UNIVERSIDADE FEEVALE

DOUGLAS NEVES CARLOS

MÉTRICAS EM PROCESSOS ÁGEIS: UM *FRAMEWORK* BASEADO EM *STORYTELLING* PARA VISUALIZAÇÕES

Novo Hamburgo

2020

DOUGLAS NEVES CARLOS

MÉTRICAS EM PROCESSOS ÁGEIS: UM *FRAMEWORK* BASEADO EM *STORYTELLING* PARA VISUALIZAÇÕES

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação pela Universidade Feevale

Orientador: Prof. Dra. Adriana Neves dos Reis

Novo Hamburgo

2020

RESUMO

A pesquisa trata da aplicação de métricas aderentes com metodologias ágeis. O tema está voltado para unir métricas objetivas com as complexidades dos ambientes ágeis de TI, e introduzir a aplicação de *storytelling* com dados para apresentar as visualizações gráficas das métricas de forma efetiva. Seguindo a metodologia DSR, o trabalho teve como objetivo desenvolver um *framework* para auxiliar a aplicação de métricas no processo de construção de software em ambientes ágeis. Baseado em métricas ágeis e lições de *storytelling* com dados para as visualizações, o artefato desenvolvido tem o propósito de apoiar as empresas que desejam promover melhorias no processo, compreender o contexto, analisar tendências, trabalhar com previsibilidade e aumentar a visibilidade do trabalho. Para avaliação da proposta foram feitas entrevistas com três profissionais do mercado de trabalho com experiência em metodologias ágeis. Apesar da pequena amostragem de entrevistados na avaliação e que existam melhorias a serem feitas no artefato, conclui-se que o *framework* é relevante para o atual mercado. Pois, vai de encontro com as necessidades de mercado e segue os valores e princípios das metodologias ágeis.

Palavras-chave: Métricas. Métodos Ágeis. Processo. *Storytelling.*

ABSTRACT

The research deals with the application of adherent metrics with agile methodologies. The theme is aimed at uniting objective metrics with the complexities of agile IT environments, and introducing the application of storytelling with data to present the graphical visualizations of the metrics effectively. Following the DSR methodology, the work aimed to develop a framework to assist the application of metrics in the process of building software in agile environments. Based on agile metrics and storytelling lessons with data for visualizations, the developed artifact aims to support companies that want to promote improvements in the process, understand the context, analyze trends, work with predictability and increase the visibility of the work. To evaluate the proposal, interviews were conducted with three professionals from the labor market with experience in agile methodologies. Despite the small sample of respondents in the evaluation and that there are improvements to be made in the artifact, it is concluded that the framework is relevant to the current market. Well, it meets the needs of the market and follows the values and principles of agile methodologies.

Keywords: Metrics. Agile Methods. Process. Storytelling.

LISTA DE ILUSTRAÇÕES

Figura 1 – Metodologia do trabalho 13

Figura 2 – Total de pontos entregues por iteração 21
Figura 3 – Cálculo do *Lead Time* através dos limites do processo 27
Figura 4 – Exemplo de cálculo do *Throughput* 27
Figura 5 – Mapa mental 33

Figura 6 – *Framework* proposto 42

Figura 7 – Exemplo de um gráfico de dispersão do *Lead Time* 47
Figura 8 – Exemplo de um gráfico do *Lead Time Breakdown* 47
Figura 9 – Exemplo de CFD 48
Figura 10 – Gráfico de *Throughput* semanal 48
Figura 11 – WIP 49
Figura 12 – Telas com *Lead Time* em períodos de tempo distintos 51
Figura 13 – Tela do *Throughput* 52
Figura 14 – Tela do CFD 52
Figura 15 – Tela do WIP 53
Figura 16 – Tela do WIP no CFD 53

LISTA DE QUADROS

Quadro 1 – Categorização das métricas 24
Quadro 2 – Requisitos do artefato 34
Quadro 3 – Requisitos do artefato 43
Quadro 4 – Perfil e repostas dissertativas do entrevistado 1 55
Quadro 5 – Perfil e repostas dissertativas do entrevistado 2 56
Quadro 6 – Perfil e repostas dissertativas do entrevistado 3 57
Quadro 7 – Média e desvio padrão das respostas dos entrevistados sobre a avaliação do *framework* 58

LISTA DE ABREVIATURAS E SIGLAS

CFD	*Cumulative Flow Diagram*
DSR	*Design Science Research*
IEEE	*Institute of Electrical and Electronics Engineers*
ISO	*International Organization for Standardization*
JSON	*JavaScript Object Notation*
RQ	Requisito
TDD	*Test Driven Development*
TI	Tecnologia da Informação
WIP	*Work In Progress*
XP	*eXtreme Programming*

LISTA DE SÍMBOLOS

λ	Lambda

SUMÁRIO

1 Introdução **10**
1.1 Objetivos do Trabalho 11
1.1.1 Objetivo geral 11
1.1.2 Objetivos específicos 11
1.2 Método do Trabalho 12
1.3 Estrutura do Trabalho 13

2 Embasamento Teórico **15**
2.1 Métodos Tradicionais e Ágeis 15
2.1.1 Manifesto Ágil 16
2.1.2 Métodos Ágeis 17
2.2 Métricas 20
2.2.1 Conceitos Gerais 20
2.2.2 Características de Métricas Ágeis 22
2.2.3 Classificações 23
2.2.3.1 Quantitativa/Qualitativa 23
2.2.3.2 Objetiva/Subjetiva 23
2.2.3.3 Categorias de Avaliação 24
2.2.4 Métricas de Processo 25
2.2.4.1 *Work In Progress* (WIP) 25
2.2.4.2 *Lead Time* 26
2.2.4.3 *Throughput* 27
2.2.5 Lei de *Little* 28
2.3 *Storytelling* com Dados 29
2.3.1 Contexto 29
2.3.2 Visual Eficaz 30
2.3.3 Saturação Visual 32
2.3.4 História 32
2.4 Considerações Finais 33

3 *Framework* Conceitual **34**
3.1 Requisitos do *framework* 34
3.2 Desenvolvimento do *framework* 35
3.2.1 Seleção das Métricas 35
3.2.2 Definição da Medição 37
3.2.3 Desenvolvimento das Visualizações 38

3.2.4 Desenvolvimento da Ferramenta 39
3.2.5 Melhoria Contínua 39
3.3 Visão Geral do *Framework* 40

4 Avaliação do *Framework* 43
4.1 Requisitos do *Framework* 43
4.2 Aplicação do *Framework* 43
4.2.1 Seleção das Métricas 44
4.2.2 Definição da Medição 46
4.2.3 Desenvolvimento das Visualizações 46
4.2.3.1 Lead Time 46
4.2.3.2 *Lead Time Breakdown* 46
4.2.3.3 Diagrama de fluxo cumulativo 47
4.2.3.4 Throughput 48
4.2.3.5 WIP 49
4.2.4 Desenvolvimento do Protótipo da Ferramenta 49
4.2.5 Melhoria Contínua 54
4.3 Entrevista 54
4.4 Perfil dos Entrevistados 54
4.5 Considerações Sobre a Avaliação 58

5 Conclusão 61

Referências 63

Apêndices 66

APÊNDICE A Roteiro da Entrevista 67

1 INTRODUÇÃO

O mercado de TI tem apresentado um movimento significativo de adoção de metodologias ágeis para organização de suas equipes (PHAM; PHAM, 2011). Essa tendência pode ser justificada pelo fato desta abordagem propiciar processos de desenvolvimento flexíveis, que permitem acompanhar o ritmo acelerado de mudanças que a construção de um sistema apresenta, pois trabalham com requisitos voláteis e focam na entrega de valor de negócio para o cliente ou usuário (SOMMERVILLE, 2011).

Hartmann e Dymond (2006) afirmam que apesar da popularização dos métodos ágeis, antigas práticas de monitoramento e controle dos projetos dificultam a avaliação do desempenho e não contribuem para a melhoria do processo. Sob o ponto de vista prático, conviver em um ambiente no qual clientes, usuários e equipe não possuem visibilidade sobre o tempo necessário para a conclusão de um determinado trabalho gera, muitas vezes, desconfiança, dúvida, ruído na comunicação e, principalmente, a sensação de que tudo poderia ser feito mais rápido. Assim, torna-se essencial estabelecer possíveis métricas, variáveis ou visualizações adequadas para analisar o progresso do trabalho (ALBINO, 2017).

Existem diversos tipos de métricas ágeis para auxiliar as equipes a compreender onde estão, e ajudar a comparar com o estado em que querem estar. O essencial é encontrar as métricas que agregam mais valor à evolução do time. Após isso, é importante dar visibilidade dos resultados das métricas para que as equipes possam acompanhá-las e melhorar a cada nova iteração (GOMES, 2013).

Dentro do contexto de dar visibilidade às métricas, para Knaflic (2017), uma visualização dos dados eficaz pode significar a diferença entre o sucesso e o fracasso na hora de interpretar e apresentar informações. O objetivo é fomentar as interações entre as pessoas, melhorar a comunicação, e aumentar o engajamento das pessoas para alcançar as metas.

Atualmente, para promover a comunicação efetiva com dados tem sido aplicada uma estratégia de visualização de dados chamada *storytelling* com dados. Essa estratégia vem sendo utilizada para apresentações de dados gerando visualizações que permitem contar histórias através de dados, mas as histórias não auxiliam apenas para apoiar a discussão e a tomada de decisões, possibilitam também a análise de processos (KOSARA; MACKINLAY, 2013).

A estratégia de *storytelling* com dados defende que existe uma história por trás das informações que precisa ser destacada visualmente e contextualmente. Suas principais lições são: entender o contexto (quem é o público-alvo e o que ele precisa saber); escolher

de um visual eficaz; evitar a saturação; e focalizar atenção do público-alvo (KNAFLIC, 2017).

Considerando este cenário, o tema do trabalho está voltado para auxiliar na aplicação de métricas objetivas em processos ágeis de desenvolvimento de *software*, almejando principalmente melhoria contínua e aumento da previsibilidade de entrega. Outro ponto do tema do trabalho é a apresentação das métricas de forma efetiva com visualizações aplicando *storytelling* com os dados.

Sendo assim, o presente trabalho tem como objetivo a elaboração de um *framework* conceitual para a apoiar o uso de métricas a serem empregadas nos processos ágeis de desenvolvimento, o qual ofereça um guia de como selecionar as métricas que melhor atendem às necessidades das empresas interessadas em monitorar o seu desempenho. A partir da aplicação do *framework*, pretende-se que as métricas tenham maior potencial para: promover melhorias no processo, compreender o contexto, analisar tendências, trabalhar com previsibilidade e aumentar a visibilidade do trabalho.

1.1 OBJETIVOS DO TRABALHO

1.1.1 Objetivo geral

Desenvolver um *framework* para auxiliar o uso de métricas no processo de construção de software em projetos com gerenciamento ágil, e a elaboração de visualizações das métricas na percepção de *storytelling* com dados.

1.1.2 Objetivos específicos

- Apontar as características necessárias para que uma métrica seja adequada aos métodos ágeis;
- Identificar um conjunto mínimo de métricas para monitorar a eficiência do processo de construção de software em ambientes que aplicam métodos ágeis;
- Apresentar as lições para aplicar *storytelling* com dados nas visualizações gráficas;
- Apontar os principais tipos de visualização de dados gráficos aplicando *storytelling* com dados;
- Propor um *framework* conceitual que oriente empresas a iniciar a utilização de métricas em processos ágeis;
- Avaliar por meio de entrevista com especialistas da área o *framework* aplicado no desenvolvimento de um protótipo de ferramenta web.

1.2 MÉTODO DO TRABALHO

É possível classificar este trabalho como uma pesquisa científica de natureza aplicada, porque segundo Prodanov e Freitas (2013), ela "objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos. Envolve verdades e interesses locais". Além disso, o trabalho tem caráter exploratório, dado que envolve um estudo sobre o tema como aporte teórico para a construção da proposta de *framework* conceitual.

Do ponto de vista dos procedimentos técnicos, é aplicada a metodologia *Design Science Research* (DSR). O propósito do DSR é o desenvolvimento de artefatos que permitam soluções satisfatórias a um determinado problema de caráter pragmático (LACERDA et al., 2013). Para Pimentel (2017) este artefato:

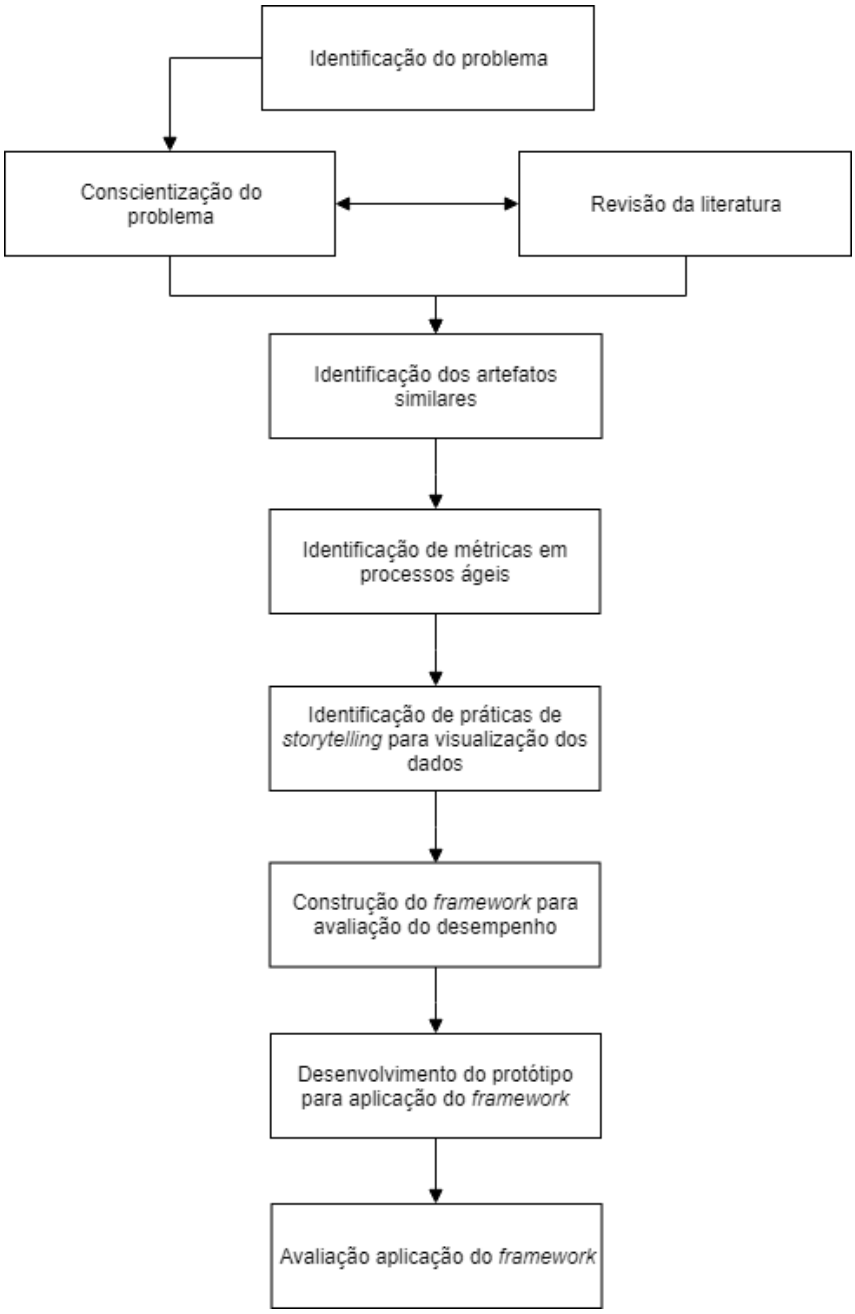
> é projetado para resolver algum problema num dado contexto a partir de conhecimentos e conjecturas sobre o mundo; e o uso do artefato, por sua vez, é o que possibilita avaliar as conjecturas que direcionaram seu desenvolvimento, o que nos possibilita produzir conhecimento sobre o mundo (ciência).

No seu processo, o DSR envolve a definição do problema, sugestões de solução, construção, validação e avaliação dos artefatos gerados pela pesquisa (DRESCH; LACERDA; ANTUNES, 2015). A partir disto, o método DSR foi escolhido porque é o adequado para pesquisas que exploram a construção de artefatos, como o *framework* proposto neste trabalho.

As etapas do trabalho e a sua direção, ilustrados na Figura 1, estão associados às etapas indicadas nas referências da literatura de DSR. As etapas também representam os objetivos específicos da pesquisa. Primeiramente o trabalho identifica o problema abordado pelo tema que é visualização de métricas ágeis. Em seguida é feita a conscientização do problema e revisão da literatura através de livros e artigos das áreas da metodologia ágil, métricas e *storytelling* com dados. A identificação dos artefatos similares é uma pesquisa com objetivo de encontrar trabalhos científicos que se propuseram a construir um material com proposta semelhante a este trabalho. A etapa de identificação de métricas em processos ágeis é efetuada com base na fundamentação teórica realizada. Da mesma maneira, a identificação de práticas de *storytelling* com dados é feita, aproveitando como principal fonte de informação a autora Knaflic (2017).

Com base nos estudos realizados, é construído a proposta de *framework* para auxiliar no uso de métricas em processos ágeis. O *framework* será composto por etapas que orientam como identificar quais métricas e práticas de *storytelling* aplicar. Após, o *framework* é avaliado por meio do desenvolvimento de um protótipo de ferramenta web e também por três profissionais do mercado através de entrevistas.

Figura 1: Metodologia do trabalho



Fonte: próprio autor

1.3 ESTRUTURA DO TRABALHO

O trabalho está estruturado em cinco capítulos. No Capítulo 1 foi apresentado a introdução ao tema do trabalho, os objetivos da pesquisa e o método para o desenvolvimento do trabalho. No Capítulo 2 consta a fundamentação teórica do trabalho, abordando metodologia de desenvolvimento ágil de *software*, métodos ágeis mais difundidos, conceitos gerais da medição de software, caraterísticas e categorização de métricas, identificação de métricas de processo e identificação de lições sobre *storytelling* com dados. No Capí-

tulo 3 é elaborada a proposta de *framework* construída a partir da pesquisa realizada. O Capítulo 4 possui a construção do protótipo seguindo as orientações do *framework*. Este capítulo ainda abrange as entrevistas com especialistas feitas para avaliar o *framework* proposto. Por fim, no Capítulo 5 são expostas as considerações finais do trabalho.

2 EMBASAMENTO TEÓRICO

A fundamentação teórica contida neste capítulo tem como objetivo elucidar o conteúdo pesquisado para a construção do *framework* para avaliação do desempenho do processo de construção de software em projetos com gerenciamento ágil. Nas próximas seções, são abordados a metodologia de desenvolvimento ágil, o Manifesto Ágil, alguns métodos ágeis, conceitos gerais sobre métricas, características e classificações de métricas, métricas de processo e lições de *storytelling* com dados.

2.1 MÉTODOS TRADICIONAIS E ÁGEIS

Os processos de software são complexos e, como todos os processos criativos, dependem de pessoas para tomar decisões e fazer julgamentos. Os processos têm evoluído de maneira a extraírem o melhor proveito dos recursos de uma organização. Normalmente os processos, ou métodos de trabalho, são classificados como tradicional ou ágil (SOMMERVILLE, 2011).

Como indica Sabbagh (2013), os métodos tradicionais são fortemente prescritivos, em geral, se caracterizam por serem processos sequenciais e sistemáticos de construção de produto, onde uma fase é iniciada apenas quando a anterior é finalizada. Outro foco dessa abordagem é o planejamento detalhado no início do projeto. Neste contexto tradicional, as mudanças são extremamente indesejadas, ao contrário dos processos ágeis. O método tradicional mais conhecido para o desenvolvimento de software é o modelo em cascata, ou *waterfall* (SABBAGH, 2013).

Com o desenvolvimento tecnológico e a globalização, as exigências em respostas rápidas às mudanças aumentaram. No mercado de empresas de médio e pequeno porte, os métodos tradicionais geravam uma grande sobrecarga, uma vez que o sequenciamento das etapas adiava a identificação e resposta às mudanças (SOMMERVILLE, 2011). Nesse contexto, na década de 90 iniciou-se um movimento, nomeado movimento "Ágil", formado por líderes e pesquisadores da área de desenvolvimento que surgiu em resposta aos métodos tradicionais (SABBAGH, 2013). O resultado do movimento ágil é o desenvolvimento de métodos, técnicas e ferramentas que apoiam uma evolução mais flexível de processo e produto, focando na qualidade do produto, no custo e na eficiência do projeto. Os métodos ágeis baseiam-se em uma abordagem incremental para o detalhamento do projeto, o desenvolvimento e a entrega de software (SOMMERVILLE, 2011).

Segundo Fowler (2005), as duas principais diferenças entre as metodologias ágeis e as tradicionais são:

- As metodologias ágeis são adaptativas, ao invés de previsíveis: as metodologias tradicionais tentam planejar uma grande parte do software em muitos detalhes com bastante antecedência. Por isto, são resistentes à mudança. Já nas metodologias ágeis, as mudanças de requisitos durante o desenvolvimento são bem-vindas. Elas tendem a ajustar o desenvolvimento do software às mudanças de requisitos, sendo mais flexíveis neste ponto.

- As metodologias ágeis são orientadas às pessoas, não aos processos: a meta das metodologias de desenvolvimento é desenvolver um processo que vai funcionar da mesma maneira, independentemente de quem esteja executando. Metodologias ágeis levam mais em conta a habilidade da equipe de desenvolvimento, e tendem por isto a se ajustar à equipe.

A filosofia por trás das metodologias ágeis está representada pelo Manifesto Ágil (SOMMERVILLE, 2011), conforme explicado no tópico seguinte.

2.1.1 Manifesto Ágil

Em fevereiro de 2001, os representantes de ideias, metodologias e processos que, em contrapartida aos métodos tradicionais, estavam trazendo valor para seus clientes, por meio de abordagens leves e empíricas para projetos de desenvolvimento de software, se reuniram no estado de Utah, Estados Unidos (SABBAGH, 2013). Durante a reunião foi criado o Manifesto Ágil que, segundo Wildt et al. (2015), é uma carta aberta aos profissionais da área que contém os valores e os princípios que fundamentam o desenvolvimento ágil de software. O Manifesto Ágil (BECK et al., 2001) possui quatro valores essenciais:

- 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 às mudanças **mais que** seguir um plano.

Para os autores do Manifesto Ágil (BECK et al., 2001) todos os itens citados são importantes, porém deve ser dado maior valor aos itens à esquerda. Além disso, por trás do Manifesto Ágil existem doze princípios que regem os seus métodos (BECK et al., 2001):

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de *software* de valor.

2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adaptam às mudanças, para que o cliente possa tirar vantagens competitivas.
3. Entregar *software* funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
4. Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente durante todo o curso do projeto.
5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6. O método mais eficiente e eficaz de transmitir informações para e entre um time de desenvolvimento, é através de uma conversa cara a cara.
7. *Software* funcional é a medida primária de progresso.
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
9. Contínua atenção à excelência técnica e bom *design*, aumenta a agilidade.
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11. As melhores arquiteturas, requisitos e *designs* emergem de times auto organizáveis.
12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Baseados nesses valores e princípios, a próxima seção expõe três abordagens mais específicas que, proporcionam bons resultados, seguindo a filosofia do Manifesto Ágil (WILDT et al., 2015). São eles: *Scrum*, *Lean Software Development* e *eXtreme Programming* (XP).

2.1.2 Métodos Ágeis

O autor Sabbagh (2013) define o *Scrum* como um *framework* ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos. *Scrum* é embasado no empirismo e aplica uma abordagem iterativa e incremental para entregar valor com frequência e, assim, reduzir os riscos do projeto.

No *Scrum*, segundo Sabbagh (2013), existem três papéis diferentes: *Time de Desenvolvimento*, *Product Owner* e *Scrum Master*. O *Time de Desenvolvimento* é um grupo multidisciplinar de pessoas, responsável por realizar o trabalho de desenvolvimento do

produto. Já o *Product Owner*, também chamado de P. O., é a pessoa responsável por garantir e maximizar, a partir do trabalho do *Time de Desenvolvimento*, o retorno sobre o investimento no produto para os clientes do projeto. Por fim, o *Scrum Master* é a pessoa que trabalha para facilitar e potencializar o trabalho do *Time de Desenvolvimento*. Ou seja, aproveitando de seu conhecimento de *Scrum*, habilidade de lidar com pessoas e técnicas de facilitação, o *Scrum Master* ajuda o *Product Owner* e *Time de Desenvolvimento* a serem mais eficientes na realização do seu trabalho (SABBAGH, 2013).

Para Wildt et al. (2015), o *Scrum* se concentra mais nos aspectos gerenciais do desenvolvimento de *software*. A partir de um conjunto de requisitos inicial, chamado de *Backlog de Produto*, o trabalho que será executado é priorizado na iteração, denominada *Sprint*. O desenvolvimento do *Backlog da Sprint* gera um incremento entregável do produto ao final de cada *Sprint*. Esse trabalho é desenvolvido com o acompanhamento diário do *Time de Desenvolvimento*, na *Reunião Diária* (ou *Stand-up Meeting*). Ao final da *Sprint*, é realizada a reunião de *Retrospectiva da Sprint*, com o objetivo de reduzir riscos e promover a melhoria contínua (WILDT et al., 2015).

Por dar menos ênfase aos aspectos técnicos, é geralmente combinada com práticas propostas pelo *eXtreme Programming*. Para Wildt et al. (2015), o *eXtreme Programming*, ou XP é uma metodologia ágil de desenvolvimento de *software* que tem como como valores a comunicação, a simplicidade, o *feedback*, a coragem e o respeito. Seu principal objetivo é promover uma maior disciplina de engenharia ágil de *software* para codificação e testes.

De acordo com Wildt et al. (2015), o XP é baseado em um conjunto de doze práticas fundamentais, que podem ser aplicadas individualmente, cada uma delas gerando melhorias no processo de desenvolvimento, mas que funcionam melhor em conjunto, uma vez que uma técnica reforça o resultado da outra. Conforme Paulk (2001), as doze práticas são:

1. Jogo do planejamento;
2. Pequenas entregas;
3. Metáfora de sistema;
4. Projeto simples;
5. Desenvolvimento orientado à testes (TDD);
6. Refatoração de código;
7. Programação em par;
8. Posse coletiva do código;

9. Integração contínua;
10. Ritmo sustentável;
11. Cliente presente;
12. Padrão de codificação.

Outra abordagem, indicada por Poppendieck e Poppendieck (2011), é o movimento *Lean Software Development*, criado com base no Sistema *Toyota* de Produção que revolucionou a indústria da manufatura, tanta pela geração de valor para o cliente, quanto para a satisfação de trabalho para o time. As características do Sistema *Toyota* de Produção serviram de apoio para traçar semelhanças com os valores ágeis dando origem aos sete princípios que guiam o *Lean Software Development* (POPPENDIECK; POPPENDIECK, 2011):

1. Eliminar desperdícios;
2. Incluir qualidade no processo;
3. Criar conhecimento;
4. Adiar comprometimentos;
5. Entregar rápido;
6. Respeitar as pessoas;
7. Otimizar o todo.

Seguindo o movimento *Lean*, conforme Albino (2017), é importante ressaltar também o *Kanban*, que é uma abordagem enxuta para a gestão do fluxo de trabalho. Literalmente, *Kanban* é uma palavra japonesa que significa "cartão visual". Na Toyota, *Kanban* é o termo usado para o sistema de sinalização visual e física que liga todo o processo produção (ALBINO, 2017).

Foi David Anderson quem desenvolveu o método *Kanban* para desenvolvimento de *software*, que combina o pensamento *Lean* e a teoria de restrições. Segundo Anderson (2011), o *Kanban* está sustentado em seis propriedades:

1. **Visualizar o fluxo de trabalho:** dividir o trabalho em partes, escrevendo cada demanda em um cartão e ter visão de todo o trabalho que está em andamento. Além disso, nomear colunas para ilustrar onde cada demanda está no fluxo de trabalho.

2. **Limitar o trabalho em progresso:** atribuir limites explícitos para a quantidade de itens que podem estar em andamento em cada etapa do fluxo de trabalho.

3. **Medir e gerenciar o fluxo de trabalho:** ter visibilidade de quanto tempo as demandas que passam pelo processo têm levado para serem concluídas e remover os impedimentos que por ventura estejam bloqueando o fluxo de trabalho.

4. **Tornar as políticas explícitas:** definir, divulgar e socializar o processo, assim todos terão uma clareza de como ele funciona e de como o trabalho realmente é feito. Exemplos de políticas são os critérios para as transições entre as etapas do fluxo de trabalho e o limite de trabalho em progresso.

5. **Desenvolver *loops* de *feedback*:** reduzir o tempo do ciclo de *feedback*, pois quanto mais tardia a entrega, mais demorada será a geração de valor daquilo que foi trabalhado.

6. **Melhorar de forma colaborativa:** exercer a filosofia de melhoria contínua baseada no *kaizen*, promovendo pequenas alterações no processo que causem menor resistência à mudança.

2.2 MÉTRICAS

Nesta seção são apresentados os fundamentos sobre avaliação de desempenho, conceitos gerais, características e classificações de métricas.

2.2.1 Conceitos Gerais

A atividade de engenharia de software responsável pela análise de desempenho é a medição de software, a qual desempenha um papel fundamental tanto para apoiar no controle de atividades do projeto quanto para monitorar planos de melhoria de processo de organizações de software (PRESSMAN; MAXIM, 2016).

Segundo Sato (2007), para adentrar na matéria de métricas primeiramente é necessário o entendimento de alguns conceitos importantes, em particular os termos medição, medida, métrica e indicador. Segue abaixo suas definições e posteriormente um cenário para exemplificar.

- **Medição:** de um modo geral, a medição é o conjunto de operações com o objetivo de determinar um valor de uma medida (ISO, 2007), permitindo quantificar medidas para manipulá-las e aprender sobre a situação. Já uma medida é uma variável para a qual um valor é atribuído como resultado de uma medição (ISO, 2007).

- **Medida:** segundo o IEEE (1990), uma medida é uma avaliação em relação a um padrão. A medida também é a avaliação de um atributo segundo um método de

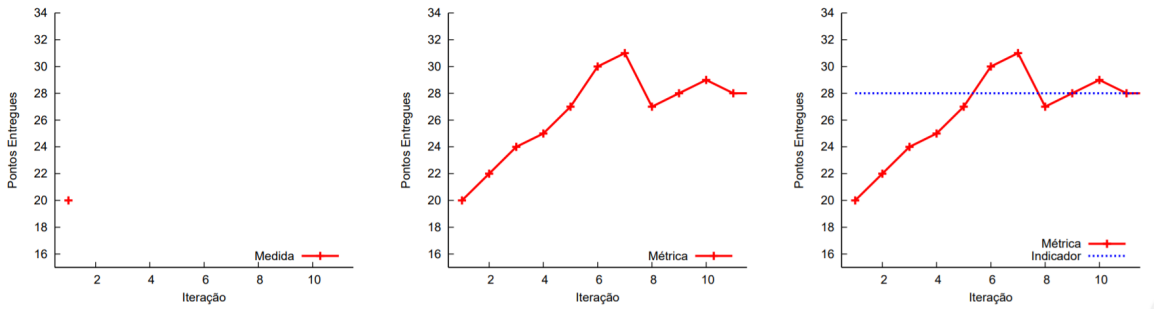
medição específico, funcionalmente independente de todas as outras medidas e capturando informação sobre um único atributo (MCGARRY et al., 2002).

- **Métrica:** o IEEE (1990) diz que métrica é um método para determinar um certo atributo em relação a um sistema, componente ou processo. Ela geralmente é formada por duas ou mais medidas. Ao coletar medidas e desenvolver métricas é possível obter indicadores (SATO, 2007).

- **Indicador:** um indicador para o IEEE (1990) é um dispositivo ou variável que pode ser configurado para um determinado estado com base no resultado de um processo ou ocorrência de uma determinada condição. Um indicador é uma métrica ou combinação de métricas que disponibilizam informações sobre o processo de software ou no próprio produto (SATO, 2007).

O exemplo fictício apresentado por Sato (2007) auxilia para demonstrar a relação entre medição, medida, métrica e indicador. Ao final da primeira iteração do desenvolvimento de um projeto, é apurado que foi entregue 20 pontos (Figura 2 (a)). Com o andamento do projeto, pode ser observado que o número de pontos entregues pela equipe aumenta aos poucos a cada iteração. Em certo ponto, essa tendência de aumento cessa e o número de pontos por iteração alcança um nível (Figura 2 (b)).

Figura 2: Total de pontos entregues por iteração



(a) Medida (b) Métrica (c) Indicador

Fonte: Sato (2007)

Neste exemplo, a Figura 2 (a) representa uma medida, que foi coleta em uma medição. A medida por si só não promove muita informação sem ter outra informação ou tendência para comparar. Na Figura 2 (b) uma métrica é representada, neste caso a velocidade de entrega da equipe. Como pode ser visto, a métrica é formada por diversas medidas como o número de pontos entregues em cada iteração do projeto. Depois que a métrica atinge um padrão de comportamento, a equipe pode considerar um patamar para sua velocidade. Com isso, o valor base ilustrado na Figura 2 (c) representa um indicador. Ele serve como base de comparação para a métrica. É importante ressaltar que uma métrica deve ser interpretada sob um ponto de vista específico. O significado de um

indicador sempre depende de um contexto, desse modo duas equipes podem interpretar a mesma métrica de maneira diferente. No cenário exposto, a equipe não tinha indicador antes, somente aconteceu uma estabilização da métrica para que um valor base fosse encontrado. Mas, supondo um outro caso, onde a mesma equipe usasse como indicador um valor base mais baixo para sua velocidade. Desta forma, a mesma métrica poderia ser interpretada como resultado de uma melhoria.

De acordo com Pressman e Maxim (2016), os Métodos Ágeis promovem um processo empírico para o desenvolvimento de software. Essa abordagem prega um ciclo constante de inspeção, adaptação e melhoria. A melhoria contínua requer mudança na organização. A avaliação das mudanças requer medição e indicadores como parâmetros. Apenas a medição não provoca mudanças, o recomendado é que a medição conduza a uma ação e não ser utilizada meramente para armazenar dados. Recomenda-se que as medições tenham uma finalidade claramente definida (PRESSMAN; MAXIM, 2016).

2.2.2 Características de Métricas Ágeis

Baseados em diferentes autores, Hartmann e Dymond (2006) sugerem uma seleção de algumas das características que devem ser consideradas para escolher uma métrica para um processo ou equipe. Uma boa métrica ágil deve:

- **Reforçar princípios ágeis:** colaboração com o cliente e entrega de valor são princípios fundamenteis para os Métodos Ágeis.
- **Medir resultados e não saídas:** ao valorizar a simplicidade, os melhores resultados podem ser aqueles que minimizam a quantidade de trabalho realizado. Medir os resultados obtidos é mais importante que medir as saídas das atividades em processo.
- **Seguir tendências e não números:** os valores representados por uma métrica são menos importantes que a tendência demonstrada. Ao medir a velocidade da equipe, é melhor se preocupar com sua estabilização do que com o valor absoluto que ela representa.
- **Responder uma pergunta específica para uma pessoa real:** toda métrica deve expor informação para um ponto de vista específico. Se outra pessoa tem outra pergunta, é melhor usar outra métrica.
- **Pertencer a um conjunto pequeno de métricas e diagnósticos:** é impossível medir tudo. Muita informação pode esconder o que realmente importa. Minimize o número de métricas e meça o que é mais importante.
- **Ser facilmente coletada:** para métricas de acompanhamento objetivas e quantitativas, o ideal é ter uma coleta automatizada.

- **Revelar, ao invés de esconder, seu contexto e suas variáveis:** uma métrica deve deixar claro os fatores que a influenciaram para evitar manipulações e facilitar a melhoria do processo.
- **Incentivar a comunicação:** uma métrica isolada de seu contexto perde o sentido. Um bom sinal é quando as pessoas comentam o que aprenderam com uma métrica.
- **Fornecer *feedback* frequentemente e regular:** para amplificar o aprendizado e acelerar o processo de melhoria, as métricas devem ser frequentemente atualizadas e disponibilizadas na *Área de Trabalho Informativa*.
- **Poder medir o valor agregado produto ou processo:** dependendo de onde estão os problemas, as métricas podem medir qualquer coisa suspeita de inibir a eficácia. Considere o público adequado para cada métrica e documente seu contexto e suposições para incentivar o uso adequado de seu conteúdo.
- **Encorajar um alto nível de qualidade:** o nível aceitável de qualidade deve ser definido pelo cliente e não pela equipe. Os Método Ágeis exigem sempre um alto nível de qualidade do software desenvolvido.

2.2.3 Classificações

Esta seção apresenta algumas das possíveis classificações, segundo Sato (2007), que as métricas podem ter seguindo diferentes critérios.

2.2.3.1 Quantitativa/Qualitativa

Uma métrica pode ser classificada como quantitativa ou qualitativa para Sato (2007). Segundo o autor, o valor de uma métrica quantitativa pertence a um intervalo de uma certa magnitude e normalmente é exposto por um número. Isto permite que medidas deste tipo possam ser comparadas entre si. De outro lado, valores de uma métrica qualitativa são aqueles representados por palavras, símbolos ou figuras ao invés de números (GILGUN, 1992 apud SATO, 2007). Um exemplo de métrica qualitativa é a satisfação do cliente final com produto entregue.

2.2.3.2 Objetiva/Subjetiva

Conforme explicado na subseção 2.2.1, uma métrica é composta por medidas que avaliam atributos de um objeto. De acordo com Sato (2007), o valor de uma métrica objetiva depende somente do objeto em questão e não do ponto de vista de quem está interpretando a métrica. Por exemplo, o número de testes unitários escritos, pois o valor é obtido diretamente de ferramentas de cobertura de testes, não dependendo de interpretação por parte do observador. Em contrapartida, o valor de uma métrica subjetiva

depende do objeto em questão e também do ponto de vista de quem está interpretando. Um exemplo para este caso é a qualidade do código, numa escala de 0% a 100%. Apesar da escala ser numérica, o resultado depende do ponto de vista de quem está avaliando o código.

2.2.3.3 Categorias de Avaliação

Pegoraro (2014) relata que o agrupamento das métricas em categorias de avaliação é importante para saber quais são os aspectos sobre o desenvolvimento de software estão sendo mais avaliados. Segundo a autora, existem poucos estudos que propõe categorização de métricas ou indicadores para projetos ágeis.

A partir de sua revisão bibliográfica, Pegoraro (2014) encontrou os trabalhos de Hartmann e Dymond (2006), Cheng, Jansen e Remmers (2009) e Poonacha e Bhattacharya (2012) que propõem formas de classificação.

Hartmann e Dymond (2006) sugerem dois tipos de categorias para a classificação, dividindo as métricas em organizacionais e diagnóstico. As métricas organizacionais englobam métricas de longo prazo, retorno de negócio sobre o investimento e satisfação dos clientes. Já a categoria de diagnóstico acaba sendo muito ampla, incluindo a maioria das métricas. Por este motivo, Pegoraro (2014) decidiu dividir a categoria de diagnóstico empregando a classificação indicada pelos autores Cheng, Jansen e Remmers (2009) e Poonacha e Bhattacharya (2012).

A proposta de Cheng, Jansen e Remmers (2009) é classificar as métricas em equipes/pessoas, tarefas do processo de desenvolvimento e qualidade. Entretanto, Poonacha e Bhattacharya (2012) propõem a classificação em pessoas, processo, cliente, qualidade e custos. A partir dessas classificações encontradas, Pegoraro (2014) selecionou para seu trabalho as categorias: organizacional, equipe, qualidade e processo, como ilustra o Quadro 1.

Quadro 1: Categorização das métricas

		Diagnóstico		
	Organizacional	**Equipe**	**Qualidade**	**Processo**
Hartmann e Dymond (2006)	X			
Cheng, Jansen e Remmers (2009)	X	X	X	X
Poonacha e Bhattacharya (2012)	X	X	X	X

Fonte: Pegoraro (2014)

Neste trabalho é utilizado a classificação proposta por Pegoraro (2014), contendo:

- **Categoria Organizacional:** contempla as métricas que avaliam aspectos gerais do negócio, financeiras, sobre satisfação do cliente e decisões de nível gerencial.

- **Categoria Equipe:** envolve métricas que avaliam questões relacionados à equipe de desenvolvimento ou seus indivíduos.
- **Categoria Processo:** abrange as métricas que avaliam aspectos relacionados ao processo de desenvolvimento iterativo e incremental, gestão das tarefas ou controles necessários que são particulares do processo de desenvolvimento ágil.
- **Categoria Qualidade:** engloba as métricas que representam questões inerentes ao controle de qualidade do produto.

Na seções seguintes são abordados métricas da categoria processo para dar continuidade ao trabalho, pois nessa categoria encontra-se as métricas relacionadas às questões do controle do processo de desenvolvimento ágil, conforme indicado pelos autores Cheng, Jansen e Remmers (2009) e Poonacha e Bhattacharya (2012).

2.2.4 Métricas de Processo

O foco da pesquisa é trabalhar com as métricas da categoria processo, dado que elas avaliam os aspectos referentes à gestão de tarefas no desenvolvimento ágil de software. Na próximas seções são apresentadas as métricas *Work In Progress*, *Lead Time* e *Throuphput* que são métricas aplicadas pelo mercado para medir o fluxo de desenvolvimento ágil e que são referenciadas pela literatura (PEGORARO, 2014). Entre as métricas selecionadas existe uma relação segundo Little (1961) que é explica na última seção.

2.2.4.1 *Work In Progress* (WIP)

Em sua definição, a métrica *Work In Progress* (em português trabalho em progresso) ou simplesmente WIP refere-se a todos os materiais e produtos parcialmente finalizados que estão em diferentes etapas do fluxo de desenvolvimento (ALBINO, 2017). Deste modo, o WIP pode ser interpretada como uma demanda importante que foi começada, mas que ainda não está fornecendo valor para o cliente (ALBINO, 2017).

No cenário do desenvolvimento de software ágil, o WIP é o número de atividades em andamento no processo produtivo, ou seja, é o termo dado ao trabalho que foi iniciado, mas não foi concluído (GUSTAFSSON, 2011). Em sua pesquisa Pegoraro (2014) diz que o objetivo da medição do WIP é identificar a capacidade de produção da equipe. Para identificar a quantidade de trabalho em progresso, segundo Albino (2017), é necessário estabelecer as etapas de entrada e saída do processo de desenvolvimento, independentemente do processo ágil adotado pela equipe. A definição desses dois limites do sistema produtivo é o ponto de partida na construção de um processo que busca previsibilidade de entrega e visibilidade do trabalho.

Anderson (2011) afirma que uma das propriedades básicas do método Kanban é a de limitar o WIP. O autor defende a necessidade de criar limitações explícitas para a quantidade máxima de itens que estarão em progresso para cada uma das etapas do fluxo de desenvolvimento. Controlar o WIP traz muitos benefícios, para Albino (2017, p. 43):

> Medir o WIP é um dos trabalhos mais importantes no monitoramento da saúde do processo de desenvolvimento de software, pois, [...] ele é um dos preditores do desempenho geral do fluxo de desenvolvimento. Gerenciando o WIP, garantimos um controle na cadência das entregas e impactamos o tempo que uma demanda passará pelo fluxo de desenvolvimento.

Então um dos principais efeitos de limitar o WIP é fornecer previsibilidade de tempo em ciclos e fazer as entregas mais confiáveis. Segundo Albino (2017), ao limitar o WIP do processo de desenvolvimento é possível:

1. Evitar a formação de gargalos que reduzirão a performance do time.
2. Ajuda a transparecer bloqueios no sistema para os envolvidos no projeto.
3. Quanto maior o número de demandas em andamento em determinada parte do processo, maior será o tempo em que uma demanda passará pelo fluxo.
4. Reduzindo o trabalho em progresso, irá melhorar a qualidade do que se passa pelo sistema. Normalmente o aumento de WIP está relacionado com fatores que invariavelmente decrescem a qualidade, aumentando o número de falhas, e dessa forma aumentando o retrabalho.

2.2.4.2 *Lead Time*

Num contexto geral, *Lead Time* representa a latência entre o início e o fim de um processo. Dentro do cenário de desenvolvimento de software, é considerado *Lead Time* o tempo (na maioria das vezes expressado em dias) que decorre entre início e o fim do processo de entrega de um item de trabalho (ANDERSON, 2011; ALBINO, 2017). Para exemplificar melhor, a Figura 3 mostra o cálculo do *Lead Time* de um item a partir dos limites de entrada e saída definidos no processo de desenvolvimento.

O objetivo desta métrica é minimizar o tempo médio de ciclo de um sistema, sob o ponto de vista do cliente final. O importante é descobrir o quão rápido o sistema entrega valor de forma repetitiva e confiável. O *Lead Time* mapeia a quantidade de tempo que a demanda gasta para passar por todo fluxo de valor. Segundo Albino (2017), medir o *Lead Time* é importante para:

1. Compreender quanto tempo a equipe tem levado para desenvolver um item de trabalho.

Figura 3: Cálculo do *Lead Time* através dos limites do processo

Fonte: Albino (2017)

2. Analisar a saúde do processo de desenvolvimento dado que altas dispersões representam algum tipo de gargalo ou aumento no tempo de passagem em alguma etapa do fluxo de desenvolvimento.

3. Identificar casos extremos e aprender com o ocorrido.

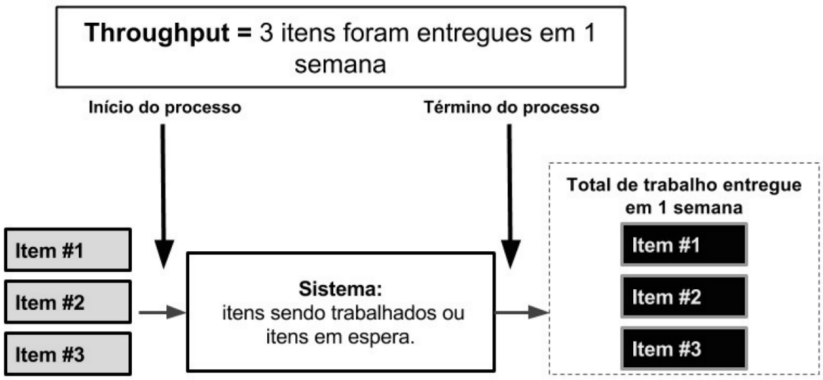
4. Analisar se a equipe tem entregue os itens de trabalho dentro de um padrão de dias ou semanas.

5. Para compreender os efeitos que as incertezas e as complexidades não mapeadas podem causar, na forma de variabilidade, no tempo necessário para a conclusão dos itens de trabalho de uma equipe de desenvolvimento de software.

2.2.4.3 *Throughput*

A métrica *Throughput* é uma medida que define quantas unidades de trabalho foram completadas em um período de tempo, ou seja, mede o rendimento da equipe de desenvolvimento. Pode-se dizer também que o *Throughput* é uma medida que determina quanto o fluxo de trabalho é capaz de processar (ALBINO, 2017).

Figura 4: Exemplo de cálculo do *Throughput*



Fonte: Albino (2017)

A Figura 4 exibe um exemplo de cálculo do *Throughput*, onde pode-se constatar que em uma semana qualquer, a vazão de tarefas entregues por uma equipe foi de três funcionalidades.

2.2.5 Lei de *Little*

As métricas de forma isolada podem proporcionar informações importantes, porém é possível combiná-las para evoluir ainda mais as análises do fluxo de trabalho (ALBINO, 2017). Surgida do estudo de sistemas de filas de Little (1961), a Lei de *Little* prevê que, em um sistema estável, a quantidade média de tempo que algo leva para atravessar um processo é igual ao número de coisas no processo dividido por sua taxa de conclusão média (LITTLE, 1961 apud ALBINO, 2017, p. 136–140).

Aplicada em processos de desenvolvimento de software, como assinalado pelo autor Albino (2017), a Lei de *Little* é uma fórmula que estabelece uma relação entre as métricas WIP, *Lead Time*, e *Throughput*. A fórmula da Lei de *Little* pode ser descrita da seguinte forma:

$$L = W/\lambda \tag{2.1}$$

Onde L é a média de *Lead Time*, W é a média de itens em WIP e λ é a média *Throughput* (ALBINO, 2017).

É importante ressaltar que há uma diferença entre a fórmula originalmente criada por Little (1961) e a fórmula citada. A versão original lida com taxa de entrada e a versão adaptada para o contexto de desenvolvimento de software trata de taxa de saída.

O resultado da aplicação da variação da lei no contexto de desenvolvimento de software nos diz que, em geral, quanto mais coisas são trabalhadas em um determinado momento (em média), mais tempo levará para que cada uma dessas coisas termine (em média). Caso seja alterado qualquer uma das variáveis da fórmula, consequentemente as outras serão afetadas. Desse modo, a equação aponta possíveis gatilhos para praticar a melhoria nos processos de desenvolvimento de *software*.

Algumas mudanças nas premissas são necessárias quando passamos a olhar a lei de *Little* sob a perspectiva de *Throughput* e não a partir da taxa de entrada. Segundo Vacanti (2015), são elas:

1. A entrada média ou taxa de chegada deve ser igual à taxa média de saída.

2. Todo o trabalho que é iniciado será eventualmente concluído e sairá do sistema. Por exemplo, tarefas que ao decorrer do desenvolvimento, por algum motivo sejam cancelados. Essas tarefas invalidam a Lei de *Little*.

3. A quantidade de WIP deve ser aproximadamente a mesma no início e no final do intervalo de tempo escolhido para o cálculo. Assim como a taxa de entrada e saída, a quantidade de trabalho em progresso deve ser constante.

4. Deve-se buscar um tempo de *Lead Time* constante. Em outras palavras, é como se o tempo médio necessário para entregar os itens que passam pelo processo seja o mesmo.

5. As métricas devem ser medidas usando unidades consistentes.

Com base na Lei de *Little*, Albino (2017) recomenda algumas diretrizes para auxiliar as equipes de desenvolvimento possam estabilizar o processo de trabalho, promovendo a previsibilidade de entrega. As diretrizes são:

- Só deve ser iniciado um novo trabalho aproximadamente na mesma proporção em que termina o que já está no processo de desenvolvimento.
- Depositar os esforços necessários para terminar qualquer trabalho iniciado e mitigar a possibilidade de se trabalhar com itens que possam ser descartáveis.
- Manter o fluxo de trabalho funcionando. Caso o trabalho da equipe por acaso ficar bloqueado, deve ser removido os impedimentos o mais rápido possível.
- Antes de iniciar um novo item de trabalho, deve ser analisado se existe algo em progresso que esteja com um *Lead Time* alto e dar foco para finalizá-lo.

2.3 *STORYTELLING* COM DADOS

Nesta seção do trabalho são apresentadas as lições que guiam a aplicação do *storytelling* com dados.

2.3.1 Contexto

Na apresentação de uma visualização de dados deve-se iniciar o trabalho dedicando atenção para entender o contexto da necessidade de se comunicar (KNAFLIC, 2017). Ainda de acordo com autora Knaflic (2017), primeiramente, é importante identificar quem é o público e como ele interpretará os dados. Isso auxilia a descobrir pontos para garantir que a mensagem seja repassada com sucesso.

Em segundo, ter em mente como que o público alvo de agir. Deve-se levar em conta como conduzir a comunicação e somente depois desse entendimento sobre o público, é que será tratado como os dados ajudarão a defender a ideia a ser transmitida (KNAFLIC, 2017). A seguir segue mais detalhes sobre o contexto segunda a autora Knaflic (2017).

- **Público:** quanto mais específico for o perfil do público, melhor será para obter sucesso na comunicação. Deve-se evitar públicos muitos vagos, como por exemplo "interessados internos ou externos". Ao tentar comunicar-se de uma só vez com diferentes pessoas, com necessidades distintas, a mensagem não atinge eficientemente nenhuma delas. Isso significa que o ideal é criar diferentes comunicações para diferentes públicos. Identificar o tomador de decisão é uma maneira de restringir o público. Quanto mais específico for o público, maior será a satisfação da necessidade dele.

- **Apresentador:** é importante analisar a relação do apresentador com o público e como que o apresentar quer ser percebido. Existem algumas considerações a serem levantadas: Será a primeira vez que apresentador e público se encontrarão por meio dessa comunicação ou já têm uma relação definida? O público já considera o apresentador um especialista ou ele precisa trabalhar para estabelecer credibilidade? Essas são considerações importantes quando se trata de determinar como a comunicação será estruturada e se e quando os dados serão usados, podendo impactar a ordem e o fluxo da história global a ser transmitida.

- **Ação:** é o ponto no qual o apresentador reflete sobre como tornar o que comunica relevante para seu público e formar um claro entendimento do motivo pelo qual dele deve se importar com o que é dito. O apresentador deve querer que o público saiba ou faça algo. Se não conseguir articular isso de forma concisa, deve-se verificar a necessidade de fazer tal comunicação.

- **Mecanismo:** é como se comunicar com público. O método a ser usado para se comunicar tem implicações em diversos fatores, incluindo o controle que o apresentador terá sobre como o público entende a informação e o nível de detalhe a ser explicitado. É importante identificar o principal veículo de comunicação a ser usado (apresentação ao vivo, documento escrito ou outro).

- **Tom:** outra consideração importante é o tom desejado para a comunicação a ser transmitida ao público. O tom escolhido é importante porque por meio dele pode-se melhorar a visualização dos dados apresentados ao público.

2.3.2 Visual Eficaz

Existem muitos gráficos diferentes e outros tipos de exibições de informações visuais, nesta seção são apresentados os 12 modelos visuais mais indicados por Knaflic (2017).

1. **Texto simples:** quando o dado a ser comunicado tem apenas um número ou dois, um simples texto pode ser uma ótima maneira de comunicar.

2. **Tabela:** as tabelas interagem com o sistema verbal das pessoas, o que significa que o público lê as tabelas. As tabelas são boas para comunicar um público misto, cujos membros procurarão, cada um, sua linha de interesse particular. O *design* das tabelas devem ficar em segundo plano, permitindo que os dados fiquem em primeiro plano. Evitar bordas grossas ou sombreados que chamem atenção. As bordas devem ser usadas para melhorar a legibilidade da tabela. Outra dica para analisar é desfocar as bordas, tornando-as cinza, ou retira-las totalmente.

3. **Mapa de calor:** usar mapas de calor é um jeito de misturar detalhes em uma tabela usando pistas visuais. O mapa de calor é um modo de visualizar dados em formato tabular, onde além das informações, é utilizado células coloridas que transmitem a magnitude relativa dos números. Essa é uma técnica para reduzir o processo mental, pois o uso de saturação de cor auxilia os olhos e o cérebro a atingir mais rapidamente os possíveis pontos de interesse.

4. **Gráfico de dispersão:** os gráficos de dispersão podem ser úteis para mostrar a relação entre duas coisas, pois eles permitem codificar dados simultaneamente em um eixo x horizontal e um eixo y vertical para ver se existe relação e qual é a relação. Se há necessidade de conduzir a atenção à relação dos dados, é indicado adicionar uma linha tracejada, projetada para facilitar a leitura mais rapidamente.

5. **Linha:** os gráficos de linhas são mais usados para registrar dados contínuos. Como os pontos são fisicamente conectados por meio da linha, isso indica uma conexão entre eles. Nesse tipo de gráfico é possível representar mais de uma série de dados. Ao representar alguma unidade de tempo é importante ser coerente na disposição dos pontos de tempo que representar.

6. **Mapa de inclinação:** os gráficos de inclinação são úteis quando têm dois períodos de tempo ou pontos de comparação e há a necessidade de mostrar rapidamente os aumentos ou diminuições relativas ou as diferenças de várias categorias entre os dois pontos de dados.

7. **Barras verticais:** os gráficos de barras verticais, ou gráficos de colunas podem ter uma ou várias séries de dados, porém evite excessos. Eles são indicados por terem menos curva de aprendizado para o público na leitura.

8. **Barras verticais empilhadas:** os casos de uso para gráficos de barras verticais empilhadas são mais limitados. Eles se destinam a permitir a comparação de totais entre categorias e ainda para visualizar as partes que compõem uma determinada categoria.

9. **Cascata:** o gráfico cascata pode ser usado para separar as partes de um gráfico de barras empilhadas a fim de focar uma de cada vez ou para mostrar um ponto de partida, aumentos ou reduções e o ponto final resultante.

10. **Barras horizontais:** o gráfico de barras horizontais é o mais indicado para representar categorias, principalmente, se os nomes de categoria são longos. Além disso, pelo modo de leitura que normalmente é processado as informações (começando à esquerda e em cima), a estrutura do gráfico de barras horizontais faz com que os olhos atingem primeiro os nomes das categorias antes dos dados. Dessa forma, ao ler os dados o público já saberá o que eles representam.

11. **Barras horizontais empilhadas:** semelhantes ao gráfico de barras verticais empilhadas, os gráficos de barras horizontais empilhadas podem ser usados para mostrar os totais em diferentes categorias, mas também dar uma ideia das partes de uma composição.

12. **Área quadrada:** o gráfico de área quadrada é indicado em casos limitados para visualizar números de magnitudes amplamente diferentes. Em outros casos, esse tipo de gráfico deve ser evitado, pois os olhos humanos não conseguem atribuir bem o valor quantitativo no espaço bidimensional, dificultando a leitura.

2.3.3 Saturação Visual

No que diz respeito às nossas comunicações visuais, o que mais importa é a carga cognitiva percebida por parte do público, isto é, o quanto ele acredita que precisará se esforçar para entender a informação da comunicação visual. Essa é a decisão a que provavelmente chegam sem grande cogitação consciente e pode fazer a diferença entre transmitir a mensagem claramente ou não (KNAFLIC, 2017).

De modo geral, deve ser minimizado a carga cognitiva percebida pelo público, pois ela faz com que as visualizações pareçam mais complicados do que realmente são. Considerando isto, a saturação visual pode contribuir para a carga cognitiva excessiva. A saturação nesse contexto, são elementos visuais que ocupam espaço, mas não aumentam o entendimento (KNAFLIC, 2017).

2.3.4 História

História é o que vincula a informação, fornecendo à apresentação ou comunicação uma estrutura para o público seguir. Pode ser usado uma ideia de começo, meio e fim, ou seja, uma estrutura de três atos, para montar as histórias que precisam ser comunicadas com os dados. Outro componente central da história é a narrativa, a qual deve-se considerar em ordem, por exemplo cronológica, e maneira, que pode ser falada, escrita ou uma combinação das duas (KNAFLIC, 2017).

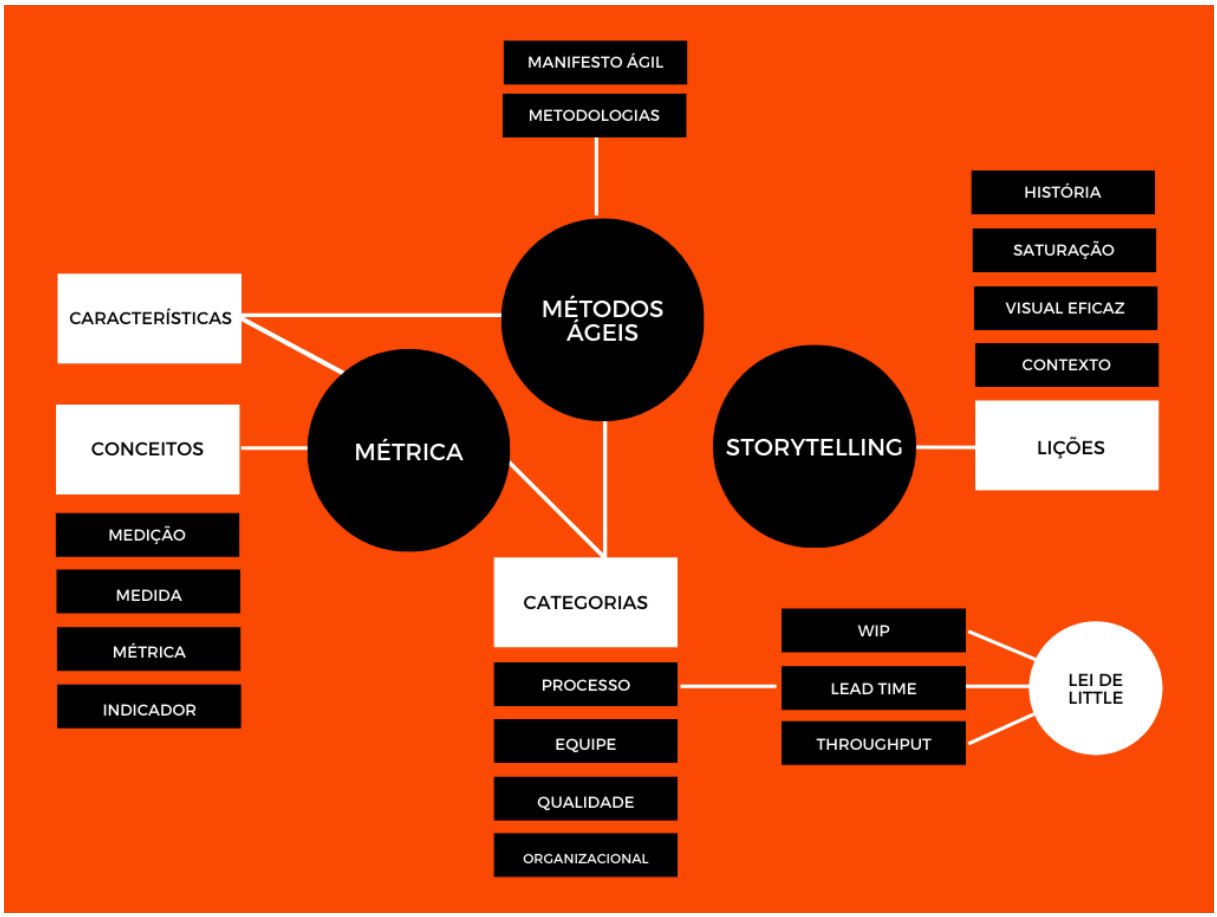
2.4 CONSIDERAÇÕES FINAIS

A etapa de embasamento teórico deste trabalho contribuiu para a compreensão do contexto do problema e dos desafios relacionados a aplicação de métricas em processos ágeis.

Segundo Fenner (2017), emprega-se o uso de mapas mentais das mais diversas formas, seja para melhoria da performance pessoal ou profissional. Através dos esquemas gráficos podemos sintetizar ideias, organizar o raciocínio, elaborar um planejamento, tomar notas, gerenciar informações, ajudar no processo de comunicação com outras pessoas, etc.

Portanto, um mapa mental (Figura 5) foi construído para demonstrar os estudos realizados que serviram de base para toda a construção teórica da proposta deste trabalho.

Figura 5: Mapa mental



Fonte: elaborado pelo autor

3 *FRAMEWORK* CONCEITUAL

Este capítulo se ocupa em apresentar a proposta de um artefato em forma de *framework* conceitual para apoiar o uso de métricas para acompanhamento de processos ágeis de desenvolvimento, por meio de visualizações de métricas com as técnicas de *storytelling* com dados. Segundo Azevedo (2016, p. 8) um *framework* conceitual deve:

> evidenciar os principais conceitos e construtos adotados na pesquisa e as relações entre eles. Essa construção deve se dar em formato de texto, mas pode ser auxiliada por um desenho ou por quadro esquemático. O importante a notar aqui é que o *framework* conceitual indica ao leitor em que conceitos a pesquisa se baseia, de que modo ela deve ser lida, com quais pesquisas anteriores ela dialoga e quais aspectos do fenômeno estão sendo analisados.

O artefato foi denominado *Framework* AMS, onde AMS é a junção das iniciais dos três pilares do trabalho: Ágil, Métricas e *Storytelling*.

Este capítulo está dividido em 3 seções. Na primeira são definidos os requisitos levantados para a construção do artefato. Na segunda seção é desenvolvida a proposta de *framework* e como cada etapa foi elaborada. E na última seção é abordado uma visão geral do *framework* com o uso de um infográfico.

3.1 REQUISITOS DO *FRAMEWORK*

Com base na literatura levantada nas seções prévias deste trabalho, foram extraídos requisitos que o artefato deve atender para ser considerado uma solução satisfatória a este estudo. O resultado é apresentado no Quadro 2, no qual "RQ#" representa a identificação atribuída a cada requisito.

Quadro 2: Requisitos do artefato

Requisito	**Descrição**
RQ01	O artefato deve indicar como selecionar as métricas para processos ágeis
RQ02	O artefato deve prescrever como elaborar um processo de medição das métricas
RQ03	O artefato deve prescrever como selecionar um modelo visual
RQ04	O artefato deve prescrever o uso das lições de *storytelling*
RQ05	O artefato deve ser simples de implementar

Fonte: elaborado pelo autor

3.2 DESENVOLVIMENTO DO *FRAMEWORK*

A proposta de *framework* elaborada neste trabalho está fundamentada nos valores e princípios do Manifesto Ágil e nas percepções extraídas do embasamento teórico, como a classificação de métricas indicado na tese de Pegoraro (2014). Para delimitar o escopo do trabalho é abordado apenas as métricas da categoria processo. Alinhado com o objetivo do estudo, também foram considerados aspectos do *storytelling* com dados.

O *framework* foi construído através da definição num ciclo lógico de atividades, apresentado através de uma sequência de fases que permita a uma equipe ágil planejar, executar, e melhorar a forma de realizar o acompanhamento de seus projetos através de métricas, respeitando as características dos projetos ágeis de desenvolvimento de software. As fases definidas para o processo são:

1. Seleção das Métricas
2. Definição da Medição
3. Desenvolvimento das Visualizações
4. Desenvolvimento da Ferramenta
5. Melhoria Contínua

3.2.1 Seleção das Métricas

Na primeira etapa as equipes devem ter um entendimento do contexto atual do processo de desenvolvimento e definir o que precisa ser medido dentro deste cenário. Isto é o que indica a primeira lição de *storytelling* com dados, que é compreender o contexto da necessidade de se comunicar. O resultado da primeira etapa é a escolha de uma ou mais métricas para o acompanhamento do processo de desenvolvimento que as equipes desejam aplicar.

Sugere-se o uso do conjunto de métricas elaborado por Pegoraro (2014) em sua tese. O conjunto detalhado de métricas foi concebido a partir da revisão de literatura sistemática e de uma pesquisa de campo com empresas desenvolvedoras de software, experientes no uso de métodos ágeis.

A pesquisa de campo aplicada foi uma entrevista qualitativa, de caráter exploratório, onde foram entrevistados profissionais de nove empresas. Dessas empresas, três são de grande porte, três de médio porte e as outras três empresas são de pequeno porte. Das nove empresas entrevistadas, sete utilizam métodos ágeis em todos os projetos e duas delas analisam o contexto dos projetos para decidir qual método mais adequado, podendo haver uma mescla entre o método ágil e tradicional. Todas as empresas possuem

equipes de desenvolvimento subdividas em grupos variando entre três e dez pessoas por equipe. Sobre o método ágil praticado, três empresas aplicam o método ágil *Scrum* e as demais adotam um conjunto de práticas ágeis onde foi possível constatar as influências dos métodos *Scrum*, *Lean* e *Kanban* associados (PEGORARO, 2014).

Levando em consideração os resultados da pesquisa de campo e da revisão de literatura, foi formulado um conjunto de métricas que é composto no total de 32 métricas para controle de projetos ágeis de software, as quais são especificadas quanto à sua definição, forma de uso e benefícios proporcionados. Para a categoria de processo, a serem consideradas na aplicação do *framework*, foram identificadas 15 métricas. São elas:

- Acurácia das estimativas (tempo);
- *Burndown* da iteração;
- *Burnup* da *release*/projeto;
- *Cycletime* (tempo de ciclo para concluir uma tarefa);
- Diagrama de fluxo cumulativo (CFD);
- Índice de histórias por iteração;
- *Leadtime* (tempo de entrega das funcionalidades ao cliente);
- Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões);
- Número de tarefas não concluídas na iteração;
- Taxa de acerto na estimativa das tarefas;
- Tempo investido em mudança;
- Tempo investido em tarefas não planejada;
- Tempo médio de resolução dos impedimentos;
- *Throughput* (número de tarefas entregues por período de tempo);
- *Work in Progress* (WIP).

Essa lista não faz uma distinção das métricas por tipo de método ágil já que existem equipes que adotam um conjunto de práticas ágeis e não um método específico, e pelo fato de diversas métricas poderem ser adotadas com qualquer método ágil.

O conjunto de métricas apresentado é amplo e atende a diversas necessidades de informação e controle para projetos ágeis de desenvolvimento de software. Portanto,

as equipes precisarão escolher quais métricas serão úteis de acordo com suas necessidades específicas. Algumas métricas possuem objetivos similares e que são intercambiáveis, podendo ser escolhida qual mais se enquadra na organização.

Também pode ser escolhido uma outra métrica além dessas listadas, porém, a métrica precisa ser cabível aos projetos de software que utilizam métodos ágeis. A métrica escolhida deve seguir os valores e princípios do Manifesto Ágil e deve-se ser observado se a métrica atende as características apontadas por Hartmann e Dymond (2006), conforme exposto na subseção 2.2.2.

Na próxima seção é apresentada a fase de medição.

3.2.2 Definição da Medição

Como já apontado na pesquisa, a medição é o conjunto de operações com o objetivo de determinar um valor de uma medida (ISO, 2007), permitindo quantificar medidas para manipulá-las e aprender sobre a situação. Desta forma, nesta etapa do *framework* as equipes devem definir os procedimentos básicos para obter as medidas das métricas selecionadas. O processo de coleta de dados definido deve ser compatível com as características dos projetos que aplicam métodos ágeis.

Conforme Pegoraro (2014), os requisitos para a definição do processo de medição são:

- Deve ser simples;
- Não deve agregar novas etapas ao processo de desenvolvimento de software;
- Deve fazer uso da estrutura organizacional existente na empresa o que implica em não criar novos cargos ou funções;
- Fornecer orientações que permitam às empresas refletir sobre os objetivos de medição;
- Auxiliar no monitoramento das métricas ao longo do projeto (novo);
- Permitir o acesso às informações a todos os interessados de forma rápida e simples.

O processo pode ser feito manualmente em planilhas, para pequenos quadros físicos, ou preferencialmente, de forma eletrônica através do uso de *softwares* de gestão/rastreamento de tarefas, como por exemplo *Jira* (*https://www.atlassian.com/br/software/jira*) e *Kanboard* (*https://kanboard.org/*).

Na próxima seção é tratado da escolha do visual a ser aplicado com os dados das métricas.

3.2.3 Desenvolvimento das Visualizações

Uma vez identificadas as métricas e os processos de suas medições, o próximo passo é determinar a melhor maneira de visualiza-las. Para o visual do gráfico deve ser escolhido o mais apropriado para comunicar a métrica. Vale ressaltar que algumas das métricas indicadas baseiam-se em visualizações predefinadas, nesses casos pode ser desconsiderado esta etapa.

Existem muitos gráficos diferentes e outros tipos de exibições de informações visuais, mas neste *framework* é indicado a escolha entre os 12 modelos visuais evidenciados Knaflic (2017). Os modelos são:

- Texto simples;
- Tabela;
- Mapa de calor;
- Gráfico de dispersão;
- Linha;
- Mapa de inclinação;
- Barras verticais;
- Barras verticais empilhadas;
- Cascata;
- Barras horizontais;
- Barras horizontais empilhadas;
- Área quadrada.

Esta lista não limita o uso de outros tipos, porém deve-se ter o cuidado de transmitir de forma objetiva os dados. Quando se trata de selecionar um tipo de gráfico, acima de tudo, deve ser escolhido um tipo que permita comunicar claramente os dados.

Na seção seguinte é abordado o desenvolvimento da ferramenta que exibirá as visualizações das métricas.

3.2.4 Desenvolvimento da Ferramenta

Após definir as visualizações das métricas, o próximo passo é construir uma aplicação ou aproveitar um *software* já usado pela equipe que forneça os meios para gerar as visualizações e torne-as visíveis a todos da equipe.

Na produção das visualizações pode ser aplicado outras lições de *storytelling* com dados como diminuir a saturação visual, como indica Knaflic (2017). De acordo com a autora, a saturação visual cria carga cognitiva excessiva que pode atrapalhar a transmissão da informação dos gráficos. De modo geral, deve-se identificar qualquer coisa que não esteja acrescentando valor informativo e remove-la. Para Knaflic (2017), exemplos de alterações que podem serem feitas nas visualizações visando a mitigação a saturação visual são:

- **Desfocar o título do gráfico:** O título precisa estar presente, mas não precisa chamar mais atenção que os dados.
- **Remover a borda e linhas de grade do gráfico:** Elas ocupam sem acrescentar muito valor. Não permita que elementos desnecessários atrapalhem os dados.
- **Colocar as linhas e legendas em segundo plano:** Elas não devem competir visualmente com os dados. Uma técnica interessante é tornar essas linhas na cor cinza.
- **Remover marcadores de dados:** Caso não tenham uma finalidade específica para a apresentação, os marcadores são item dispensáveis.

A próxima seção explica como o *framework* recomenda o uso das visualizações das métricas para promover a melhoria contínua.

3.2.5 Melhoria Contínua

Um dos pontos-chave de um time de alto desempenho está nas suas reuniões de retrospectiva, pois são identificados progressos a serem realizados no seu trabalho para aprimorar suas práticas. Essa reunião exige uma profunda análise do próprio trabalho para que se possam encontrar melhorias (WILDT et al., 2015). Isso reflete o princípio ágil que defende que em intervalos regulares, a equipe deve refletir em como ficar mais efetivo, então, se ajustar e otimizar seu comportamento de acordo (BECK et al., 2001).

Em sua obra Albino (2017) recomenda que em reuniões de retrospectiva devem ser expostas as visualizações das métricas desenvolvidas, com a finalidade de apoiar a equipe em exercer a filosofia de melhoria contínua baseada em dados, a partir de análises feitas de forma colaborativa. Com isso pode-se tomar decisões para promover pequenas alterações no processo que causem menor resistência à mudança, com os objetivos de

identificar pontos de melhoria no trabalho (inspeção) e de traçar planos de ação para executar essas melhorias (adaptação) (ALBINO, 2017).

Em processos ágeis de desenvolvimento maduros é dado muito valor na adoção de métricas e suas análises colaborativas como forma de apoio para evolução contínua da equipe. Como aponta o estudo de Liechti, Pasquier e Reis (2017) as métricas devem trazer à tona a conscientização e a discussão da "história contada" através dos dados.

Por fim, a seção seguinte apresenta a visão do geral da proposta de *framework* proposto.

3.3 VISÃO GERAL DO *FRAMEWORK*

Para ilustrar graficamente o artefato construído, foi criado um infográfico de como aplicar o *Framework* AMS. A Figura 6 apresenta o resultado, a sequência de etapas a serem seguidas para a implementação de ferramentas com o objetivo de acompanhar um processo de desenvolvimento de software.

Na primeira etapa, Seleção das Métricas, o primeiro passo para aplicar o *framework* é entender o cenário atual, apoioado no que a autora Knaflic (2017) afirma sobre o entendimento do contexto, vide subseção 2.3.1. O segundo passo é definir os objetivos embasado pelas necessidades identificadas no contexto da equipe. Com isso, o terceiro passo é selecionar uma ou mais métricas, que atendem as necessidades do time, a partir do conjunto de métricas de processo indicado por Pegoraro (2014).

Como visto no conceitos gerais da fundamentação, para se obter as métricas é necessário as suas medidas, e essas são aferidas através de medições. Para tanto, a segunda etapa do *Framework* AMS é a Definição da Medição. O processo de medição deve respeitar os requisitos indicados por Pegoraro (2014) na subseção 3.2.2.

A terceira etapa do artefato é o Desenvolvimento das Visualizações, onde é escolhido o tipo de visualização gráfica mais eficaz para cada métrica. Deve ser seguido as sugestões de tipos de exibições de Knaflic (2017)(subseção 2.3.2).

Após a escolha do visual mais eficaz, a quarta etapa é o Desenvolvimento da Ferramenta. Nesta etapa, é implementado uma aplicação, ou aproveitado um *software* já usado pela equipe, para construir as visualizações gráficas. Esta aplicação tornará os resultados das métricas acessíveis a todos da equipe. Durante o desenvolvimento dos gráficos devem ser aplicadas as lições de *storytelling* para diminuir a saturação visual, vide subseção 2.3.3, e dar foco aos resultados.

Por fim, a última etapa é a Melhoria Contínua. Seguindo o último princípio do Manifesto Ágil (subseção 2.1.1), o *framework* orienta o uso da visualizações gráficas das métricas construídas durante as reuniões de retrospectiva da equipe. Desta forma, os

resultados das métricas devem ser expostos, com o propósito de realizar análises e/ou discussões em conjunto sustentadas em dados.

O objetivo principal do *framework* é servir de apoio para que equipes possam encontrar as métricas que agregam mais valor à evolução do time, após isso, é importante dar visibilidade dos resultados das mesmas para que as equipes possam acompanhá-las e melhorar a cada nova iteração.

Figura 6: *Framework* proposto

Fonte: elaborado pelo autor

4 AVALIAÇÃO DO *FRAMEWORK*

A etapa de avaliação do *framework* tem como função identificar se a solução construída atende ao problema deste trabalho. Para isso, é verificado o atendimento aos requisitos estipulados anteriormente no Capítulo 3. Após, é feito a avaliação do arfeto *Framework* AMS com a aplicação do mesmo, com dados cedidos de uma fundação pública de tecnologia da informação, e por meio de entrevistas com três profissionais do mercado de trabalho.

4.1 REQUISITOS DO *FRAMEWORK*

A primeira fase da avaliação foi rever os requisitos definidos anteriormente, apurando cada um deles para constatar se estavam ou não de acordo com a construção do *framework*. Para tal foi elaborado o Quadro 3, onde é feito um cruzamento entre os requisitos levantados e os itens do trabalho que os atendem.

Quadro 3: Requisitos do artefato

Req.	**Descrição**	Item
RQ01	O artefato deve indicar como selecionar as métricas para processos ágeis	subseção 3.2.1
RQ02	O artefato deve prescrever como elaborar um processo de medição da(s) métrica(s)	subseção 3.2.2
RQ03	O artefato deve prescrever como selecionar um modelo visual	subseção 3.2.3
RQ04	O artefato deve prescrever o uso das lições de *storytelling*	subseção 3.2.3 e subseção 3.2.4
RQ05	O artefato deve ser simples de implementar	figura 6

Fonte: elaborado pelo autor

Pode ser observado que todos os requisitos apontados foram respeitados, porém foi identificado um ponto que precisa ser melhorado que é a apresentação explícita das entradas e saídas de cada etapa da solução.

4.2 APLICAÇÃO DO *FRAMEWORK*

Para a avaliação do *framework* foram utilizados os dados cedidos por uma fundação pública de tecnologia da informação de uma prefeitura da região metropolitana de Porto Alegre. Esta fundação possui um setor que desenvolve projetos de *software* de acordo com as necessidades e demandas solicitadas pelo ente público.

Essas informações foram solicitadas à diretoria responsável pela equipe de desenvolvimento, que forneceu e contextualizou o processo de desenvolvimento praticado no

projeto. A equipe utiliza o método *Kanban*, para gerenciar o fluxo de trabalho, juntamente com práticas de *eXtreme Programming* em ciclos iterativos e entregas incrementais.

O fluxo de trabalho do setor de desenvolvimento é composto pelas seguintes fases:

- *Review*: etapa que os analistas elaboram as histórias de usuário junto ao cliente final.
- *Backlog*: fila de histórias de usuário prontas para serem estimadas pela equipe desenvolvimento.
- *Backlog* desejado: fila de histórias de usuário estimadas aguardando a prioridade para entrar no fluxo de desenvolvimento.
- Aguardando: etapa de entrada no processo de desenvolvimento onde os itens de trabalho que devem ser entregues aguardam a execução.
- Executando: fase de execução da tarefa.
- *Code Review*: etapa de revisão código implementado.
- Validando: estágio de validação do critérios de aceitação pelo analista que relatou a tarefa.
- Pronto: funcionalidades válidas para publicar no ambiente de homologação.
- Homologação: fase de publicação do incremento de software em ambiente específico para a homologação do cliente final.
- Produção: itens homologados e publicados no ambiente final de produção.

Com base neste cenário foi realizado o exercício de aplicar *framework*, as seções seguintes expõem o trabalho realizado seguindo as etapas definidas do *Framework* AMS.

4.2.1 Seleção das Métricas

Seguindo o *framework*, foi analisado o cenário com as premissas apresentadas no último tópico sobre o processo da fundação. Dessa forma foram identificadas as deficiências que a equipe tinha na gestão do projeto em questão, e assim os objetivos foram traçados, são eles:

- Saber o tempo de entrega das funcionalidades;
- Fornecer previsão de entrega do produto;
- Visualizar o trabalho em progresso.

Analisando os pontos a serem melhorados na gestão do processo foi consultado o conjunto de métricas indicado pelo *Framework* AMS e encontrou-se as métricas:

- Acurácia das estimativas (tempo);
- *Burndown* da iteração;
- *Burnup* da *release*/projeto;
- *Cycletime* (tempo de ciclo para concluir uma tarefa);
- Diagrama de fluxo cumulativo (CFD);
- Índice de histórias por iteração;
- *Leadtime* (tempo de entrega das funcionalidades ao cliente);
- Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões);
- Taxa de acerto na estimativa das tarefas;
- *Throughput* (número de tarefas entregues por período de tempo);
- *Work in Progress* (WIP).

A partir dessa lista verificou-se a possibilidade de extrair as medidas de tais métricas dos dados fornecidos. Os dados estavam estruturados em JSON (acrônimo de *JavaScript Object Notation*), que é um formato compacto de chave e valor, usado na troca de dados simples e rápida entre sistemas. Os dados correspondiam a um conjunto de datas associadas ao número de itens que constavam em cada etapa do processo de desenvolvimento.

A partir disso as métricas selecionadas para a continuação da aplicação do *framework* foram:

1. WIP: pois possibilita a equipe de visualizar o número de atividades em andamento no processo produtivo;
2. *Lead Time*: mapeia a quantidade de tempo que uma demanda gasta para passar por todo fluxo de desenvolvimento;
3. *Throughput*: mede o rendimento da equipe pelo número de tarefas entregues em um período de tempo;
4. Diagrama de fluxo cumulativo (CFD): exibe progresso do trabalho, identifica gargalos no fluxo e apresenta o estado atual do projeto.

Essas métricas atendem as características sugeridas por Hartmann e Dymond (2006), porque são métricas importantes que oferecem visibilidade do fluxo de trabalho, fornecem previsibilidade de entrega e servem como apoio na aplicação da melhoria contínua das equipes.

4.2.2 Definição da Medição

Para a aplicação das métricas *Work In Progress*, *Lead Time* e *Throughput* é necessária a existência de um quadro de acompanhamento diário das tarefas, onde é armazenado para cada item de trabalho o tempo que permaneceu em cada fase do quadro, até a sua conclusão.

Para o projeto em questão a coleta de dados foi feita de forma automatizada por meio do uso do *software Kanboard*, que faz a gestão de tarefas dentro do processo de desenvolvimento.

4.2.3 Desenvolvimento das Visualizações

Com as métricas e a medição definidas a próxima fase foi de elaborar as visualizações com base na literatura e lições de *storytelling*. Nas subseções seguintes são apresentadas a identificação do modelo gráfico de cada métrica.

4.2.3.1 Lead Time

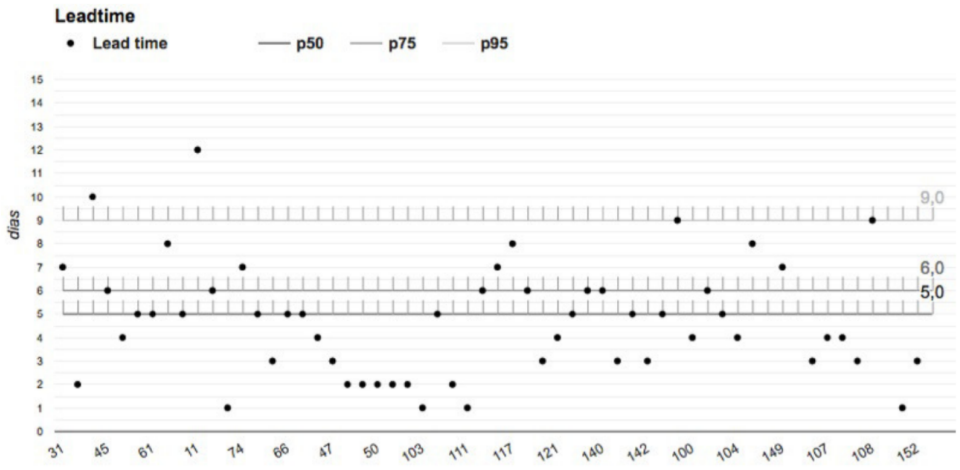
De acordo, com Albino (2017), as informações coletas de *Lead Time* podem ser apresentadas em um gráfico de dispersão para visualizar a variância do tempo de entrega, sendo que num processo estável a variância deve ser baixa. Com base no gráfico de dispersão e análise estatística, pode-se obter informações como média, moda, mediana, percentis que representam 50%, 75% e 95% do conjunto, assim podendo dar visibilidade sobre a tendência do tempo de entrega.

A Figura 7 exibe um exemplo de gráfico de dispersão, no eixo vertical é representado o número de dias que um item demorou para ser entregue e no eixo horizontal aponta o identificador de cada item trabalhado no período que está sendo analisado. As linhas *p50*, *p75* e *p95* reproduzem respectivamente o corte dos percentis de 50%, 75% e 95% do conjunto de tarefas.

4.2.3.2 *Lead Time Breakdown*

Outra proposta de visualização para o *Lead Time*, o *Lead Time Breakdown* que é um gráfico permite ao time avaliar, comparar e analisar em detalhes a latência entre o início e o término de cada uma das etapas do processo de desenvolvimento para cada item de trabalho que passa pelo fluxo (ALBINO, 2017).

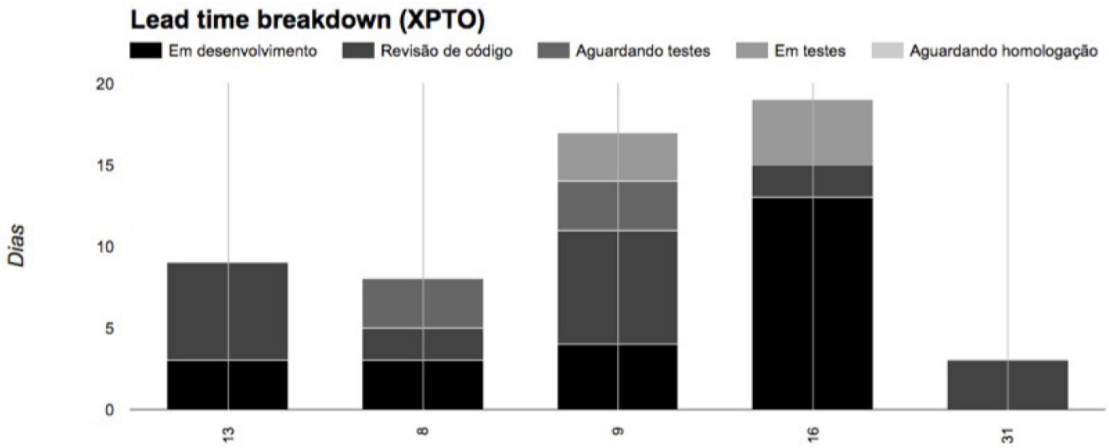
Figura 7: Exemplo de um gráfico de dispersão do *Lead Time*



Fonte: Albino (2017)

A forma de criar a visualização é simples. Como pode ser observado na Figura 8, no eixo vertical são expostos os dias de trabalho. Já no eixo horizontal, são exibidos os itens de trabalho em progresso ou já entregues pela equipe. Para representar cada uma das etapas do processo, é usado o gráfico de barra acumulado.

Figura 8: Exemplo de um gráfico do *Lead Time Breakdown*



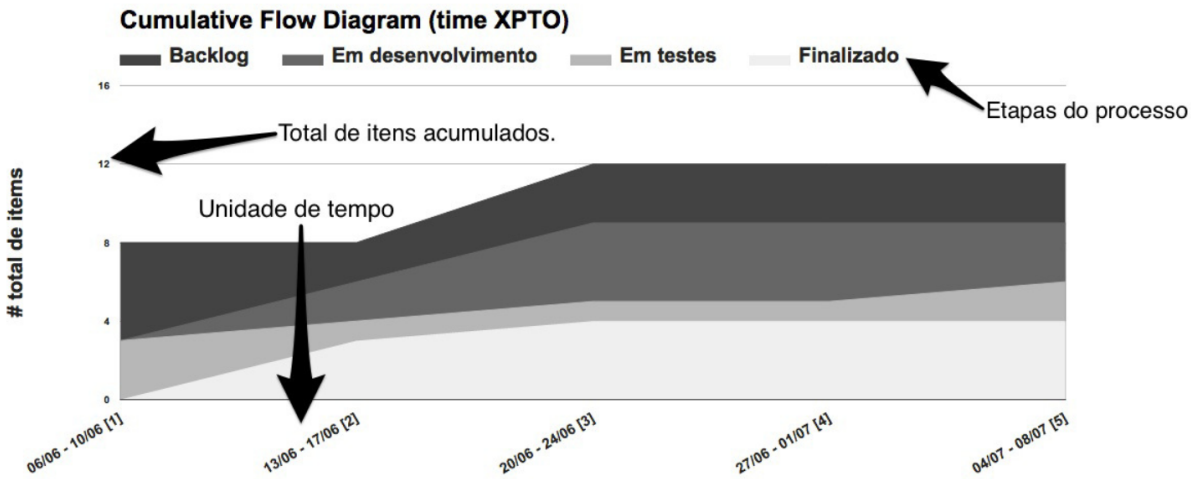
Fonte: Albino (2017)

4.2.3.3 Diagrama de fluxo cumulativo

O Diagrama de fluxo cumulativo ou CFD, do inglês *Cumulative Flow Diagram*, é um tipo de visualização que trata de entradas e saídas de um processo. O CFD é uma forma de compreender o fluxo de trabalho, pois o gráfico oferece um panorama geral do que está acontecendo durante o desenvolvimento de um projeto ou produto (ALBINO, 2017).

A Figura 9 exibe um exemplo de CFD, onde o eixo horizontal representa um período de tempo e o eixo vertical indica, de forma acumulada, o número de itens no processo. Cada área pintada no gráfico está relacionada a uma etapa do fluxo de trabalho e as curvas são basicamente o número de itens acumulados em tais etapas.

Figura 9: Exemplo de CFD

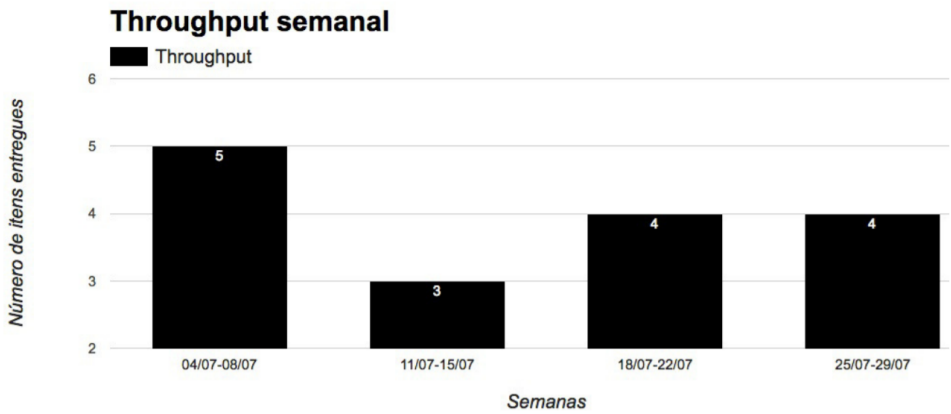


Fonte: Albino (2017)

4.2.3.4 Throughput

Para o autor Albino (2017), uma das formas de representar a métrica *Throughput* é a partir de um gráfico de colunas. No exemplo da Figura 10, é criada a visualização ao longo de um período (semanas) de qual a quantidade de entregas de um time de desenvolvimento. Sendo que o eixo vertical retrata o número de entregas e o eixo horizontal demonstra a unidade de tempo.

Figura 10: Gráfico de *Throughput* semanal

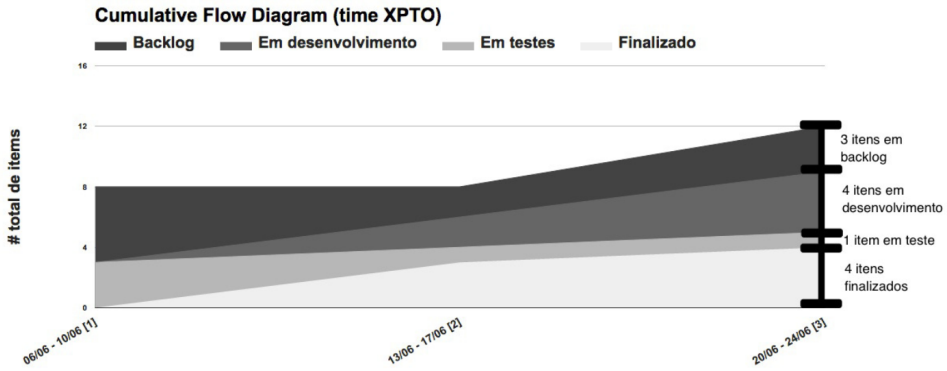


Fonte: Albino (2017)

4.2.3.5 WIP

Conforme Albino (2017), uma das maneiras de visualizar o WIP é através do CFD. Segundo o autor, dado que a linha superior representa o total de itens que entraram no processo de desenvolvimento do time, e a linha inferior o acumulado de itens entregues, o WIP pode ser aferido com a diferença vertical entre as linhas que delimitam o trabalho em progresso. A Figura 10 ilustra um exemplo.

Figura 11: WIP



Fonte: Albino (2017)

4.2.4 Desenvolvimento do Protótipo da Ferramenta

Este estudo optou por desenvolver um protótipo de ferramenta que proponha centralizar as visualizações das métricas e que fosse de fácil acesso a todos os envolvidos do projeto num cenário hipotético.

Para o desenvolvimento foi selecionado o ambiente *web*, tornando a aplicação multiplataforma, ou seja, acessível de diferentes dispositivos e sistemas operacionais, sem a necessidade de um processo de instalação, por exemplo.

Foram empregadas as tecnologias PHP (*Hypertext Preprocessor*), HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JS (*JavaScript*) para a construção de páginas para interação com usuários. Tais tecnologias são amplamente utilizadas no âmbito profissional, com fácil acesso a materiais de referência.

Para facilitar o desenvolvimento do protótipo, utilizou-se o *Laravel*, *framework* livre e de código aberto escrito na línguagem de programação PHP. O *Laravel* foi adotado pois fornece importantes funcionalidades para o início do desenvolvimento de um sistema.

Uma funcionalidade essencial é o sistema de rotas, que mapeia as requisições HTTP (*Hypertext Transfer Protocol*) do sistema para as áreas responsáveis por tratá-las, melhorando o desempenho do sistema. Neste caso, foi criado uma rota para cada uma das métricas selecionadas que foram mencionadas anteriormente. O tratamento das rotas consistiu em retornar o *template* da página de cada uma das métricas.

O *template* retornado pela rota foi construído usando outra funcionalidade essencial do *Laravel* que é o *Blade Engine*. Este é um sistema de *templates* responsável pela padronização da parte visual da aplicação, possibilitando maior otimização na construção de *templates* e facilitando a reutilização de códigos.

Então, foi criado um *template* base com o layout geral, que é reaproveitado em todas as páginas do protótipo e para cada métrica foi adicionado um *template* que estende a base contendo apenas os trechos específicos de cada interface que precisavam ser implementados.

Ainda com relação a *interface* gráfica, foi escolhido a implementação de componentes da biblioteca *Bootstrap*, que é desenvolvida com as tecnologias HTML, CSS e JS, visto que os componentes disponibilizam elementos bem estruturados em sua diagramação e também implementam *layout* responsivo.

Para a elaboração dos gráficos a ferramenta adotada foi o *Highcharts*, pois facilita a configuração de gráficos interativos em páginas da *web*. O *Highcharts* é uma API (*Application Programming Interface*) de gráficos de manuseio simples e flexível, mas destaca-se que seu uso é livre apenas para projetos educacionais. A escolha dessa ferramenta se deve ao fato que ela apresenta uma funcionalidade diferenciada, pois possibilita observar um período de tempo personalizado, sendo fácil a manipulação de um período específico, conforme a necessidade do usuário.

Durante o desenvolvimento dos gráficos das métricas foram aplicadas algumas lições de *storytelling* com dados. A primeira prática foi diminuir a saturação visual, que consistiu em remover os elementos dispensáveis que eram carregados por padrão no uso do *Highcharts* e que sobrecarregavam o visual do gráfico. Removendo a saturação, permite ao usuário uma melhor observação das métricas, focando no que é relevante, que são os dados das métricas.

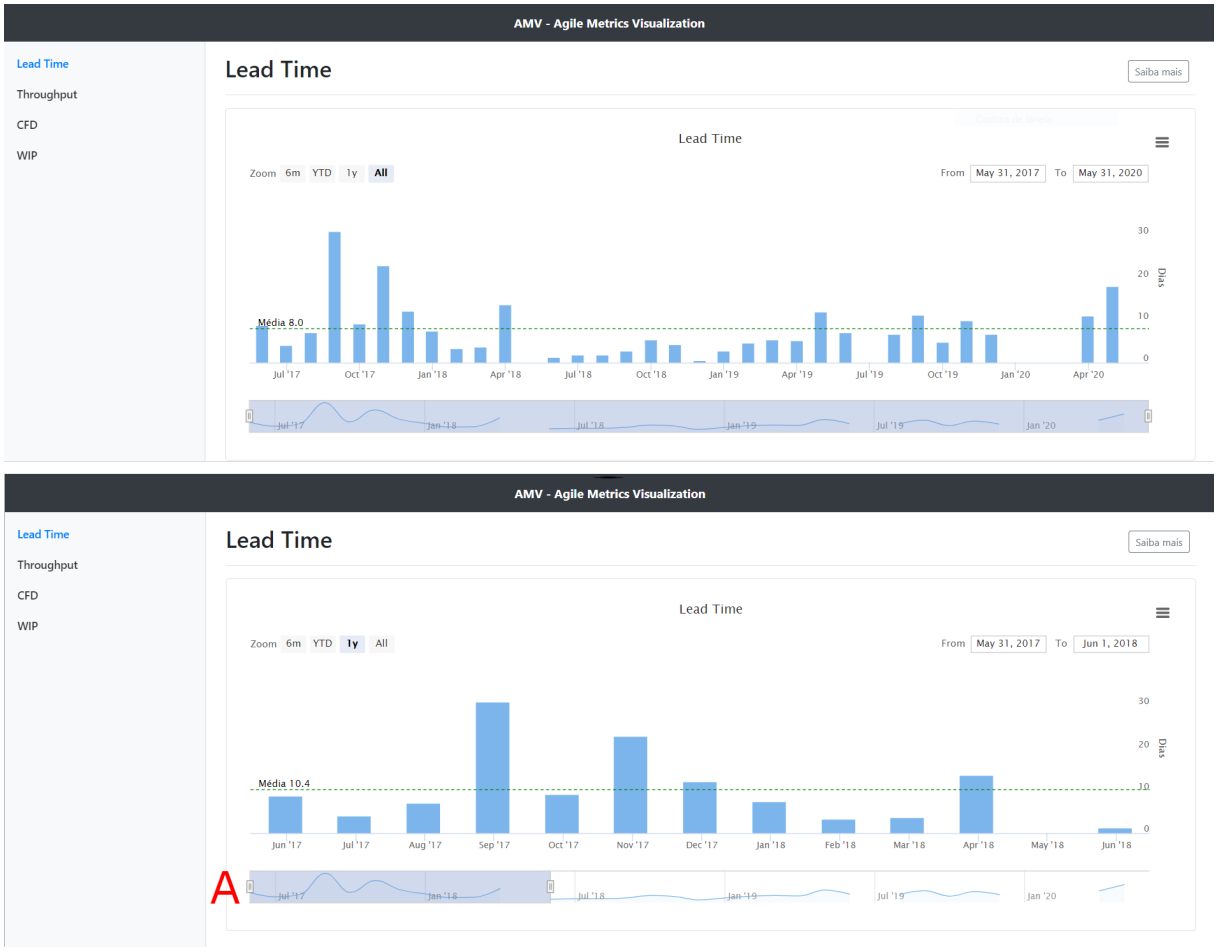
Na implementação de cada um dos gráficos foram utilizados os dados fornecidos em formato de JSON. Os dados eram compostos por um conjunto de datas associadas ao número de itens de trabalho que constavam em cada etapa do processo de desenvolvimento. Para obter o *dataset* necessário para o *Highcharts* gerar as visualizações, foi elaborado um algoritmo em *Javascript* que manipulou o JSON.

Com base nisso, para o *dataset* do CFD foi necessário realizar a soma acumulativa da quantidade de itens em cada etapa do processo de desenvolvimento. A métrica WIP pode ser aferida através da média mensal da soma dos itens que estavam nas etapas Aguardando (fase de início do processo), Executando, *Code Review* e Validando (última fase de desenvolvimento). Já o *Throughput* foi calculado pela média mensal da soma dos itens entregues que acumulavam nas etapas do processo Pronto, Homologação e Produção. O *Lead Time* foi a única métrica que não pode calculada diretamente. Então, para chegar

ao resultado do foi aplicada a Lei de *Little*, seguindo a fórmula $L = W/\lambda$, e assim, pode-se obter a média mensal de *Lead Time*.

Na Figura 12 é possível observar a tela desenvolvida com o gráfico de barras verticais para o *Lead Time* em dois momentos distintos. O primeiro é exibindo todo o histórico de dados disponível, e o segundo a tela mostra os dados referentes ao primeiro ano de execução do projeto. Na figura é possível observar o componente que permite a navegação personalizada ao longo de todo o período de data dos dados, a marcação em vermelho com a letra "A"foi adicionada sobre a imagem para facilitar a identificação do componente.

Figura 12: Telas com *Lead Time* em períodos de tempo distintos

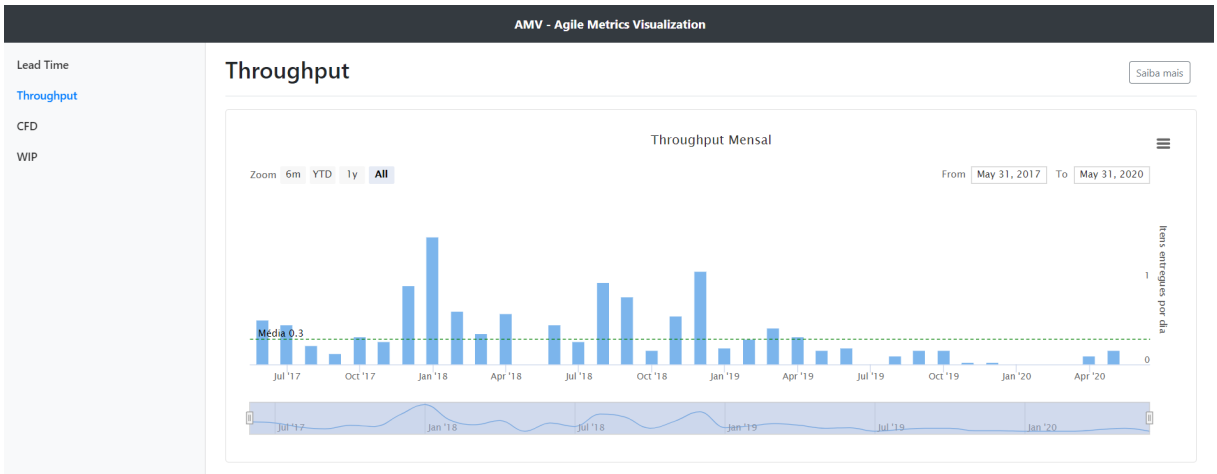


Fonte: elaborado pelo autor

A tela do protótipo com o *Throughput* é visto na Figura 13, foi construído um gráfico de barras verticais para esta métrica também. Na figura pode ser observado todo o histórico de dados disponível, e com o componente de navegação personalizada é possível selecionar um período específico para visualizar o gráfico.

A visualização do CFD é um gráfico do tipo área, exibindo a quantidade acumu-

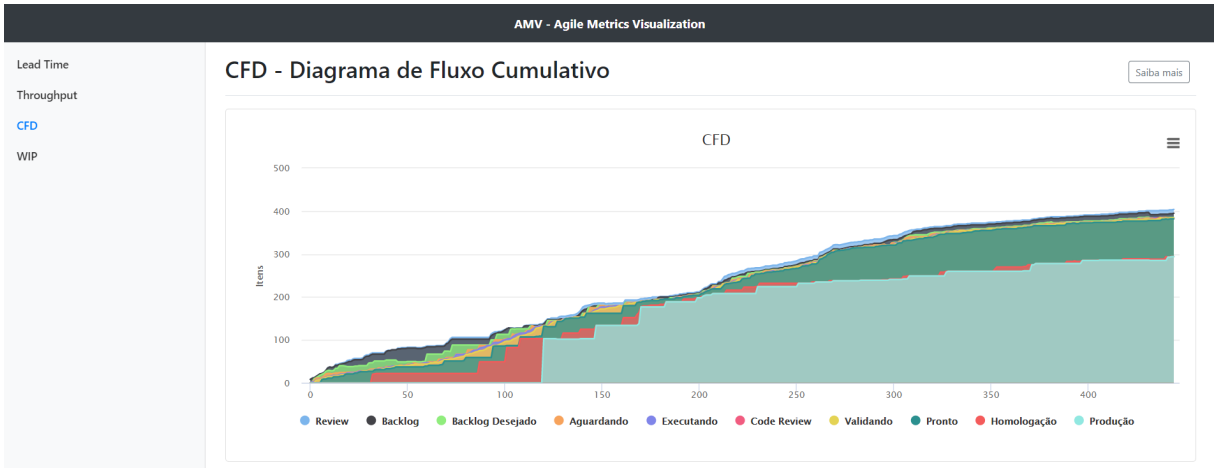
Figura 13: Tela do *Throughput*



Fonte: elaborado pelo autor

lada em cada etapa do processo ao longo do período de tempo, como mostra a Figura 14.

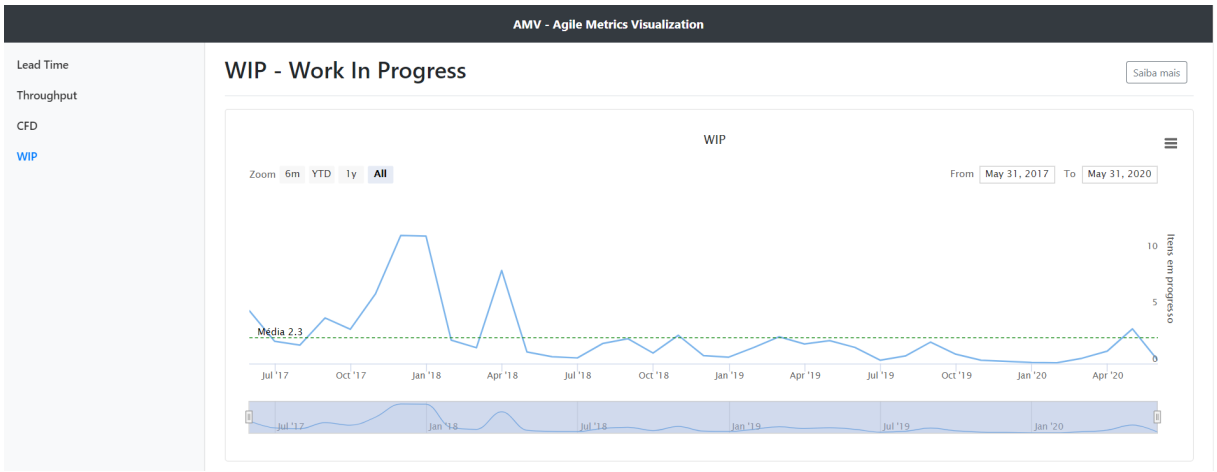
Figura 14: Tela do CFD



Fonte: elaborado pelo autor

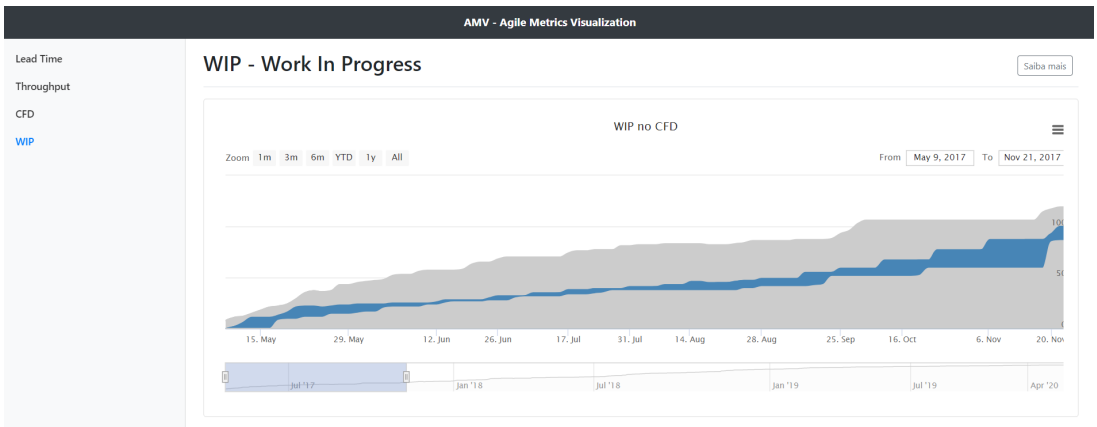
Para a métrica WIP foi implementado duas visualizações, uma em linhas e a outra reutilizando a estrutura do gráfico CFD, conforme é ilustrado respectivamente na Figura 15 e Figura 16. Considerando o cenário dos dados foi definido a etapa Aguardando como início e a Validando como término do processo para alcançar os gráficos de WIP.

Figura 15: Tela do WIP



Fonte: elaborado pelo autor

Figura 16: Tela do WIP no CFD



Fonte: elaborado pelo autor

4.2.5 Melhoria Contínua

Neste cenário desenvolvido, a etapa de melhoria contínua do *framework* não se aplica, pois ele já foi finalizado. Essa etapa no *framework* indica que a ferramenta desenvolvida deve ser empregada nas reuniões de restrospectiva, com a finalidade de apoiar a equipe em exercer a filosofia de melhoria contínua baseada em dados, a partir de análises feitas de forma colaborativa.

4.3 ENTREVISTA

A entrevista elaborada para realizar a avaliação do *framework* ocorreu abordando três profissionais que trabalham com metodologias ágeis e que podiam trazer contribuições sobre a proposta deste trabalho e verificar a sua validade e aplicação no mercado de trabalho atual.

Para isso, foi realizado um roteiro para a entrevista dessas pessoas, e não um simples questionário fechado, uma vez que foi necessário apresentar o *framework* desenvolvido nesta pesquisa, para depois colher as respostas das questões.

O roteiro da entrevista está disponível no Apêndice A. O mesmo é composto por perguntas iniciais, para traçar o perfil dos entrevistados, e traz nove questões ao total. Destas questões, seis são questões avaliativas e três são dissertativas, onde os usuários opinaram sobre a ferramenta e traziam contribuições para melhoria. Para as questões avaliativas foi usado uma escala ordinal de 1 a 5, sendo 1 o valor para discordo totalmente e 5 o valor para concordo totalmente.

4.4 PERFIL DOS ENTREVISTADOS

Na entrevista não há informações nominais, sendo que os três participantes são profissionais com experiência como *scrum master* e ou gerenciamento de projetos. Foram elaborados os quadros: Quadro 4, Quadro 5 e Quadro 6 para sintetizar o perfil dos entrevistados juntamente com as suas respostas dissertativas.

Embora o número de entrevistados seja pequeno, verifica-se que eles possuem uma larga experiência no campo das metodologias ágeis. Seguem os quadros elaborados.

Quadro 4: Perfil e repostas dissertativas do entrevistado 1

Perfil do entrevistado	Especialista do mercado
Descrição da empresa	Empresa pública de médio porte que possui experiência em métodos ágeis.
Descrição do entrevistado	Possui experiência como engenheiro de software, trabalhando há 5 anos com metodologias ágeis, sendo que já atuou em alguns projetos como *scrum master*
Qual a sua opinião em relação ao *framework* proposto?	"Ele é bom, aplica bem as métricas, eu gostei do fato dele ser um incremento em cima de algo que já existe, e não uma mudança de paradigma. Tem a questão dos dados que dificulta aplicar ele totalmente, no decorrer tu terá que aplicar o fluxo inteiro, pra ti ir construindo a questão do *storytelling*, a questão de métricas é super importante e nenhuma das métricas é individual, todas elas são em equipe, e isso vai na linha de métodos ágeis, não medir a pessoa e sim a equipe."
Qual a sua opinião sobre os instrumentos de apoio (mapa mental e infográfico)?	"Ficou bem claro, o design tá bonito e o layout também, e tu conseguiu explicar tudo em uma tela só."
Qual seria a utilidade do framework proposto para as empresas?	"A utilidade é grande porque no método ágil as métricas são muito importantes e o pessoal sempre têm muitas dúvidas em como aplica as métricas, então o framework que foca nessa questão de métricas e indicadores e tudo mais em cima de algo que já existe é útil."

Fonte: elaborado pelo autor

Quadro 5: Perfil e repostas dissertativas do entrevistado 2

Perfil do entrevistado	Especialista do mercado
Descrição da empresa	Empresa privada de médio porte que possui experiência em métodos ágeis.
Descrição do entrevistado	Atua em cargo de gestão, exerce a função de líder técnico de equipe de desenvolvimento e scrum master, utilizando métricas ágeis nos projetos a 5 anos.
Qual a sua opinião em relação ao *framework* proposto?	"Ele contempla a metodologia como um todo e ele é simples, para uma proposta inicial ele está bom, mas eu acrescentaria algumas opções. No contexto geral ele é bom, seguindo os passos que você colocou se consegue obter um resultado."
Qual a sua opinião sobre os instrumentos de apoio (mapa mental e infográfico)?	"Eu achei que ficou bem clean, ficou simples e contendo todos os argumentos necessários para uma explicação."
Qual seria a utilidade do *framework* proposto para as empresas?	"Seria para ganho de performance do time, gestão de tempo do time e visão de entrega."

Fonte: elaborado pelo autor

Quadro 6: Perfil e repostas dissertativas do entrevistado 3

Perfil do entrevistado	Especialista do mercado
Descrição da empresa	Empresa privada de médio porte que possui experiência em métodos ágeis.
Descrição do entrevistado	Atua como Agile Coach, é pós-graduado em metodologias ágeis, possui 12 anos de experiência na área.
Qual a sua opinião em relação ao framework proposto?	"Eu achei ele muito bom, são métricas que o mercado está utilizando bastante, o que tu falou aqui conectou bastante com as métricas que utilizamos na empresa, a única dificuldade que eu vejo é a das pessoas é de interpretarem isso, elas tem bastante dificuldade de compreender o que são essas métricas, como tomar decisões em cima disso e o que fazer com esses dados."
Qual a sua opinião sobre os instrumentos de apoio (mapa mental e infográfico)?	"Gostei do mapa mental, achei que ele ajudou mais que o infográfico do framework, eu acho que poderia ser menos texto e mais figuras, algo assim, eu tive mais atenção no primeiro momento."
Qual seria a utilidade do framework proposto para as empresas?	"Justamente a tomada de decisões, previsibilidade, planejamento, pois isso são pilares para a tomada de decisões. Como trabalhar com a previsão de entrega, eu posso avaliar o Throughput do meu time, o WIP, para saber se não estou me comprometendo com muita coisa, uma coisa acaba conectando com a outra, e o CFD vai me ajudar a ter uma visão total do meu processo para saber se eu estou tendo algum tipo de gargalo. Essas métricas são básicas para qualquer time de tecnologia ou qualquer empresa que trabalha com algum tipo de processo, ele é super importante para a tomada de decisões, sendo para melhoria do processo, melhoria contínua, seja previsibilidade de entrega."

Fonte: elaborado pelo autor

4.5 CONSIDERAÇÕES SOBRE A AVALIAÇÃO

Após a coleta dos dados iniciais, foi apresentado ao entrevistado o embasamento teórico do trabalho utilizando o mapa mental, e a solução proposta, apresentada através do infográfico, neste caso o *framework*.

A partir disso, foram submetidas as perguntas avaliativas para o entrevistado avaliar as questões elaboradas. Apesar do número reduzido de entrevistados, foi criado o Quadro 7 com a média do resultado das questões e o desvio padrão, que consiste na medida de dispersão em torno da variável.

Quadro 7: Média e desvio padrão das respostas dos entrevistados sobre a avaliação do *framework*

Pergunta	**Média das respostas**	**Desvio padrão**
Quanto à compreensão do framework, você o considera simples de ser entendido.	4,6	0,6
Quanto à aplicação do framework, você o considera fácil de ser aplicado.	4,3	0,6
Quanto à aplicação do framework, você o considera ser utilizado em um cenário real.	4,6	0,6
Quanto ao alinhamento com a abordagem ágil, o framework respeita os valores e princípios ágeis.	5	0
Quanto ao método storytelling com dados, o framework emprega as lições apresentadas.	3,3	0,3
Quanto aos benefícios do framework, é fornecido orientações que permitem a empresa refletir sobre os seus objetivos e necessidades de monitoramento do processo através do uso das visualizações das métricas.	4,6	0,6

Fonte: elaborado pelo autor

Com base na média das respostas apresentadas, foi possível verificar as seguintes considerações:

- Os entrevistados responderam que a compreensão do *framework* é simples e fácil de ser compreendida, nenhum dos entrevistados reportou algum problema em sentido contrário;
- Com base na aplicação do *framework*, segundo os entrevistados, seria de fácil aplicação. Porém, com algumas ressalvas quanto a ter mais opções partindo do método ágil que é adotado e não somente tratando os métodos de uma maneira geral. Outra ressalva foi ter um cuidado especial com relação à coleta de dados, na etapa Definição da Medição;

- Quando questionados sobre a aplicação do *framework* em um cenário real, os entrevistados relataram que seria possível, tendo em vista as necessidades atuais do mercado;
- Quanto ao alinhamento do *framework* em referência aos valores e princípios ágeis, os entrevistados julgaram compatível, de forma unânime;
- No que diz respeito ao emprego do método *storytelling* com dados, os entrevistados consideraram que o objetivo foi parcialmente cumprido;
- Sobre benefícios trazidos que são demonstrados no *framework*, todos os entrevistados reconheceram que o uso de métricas possibilitam um maior planejamento, acompanhamento e visibilidade do trabalho da equipe em relação aos projetos.

Após as questões avaliativas respondidas pelos entrevistados, foram formuladas três perguntas dissertativas que possibilitaram que o avaliador expressasse sua opinião de forma mais detalhada. Estas respostas constam nos quadros: Quadro 4, Quadro 5 e Quadro 6.

Na primeira pergunta, questiona-se a opinião geral dos participantes sobre *framework*. Os avaliadores o estimaram bom, de fácil entendimento, objetivo, de forma que seguindo os passos estabelecidos consegue-se chegar em um resultado, e que também está de acordo com as necessidades do mercado. Mas destacaram que existem coisas a serem acrescentadas ou alteradas, como por exemplo, que o *framework* seja mais explicativo, e possibilite uma maior compreensão para quem está tendo um primeiro contato com a metodologia ágil.

Quanto aos instrumentos aplicados nas entrevistas, que foram o mapa mental (Figura 5) e o infográfico (Figura 6), os avaliadores demonstraram uma satisfação, referindo que as informações estão bem sintetizadas e contém os argumentos necessários para a aplicação do *framework*. Entretanto, um dos avaliadores sugeriu que o infográfico tivesse mais elementos visuais para melhor ilustrar as etapas, pois em comparação ao mapa mental o mesmo ficou mais atrativo.

Sobre a utilidade do *framework*, todos consideraram de forma positiva, uma vez que o uso das métricas em processos ágeis é de grande importância para equipes de desenvolvimento de software, porque auxiliam na tomada de decisões, no planejamento do projeto e na visualização do trabalho em progresso.

Com base em todas as respostas, concluiu-se que embora existam aprimoramentos a serem feitos no *framework*, há um reconhecimento unânime por parte dos avaliadores. A proposta traz benefícios às equipes de desenvolvimento que empregam o uso de métodos ágeis, tais como, previsibilidade, planejamento e visibilidade, sendo uma ótima ferramenta a ser utilizada nos projetos.

Portanto, embora haja algumas dificuldades na aplicação do *framework*, seja com relação à coleta dos dados ou a dificuldade para iniciantes com a abordagem ágil, verificou-se a validade do *framework* proposto para estes três entrevistados, uma vez que vai de encontro com as necessidades de mercado e segue os valores e princípios das metodologias ágeis.

5 CONCLUSÃO

Primeiramente identificou-se que o mercado de TI vem adotando cada vez mais metodologias ágeis para a organização de suas equipes e desenvolvimento de seus projetos. As tradicionais práticas de monitoramento e controle dos projetos não atendem as necessidades para processos ágeis de desenvolvimento. Visto isso, este trabalho abordou o uso de métricas ágeis para auxiliar a aplicação de métricas aderentes aos métodos ágeis de desenvolvimento de *software*. O trabalho compreendeu também a elaboração de visualizações das métricas aplicando as lições de *storytelling* com dados.

Com base nisso, o objetivo do trabalho foi desenvolver um *framework* conceitual para a aplicação de métricas que são adotadas na avaliação de desempenho de processos ágeis no desenvolvimento de software.

Para realizar a construção deste artefato foi realizada uma pesquisa que apresentou a fundamentação teórica necessária para a elaboração do *framework*. Desta forma, constatou-se quais características as métricas devem ter para se adequarem em ambientes ágeis. Por exemplo, as métricas devem reforçar os princípios ágeis, incentivar a comunicação e fornecer *feedbacks* frequentes.

Essa pesquisa focou nas métricas da categoria de processo, pois abrange as métricas que avaliam aspectos relacionados ao desenvolvimento iterativo e incremental e gestão dos itens de trabalho. Para tal, foi dado ênfase para três métricas: WIP, que visa identificar a capacidade de produção da equipe; *Lead Time*, que tem por objetivo de compreender quanto tempo a equipe tem levado para desenvolver um item de trabalho; e *Throughput* que mede o rendimento da equipe de desenvolvimento. E após foi descrito a relação entre essas métricas definida pela Lei de *Little*. Para a construção das visualizações foram utilizados as lições trazidas pela técnica de *storytelling* com dados.

O *framework* proposto, nomeado *Framework* AMS, define um ciclo lógico de atividades, apresentado através de uma sequência de fases que permite a uma equipe ágil planejar, executar, e melhorar a forma de realizar o acompanhamento de seus projetos através de métricas, respeitando as características dos projetos ágeis de desenvolvimento de *software*.

O objetivo principal do *framework* é servir de apoio para que equipes possam aplicar as métricas que agregam mais valor à evolução do time. Após isso, é importante dar visibilidade dos resultados das métricas para que as equipes possam acompanhá-las e melhorar o processo a cada nova iteração.

Para avaliar o *Framework* AMS desenvolveu-se um roteiro de entrevista que foi submetido a um grupo de três especialistas na área com experiência em metodologias

ágeis.

Inicialmente, foi coletado dados para traçar o perfil dos entrevistados. Após sucedeu a apresentação do trabalho desenvolvido em dois momentos. Primeiro, com o uso do mapa mental criado, foi demonstrado a motivação e o embasamento teórico desenvolvido. O segundo momento foi apresentar o *Framework* AMS por meio do infográfico elaborado e exemplificando o uso do mesmo com o protótipo de ferramenta *web* implementada.

No protótipo desenvolvido a partir de dados cedidos por uma fundação pública de TI foram aplicadas as métricas estudadas WIP, *Lead Time* e *Throughput*. Os gráficos gerados no protótipo passaram por um refinamento das visualizações das métricas conforme as lições do *storytelling* com dados.

Então foi aplicado questões avaliativas e dissertativas aos entrevistados, que demonstraram uma boa aceitação da proposta desenvolvida. Baseado nas respostas, concluiu-se que embora existam aprimoramentos a serem feitos no *framework*, há um reconhecimento pelos avaliadores que ele oferece benefícios às equipes de desenvolvimento que empregam o uso de método ágeis. Tais como, previsibilidade, planejamento e visibilidade, sendo assim, uma ótima ferramenta a ser aplicada nos projetos.

Portanto, com base nos resultados obtidos é possível indicar que o *Framework* AMS cumpriu com o objetivo estabelecido neste trabalho, que é criar um artefato para auxiliar as empresas na aplicação de métricas em processos ágeis.

Embora existam melhorias a serem feitas no *framework* e apesar da pequena amostragem de entrevistas, ele pode ser considerado relevante, pois atende a necessidade das equipes de desenvolvimento. Uma vez que o uso de métricas ágeis, juntamente com as suas visualizações aplicando o *storytelling*, oferece uma melhor visão do projeto em andamento para que a equipe possa ter um melhor desempenho, estar em constante aperfeiçoamento e facilitando a tomada de decisões.

Como sugestões para trabalhos futuros ficam, a avaliação do *framework* com uma amostragem maior de entrevistados e a validação do artefato por meio de execução de um estudo de caso, aplicando o *Framework* AMS em uma empresa de desenvolvimento.

REFERÊNCIAS

ALBINO, R. D. *Métricas Ágeis: obtenha melhores resultados em sua equipe*. São Paulo: Casa do Código, 2017.

ANDERSON, D. J. *Kanban: Mudança evolucionária de Sucesso para seu Negócio de Tecnologia*. [S.l.]: Blue Hole Press, 2011.

AZEVEDO, D. Revisão de literatura, referencial teórico, fundamentação teórica e framework conceitual em pesquisa – diferenças e propósitos. Working paper, 2016. Disponível em: <https://unisinos.academia.edu/DeboraAzevedo/Papers>. Acesso em: 31 mai. 2020.

BECK, K. et al. Manifesto para desenvolvimento Ágil de software. 2001. Disponível em: <http://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 11 nov. 2018.

CHENG, T.-H.; JANSEN, S.; REMMERS, M. Controlling and monitoring agile software development in three dutch product software companies. *2009 ICSE Workshop On Software Development Governance*, IEEE, Washingtion, DC, p. 29–35, 2009. Disponível em: <https://www.computer.org/csdl/proceedings/sdg/2009/3736/00/05071334.pdf>. Acesso em: 15 nov. 2018.

DRESCH, A.; LACERDA, D. P.; ANTUNES, J. A. V. *Design Science Research: método de pesquisa para avanço da ciência e tecnologia*. São Paulo: Bookman, 2015.

FENNER, G. *Mapas mentais: potencializando ideias*. Rio de Janeiro: Brasport, 2017.

FOWLER, M. The new methodology. 2005. Disponível em: <https://www.martinfowler-.com/articles/newMethodology.html>. Acesso em: 15 nov. 2018.

GOMES, A. F. *Agile desenvolvimento de software com entregas frequentes e foco no valor de negócio*. São Paulo: Casa do Código, 2013.

GUSTAFSSON, J. Model of agile software measurement: A case study. University of Gothenburg, Gotemburgo, Suécia, 2011. Disponível em: <http://publications.lib-.chalmers.se/records/fulltext/143815.pdf>. Acesso em: 15 nov. 2018.

HARTMANN, D.; DYMOND, R. Appropriate agile measurement: using metrics and diagnostics to deliver business value. *IEEE transactions on knowledge and data engineering*, IEEE, Minneapolis, MN, p. 126–134, 2006. Disponível em: <https://ieeexplore.ieee.org/document/1667571/>. Acesso em: 7 out. 2018.

IEEE. Ieee standard glossary of software engineering terminology. IEEE, 1990.

ISO, I. O. f. S. ISO/IEC 15939: Software engineering - software measurement. ISO, 2007.

KNAFLIC, C. N. *Storytelling com dados: um guia sobre visualização de dados para profissionais de negócios*. Rio de Janeiro: Alta Books, 2017.

KOSARA, R.; MACKINLAY, J. Storytelling: the next step for visualization. *IEEE transactions on knowledge and data engineering*, IEEE, v. 46, n. 5, p. 44–50, 2013. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/MC.2013.36>. Acesso em: 7 out. 2018.

LACERDA, D. P. et al. Design science research: método de pesquisa para a engenharia de produção. *Gestão e Produção*, 2013. Disponível em: <http://www.scielo.br/pdf/gp-/2013nahead/aop_gp031412.pdf>. Acesso em: 7 out. 2018.

LIECHTI, O.; PASQUIER, J.; REIS, R. Beyond dashboards: On the many facets of metrics and feedback in agile organizations. *ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, IEEE, 2017.

LITTLE, J. A proof for the queuing formula: L=λw. *Operations Research*, $v.\ 9, n.\ 3, p.383-$ $-387, 1961$.

MCGARRY, J. et al. *Practical Software Measurement: Objective Information for Decision Makers*. [S.l.]: Addison-Wesley Professional, 2002.

PAULK, M. C. Extreme programming from a cmm perspective. *IEEE transactions on knowledge and data engineering*, IEEE, v. 18, n. 6, p. 19–26, 2001. Disponível em: <https://ieeexplore.ieee.org/document/965798>. Acesso em: 17 nov. 2018.

PEGORARO, R. A. Métricas de avaliação para abordagens ágeis em projetos de software. Tese (Doutorado em Engenharia) - Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul, 2014. Disponível em: <https://lume.ufrgs.br/handle/10183/101177>. Acesso em: 7 out. 2018.

PHAM, A.; PHAM, P. van. *Scrum em ação - gerenciamento e desenvolvimento ágil de projetos de software*. São Paulo: Editora Novatec, 2011.

PIMENTEL, M. Design science research e pesquisas com os cotidianos escolares para fazer pensar as pesquisas em informática na educação. *Anais do XXVIII Simpósio Brasileiro de Informática na Educação*, 2017. Disponível em: <http:/-/www.br-ie.org/pub/index.php/sbie/article/view/7570/5366>. Acesso em: 7 out. 2018.

POONACHA, K.; BHATTACHARYA, S. Towards a framework for assessing agility. *45th Hawaii International Conference on System Sciences*, IEEE, p. 5329–5338, 2012. Disponível em: <https://ieeexplore.ieee.org/document/6149539>. Acesso em: 15 nov. 2018.

POPPENDIECK, M.; POPPENDIECK, T. *Implementando o desenvolvimento Lean de software*. [S.l.]: Bookman, 2011.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. Porto Alegre: AMGH, 2016.

PRODANOV, C. C.; FREITAS, E. C. de. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico*. 2. ed. Novo Hamburgo: Universidade Feevale, 2013. Disponível em: <http://www.feevale.br/Comum/midias/8807f05a-14d0-4d5b-b1ad-1538f3aef538/E-book%20Metodologia%20do%20Trabalho%20Cientifico.pdf>. Acesso em: 7 out. 2018.

SABBAGH, R. *Scrum: Gestão ágil para projetos de sucesso*. São Paulo: Casa do Código, 2013.

SATO, D. T. Uso eficaz de métricas em métodos ágeis de desenvolvimento de software. Dissertação de Mestrado (Mestre em Ciências - Ciências de Computação e Matemática Computacional) - Intituto de Ciências Matemáticas e de Computação - ICMC/USP, 2007. Disponível em: <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-06092007-225914/pt-br.php>. Acesso em: 15 nov. 2018.

SOMMERVILLE, Y. *Engenharia de software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Disponível em: <http://www.feevale.br/Comum/midias/8807f05a-14d0-4d5b-b1ad-1538f3aef538/E-book%20Metodologia%20do%20Trabalho%20Cientifico.pdf>. Acesso em: 7 out. 2018.

VACANTI, D. S. *Actionable Agile Metrics for Predictability*. [S.l.]: Daniel S. Vacanti, Inc., 2015.

WILDT, D. et al. *eXtreme Programming: Práticas para o dia a dia no desenvolvimento ágil de software*. São Paulo: Casa do Código, 2015.

Apêndices

APÊNDICE A – ROTEIRO DA ENTREVISTA

Perfil do entrevistado

- Atividade profissional:
- Número aproximadamente de funcionários:
- Idade:
- Formação:
- Possui experiência profissional com metodologias ágeis:
- Desde que ano:

Perguntas iniciais

- Na empresa é empregado o uso de métricas para acompanhamento dos projetos desenvolvidos?
- Se sim, quais métricas?

Roteiro de apresentação do *framework* proposto.

- Apresentação do mapa mental expondo o embasamento utilizado para o desenvolvimento do *framework*.
- Apresentação do *framework* proposto utilizando o infográfico e o protótipo desenvolvido.

Avaliação do *framework* proposto considerando a escala ordinal de 1 a 5, sendo 1 o valor para discordo totalmente e 5 o valor para concordo totalmente:

- Quanto à compreensão do *framework*, você o considera simples de ser entendido.
- Quanto à aplicação do *framework*, você o considera fácil de ser aplicado.
- Quanto à aplicação do *framework*, você o considera ser utilizado em um cenário real.

- Quanto ao alinhamento com a abordagem ágil, o *framework* respeita os valores e princípios ágeis.
- Quanto ao método *storytelling* com dados, o *framework* emprega as lições apresentadas.
- Quanto aos benefícios do *framework*, é fornecido orientações que permitem a empresa refletir sobre os seus objetivos e necessidades de monitoramento do processo através do uso das visualizações das métricas.

Perguntas Finais

- Qual a sua opinião em relação ao *framework* proposto?
- Qual a sua opinião sobre os instrumentos de apoio (mapa mental e infográfico)?
- Qual seria a utilidade do *framework* proposto para as empresas?

Agradecimentos