais abrangentes que não restringissem os mesmos a poucos casos optou-se pela seguinte *String* de busca: *(kata) AND (improvement OR coaching) AND (software OR manufacturing OR healthcare)*.

Esta *String* se mostrou satisfatória justamente por relacionar os temas ao problema de pesquisa, e por não restringirem os resultados em um determinado contexto específico, já que os temas aqui abordados ainda são um tanto novos na literatura.

Não foi utilizada a expressão “*Toyota Kata*”, para que se pudesse verificar dentre todos os estudos encontrados, que se por acaso algum deles não utilizasse essa expressão, mas relatasse sobre estas rotinas, fosse encontrado na busca e posteriormente analisado.

1.5 CRITÉRIOS DE INCLUSÃO

Devido ao problema da pesquisa, e visando encontrar estudos que contemplem tanto a parte teórica a respeito das rotinas *Toyota Kata*, quanto a parte empírica de sua aplicação, os critérios para a inclusão de estudos a princípio estão voltados para revisões tradicionais, pesquisação e estudos de caso.

Estes estudos são tanto de periódicos, conferências ou livros conceituados e referenciados nos temas. Como os temas são novos ainda, optou-se por não incluir apenas estudos passados por revisão por pares. Os idiomas selecionados foram o inglês e o português.

Por fim, como critério de exclusão, optou-se por retirar os estudos que mencionavam o *Kata* vinculado a arte marcial, áreas medicinais, ou outras áreas que aplicam o *Kata* ancestral, e não as rotinas *Toyota Kata* utilizadas pela Toyota.

1.6 CRITÉRIOS DE QUALIFICAÇÃO

Quanto aos critérios de qualificação dos trabalhos, optou-se por identificar a quantidade de citações dos mesmos, o fator JCR, bem como o Qualis-Capes do periódico/evento de publicação a fim de identificar se estes eram mais ou menos relevantes. Esses itens constam no arquivo RBS para melhor identificação.

1.7 MÉTODO E FERRAMENTAS

O método de busca e as ferramentas utilizadas seguiram conforme uma adaptação de Conforto, Amaral e Silva (2011), para melhor representar os resultados. Foi utilizado apenas um arquivo a fim de facilitar todo o gerenciamento. Este arquivo foi separado por 3 planilhas a fim de caracterizar os filtros da RBS, da mesma forma como foi feito na ferramenta Mendeley (2018).

1.8 CRONOGRAMA

Para esta revisão foi estipulado um prazo de término de 12 meses, contados a partir de janeiro de 2017, com data de corte nos estudos até março de 2018.

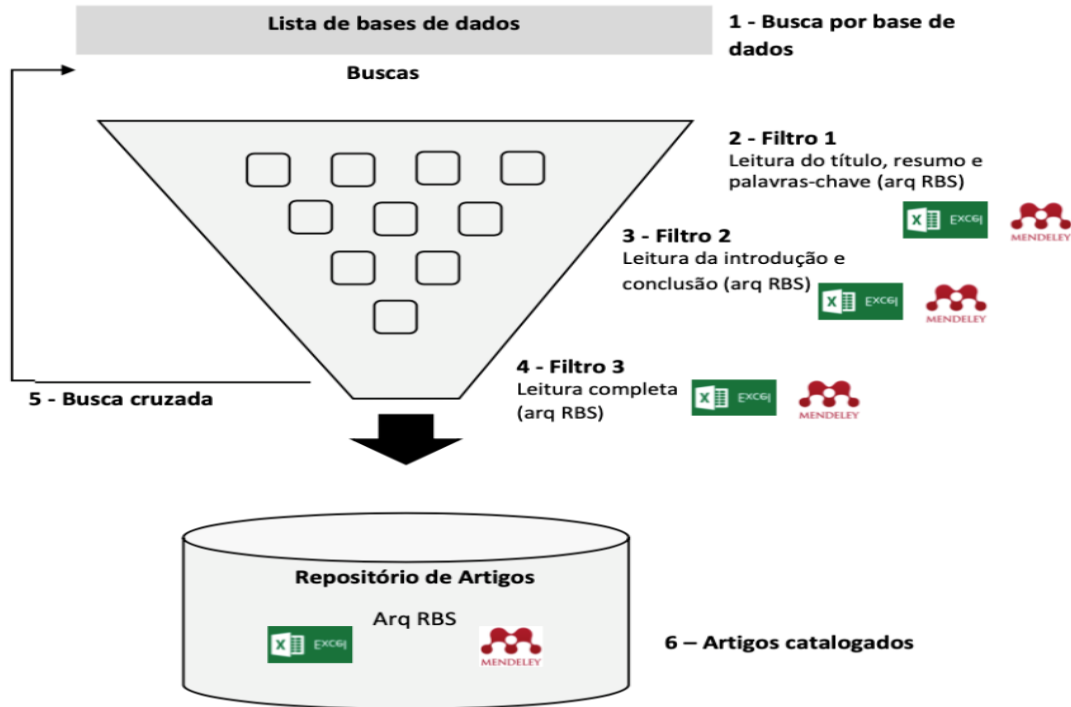
1.9 BUSCAS, ANÁLISES DOS RESULTADOS E DOCUMENTAÇÃO

As etapas de busca, análise dos resultados e documentação seguem um processo iterativo contendo 6 passos. Este processo pode ser melhor visto conforme mostra a Figura 57.

Na primeira etapa são realizadas as buscas, compreendendo os passos 1 e 5. Na segunda etapa, é realizada a leitura e análise dos resultados, ou seja, os filtros de leitura, passos 2, 3 e 4. Na terceira etapa,

é realizada a documentação e arquivamento dos estudos selecionados nos filtros, bem como os resultados das buscas e filtros de leitura, seguindo os passos 2, 3, 4 e 6.

Figura 57 – RBS



Fonte: Adaptado de Conforto, Amaral e Silva (2011).

Os dados registrados no arquivo RBS, desde o primeiro passo, até o passo 6 são importantes para refinar as buscas e posteriormente serão úteis para argumentação teórica e embasamento da síntese da teoria sobre o assunto pesquisado. Além disso, é útil para outros pesquisadores que irão pesquisar temas correlacionados, propondo um ponto inicial que reduz o tempo da RBS e possibilita melhor direcionamento e foco na pesquisa.

Conforme apresentado na Figura 8, a primeira etapa consiste na busca por periódico nas fontes primárias já identificadas. Para cada fonte primária é realizada uma busca, utilizando a *String* de busca. Os resultados são registrados utilizando o arquivo RBS, sendo que no mesmo são preenchidos os seguintes campos: ID; Nome da fonte primária; Citações; Ano do estudo; Autor(es) do estudo; Título do estudo; Periódico/Evento de publicação.

Na segunda etapa, o primeiro passo é a realização da leitura do título, resumo e palavras-chave. Desta forma os trabalhos que não passarem por esse filtro por não se adequarem ao objetivo do estudo são marcados como “inaptos” na coluna “filtro” do arquivo RBS.

O segundo passo desta etapa é a leitura da introdução e conclusão dos trabalhos. Da mesma forma que no passo anterior, se os mesmos não se adequarem ao objetivo do estudo, estes também são marcados como “inaptos”.

Por fim, o último passo é a leitura completa do artigo, e desta forma os trabalhos que se mostrarem adequados ao objetivo do estudo são marcados como aptos, já os demais são identificados da mesma forma como inaptos.

Durante todas as análises os estudos são registrados no arquivo RBS, e os que passarem pelo último filtro, são documentados posteriormente na ferramenta Mendeley (2018).

1.10 ALERTAS

Esta etapa consiste na inserção de alertas nos periódicos de interesse do estudo a fim de se manter atualizado nos temas relacionados ao trabalho, e assim, ter a possibilidade de manter o mesmo também atualizado conforme o tempo passa.

Desta forma, foram inseridos alertas nos periódicos relevantes aos temas aqui estudados que passaram pelos três filtros da RBS, para que sempre que um estudo relevante fosse identificado nestes periódicos, os mesmos fossem de conhecimento e possível estudo posterior.

1.11 CADASTRO E ARQUIVO

Nesta etapa, os estudos que passaram pelos três filtros da RBS foram incluídos em uma ferramenta de análise Mendeley (2018) para que os mesmos pudessem ser arquivados da forma mais adequada possível, a fim de garantir seus registros e sua rastreabilidade.

1.12 SÍNTESE E RESULTADOS

Esta etapa é a responsável por realizar uma síntese dos resultados encontrados na RBS, a fim de explicitar o conhecimento no que tange os temas abordados, e como estes estão sendo tratados na literatura de forma resumida.

Para isso a pesquisa foi desenvolvida primeiramente por meio de uma revisão bibliométrica orientada pela busca de evidências no tema principal deste trabalho, sendo ele as rotinas do *Toyota Kata*.

As palavras-chaves destacadas conforme já informado anteriormente e que melhor se destacam para este trabalho foram: “Kata”, “improvement”, “coaching” e “Toyota”.

Assim, a primeira pesquisa foi realizada na *Scopus*, *Web of Science* (WOS) e Capes no dia 03/12/2016 utilizando o seguinte filtro de palavras-chave: “Kata”. A amostra de resultados trouxe 1054 estudos na base *Scopus*, 624 na WOS e 40694 na Capes.

Após esse resultado, foram inseridas as seguintes palavras-chave para o prosseguimento da busca: “improvement” ou “coaching” ou “Toyota”. Com isso a mesma trouxe os seguintes resultados: 52 trabalhos na *Scopus*, 23 na WOS, e 2 na Capes, totalizando assim 77 artigos.

Por fim, foram inseridas as palavras-chave finais a fim de restringir o contexto as áreas de estudo que mais utilizam *Toyota Kata*, procurando não afunilar tanto – utilizando apenas a área de *software* –, pois desta forma, outros estudos que possam servir de alguma maneira, ficariam de fora da busca. Com isso, as palavras utilizadas foram: “software” ou “manufacturing” ou “healthcare”, resultando assim 13 estudos na *Scopus*, 2 na WOS e 0 na Capes, totalizando então 15 estudos.

Assim sendo, foi feito o upload dos “artigos” em uma ferramenta específica Mendeley (2018), para exclusão dos possíveis duplicados, restando assim 15 artigos para análise, pois nenhum estava duplicado. A partir de então foram realizados os filtros que sustentam a RBS.

Filtro 1: Análise do título, resumo e palavras-chave para identificar a conexão entre os temas desejados, restando nesta etapa apenas 15 artigos com a respectiva aderência ao objetivo desse estudo. O critério de

exclusão utilizado foi retirar os artigos que falavam de *Kata* como uma arte marcial, como por exemplo no karatê ou judô, bem como, estudos a respeito de áreas medicinais, e outras áreas, que não se referiam as rotinas da Toyota, pois este não é o objetivo deste estudo.

Filtro 2: Leitura da introdução e conclusão. Após a leitura complementar da introdução e conclusão foram retirados mais 2 artigos por não explicitarem de forma clara um processo de utilização das rotinas *Toyota Kata*.

Filtro 3: Por fim, ao passar pelas duas primeiras análises, a última etapa foi a leitura completa de cada artigo para confirmar a aderência aos temas pesquisados, e a partir deste último ponto foi realizado então o último descarte dos artigos, a fim de identificar quais dos mesmos atendiam ao objetivo do estudo, sendo que ao final restaram apenas 11 artigos. Nesta fase a intenção foi retirar os estudos que não explicitavam nenhum tipo de oportunidade ou barreira na implantação das rotinas *Toyota Kata*, ou não transpareciam a utilização de pelo menos uma das rotinas.

Diante dos resultados encontrados, o próximo passo para facilitar uma melhor análise, bem como, explicitar posteriormente uma melhor discussão, o Quadro 8 mostra de forma objetiva uma síntese da RBS no que tange os aspectos desejados por este estudo.

Quadro 8 – Síntese da RBS

(Continua)

Autor(es)	Ano	Modelo de implantação?	Oportunidades	Barreiras
Soltero, C.	2011	Nenhum	Adaptabilidade; Integração entre TWI e <i>Toyota Kata</i>	Falta de um mapeamento de fluxo de valor atual
Soltero, C.	2012	Nenhum	Fomentar uma cultura de inovação por toda organização; Abordar a melhoria contínua com viés na inovação para o cliente externo.	Falta de experimentação no cliente; Falta de um mapeamento de fluxo de valor atual
Soltero, C.	2012	Nenhum	Inspiração criativa; Processo padronizado de resolução de problemas; Melhoria nos padrões comportamentais; Impactos das melhorias obtidas; Ciclos curtos no uso do <i>Kata</i> de Melhoria	Falta de um mapeamento de fluxo de valor atual

Quadro 8 – Síntese da RBS

(Continuação)

Autor(es)	Ano	Modelo de implantação?	Oportunidades	Barreiras
Reverol, J.	2012	Nenhum	A implantação do <i>Kata de Coaching</i> pode facilitar a implantação de outros <i>Katas</i> ; O pré-trabalho pode significar o sucesso da implantação das rotinas <i>Toyota Kata</i> ; desenvolver colaboradores mais adaptáveis e valiosos para a organização	Preparar a estratégia da empresa e identificar a cadeia de valor antes de aplicar <i>Toyota Kata</i> ; Tempo adequado para <i>coach</i> e <i>aprendiz</i> trabalharem o <i>Kata de Coaching</i> ; não seguir as etapas das rotinas <i>Toyota Kata</i> como estabelecidas na literatura
Casten, et al.	2013	Nenhum	Compartilhamento de conhecimento; Rotina de aprendizado e melhoria constantes; Desenvolvimento de lideranças; Adaptabilidade	Preparação do ambiente na indústria da construção é diferente e até desprezada, comparada ao ambiente da manufatura; Necessidade de maior experimentação das rotinas <i>Toyota Kata</i>
Tillmann, P; Ballard, G; Tommelein, I.	2014	Nenhum	Ambiente colaborativo; Desenvolvimento de liderança; Criação de um ambiente com segurança psicológica	Falta de consenso da equipe na utilização de rotinas e métodos; Conflito de interesses entre os que apoiam e os que não apoiam a iniciativa

Quadro 8 – Síntese da RBS

(Continuação)

Autor(es)	Ano	Modelo de implantação?	Oportunidades	Barreiras
Toivonen, T.	2015	Nenhum	<p>Contrapõe os preconceitos cognitivos; Cria alinhamento na organização do nível estratégico ao nível operacional; Envolve os funcionários através de desafios comuns e sucessos frequentes em relação a ele; A abordagem comum para a melhoria e gestão facilita a colaboração eficaz em toda a organização; Implementação Fractal permite a utilização da inteligência coletiva das organizações para atividades de melhoria contínua; Um número maior de ideias mais criativas para resolver problemas; Uma abordagem sistemática para estender os esforços de melhoria aos aspectos intangíveis, que são cada vez mais importantes em muitas indústrias hoje; Uma abordagem sistemática para muitas das partes difíceis das rotinas <i>Toyota Kata</i>, como compreender a situação atual e escolher o desafio certo</p>	<p>Cria pressão sobre os treinadores que deveriam ser muito competentes no ensino do método para torná-lo acessível para todos na organização; A quantidade de trabalho para criar um nível sustentável de competência e cultura de apoio</p>

Quadro 8 – Síntese da RBS

(Conclusão)

Autor(es)	Ano	Modelo de implantação?	Oportunidades	Barreiras
Merguerian, et al.	2015	Nenhum	Aplicar <i>Toyota Kata</i> primeiramente nos processos que agregam valor; Automatizar a coleta e os processos para melhorar os índices expostos	Coleta de dados manual; Processos manuais; Nem todos colaboradores foram incluídos para a estimativa de custos
Iberle, K.	2015	Nenhum	Passos curtos nos ciclos PDCA se encaixam melhor na carga de trabalho das pessoas; Reuniões de <i>coaching</i> de 2 a 3 vezes por semana; As perguntas no Kata de <i>coaching</i> empurram o <i>coach</i> e o aprendiz na direção certa.	Baixa qualidade na formulação de hipóteses; Falta de consciência em relatar o aprendizado; Reuniões não serem curtas e em horário regular; Falta de reuniões regulares entre <i>coach</i> e aprendiz; Falta de verificação no <i>gemba</i> ; Reunião de <i>coaching</i> apenas uma vez na semana;
Ehni, M; Kersten, W.	2015	Rother (2010) e Bleicher (2011)	Definição da condição alvo; Utilização do <i>Hoshin Kanri</i> ; Utilização do mapa de fluxo de valor	
Villalba-Diez et al.	2016	Nenhum		Falta de utilização do PDCA de forma sistêmica

Fonte: Elaborado pelo autor.

Como resultados principais da RBS, pode-se identificar os três pontos primordiais para a análise e utilização deste estudo, sendo eles os modelos de implantação que os artigos da revisão utilizaram, as oportunidades que os mesmos relataram durante a implantação, bem como, as barreiras enfrentadas.

No entanto, com base na leitura destes 11 estudos resultantes da RBS, nenhum deles foi claramente explícito quanto ao modelo de implantação utilizado para as rotinas *Toyota Kata*. Desta forma pode-se presumir que o modelo utilizado foi o elucidado por Rother (2010), o qual será devidamente detalhado posteriormente.

Estes três pontos serão importantes para este estudo para que se aproveitem as oportunidades relatadas, minimizem as barreiras, bem como, os modelos explicitados sirvam de base para a construção do modelo de referência que será o resultado final deste trabalho, o qual será visto posteriormente.