



**ANÁLISE DE MÉTRICAS LEANS E ÁGEIS PARA AUXILIAR NA TOMADA DE  
DECISÃO POR MEIO DE UMA FERRAMENTA DESENVOLVIDA PARA  
INTEGRAR A UMA PLATAFORMA DE PORTFÓLIO E DE PROJETOS**

***LEANS AND AGILE METRICS ANALYSIS TO AID IN DECISION MAKING  
THROUGH A TOOL DEVELOPED TO INTEGRATE A PORTFOLIO AND PROJECT  
PLATFORM***

Leonardo Ferreira - mail@leonardoferreira.com.br

José Eduardo Ribeiro - jeribeiro@uniara.com.br

Fabiana Florian - fflorian@uniara.com.br

Universidade de Araraquara (UNIARA) - Araraquara - SP - Brasil

**RESUMO**

Este artigo tem como principal objetivo evidenciar a importância de métricas Leans e ágeis, e a partir delas tomar decisões sobre as próximas entregas de um time de desenvolvimento de software. Durante três meses uma equipe foi submetida a extração de métricas com o auxílio de uma ferramenta de código aberto criada especificamente para isso. Ao longo do período observado é possível verificar a vazão e o tempo médio para desenvolver tarefas de acordo com o esforço e complexidade. Esses dados auxiliaram para que estimativas de entrega de novas funcionalidades fossem feitas sobre dados históricos, fornecendo maior assertividade no processo de desenvolvimento de software.

**Palavras Chaves:** Métricas. Jira. Lean. Estimativa.

**ABSTRACT**

This article has as main objective to highlight the importance of Leans and Agile metrics, and from them to make decisions about the next deliveries of a software development team. For three months a team was subjected to extraction of metrics with the aid of an open source tool created specifically for this. Throughout the observed period it is possible to verify the flow and the average time to develop tasks according to the effort and complexity. These data helped to make estimates of the delivery of new functionalities made on historical data, providing greater assertiveness in the software development process.

**Keywords:** Metrics. Jira. Lean. Estimates.



## 1 INTRODUÇÃO

Um processo de desenvolvimento de software pode ser visto como um conjunto de atividades organizadas e utilizadas para definir, desenvolver, testar e manter um *software*.

Desde o começo do desenvolvimento de *software* essas ações vêm sendo guiadas por um processo, seja esse processo dos modelos mais tradicionais, ou com modelos ágeis de desenvolvimento.

A metodologia ágil de desenvolvimento de software, consolidou-se nos últimos anos como uma alternativa para atender projetos de forma dinâmica e flexível em relação a escopo visando o sucesso do projeto. É utilizada uma abordagem de planejamento iterativa com iterações cada vez menores para que seja possível uma entrega com ciclos cada vez mais constantes (SUTHERLAND, 2014).

O *Scrum* é um dos mais conhecidos *frameworks* baseado em metodologias ágeis, que descreve as iterações como *Sprints* que devem durar entre 1 a 4 semanas e algumas cerimônias como *Sprint Review* onde é feita uma revisão do que foi entregue. *Sprint Retrospective* onde é feita uma retrospectiva da *Sprint* para refinamento e aprendizado para as seguintes iterações. *Sprint Planning* onde é feito o planejamento das histórias que deverão ser entregues e *Daily Scrum*, realizada diariamente e tem como principal objetivo a identificação de possíveis impedimentos que possam afetar a entrega da *Sprint* (SUTHERLAND, 2014).

Existem também metodologias para desenvolvimento de software que seguem os princípios *Lean*.

*Lean* é uma metodologia de produção fabril criada após a segunda guerra mundial por Taiichi Ohno na Toyota. Depois da segunda guerra mundial a Toyota precisava de uma metodologia de trabalho para concorrer com o mercado de veículos ocidental, Taiichi Ohno então desenvolveu o Sistema Toyota de Produção que veio depois a se tornar o sistema *Lean* de produção. Esses sistemas seguem os princípios *Just-In-Time* que dizem que um item só deverá chegar na linha de montagem no momento de ser utilizada, redução de desperdícios e é caracterizado como um sistema puxado que diz que os produtos devem ser puxados pelo cliente e não empurrados para o fim da linha de produção (JONES; WOMACK, 2004).

O *Kanban* é uma metodologia para desenvolvimento de software baseada no *Lean* idealizado em 2006 por David Anderson que tem como principal tarefa “traduzir” o Sistema Toyota de Produção e o *Lean* para o trabalho criativo. A grande revolução para as metodologias fabris foi o controle de estoque, muito estoque significava desperdício, por ter produtos parados que poderiam acabar não sendo utilizados, quando isso foi traduzido para processos criativos por David Anderson, chegou-se à conclusão que o estoque quando se trabalha com processos criativos é o que começa mas não termina, pois isso implica em guardar informações sobre o estado atual de uma determinada tarefa e reflete o que acontecia com os estoques nas fábricas, mais itens do que o necessário e itens que talvez não fossem utilizados, ou seja, desperdício. Desse pensamento extrai-se o principal ensinamento do *Kanban* "Pare de Começar e Comece a Terminar" (ANDERSON, 2011).



Um sistema *Lean*, baseado em melhoria contínua vem se tornando mais adequado em situações onde mudanças são necessárias de forma cada vez mais rápidas para acompanhar o mercado e a otimização do fluxo, performance e eliminação de desperdícios podem ser evidenciadas por métricas *Leans* e ágeis.

## 1.1 OBJETIVO

O Objetivo deste trabalho é mostrar a importância de métricas *Leans* e Ágeis, para auxiliar a tomada de decisões para gestores e equipes de desenvolvimento em previsões e lançamento de novas funcionalidades. Métricas são contabilizadas a partir de um *software* desenvolvido e integrado a ferramenta de portfólio e de projetos - Jira, que obtêm os dados e fornece as principais métricas utilizando um sistema *Kanban*.

## 1.2 JUSTIFICATIVA

Devido à concorrência o tempo de comercialização (*time to market*) está sendo cada vez menor, as empresas procuram se diferenciar com novas funcionalidades, de forma rápido e seguro, promovendo validação de hipóteses. Para isso, muitas vezes se faz necessário obter uma estimativa de entrega de funcionalidades antes mesmo de começar a produzir incrementos no *software*, de tal forma que se feito de maneira assertiva, a partir de coleta de dados, baseado em históricos, estimativas podem ser realizadas e prever assertividades ou incertezas durante o processo de desenvolvimento da equipe (ALBINO, 2010).

No mercado há muitas ferramentas para controle de processo. A ferramenta Jira da empresa Atlassian é uma das mais conhecidas (ATLASSIAN, 2018), porém para extração de métricas muitos plugins para essa ferramenta são pagos ou não oferecem tudo o que é necessário para uma boa análise das métricas. Como por exemplo os plugins eazyBI (EAZYBI, 2018) e JReport (SHANGHAI DIGITAL TALENT TECHNOLOGY, 2018) que oferecem algumas métricas baseados no Jira porém tem um valor adicional.

Para contornar tais problemas, foi criado o projeto Jiratório e desenvolvido uma ferramenta com código aberto com o objetivo de fornecer uma plataforma gratuita para extração de métricas e relatórios do Jira e auxiliar gestores e desenvolvedores em tomadas de decisões. Este trabalho não descreve detalhes do desenvolvimento da ferramenta Jiratório e sim os benefícios extraídos por essa ferramenta, em comparação aos plugins de mercado.

## 2 REVISÃO BIBLIOGRÁFICA

Métricas de processo ajudam a identificar gargalos e permitem que melhorias sejam feitas sobre os pontos identificados e também auxiliam na previsibilidade do trabalho a ser realizado e dos itens a serem entregues. Boas métricas de processo não devem simplificar o time de desenvolvimento a números ou criar comparações entre outros time ou inclusive indivíduos do mesmo time. O conceito errado no entendimento de métricas por parte de

gestores, traz desconforto e não auxilia no processo de melhoria contínua, apenas colabora como dados de cobrança (ALBINO, 2010). Dessa forma, a seguir são descritas definições de métricas para entendimento e conceitos fundamentais.

## 2.1 LEAD TIME

*Lead Time*, define-se como o tempo total em que uma determinada tarefa leva para passar por todo o fluxo, ou seja, o tempo em que a tarefa leva entre entrar na esteira de produção até que seja entregue para o cliente que a solicitou (SILVA, 2014).

Figura 1 - *Lead Time*



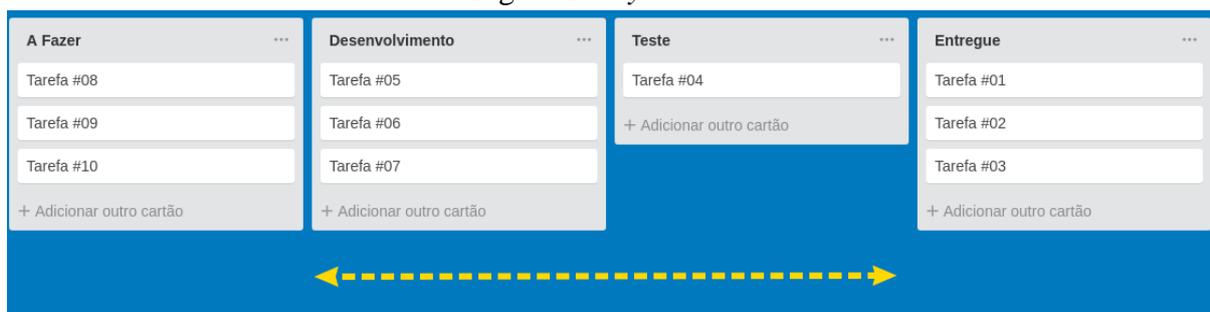
Fonte: Produzida pelos autores.

O *Lead Time* é normalmente utilizado para responder o tempo médio para entregar uma tarefa. Essa métrica é útil para medir a saúde do processo, pois tarefas com grande média de tempo podem indicar a falta de clareza sobre a mesma. *Lead Time* não serve apenas como medida de tempo, também pode ser vista como uma medida de custo, pois quanto maior tempo para uma determinada tarefa ser entregue maior é custo para produzi-la (ALBINO, 2010).

## 2.2 CYCLE TIME

O *Cycle Time* pode ser visto como o tempo que uma determinada tarefa leva entre partes do processo, normalmente utilizada para medir gargalos entre as partes do fluxo (SILVA, 2014).

Figura 2 - *Cycle time*



Fonte: Produzida pelos autores



O *Cycle Time* muitas vezes é visto como um termo similar em relação ao *Lead Time* com a diferença de ser uma medida entre parte do processo e não do começo ao fim do mesmo, porém pode ter definições diferentes por alguns autores (THOMAS, 2015).

Para Taiichi Ohno criador do sistema Toyota de produção *Cycle Time* é definido como o tempo médio do ciclo, podendo ser descrito pela equação: Tempo de ciclo = Horas de operação diária / Quantidade diária (OHNO, 1988).

George Dinwiddie define como o tempo necessário para que cada tarefa progrida de um ponto de entrega de valor do fluxo para outro (DINWIDDIE'S, 2014).

Mary e Tom Poppendieck, pioneiros no desenvolvimento de software *Lean*, não fazem menção ao *Lead Time* em seu livro "*Lean Software Development: An Agile Toolkit*" e citam *Cycle Time* como tempo médio para uma tarefa ir de um lado a outro do processo (POPPENDIECK; POPPENDIECK, 2003).

Por essas e outras definições, muitas vezes divergentes, alguns autores preferem se referir ao *Cycle Time* (tempo entre um determinado ciclo do processo) também como *Lead Time*, porém nomeando-os diferente, como por exemplo "*Lead Time* de desenvolvimento até os testes", possibilitando um entendimento mais natural de que métrica se refere (THOMAS, 2015).

### 2.3 THROUGHPUT

O *Throughput* é a quantidade de tarefas entregues por um determinado tempo, ou seja, é a vazão das tarefas no sistema em um período de tempo. A medida de tempo utilizada pode e deve ser variada de acordo com cada caso, a medição pode fazer sentido por semana, mês, *Sprint* ou outras unidades, isso depende do que faz mais sentido para cada time (ALBINO, 2010).

### 2.4 CLASSES DE SERVIÇO

No Kanban, cada tarefa tem suas determinadas características e essas características que definem cada classe de serviço e sua prioridade sobre as outras, isso também pode ser visto como o tipo de cada tarefa. Cada classe de serviço tem sua prioridade levando em consideração o custo adicional que cada tipo tem, por exemplo uma tarefa normal tem como custo adicional R\$ 0, pois todo o valor gasto para essa tarefa já foi considerado na concepção do projeto, porém para tarefas do tipo erro o custo adicional é entre R\$ 0 a R\$ 1000 por hora que esse erro fica no ambiente de produção, caracterizando que a classe de serviço de erro deve ter prioridade superior a classe de serviço normal (BOEG, 2012).

É comum que as métricas de *Lead Time* e *Throughput* possam ser avaliadas sobre classe de serviço também, pois pode se fazer necessário saber qual é a vazão de tarefas do tipo normal e a vazão de tarefas do tipo erro separadamente da mesma forma que pode ser



necessário saber qual é o tempo médio para a entrega de uma tarefa do tipo erro e o tempo médio de uma tarefa do tipo normal.

## 2.5 TAMANHO DE TAREFAS

Cada tarefa tem uma complexidade, e essa complexidade pode ser vista como o tamanho do esforço e incerteza que uma determinada tarefa possui.

O *Scrum* define esses tamanhos de tarefas como pontos de história (*story points*), isso faz com que em uma determinada *Sprint*, o time negocie a quantidade de pontos que julga encaixar perfeitamente no tempo de duração da *Sprint* baseado no histórico do time, isso significa que se durante as últimas 3 *Sprints* o time obteve 32 pontos em média e entregou dentro do prazo, na próxima *Sprint*, se o time assim estiver confortável pode aceitar entregar em torno de 32 pontos (SUTHERLAND, 2014). Entenda-se como pontos, alguma medida que define incertezas, no *Scrum* a estimativa mais usual é por meio de *Planning Poker*, um jogo de estimativas baseado na sequência de Fibonacci. Outra forma de estimar é por meio de *TShirt-Size*, baseado em tamanhos de camisas, que define cada tarefa como P, M ou G (BOEG, 2012).

No Kanban as tarefas também podem ter seus pesos definidos e essa definição pode ser feita de várias maneiras, com uma sequência baseada em Fibonacci como o *Planning Poker*, neste projeto foi utilizado a estimativa por meio de *TShirt-Size*.

As métricas de *Throughput* e *Lead Time* também podem ser avaliadas sobre os tamanhos das tarefas, pois assim é possível dizer qual é a vazão de tarefas pequenas e compará-la a vazão de tarefas médias ou grandes da mesma forma que torna-se possível medir o tempo de entrega das tarefas baseado no seu tamanho.

## 3 MÉTRICAS EXTRAÍDAS POR MEIO DA FERRAMENTA JIRATÓRIO

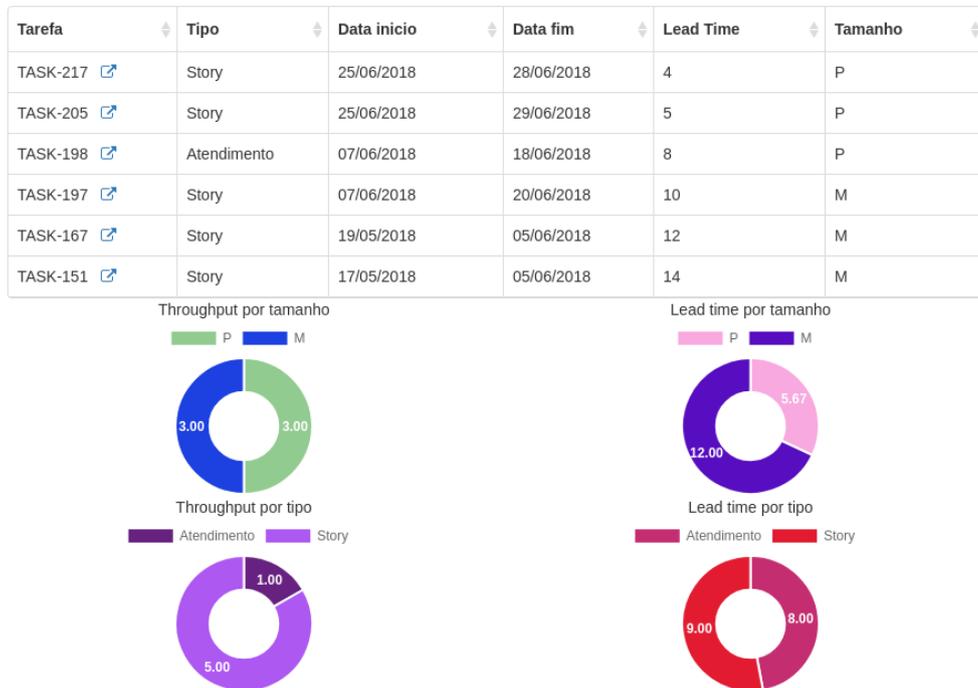
Para a validação da ferramenta durante 3 meses foram extraídas métricas de um time de desenvolvimento de software em uma empresa de consultoria de software na cidade de Araraquara, o nome da empresa não será mencionado a fim de preservar e não expor a Empresa. O time de desenvolvimento trabalha com desenvolvimento ágil por aproximadamente 2 anos, essas métricas serviram tanto para validação e teste da ferramenta quanto como base para conclusões que podem ser tomadas para futuras entregas. Embora ocorreram mudanças na equipe, com entrada e saída de profissionais, o objetivo é avaliar a ferramenta para auxiliar gestores e equipes de desenvolvimento com a tomada de decisões de projeto e produto.

Com base nas métricas *Leans* e *Ágeis* a ferramenta exporta quatro gráficos e uma listagem para análise. A listagem exibe o identificador único da tarefa, o seu tipo (classe de serviço) a data de início, data final, *Lead Time* de cada tarefa e seu respectivo tamanho (P, M ou G). Nos gráficos é feita a exibição das métricas de *Lead Time* e *Throughput* separado por



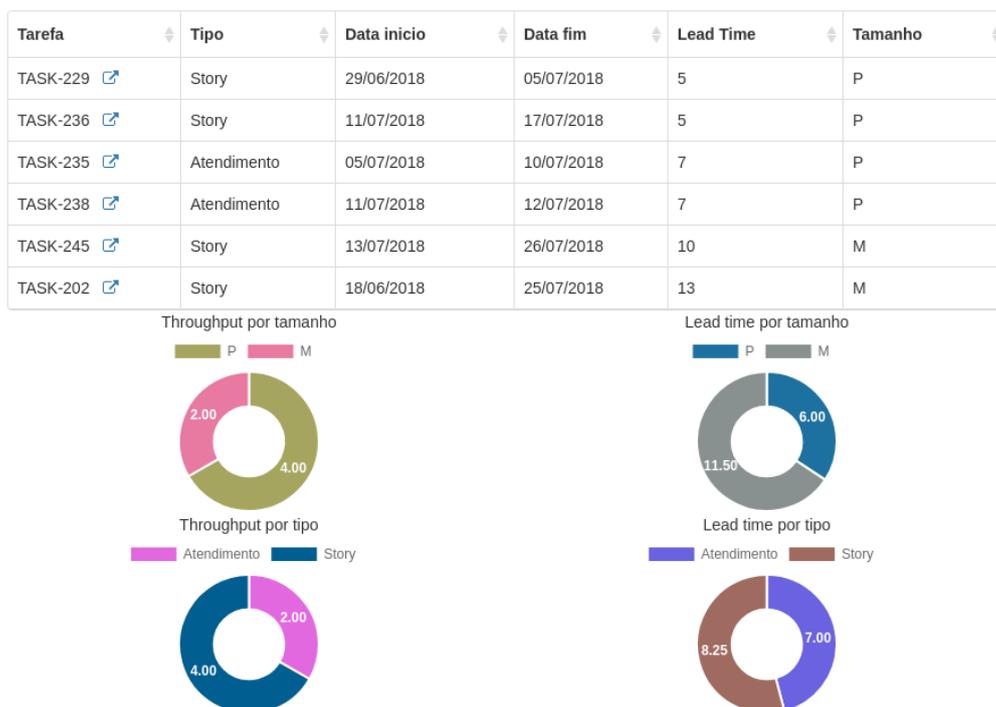
tipo e estimativa. Além dessas métricas, a ferramenta também permite visualizar outras métricas mais avançadas, porém esta análise inicial foi apenas sobre as métricas mais básicas.

Figura 3 - Métricas extraídas no primeiro mês



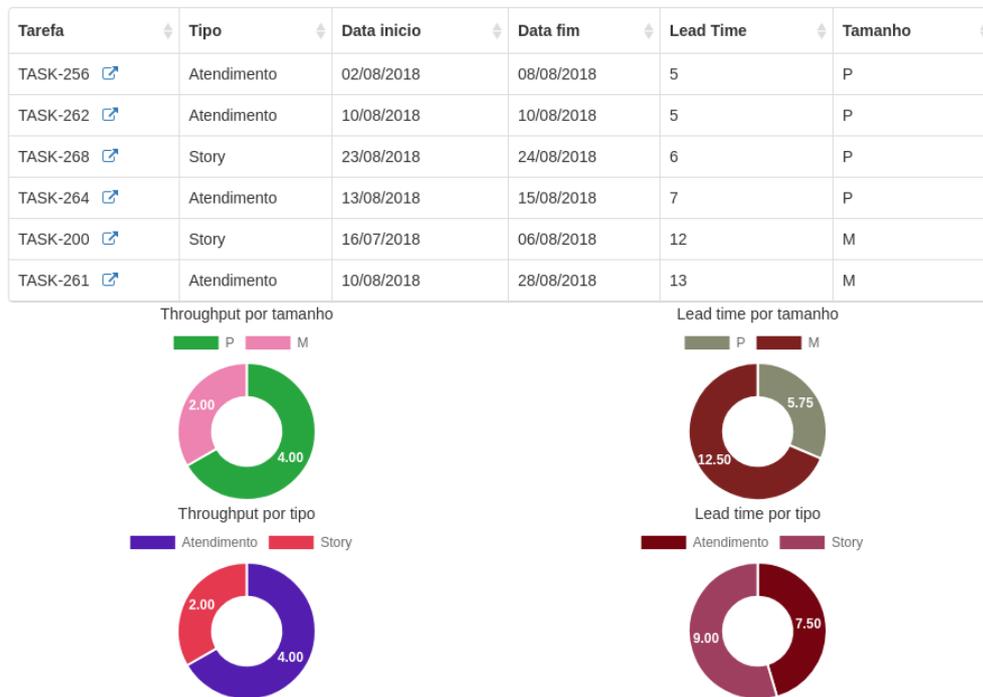
Fonte: produzida pelos autores

Figura 4 - Métricas extraídas no segundo mês



Fonte: produzida pelos autores

Figura 5 – Métricas extraídas no terceiro mês



Fonte: produzida pelos autores

No primeiro mês, exibido na Figura 3, é possível ver que *Throughput* da equipe foi de 6 tarefas e teve um *Lead Time* médio de 8,83 dias. Na análise por tamanho de tarefa, foi exibido que o time possui *Lead Time* médio de 12 dias para tarefas de tamanho M e 5,67 dias para tarefas de tamanho P, e o *Throughput* de 3 para tarefas P e também 3 dias para tarefas M. Na análise por tipo de tarefa, o *Throughput* para o tipo Atendimento é de 1 e para o tipo Story é de 5, o *Lead Time* médio para Atendimento é de 9 dias e para Story é de 8 dias.

No segundo mês, exibido na Figura 4, o *Throughput* da equipe se manteve em 6 tarefas e o *Lead Time* médio foi de 7,83 dias. Na análise por tamanho de tarefa, o *Lead Time* médio foi de 11,50 dias para tarefas de tamanho M e 6 dias para tarefas de tamanho P, e o *Throughput* é de 4 para tarefas P e 2 para tarefas M. Na análise por tipo de tarefa o *Throughput* para o tipo Atendimento é de 2 e para o tipo Story é de 4, o *Lead Time* médio para Atendimento é de 7 dias e para Story é de 8,25 dias

No terceiro mês, exibido na Figura 5, o *Throughput* da equipe foi de 6 tarefas e o *Lead Time* médio foi de 8 dias. Na análise por tamanho de tarefa, o *Lead Time* médio foi de 12,5 dias para tarefas de tamanho M e 5,57 dias para tarefas de tamanho P, e o *Throughput* foi de 4 para tarefas P e 2 para tarefas M. Na análise por tipo de tarefa, o *Throughput* para o tipo Atendimento é de 2 e para o tipo Story é de 4, o *Lead Time* médio para Atendimento é de 7,50 dias e para Story é de 9 dias.



#### 4 RESULTADOS E DISCUSSÃO

Na análise dos últimos 3 meses as métricas mostram que o time normalmente entrega 6 tarefas e cada tarefa leva em média 8 dias para ser entregue.

Analisando por tamanho de história, pode-se dizer que o time entrega em média 4 histórias P e 3 histórias M por mês, e o *Lead Time* das histórias P são de 5,74 dias e das histórias M são 12 dias.

Na análise por classe de serviço, as conclusões podem não ser 100% assertivas, pois houve uma grande variação entre os meses analisados, mas pode-se afirmar que o time analisado entrega por mês 2,33 tarefas do tipo Atendimento e 3,66 tarefas do tipo *Story* e que tarefas do tipo Atendimento levam em média 8 dias, enquanto tarefas do tipo *Story* levam em média 8,25 dias.

Com essas métricas é possível dizer que se a próxima *feature* do time depende de 24 tarefas M e 12 tarefas P, essa *feature* ficará finalizada em média até 6 meses, isso considerando apenas tarefas do tipo *Story*. Dessa forma, essa ferramenta e as métricas expostas auxiliam gestores e times de desenvolvimento, a ser mais assertivos em previsões e estimativas utilizando práticas ágeis.

#### REFERÊNCIAS

ALBINO, R. D. **Métricas Ágeis**: Obtenha melhores resultados em sua equipe. [S.l.]: Casa do Código, 2010.

ANDERSON, D. **Kanban**: Mudança evolucionária de sucesso para seu negócio de tecnologia. [S.l.]: Blue Hole Press, 2011.

ATLASSIAN. **Jira | Software de rastreamento de projetos e problemas | Atlassian**. 2018. Disponível em: <<https://br.atlassian.com/software/jira>>. Acesso em: 01 de Setembro de 2018.

BOEG, J. **Kanban em 10 Passos**. [S.l.]: InfoQ Brasil, 2012.

DINWIDDIE'S, G. **Relationship of Cycle Time and Velocity**. 2014. Disponível em: <<http://blog.gdinwiddie.com/2014/12/10/relationship-of-cycle-time-and-velocity/>>. Acesso em: 01 de Setembro de 2018.

EAZYBI. **EazyBI Reports and Charts for Jira**. 2018. Disponível em: <<https://marketplace.atlassian.com/apps/1211051/eazybi-reports-and-charts-for-jira>>. Acesso em: 01 de Setembro de 2018.

JONES, D.; WOMACK, J. **A Máquina que Mudou o Mundo**. [S.l.]: Elsevier, 2004.



OHNO, T. **Toyota Production System**: Beyond large-scale production. [S.l.]: Productivity Press, 1988.

POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development**: An agile toolkit. [S.l.]: Pearson, 2003.

SHANGHAI DIGITAL TALENT TECHNOLOGY. **JReport - Reports and Charts for Jira**. 2018. Disponível em: <<https://marketplace.atlassian.com/apps/1217877/jreport-reports-and-charts-for-jira>>. Acesso em: 01 de Setembro de 2018.

SILVA, J. N. da. **Uma ferramenta para cálculo de Métricas Lean**. Monografia (Graduação em Sistema de informação) — UFRPE, Recife, 2014.

SUTHERLAND, J. **SCRUM**: A arte de fazer o dobro de trabalho na metade do tempo. [S.l.]: Leya Brasil, 2014.

THOMAS, S. **Lead time versus Cycle Time – Untangling the confusion**. 2015. Disponível em: <<http://itsadeliverything.com/lead-time-versus-cycle-time-untangling-the-confusion>>. Acesso em: 01 de Setembro de 2018.