

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MATHEUS MACIEL SOARES

Análise comparativa de ferramentas utilizadas para Kanban

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Dra. Karin Becker

Porto Alegre
2017

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Profa. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Profa. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

Kanban é um método de desenvolvimento de software bastante popular. Em decorrência disso, diversas ferramentas foram criadas para auxiliar a adoção deste método. Entretanto, faltam critérios claros para auxiliar na escolha da ferramenta mais apropriada dentro deste universo.

Este trabalho descreve e avalia quatro ferramentas populares utilizadas para apoio ao uso de Kanban. Para isto, o método é descrito, de forma a se extrair critérios chave para seu uso, que devem ser contemplados pelas ferramentas. Por fim, estas ferramentas são comparadas entre si.

Palavras-chave: Kanban. Métodos ágeis.

Comparative analysis of Kanban tools

ABSTRACT

Kanban is a popular method for software development. As a result, numerous tools have been created to support its adoption. However, there is a lack of clear criteria to help in the choice of the most suitable tool amongst this universe.

This work describes and assesses four popular tools used for supporting Kanban's usage. To this end, the method is described, so as to extract key criteria for its usage, which must be satisfied by the tools. Finally, the tools are compared with one another.

Keywords: Kanban. Agile methods.

LISTA DE FIGURAS

Figura 2.2.1 – Exemplo de quadro de Kanban	11
Figura 2.2.2 – Quadro de Kanban com raias	12
Figura 2.2.3 – Quadro de Kanban com cores indicando agrupamentos de itens.....	13
Figura 2.2.4 – Exemplo de gráfico de fluxo cumulativo.....	15
Figura 2.2.5 – WIP do sistema em um GFC	16
Figura 2.2.6 – <i>Lead time</i> em um GFC.....	17
Figura 2.7 – <i>Throughput</i> em um GFC.....	17
Figura 2.8 – Quadro de Kanban com políticas explícitas	18
Figura 2.9 – Quadro de Kanban com subcoluna para itens prontos	19
Figura 3.1 – Quadro de Kanban a ser reproduzido nas ferramentas avaliadas	24
Figura 3.2 – Exemplo de quadro no Trello	25
Figura 3.3 – Cartão no Trello	26
Figura 3.4 – Reprodução do Kanban de exemplo com o Trello.....	28
Figura 3.5 – Exemplo de GFC no Correlo	29
Figura 3.6 – Exemplo de quadro de Kanban no Jira.....	31
Figura 3.7 – Cartão no Jira	32
Figura 3.8 – Reprodução do Kanban de exemplo com o Jira, com indicação de agrupamento de itens através de raias.....	34
Figura 3.9 – Reprodução do Kanban de exemplo utilizando o Jira, com indicação de agrupamento de itens através de cores.....	35
Figura 3.10 – GFC no Jira.....	36
Figura 3.11 – Visão detalhada de um item no CA Agile Central.....	38
Figura 3.12 – Reprodução do Kanban de exemplo no CA Agile Central.....	40
Figura 3.13 – Relatório de <i>lead time</i> gerado pelo CA Agile Central.....	41
Figura 3.14 – Relatório de <i>throughput</i> gerado pelo CA Agile Central.....	42
Figura 3.15 – GFC no CA Agile Central	43
Figura 3.16 – Exemplo de quadro de Kanban no LeanKit.....	45
Figura 3.17 – Visão detalhada de um cartão no LeanKit	46
Figura 3.18 – Reprodução do Kanban de exemplo com o LeanKit.....	47
Figura 3.19 – Visualização de políticas de processo no LeanKit	48
Figura 3.20 – GFC gerado pelo LeanKit.....	49

LISTA DE TABELAS

Tabela 3.1 – Comparação entre as ferramentas avaliadas.....	50
---	----

LISTA DE ABREVIATURAS E SIGLAS

GFC	Gráfico de Fluxo Cumulativo
STP	Sistema Toyota de Produção
WIP	Work In Progress

SUMÁRIO

1 INTRODUÇÃO	9
2 KANBAN.....	10
2.1 Histórico	10
2.2 Propriedades básicas do Kanban como método de desenvolvimento de software....	11
2.2.1 Visualização do fluxo de trabalho	11
2.2.2 Limitação do trabalho em progresso.....	13
2.2.3 Medição do fluxo	14
2.2.4 Explicitação das políticas do processo.....	18
2.2.5 Uso de modelos para reconhecimento de oportunidades de melhoria.....	19
2.3 Trabalhos relacionados	20
2.4 Considerações finais.....	20
3 AVALIAÇÃO DE FERRAMENTAS PARA USO DE KANBAN	21
3.1 Contexto.....	21
3.2 Ferramentas avaliadas	21
3.3 Requisitos avaliados	22
3.4 Forma de avaliação	23
3.5 Trello	24
3.5.1 Reprodução do Kanban de exemplo	27
3.5.2 Métricas.....	28
3.5.3 Avaliação.....	29
3.6 Jira.....	30
3.6.1 Reprodução do Kanban de exemplo	33
3.6.2 Métricas.....	35
3.6.3 Avaliação.....	36
3.7 CA Agile Central.....	37
3.7.1 Reprodução do Kanban de exemplo	39
3.7.2 Métricas.....	40
3.7.3 Avaliação.....	43
3.8 LeanKit.....	44
3.8.1 Reprodução do Kanban de exemplo	46
3.8.2 Métricas.....	48
3.8.3 Avaliação.....	49

3.9 Avaliação comparativa.....	50
4 CONCLUSÃO	51
REFERÊNCIAS.....	52

1 INTRODUÇÃO

A utilização de Métodos Ágeis em projetos de desenvolvimento de software é percebida como geradora de diversos benefícios quando comparadas com outros métodos, como aumento de produtividade, aumento de qualidade e maior adaptabilidade a mudanças (MELO; et al., 2012). Assim, não é surpresa que esta família de métodos seja amplamente popular, com relatos de até 94% de empresas de desenvolvimento de software os utilizando em algum grau (VERSIONONE, 2017).

Dentro desta família de métodos, encontra-se o Kanban. O Kanban é um método inspirada no Sistema Toyota de Produção (STP), originário na produção industrial, que preza pela minimização de desperdícios e eliminação de gargalos. Para isso, utiliza a divisão do trabalho necessário para a conclusão do desenvolvimento de um ítem em diversas etapas, a visualização do trabalho em andamento dentro dessas etapas e a limitação da quantidade de itens processados simultaneamente (WIP) em cada estágio do processo. Esta limitação do WIP visa a obtenção de um ritmo constante e sustentável, que possa ser mantido por um período indefinido de tempo, sem sobrecargas no time de desenvolvimento.

Dada a popularidade do Kanban, diversas ferramentas surgiram com o intuito de apoiar a utilização deste método, como Trello, Jira, LeanKit, Kanban Tool, Kanban Flow, Kanbanize, Kanbanchi, Volverro, CA Agile Central, Kanbanery, SwiftKanban, AgileZen e Cleverflow. Entretanto, faltam critérios que possam ajudar na escolha da ferramenta mais apropriada.

Este trabalho tem a intenção de fazer uma análise comparativa entre diferentes ferramentas disponíveis para a utilização do Kanban enquanto ferramenta para controle de cadência de processo e aumento de vazão, no âmbito de desenvolvimento de software. Para isso, foram identificados os aspectos chave necessários para a adoção de Kanban como apoio ao desenvolvimento de software. Após, as ferramentas foram comparadas em relação às suas capacidades de satisfazerem estes requisitos básicos.

O restante deste trabalho está estruturado como segue:

- O Capítulo 2 abordará o uso de Kanban para desenvolvimento de software, incluindo quais são as características necessárias para um sistema deste tipo;
- No Capítulo 3, será feita uma descrição de diversas ferramentas utilizadas como suporte para Kanban, e estas serão avaliadas e comparadas em relação aos requisitos essenciais identificados no Capítulo 2;
- No Capítulo 4, será feita a conclusão sobre este trabalho.

2 KANBAN

Neste capítulo, o sistema Kanban será apresentado, incluindo seu histórico, seus conceitos e suas propriedades.

2.1 Histórico

Originalmente, Kanban foi o nome adotado para um dos componentes do STP, o processo desenvolvido pela Toyota para a produção de carros de forma eficiente.

O STP, também conhecido como o Método Toyota, se baseia em só produzir peças conforme elas são necessárias para um estágio posterior na linha de produção, evitando superprodução e assim diminuindo desperdícios. A principal característica desse sistema é ser construído a partir do conceito de “*Pull*”, em que o trabalho é “puxado” de uma etapa anterior para um estágio posterior da linha de produção quando há capacidade disponível, ao invés de “empurrado” (“*Push*”) para ela:

[...] o Método Toyota não se trata de gerenciar estoque; ele se trata de eliminá-lo. Desde o começo, Toyota começou a pensar em termos de um estoque de “*Pull*” baseado na demanda imediata dos clientes, ao invés de utilizar um sistema de “*Push*” que adivinha a demanda dos clientes. No Método Toyota, “*Pull*” significa o estado ideal da manufatura just-in-time: dar aos clientes (que podem ser o próximo passo do processo de produção) o que ele ou ela quer, quando ele ou ela o quiser, e na quantidade que ele ou ela quiser. (LIKER, 2004)

Kanban dentro do contexto do STP se refere a sinalizações visuais para a produção de mais peças. De fato, a palavra “Kanban” em japonês significa “quadro indicador”. No STP, uma peça só pode ser produzida caso haja uma sinalização visual partindo de um estágio posterior da linha de produção indicando que mais peças são necessárias.

A ideia de Kanban enquanto um método para o desenvolvimento de software inspirada no STP foi criada e primeiramente colocada em prática em um projeto na empresa Corbis, entre os anos de 2006 e 2007.

2.2 Propriedades básicas do Kanban como método de desenvolvimento de software

De acordo com Anderson (2010), a utilização de Kanban enquanto sistema de desenvolvimento de software tem cinco propriedades principais:

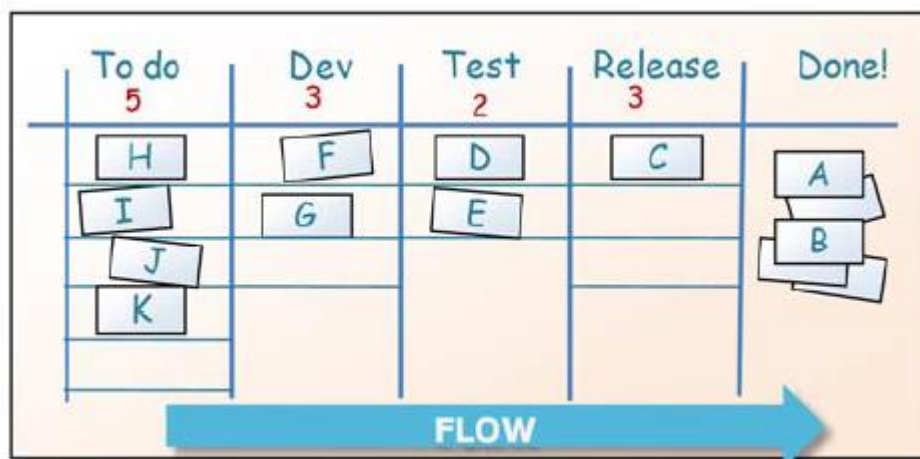
- Visualização do fluxo de trabalho;
- Limitação do trabalho em progresso (WIP);
- Medição do fluxo;
- Explicitação das políticas do processo;
- Uso de modelos para reconhecimento de oportunidades de melhoria.

2.2.1 Visualização do fluxo de trabalho

O processo de desenvolvimento de software é visto de forma análoga a uma linha de produção industrial. Assim, este é decomposto em uma sequência de etapas; para que um item seja finalizado, ele deve passar por cada um destes estágios, em ordem.

O Kanban tem como objetivo permitir a fácil visualização de cada um destes estágios, juntamente com o trabalho em andamento em cada um deles. Para que este requisito seja atendido, se utiliza um quadro, denominado de quadro de Kanban. Neste quadro, os itens de trabalho entre os estágios, tipicamente da esquerda para a direita, como exemplificado na Figura 2.1. Neste exemplo, o trabalho de desenvolvimento foi decomposto em cinco etapas: *To do* (a fazer), *Dev* (desenvolvimento), *Test* (teste), *Release* (entrega) e *Done* (finalizado).

Figura 2.2.1 – Exemplo de quadro de Kanban



Fonte: Kniberg; Skarin (2009, p. 5).

É importante notar que a decomposição do processo em uma sequência de etapas é condição inicial e mandatória para a adoção do Kanban, pois a visualização do fluxo de trabalho é a forma de controlar a execução e cadência do processo de desenvolvimento.

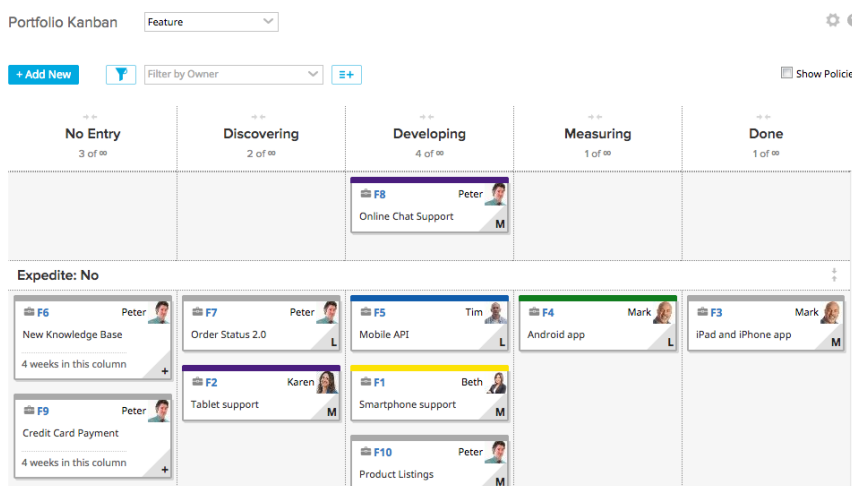
Observe-se também que o processo de desenvolvimento é auto-contido e existe independentemente do quadro de Kanban, sendo o quadro a concretização do processo de uma maneira visual. Assim, ambos frequentemente são vistos como sinônimos.

Em alguns casos, é necessário agrupar diferentes tipos de itens de trabalho relacionados, para que estes sejam visualizados de forma simples. Exemplos destes agrupamentos são em relação ao cliente que requisitou o trabalho, sua prioridade, ou ao tipo dos itens. Nestes casos, o quadro de Kanban deve possuir uma indicação sobre os diferentes tipos de grupos.

Estes agrupamentos são normalmente representados em um quadro de Kanban através de duas formas: uma subdivisão horizontal, conhecida como raia, ou uma indicação visual nos itens, como, por exemplo, diferentes colorações.

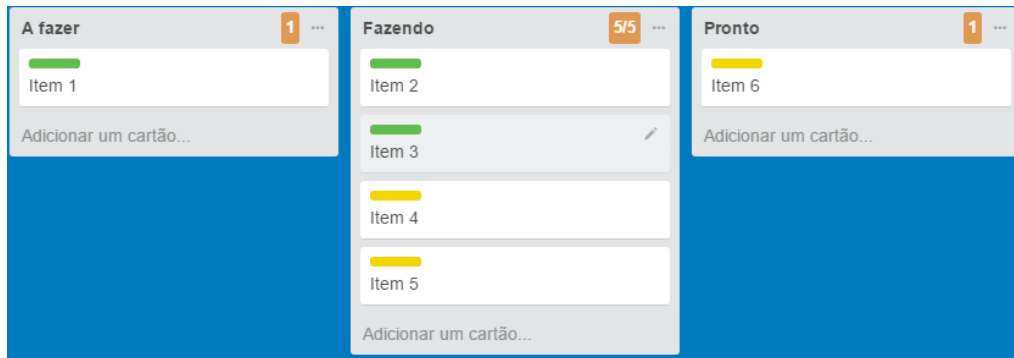
A Figura 2.2 demonstra um exemplo de um quadro de Kanban com raias e cores, e a Figura 2.3 exemplifica um quadro de Kanban com agrupamento dos itens de acordo com cores apenas.

Figura 2.2.2 – Quadro de Kanban com raias



Fonte: Q1 2015.

Figura 2.2.3 – Quadro de Kanban com cores indicando agrupamentos de itens



Fonte: elaborada pelo autor.

2.2.2 Limitação do trabalho em progresso

Para cada uma das etapas do processo de desenvolvimento, o número de tarefas simultâneas (WIP) permitidas é limitado. De acordo com Anderson & Carmichael (2016), utilizar e respeitar limites de WIP mudam um sistema de “*push*” para um sistema de “*pull*”, em que novos itens não são começados até que trabalho seja completado. Isso é importante pois ter muito trabalho parcialmente completo é um desperdício e caro, impedindo que organizações respondam rapidamente aos clientes e a mudanças nas circunstâncias.

Esta limitação tem a intenção de reduzir o desperdício por superprodução, e evitar que muitas tarefas sejam trabalhadas em paralelo sem que nada seja finalizado. Um resultado desta limitação é que se pare de começar itens novos até que se conclua os itens em andamento (VASHISHTHA, 2014).

Este limite pode ser diferente para cada um dos estágios do Kanban e usualmente é indicado no topo da coluna, junto ao nome do estágio. Esta limitação pode ser visualizada na Figura 2.1. É importante notar nesta figura que a última coluna (Done) não tem limite de WIP, pois representa os itens já finalizados, e, portanto não há trabalho sendo realizado nos itens desta coluna.

De acordo com Anderson (2010), não há uma fórmula mágica para a escolha dos limites de WIP para cada estágio do Kanban; estes devem ser escolhidos empiricamente, de acordo com uma estimativa da capacidade de processamento do sistema, revisados frequentemente e ajustados quando necessário.

2.2.3 Medição do fluxo

Para avaliar o fluxo de um sistema, é preciso medir os resultados obtidos a partir dele. Para isso, há dois métodos complementares: o cálculo de métricas e a construção de gráficos de acompanhamento.

2.2.3.1 Métricas

Existem duas métricas que são comumente utilizadas para avaliar o fluxo em um sistema Kanban: *lead time* (tempo de provisionamento) e *throughput* (vazão). As revisões nos valores de WIP devem ter como objetivo melhorar estas métricas, maximizando o *throughput* e minimizando o *lead time*.

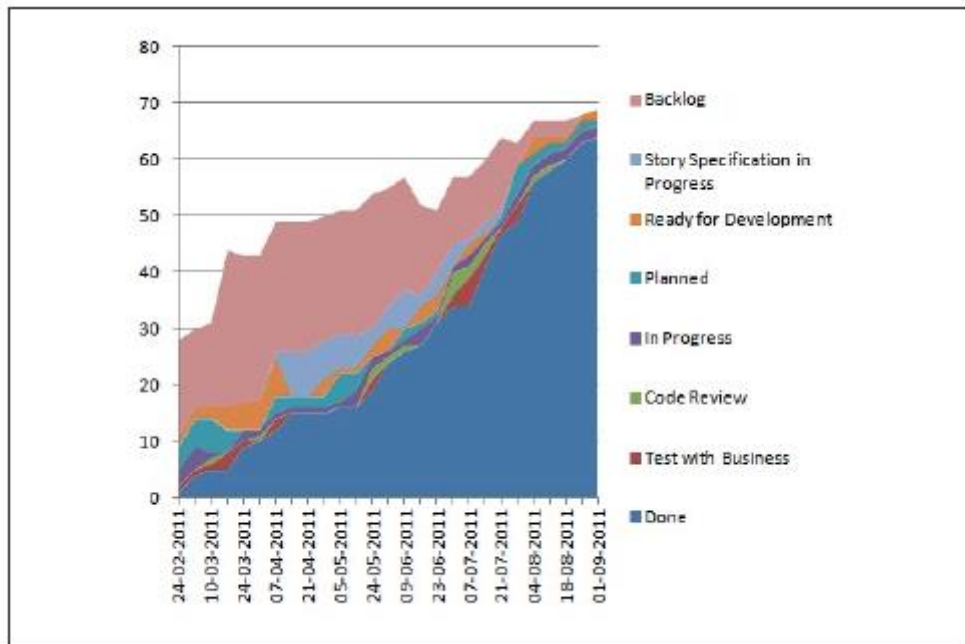
O *lead time* é medido como o tempo entre um item entrar no sistema Kanban e ser finalizado. Como normalmente há mais do que um item ativo no sistema ao mesmo tempo, a mensuração é feita como a média dos *lead times* dos diversos itens que passaram pelo sistema. O *throughput*, por outro lado, mede a quantidade de itens finalizados em um certo intervalo de tempo (um mês, por exemplo).

2.2.3.2 Gráficos de acompanhamento

Um dos gráficos mais utilizados para a visualização do fluxo em um sistema de Kanban é o gráfico de fluxo cumulativo (GFC).

Um GFC mostra o estado do trabalho para cada estágio do sistema pelo tempo, permitindo assim uma fácil visualização das métricas de fluxo de um Kanban. Nele, o eixo horizontal indica o tempo (em um período qualquer, como dias, semanas ou meses) e o eixo vertical indica a quantidade acumulada de itens. Uma cor é escolhida para representar cada estágio do Kanban, sendo que os estágios são mostrados de forma inversa à sua ordem no Kanban. Assim, o último estágio do Kanban, que indica itens finalizados, é posicionado o mais abaixo no gráfico, enquanto o primeiro estágio, que indica itens a fazer, é posicionado o mais acima. Um exemplo de GFC pode ser visualizado na Figura 2.4.

Figura 2.2.4 – Exemplo de gráfico de fluxo cumulativo



Fonte: Boeg (2010, p.45).

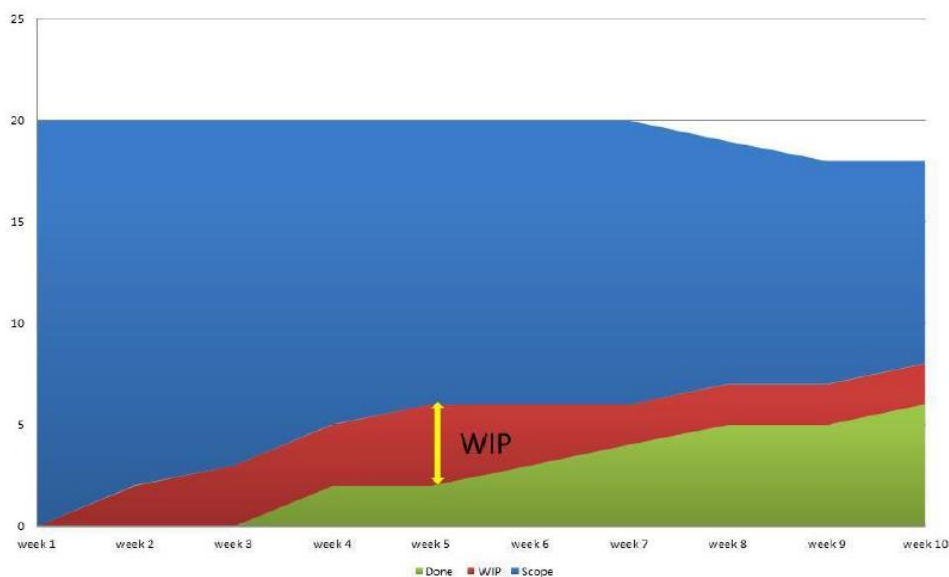
Neste exemplo, o intervalo de tempo escolhido é de duas semanas, e as etapas do Kanban são *Backlog* (a fazer), *Story Specification in Progress* (especificação da história em progresso), *Ready for Development* (pronto para desenvolvimento), *Planned* (planejado), *In Progress* (em progresso), *Code Review* (revisão de código), *Test with Business* (teste com o setor de Negócios) e *Done* (pronto). É possível notar a evolução do trabalho no sistema, com cada vez mais itens sendo finalizados e a lista de itens a fazer diminuindo.

Neste gráfico, é possível visualizar diversas métricas sobre o sistema ao longo do tempo, como o WIP do sistema em um dado momento, o *lead time* e o *throughput*.

O WIP do sistema é a quantidade de itens em andamento somando todos os estágios do Kanban em que há trabalho sendo feito. Assim, são desconsiderados os itens prontos e os ainda a fazer (primeira e última colunas do Kanban). Note que, neste contexto, o WIP se refere ao sistema como um todo, não devendo ser confundido com o WIP de cada uma das etapas do processo e seus limites associados.

A Figura 2.5 demonstra como esta informação pode ser lida em um GFC, considerando um sistema com apenas três estágios: *Scope* (a fazer), *WIP* (em andamento) e *Done* (feito).

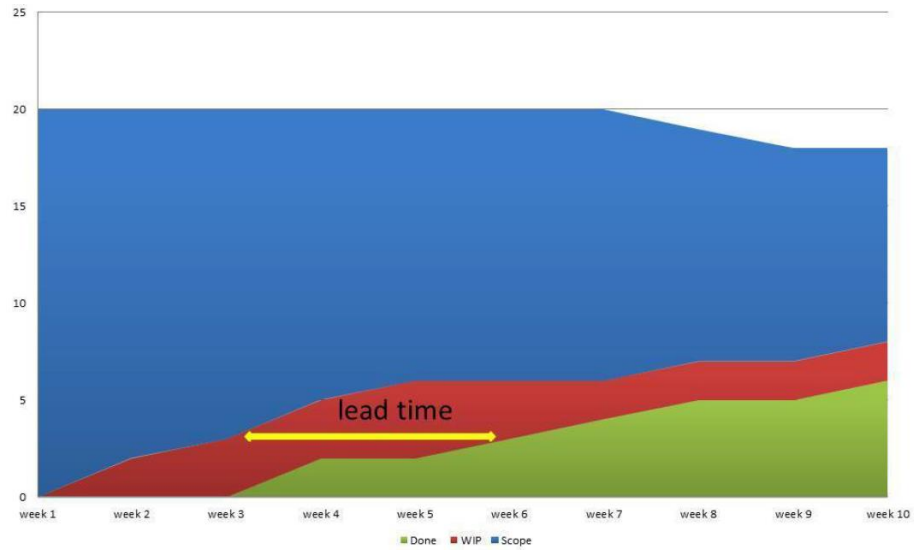
Figura 2.2.5 – WIP do sistema em um GFC



Fonte: Caroli (2016, p.12).

Como indicado pela flecha amarela, o WIP do sistema em um determinado momento é dado pela diferença entre os fluxos de itens a fazer e de itens prontos naquele momento (fluxos azul e verde, respectivamente). Neste exemplo, há somente um estágio em que há trabalho sendo feito no sistema, mostrado na cor vermelha, mas este não é um pré-requisito para a visualização do WIP do sistema em um GFC. Em casos como o do exemplo da Figura 2.4, esta medição compreenderia todos os estágios do sistema, exceto pelas etapas inicial e final, em que nenhum trabalho é feito.

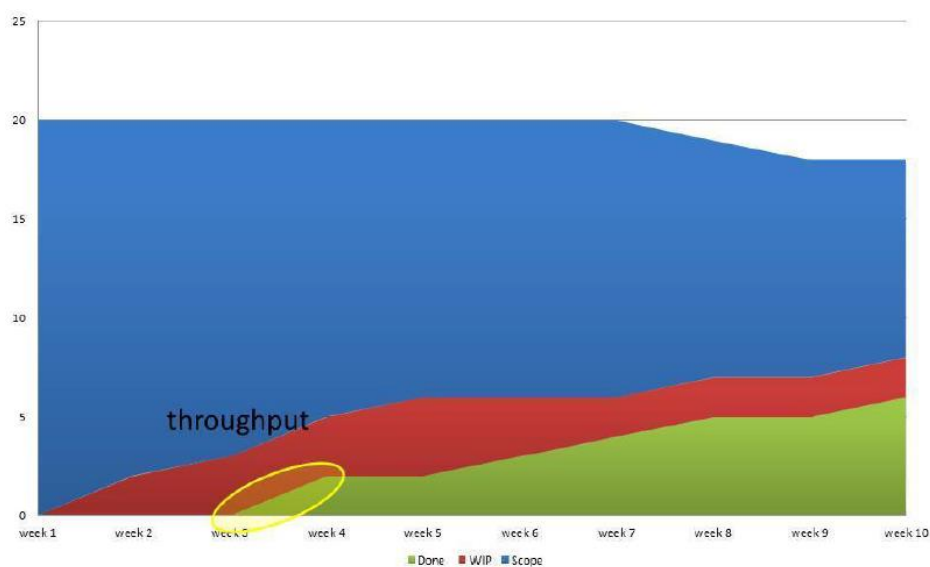
Já o *lead time* pode ser visualizado ao se traçar uma linha paralela ao eixo horizontal a partir do fluxo de itens a fazer em um determinado ponto do tempo até que ele encontre o fluxo de itens prontos. O tamanho desta linha indicará o *lead time* médio para os itens começados no ponto de tempo escolhido inicialmente. A Figura 2.6 exemplifica esta visualização considerando os mesmos três estágios.

Figura 2.2.6 – *Lead time* em um GFC

Fonte: Caroli (2016, p.13).

Neste exemplo, a medição é feita para a terceira semana (*week 3*). A linha horizontal só chega à uma intersecção com o fluxo verde, que indica itens finalizados, aproximadamente na sexta semana (*week 6*). Assim, o *lead time* médio dos itens que foram começados na terceira semana é de aproximadamente três semanas.

O *throughput* em um determinado intervalo de tempo é visto simplesmente como o aumento no tamanho do fluxo de itens prontos entre o início e o final deste intervalo. A Figura 2.7 indica como ele é visualizado num GFC.

Figura 2.7 – *Throughput* em um GFC

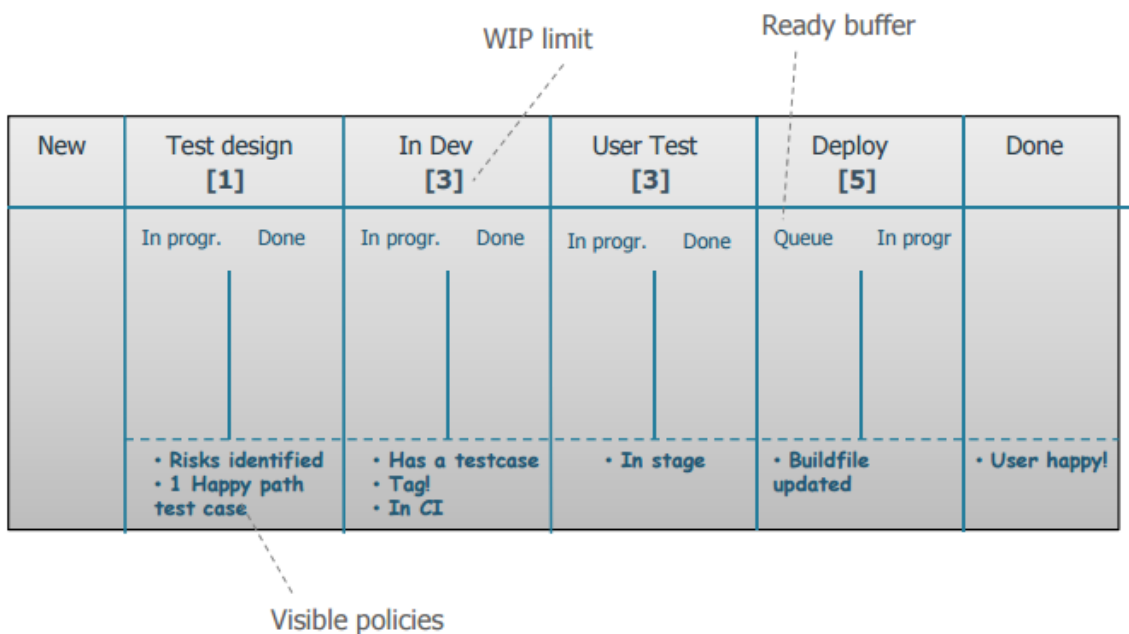
Fonte: Caroli (2016, p.13).

Neste exemplo, o *throughput* da terceira semana (*week 3*) é visto como a diferença entre o fluxo de itens prontos no início e no final da semana. Neste caso, não havia nenhum item pronto no começo do período, mas no final dele já haviam dois itens prontos. Assim, o *throughput* desta semana foi de 2.

2.2.4 Explicitação das políticas do processo

As políticas do processo definem, para cada um dos estágios do Kanban, os critérios mínimos para que um item possa avançar para o próximo estágio, isto é, o que precisa ser feito para o trabalho daquela etapa do sistema ser considerado pronto e bem feito. Estas políticas devem estar explicitadas no quadro, junto a cada uma das colunas do Kanban.

Figura 2.8 – Quadro de Kanban com políticas explícitas

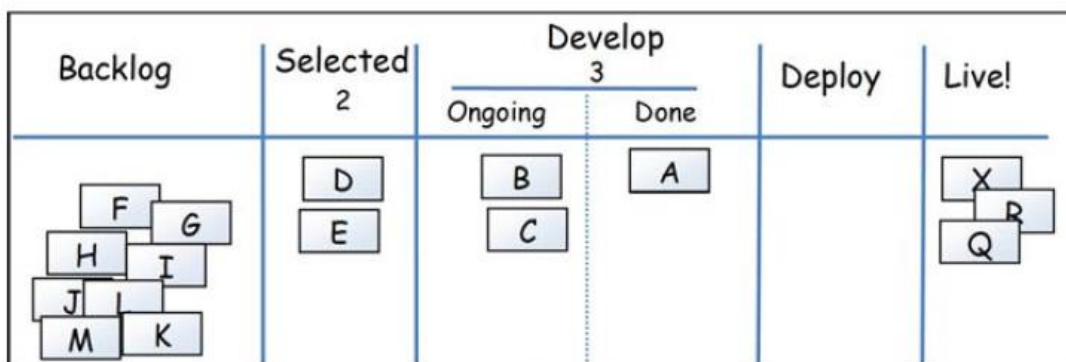


Fonte: Skarin (2010, p. 5).

Na Figura 2.8, estas políticas podem ser visualizadas na parte inferior de cada coluna. Como exemplo, a coluna de *Test Design* tem como política que os riscos devem ser identificados e um caso de teste deve ser desenvolvido. Assim, um item só será considerado pronto para avançar ao próximo estágio (*In Dev*) quando estes requisitos forem alcançados.

Como o STP, o Kanban prescreve que existam indicações visuais de que um item de trabalho está pronto para avançar para o próximo estágio do processo, ou seja, de que as políticas do estágio foram atingidas. Uma das formas de alcançar este requisito é através da subdivisão de colunas em duas. Neste caso, a subcoluna da esquerda representa os itens sendo trabalhados e a subcoluna da direita representa os itens que estão prontos para serem movidos para a etapa subsequente. Um exemplo deste tipo de segmentação é demonstrado na Figura 2.9.

Figura 2.9 – Quadro de Kanban com subcoluna para itens prontos



Fonte: Kniberg; Skarin (2009, p.42).

É importante notar que, apesar da subdivisão da coluna denominada *Develop* (desenvolvimento) entre itens *Ongoing* (em andamento) e *Done* (finalizados), o limite de WIP ainda é válido para a coluna como um todo, ou seja, o limite engloba tanto itens em andamento quanto prontos desta coluna. Portanto, somente quando o item A for movido para a próxima etapa (*Deploy*), um item do estágio *Selected* (D ou E) pode passar à fase *Develop*.

2.2.5 Uso de modelos para reconhecimento de oportunidades de melhoria

O processo utilizado por um sistema Kanban deve estar em constante evolução. Como auxílio para se atingir este objetivo, são usados modelos que ajudam a identificar oportunidades de melhoria. Modelos como o de Teoria das Restrições, Pensamento Sistêmico e o conceito de *muda* (identificação e eliminação de pontos de desperdício), importado do Método Toyota, são comumente utilizados juntamente ao Kanban.

2.3 Trabalhos relacionados

Augustine (2017) e Burger (2016) também fizeram avaliações sobre ferramentas utilizadas para Kanban. Estas avaliações focam em aspectos como o design, suporte ao usuário, preço e facilidade no uso das aplicações consideradas.

2.4 Considerações finais

O Kanban aplicado ao desenvolvimento de software tem cinco propriedades principais. Entretanto, uma destas propriedades, o uso de modelos para avaliar possibilidades de melhoria, está fora do escopo deste trabalho, por não ser relacionada ao quadro de Kanban.

Desta forma, restam quatro propriedades principais, que foram avaliadas neste trabalho: visualização do fluxo de trabalho, limitação do WIP, medição do fluxo e explicitação das políticas do processo.

Os critérios utilizados nas avaliações de Augustine e Burger não serão utilizados pois os mesmos não abrangem a capacidade das ferramentas de satisfazer os critérios chave para uso de Kanban como método de desenvolvimento de software.

3 AVALIAÇÃO DE FERRAMENTAS PARA USO DE KANBAN

Neste capítulo, ferramentas populares para o uso de Kanban serão descritas e avaliadas.

3.1 Contexto

O Kanban é um dos principais integrantes de uma família de métodos de desenvolvimento de software extremamente popular, os chamados Métodos Ágeis. Por este motivo, diversas ferramentas foram criadas para facilitar o seu uso. Os sites Listly¹ e TheTopTens² listam ferramentas disponíveis com este intuito, totalizando 37 e 36 ferramentas, respectivamente. Dentro deste grande universo de utensílios, há uma grande variação entre as funcionalidades oferecidas pelas mesmas, o que torna difícil para que um usuário escolha a mais apropriada para suas necessidades.

A avaliação proposta nesse trabalho visa identificar, entre as principais ferramentas disponíveis para o apoio ao uso de Kanban no âmbito do desenvolvimento de software, quais dos critérios básicos para o uso de Kanban enquanto ferramenta para controle de cadência e aumento de vazão são satisfeitos por cada uma.

3.2 Ferramentas avaliadas

Há uma série de ferramentas disponíveis para o suporte ao uso de Kanban para desenvolvimento de software. Seja por serem citadas como ferramentas populares ou por terem destaque em diversas avaliações, como as de Augustine (2017) e Burger (2016), citadas anteriormente, as seguintes ferramentas foram consideradas: Trello, Jira, LeanKit, Kanban Tool, Kanban Flow, Kanbanize, Kanbanchi, Volverro e CA Agile Central.

Entre as ferramentas consideradas, foram escolhidas para avaliação neste trabalho as seguintes:

- Trello
- Jira
- CA Agile Central

¹ <https://list.ly/list/InV-kanban-tools>

² <https://www.thetoptens.com/online-kanban-tools-for-business>

- LeanKit

Esta escolha foi feita por estas serem ferramentas extremamente populares. Trello tem mais de um milhão de usuários ativos por dia, de acordo com Konrad (2016), e é descrito por Burger (2016) como o padrão de indústria para ferramentas de Kanban. Jira, de acordo com o relatório VersionOne (2017), é a ferramenta mais popular para gerência de projetos ágeis, sendo utilizada por mais de 50% das empresas avaliadas. O mesmo relatório indica uma alta popularidade do CA Agile Central e do LeanKit, utilizados por 13% e 5% das empresas, respectivamente.

3.3 Requisitos avaliados

Como resumido na Seção 2.3, há quatro propriedades principais do Kanban que foram analisadas neste trabalho. Mais especificamente, os critérios relacionados a essas propriedades que foram avaliados são os seguintes:

1. Decomposição do processo em diversas etapas, mostradas no quadro de Kanban;
2. Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes;
3. Limitação do WIP para cada coluna do Kanban;
4. Medição do *lead time*;
5. Medição do *throughput*;
6. Geração do GFC;
7. Políticas do processo visíveis junto a cada coluna do Kanban;
8. Divisão de cada coluna do Kanban em que há trabalho sendo feito em duas subcolunas, uma para itens em andamento e outra para itens prontos.

Também foram avaliados requisitos relacionados com o uso de Kanban como uma ferramenta de apoio ao desenvolvimento de software, utilizando-se qualquer método:

1. Criação de diferentes projetos e times;
2. Separação dos itens entre os projetos ou times;
3. Capacidade de descrever os itens de trabalho;
4. Atribuição de responsável para cada item;
5. Interação entre membros do time de desenvolvimento em relação a um item de trabalho;
6. Priorização dos itens de trabalho.

A divisão dos critérios de avaliação em dois grupos ocorreu pelo motivo de que o segundo grupo de requisitos não é específico para o uso de Kanban. Assim, o mesmo pode ser reutilizado, em trabalhos futuros, para a avaliação de ferramentas de apoio ao uso de outros métodos de desenvolvimento de software.

Outros quesitos, como o preço de cada aplicação, não foram considerados nesta avaliação, por não serem relacionados às características funcionais das ferramentas.

3.4 Forma de avaliação

Foi feita uma avaliação geral de cada ferramenta, em relação aos requisitos para o uso de uma ferramenta de Kanban e de desenvolvimento de software. Para avaliar as características de aderência ao Kanban, foi projetado um quadro de Kanban, mostrado na Figura 3.1, que possui as características essenciais avaliadas: processo decomposto em diversas etapas, agrupamento de itens semelhantes, limitação do WIP para cada estágio do processo, políticas de cada etapa do processo visíveis junto a cada coluna do Kanban e subdivisão de colunas em duas, uma para itens em andamento e outra para itens prontos. A avaliação das ferramentas inclui assim a tentativa de reprodução, em cada uma das ferramentas, deste quadro.

Figura 3.1 – Quadro de Kanban a ser reproduzido nas ferramentas avaliadas

A fazer	Análise (2)		Desenvolvimento (3)		Teste (2)	Pronto
	Em progresso	Finalizado	Em progresso	Finalizado		
URGENTE						
Item 1	Item 5 Item 6			Item 7	Item 10	Item 12
NAO URGENTE						
Item 2			Item 8	Item 9	Item 11	Item 13
Item 3						Item 14
Item 4						
	Plano de teste criado Riscos identificados		Todos os testes unitários passaram Código integrado ao sistema de teste		Todos os casos de teste passaram	

Fonte: Elaborada pelo autor.

Neste exemplo, o processo é dividido nas etapas de *A fazer*, *Análise*, *Desenvolvimento*, *Teste* e *Pronto*. *Análise*, *Desenvolvimento* e *Teste* têm limites de WIP de 2, 3 e 2, respectivamente. Adicionalmente, as colunas de *Análise* e *Desenvolvimento* possuem subdivisão para itens *em progresso* e *finalizados* dentro destes estágios. As políticas do processo para os estágios de *Análise*, *Desenvolvimento* e *Teste* estão afixadas abaixo de cada uma delas. Ainda, existem dois grupos de itens, os urgentes e os não urgentes.

Além da reprodução deste quadro de modelo, também foram avaliados os requisitos de cálculo de *throughput*, cálculo de *lead time* e geração de GFC, que são relacionados a medição de fluxo do sistema e, portanto, não se manifestam em um quadro de Kanban, e os requisitos relevantes para qualquer método de desenvolvimento de software, não específicos para Kanban.

3.5 Trello

Trello³ é uma ferramenta web gratuita. É hospedada em seu próprio servidor, não sendo necessário nenhum tipo de infraestrutura própria para utilizá-lo. Possui também uma versão paga, que trás melhorias de segurança e de suporte.

Usando Trello, é possível criar diversos projetos, e cada um destes pode ter vários quadros. Um quadro é visto como uma sequência de *listas*, onde cada lista corresponde a um estágio do processo definido. Também pode-se configurar quais usuários têm acesso a cada

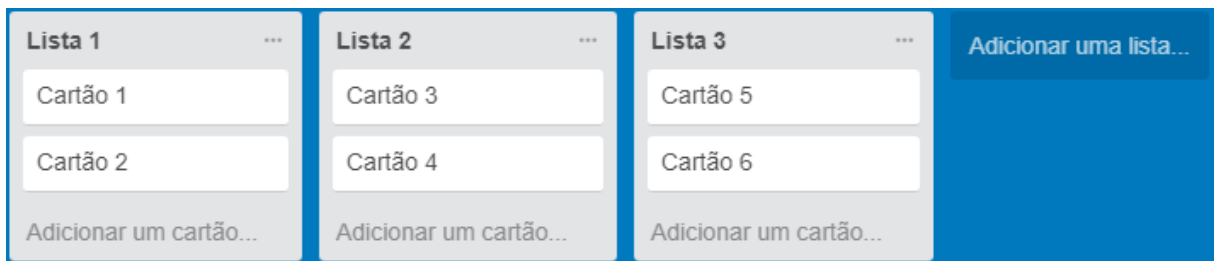
³ <https://trello.com/>

projeto, definindo-se assim times de desenvolvimento. Um mesmo usuário pode ter acesso a diversos projetos, portanto fazendo parte de vários times.

Os itens de trabalho são vistos como cartões, e são colocados em alguma das listas. É possível criar tantas listas quanto necessário e ordená-las no quadro, escolhendo-se qual será a ordem das etapas do sistema que cada item deve percorrer para ser considerado pronto. A priorização dos itens de trabalho pode ser alcançada através da ordem dos mesmos nas listas. Assim, quanto mais acima na lista um cartão está, mais prioritário ele é. Para movimentar um cartão, seja para trocá-lo de lista (estágio) ou para mudar a sua prioridade em relação aos outros cartões, deve-se arrastá-lo para a posição desejada.

Um exemplo de quadro no Trello pode ser visto na Figura 3.2.

Figura 3.2 – Exemplo de quadro no Trello

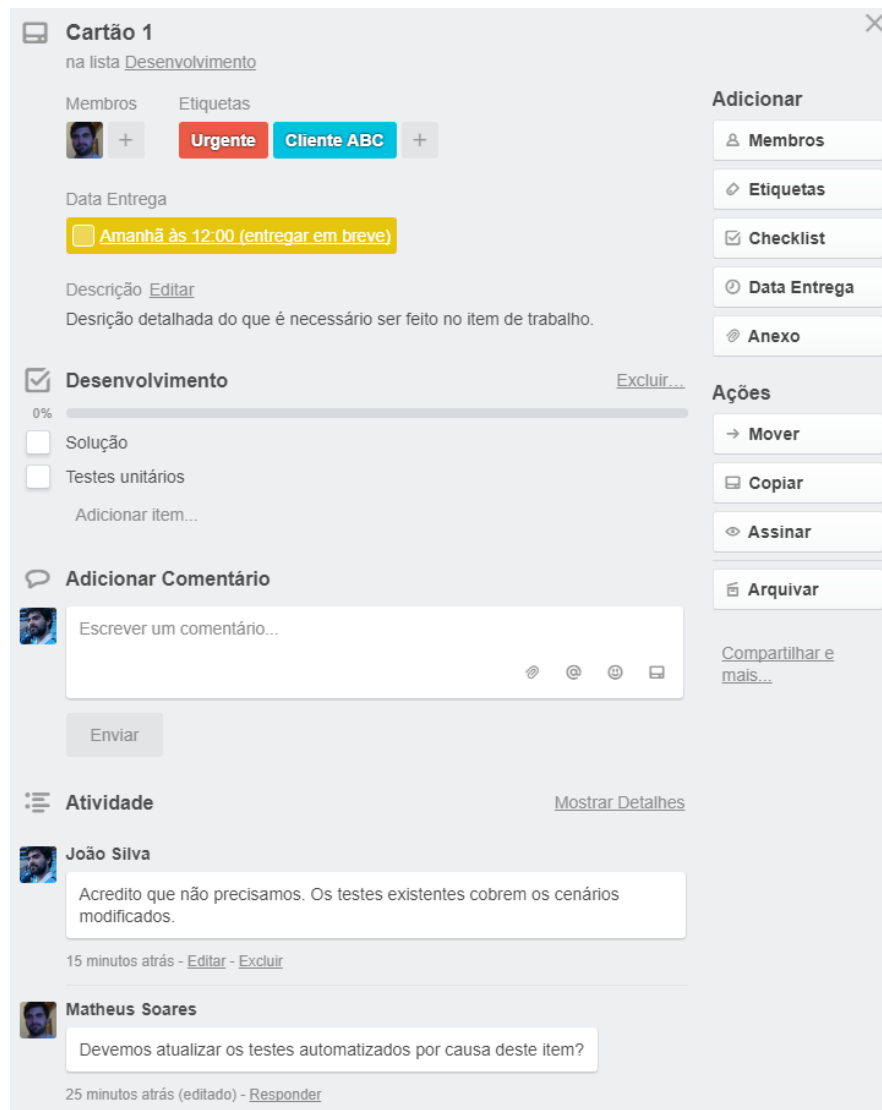


Fonte: elaborada pelo autor.

Um cartão (item de trabalho) recebe um título e uma descrição do que deve ser feito e pode ser atribuído a uma ou mais pessoas do time de desenvolvimento. Também é possível definir uma data de entrega para cada cartão, anexar arquivos e inserir comentários no item, possibilitando assim a troca de informações de toda natureza entre membros do time de desenvolvimento. Ainda, é possível atribuir etiquetas para cada cartão. Etiqueta é um conceito do Trello que corresponde a um conjunto de cor e rótulo, permitindo a categorização dos cartões. Cartões que recebem etiquetas são mostrados com as respectivas cores no quadro. Além disso, também pode-se criar um ou mais *checklists* (listas de coisas a se fazer) em cada cartão.

A Figura 3.3 mostra a visão detalhada de um cartão no Trello, contendo título, descrição, etiquetas, membro do time de desenvolvimento responsável pelo cartão no momento, data de entrega, um *checklist* e comentários de membros do time de desenvolvimento. Neste exemplo, as etiquetas são usadas para indicar a urgência do cartão (Urgente) e o cliente que requisitou este trabalho (Cliente ABC). O *checklist* indica atividades que devem ser completas para a conclusão do item.

Figura 3.3 – Cartão no Trello



Fonte: elaborada pelo autor.

O Trello conta com possibilidade de expansão através de uma funcionalidade denominada de *Power Up*. Com esta funcionalidade, habilita-se a integração do Trello com diversas ferramentas externas, como o Dropbox e o Google Drive (ferramentas para armazenamento de arquivos), o Github (sistema de versionamento de código), o Slack (ferramenta para colaboração entre membros de um time), o SurveyMoney (sistema de enquetes) e o Corrello (sistema que oferece métricas e gráficos sobre o estado e progresso dos projetos no Trello). É importante ressaltar que estas ferramentas possuem suas próprias políticas comerciais, independentes do Trello, sendo necessário pagar para ter acesso a várias delas.

Outra possibilidade de expansão das funcionalidades do Trello é através de extensões para os navegadores Google Chrome, Mozilla Firefox e Safari. De acordo com Ryder (2017) e Shanbhag (2016), algumas das extensões populares disponíveis são as seguintes:

- CardCounter for Trello: disponível para o navegador Google Chrome, adiciona a contagem de itens em cada lista ao quadro. Também possibilita que se estabeleça limites de WIP para cada lista.
- Ultimello: disponível para o Google Chrome, permite criar conexões entre cartões (para indicar que um cartão depende de outro para ser finalizado, por exemplo).
- Export for Trello: também disponível apenas para o navegador Google Chrome, permite exportar os dados de um quadro para um arquivo de planilha no formato XLS.
- Card Numbers for Trello: disponível para os navegadores Google Chrome e Safari. Cria um número identificador único para cada cartão e o mostra no quadro junto ao título do cartão, facilitando a identificação de cada item.

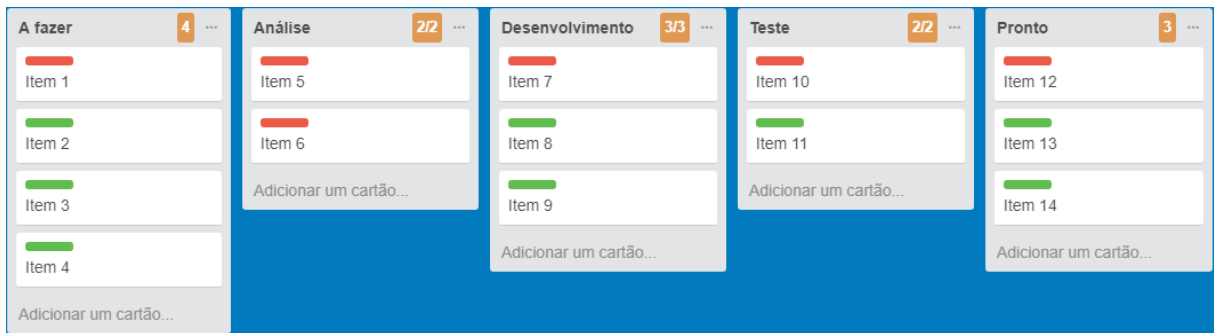
3.5.1 Reprodução do Kanban de exemplo

No Trello, é possível decompor o processo em diversas etapas e agrupar itens de trabalho semelhantes através das mesmas etiquetas. A ferramenta em sua versão padrão não permite definir limitação de WIP para cada lista (estágio do sistema), mas este recurso é oferecido através da extensão CardCounter for Trello⁴. Não é possível criar subdivisões verticais nas colunas para diferenciar entre itens em processamento e itens prontos para avançar para o próximo estágio, nem afixar as políticas do processo juntamente às colunas do quadro. Entretanto, diferentemente da limitação de WIP, não se encontrou nenhuma extensão que oferecesse estas funcionalidades. Apesar de não ser possível agrupar itens no quadro através de raias, isto pode ser feito através de etiquetas.

A Figura 3.4 ilustra a reprodução do quadro de Kanban de exemplo através do Trello. Neste quadro, a etiqueta vermelha foi utilizada para representar itens urgentes, e a verde, itens não urgentes. Os limites de WIP são mostrados à direita do nome de cada coluna. Como nas colunas de A Fazer e de Pronto não há limitação, apenas o número de itens presentes em cada um destes estágios é indicado.

⁴ <https://chrome.google.com/webstore/detail/cardcounter-for-trello/miejdnaildjcmahbhmfngfdoficmkdhi>

Figura 3.4 – Reprodução do Kanban de exemplo com o Trello



Fonte: elaborada pelo autor.

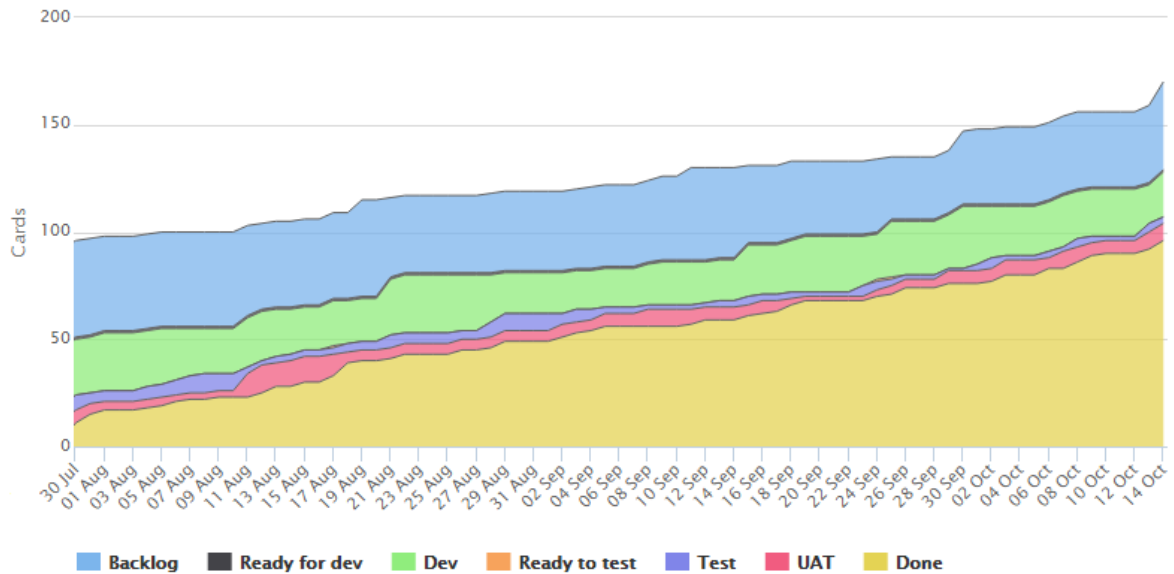
3.5.2 Métricas

O Trello não oferece suporte nativo à medição de *throughput* e *lead time*, nem geração de GFC. Entretanto, um Power Up, chamado Correlo⁵, permite o cálculo de *lead time* e de *throughput* e a geração de GFC. Como o Correlo é uma ferramenta paga, neste trabalho se utilizou uma versão de testes, que permite o acesso por um período limitado de tempo.

No Correlo, o intervalo de tempo em que o *throughput* e o *lead time* são calculados pode ser escolhido pelo usuário. O GFC é mostrado também em relação a um período de tempo definido pelo usuário. Um exemplo de GFC gerado pelo Correlo pode ser visto na Figura 3.5.

⁵ <https://getcorrelo.com/Onboard/start>

Figura 3.5 – Exemplo de GFC no Correlo



Fonte: Dashboards (s.d.).

3.5.3 Avaliação

O Trello por si só possui poucas das características avaliadas como essenciais ao Kanban. Apesar disso, a ferramenta tem como ponto positivo o fato de que há opções para estendê-la, assim sendo possível alcançar a maioria dos requisitos essenciais para um instrumento de suporte ao uso de Kanban.

Utilizando-se a integração com o Corrielo e a extensão CardCounter for Trello, apenas os requisitos de políticas de processo visíveis e de subdivisão vertical de colunas não são contemplados.

Em relação aos requisitos relevantes para qualquer método de desenvolvimento de software, todos são contempladas pelo Trello:

- É possível criar diferentes times e projetos e distribuir os itens de trabalho entre estes times;
- Os itens de trabalho (cartões) podem ser descritos e atribuídos para membros do time de desenvolvimento;
- É possível que membros do time interajam entre si à medida que o item vai sendo processado, através da seção de comentários dos cartões;
- É possível priorizar os cartões, através da ordenação deles no quadro.

3.6 Jira

Jira⁶ é uma ferramenta que pertence à empresa Atlassian. É uma solução não apenas para sistemas Kanban, mas também Scrum, outro método ágil popular. É um sistema proprietário, sendo assim necessário adquirir uma licença para a utilização do mesmo. Para esta avaliação, utilizou-se uma versão de teste, que permitiu o acesso ao sistema por um período limitado de tempo.

Possui uma versão web e também uma versão *on-premise* (local), que requer que a aplicação seja instalada em um servidor próprio. Enquanto a versão web tem como vantagens o fato de o usuário não precisar se preocupar com infraestrutura, manutenção ou atualizações da aplicação, esta tem uma limitação de customização: não é possível modificar o código fonte da aplicação e há uma quantidade reduzida de extensões que podem ser instaladas. A versão *on-premise*, por outro lado, permite tais customizações, mas traz as desvantagens relacionadas à infraestrutura.

Pode-se configurar diversos projetos, mas cada projeto pode ter apenas um quadro de Kanban. Assim, um quadro de Kanban representa um projeto por inteiro. Times de desenvolvimento são definidos através da configuração de quais usuários têm acesso a um projeto.

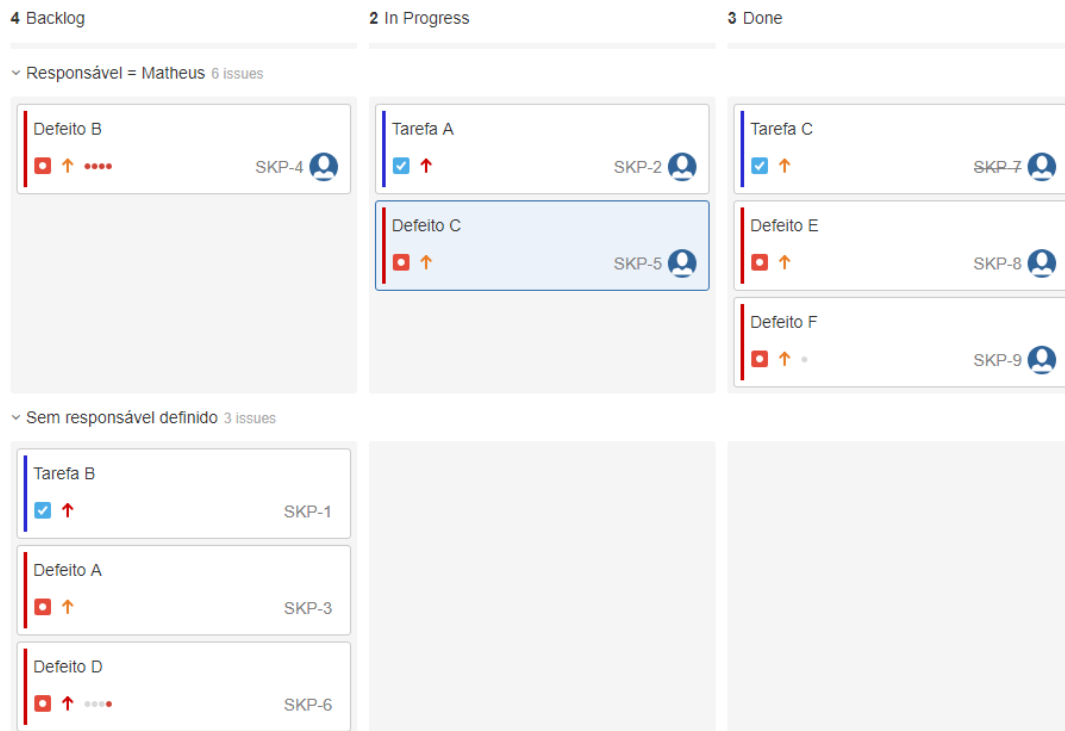
Os estágios do sistema são definidos pelo usuário e podem ser diferentes para cada quadro. Assim como no Trello, os itens também são vistos como cartões. A priorização dos cartões pode ocorrer de forma similar ao Trello, através da ordenação dos itens nas colunas do Kanban. Entretanto, o Jira também oferece outra alternativa: um atributo chamado *Priority* (prioridade). Este atributo pode receber os valores *Highest* (mais alta), *High* (alta), *Medium* (média), *Low* (baixa) e *Lowest* (mais baixa). Para se movimentar um item no quadro, seja para movê-lo para outra coluna do Kanban ou para reordenar (priorizar) os cartões, deve-se arrastá-lo para a posição desejada.

O Jira permite a categorização de itens em um quadro de duas maneiras: coloração dos cartões ou raias no quadro de Kanban. Pode-se também utilizar uma combinação das duas formas. É possível definir regras para categorização automática dos itens, atribuindo a eles uma certa coloração ou raia, de acordo com alguma de suas características (prioridade, data de criação ou data limite para entrega, por exemplo).

⁶ <https://jira.atlassian.com/>

Um exemplo de quadro de Kanban no Jira pode ser visto na Figura 3.6. Neste exemplo, existe uma raia para os itens cujo responsável é o usuário Matheus e outra para itens sem responsável definido. Há uma regra de coloração automática dos cartões: os defeitos recebem a cor vermelha, enquanto tarefas recebem a cor azul.

Figura 3.6 – Exemplo de quadro de Kanban no Jira



Fonte: elaborada pelo autor.

Um cartão recebe um título e uma descrição opcional. Diferentemente do Trello, em que um cartão é um item genérico, no Jira os cartões recebem um tipo, representando itens de desenvolvimento de software. Os tipos possíveis de itens são defeito, tarefa, subtarefa, épico e história de usuário. Apenas um membro do time de desenvolvimento pode ser atribuído como responsável por cada cartão. Também pode-se atribuir outras informações aos itens, como data limite para entrega, prioridade e arquivos anexos. Adicionalmente, a colaboração entre membros do time é possibilitada através de comentários nos cartões.

Figura 3.7 – Cartão no Jira

Sample Kanban Project / SKP-4

Defeito B

Edit Comment Assign Backlog Selected for Development Workflow Admin

Details

Type:	■ Bug	Status:	BACKLOG (View workflow)
Priority:	↑ Medium	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Labels:	None		

Description

Descrição do defeito B.

Attachments

Drop files to attach, or browse.

Activity

All **Comments** Work log History Activity Transitions

- Matheus Soares added a comment - 1 minute ago
Alguém sabe como resolver este defeito?
- Joao Silva added a comment - Just now
Matheus Soares eu sei!

Fonte: elaborada pelo autor.

A Figura 3.7 mostra a visão detalhada de um cartão no Jira. Nesta figura, o tipo do cartão é defeito (*Bug*), e este tem uma prioridade média (*Medium*). Há uma discussão entre membros do time a respeito do item na seção de comentários (*Comments*). A descrição do item encontra-se no campo *Description*. O campo *Resolution* (solução) é usado para indicar como o defeito foi resolvido. Neste caso, ele ainda não foi resolvido, e portanto este campo tem o valor *Unresolved* (não resolvido). Os campos *Affects Versions* e *Fix Version/s* são opcionalmente utilizados para indicar quais versões do produto são afetadas pelo defeito e em que versões o mesmo foi resolvido, respectivamente. O botão *Assign* (atribuir) pode ser usado para selecionar qual membro do time é responsável por um cartão. O botão *Workflow* (fluxo de trabalho) pode ser usado para selecionar em qual estágio do processo o item se encontra.

Como mencionado anteriormente, o Jira pode ser estendido através da instalação de extensões ou pela modificação do código fonte. Enquanto a segunda alternativa só está disponível para a versão *on-premise* da aplicação, a primeira é válida também para a versão web. Tais extensões podem ser compradas na loja virtual da Atlassian, chamada de Atlassian

Marketplace⁷, e permitem a modificação do sistema de diversas formas. Como exemplo, algumas extensões populares permitem a integração com o Github⁸ (sistema de versionamento de código), com o Draw.io⁹ (ferramenta para desenho de diagramas), com o Slack¹⁰ (ferramenta para colaboração entre membros de um time) e com o TestRail¹¹ (sistema de gerenciamento de testes).

3.6.1 Reprodução do Kanban de exemplo

O resultado da reprodução do Kanban de exemplo através da ferramenta Jira pode ser visualizado na Figura 3.8. Como pode ser visto nesta figura, é possível decompor o processo em diversas etapas, sendo possível delimitar o WIP de cada uma delas. Não é possível afixar as políticas do processo junto ao quadro, nem subdividir colunas entre itens em andamento e itens prontos para o próximo estágio. Raias são utilizadas para agrupamento de itens similares.

⁷ <https://marketplace.atlassian.com/addons/app/jira>

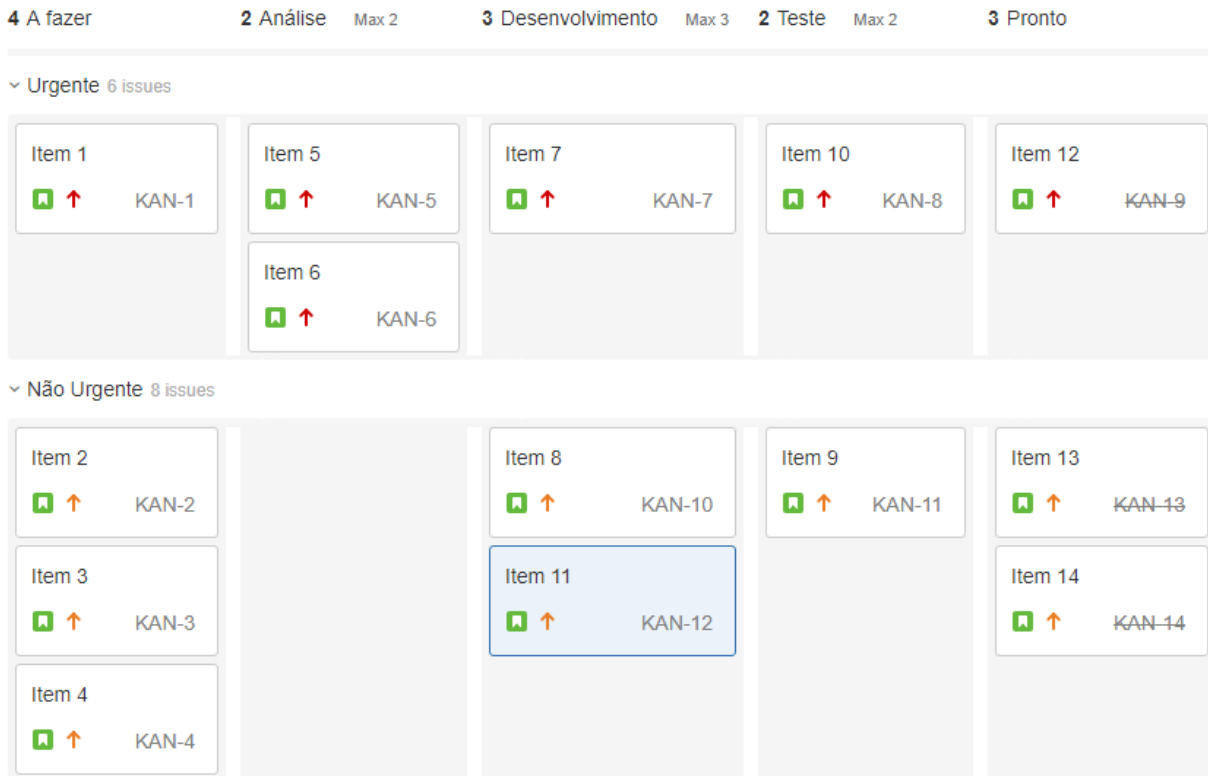
⁸ <https://marketplace.atlassian.com/plugins/com.osowskit.jira.github.app/cloud/overview>

⁹ <https://marketplace.atlassian.com/plugins/com.mxgraph.jira.plugins.drawio/cloud/overview>

¹⁰ <https://marketplace.atlassian.com/plugins/jiraslackintegration/cloud/overview>

¹¹ <https://marketplace.atlassian.com/plugins/com.testrail.jira.testrail-plugin/cloud/overview>

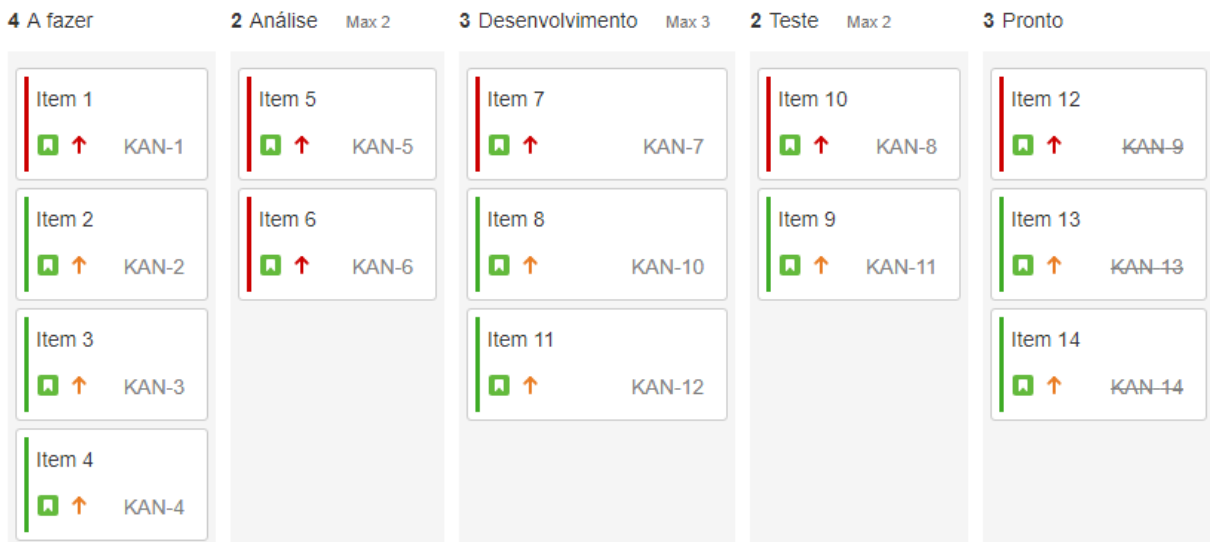
Figura 3.8 – Reprodução do Kanban de exemplo com o Jira, com indicação de agrupamento de itens através de raias



Fonte: elaborada pelo autor.

Adicionalmente, há a opção de agrupamento de itens através de regras de coloração automática. A Figura 3.9 indica o resultado do quadro utilizando-se esta técnica para diferenciação de grupos de itens ao invés de raias. Neste exemplo, definiu-se uma regra de coloração vermelha para itens urgentes e verde para itens não urgentes.

Figura 3.9 – Reprodução do Kanban de exemplo utilizando o Jira, com indicação de agrupamento de itens através de cores



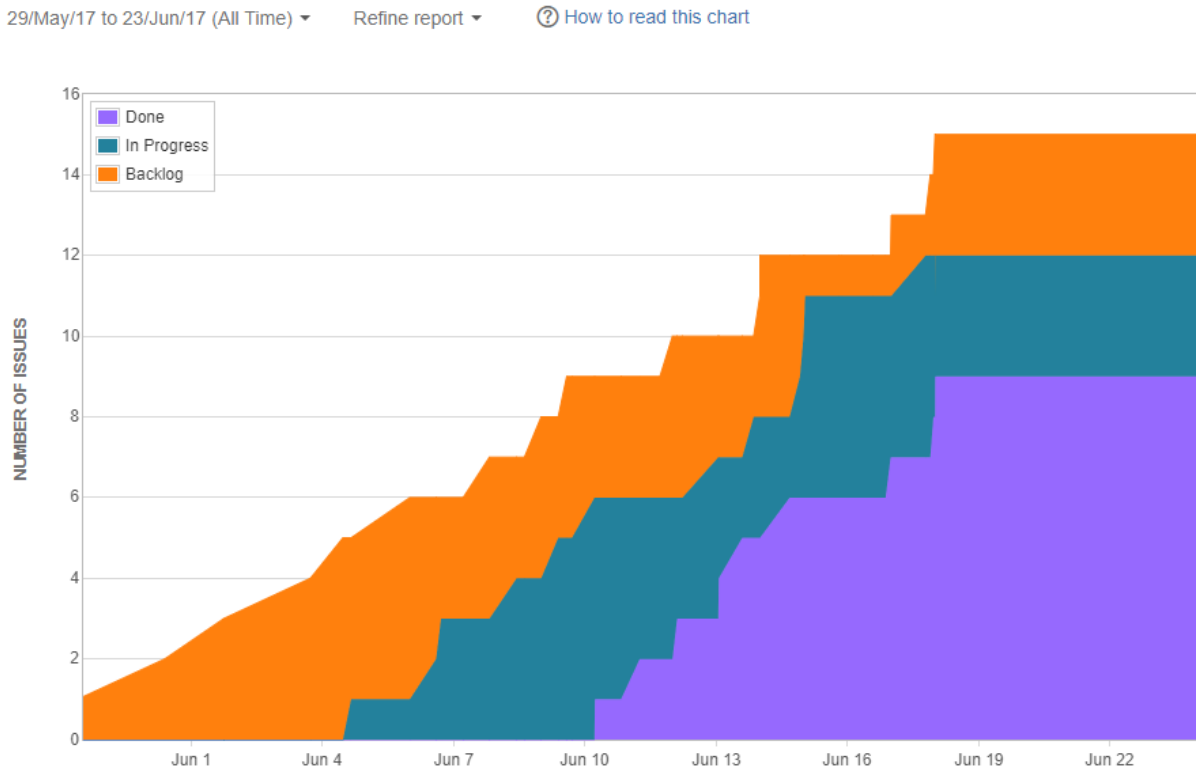
Fonte: elaborada pelo autor.

3.6.2 Métricas

O Jira oferece suporte à geração de GFC, que é mostrado em relação a um intervalo definido pelo usuário. Não há medição explícita de *lead time* ou *throughput*, sendo apenas possível inferí-los através da leitura do GFC.

Um exemplo de GFC gerado pelo Jira pode ser visto na Figura 3.10. Neste exemplo, há três estágios no sistema: *Backlog* (a fazer), indicado pela cor laranja, *In Progress* (em andamento), indicado pela cor azul, e *Done* (pronto), indicado pela cor roxa. O GFC mostra a evolução do sistema no intervalo de tempo escolhido pelo usuário, que neste caso é de 29 de maio de 2017 a 23 de junho de 2017.

Figura 3.10 – GFC no Jira



Fonte: elaborada pelo autor.

3.6.3 Avaliação

O Jira oferece seis das oito funcionalidades consideradas essenciais para um sistema de Kanban: é possível decompor o processo em estágios, limitar o WIP destes estágios, agrupar itens semelhantes através de raias ou coloração, gerar o GFC e inferir o *lead time* e o *throughput* a partir do GFC.

Apesar do Jira contar com uma extensa loja virtual, em que extensões podem ser adquiridas e integradas ao sistema, não se encontrou uma extensão capaz de prover as funcionalidades faltantes adotadas neste trabalho: afixação das políticas do processo junto ao quadro e divisão de colunas em duas subcolunas, uma para itens em andamento e outra para itens prontos.

Todas as características identificadas como básicas para uma ferramenta de suporte a desenvolvimento de software, independentemente do método utilizado, são contempladas pelo Jira:

- Pode-se criar diferentes times e projetos, distribuindo-se os itens entre os times;
- Os cartões podem ser descritos e atribuídos para um membro do time;

- A priorização pode ser feita tanto através do atributo de prioridade dos itens quanto pela ordenação dos mesmos no Kanban;
- Indivíduos podem interagir através de comentários nos cartões.

3.7 CA Agile Central

CA Agile Central¹² é uma ferramenta web. Assim como o Jira, não é focada apenas em Kanban, possuindo suporte também para Scrum. Também é um sistema proprietário, cujo uso é pago. Para esta avaliação, utilizou-se uma versão de testes, que permitiu o acesso por tempo limitado ao sistema.

No CA Agile Central, pode-se definir diversos projetos, cada um correspondendo a um time de desenvolvimento. Pode-se definir quais usuários fazem parte de cada time, sendo possível que um mesmo usuário participe de diversos times. Em cada projeto, pode-se definir diversos quadros de Kanban. Os estágios do processo são personalizados pelo usuário e podem ser diferentes para cada quadro.

De forma similar ao Jira e ao Trello, a priorização dos itens pode ser feita através da ordenação dos mesmos no quadro de Kanban. Também de forma similar às outras ferramentas, itens são movidos no quadro de Kanban ao arrastá-los para a posição desejada.

Pode-se categorizar itens no quadro de Kanban através de raias ou da coloração dos itens. Não há, entretanto, uma forma de colorir automaticamente os itens de acordo com alguma de suas propriedades, sendo necessário que a coloração seja feita manualmente caso se deseje utilizar esta forma de agrupamento.

Assim como no Jira, itens recebem um tipo, que representa um item de desenvolvimento de software. No caso do CA Agile Central, itens podem ser defeitos ou histórias de usuário. Itens recebem um título e uma descrição, esta última opcional. Também pode-se definir qual membro do time é o responsável pelo item no momento. Outras informações que podem ser colocadas em um item de trabalho são arquivos anexos e indicação de que o item está bloqueado (junto com a razão para este bloqueio). A comunicação entre usuários é possível a partir de comentários na seção de discussões de cada item. Há ainda um indicador de *Ready to Pull* (pronto para puxar), que indica se um item está pronto para avançar ao próximo estágio do sistema. Como mencionado anteriormente, um

¹² <https://rally1.rallydev.com/>

item também pode receber uma cor. Além disso, pode-se selecionar de qual projeto o item faz parte, possibilitando que um item seja enviado para outro quadro.

Pode-se criar tarefas dentro de um item, indicando coisas que devem ser feitas para a sua conclusão. Também pode-se definir uma hierarquia de itens, indicando que um item faz parte de outro, assim como identificar dependências entre itens de trabalho e sinalizar os defeitos relacionados a um item. Finalmente, o CA Agile Central permite que se crie atributos personalizados para os itens, possibilitando assim que qualquer tipo de informação desejada seja colocada no item de trabalho. A Figura 3.11 mostra a visão detalhada de um item no CA Agile Central.

Figura 3.11 – Visão detalhada de um item no CA Agile Central

Fonte: elaborada pelo autor.

Neste exemplo, há um usuário responsável pelo item, indicado no campo *Owner*. O item tem um título, mostrado no topo da imagem, e uma descrição, no campo chamado *Description*. Há um espaço para arquivos anexos (*Attachments*) e um campo personalizado: *Kanban State* (estado do Kanban), indicando a coluna do quadro de Kanban em que o item

está. O campo *Project* indica em qual projeto o item está. Ainda, pode-se ver que o item recebeu a cor azul. Através dos botões *Tasks*, *Children*, *Defects*, *Discussion* e *Dependencies*, pode-se acessar as tarefas do item, os itens “filhos”, os defeitos relacionados ao item, a seção de discussões e as dependências do mesmo, respectivamente.

O código fonte da aplicação não é disponibilizado, não sendo assim possível estender o CA Agile Central desta forma. Também não se encontrou nenhuma outra forma para a extensão da ferramenta, seja por extensões próprias do produto ou por complementos para navegadores.

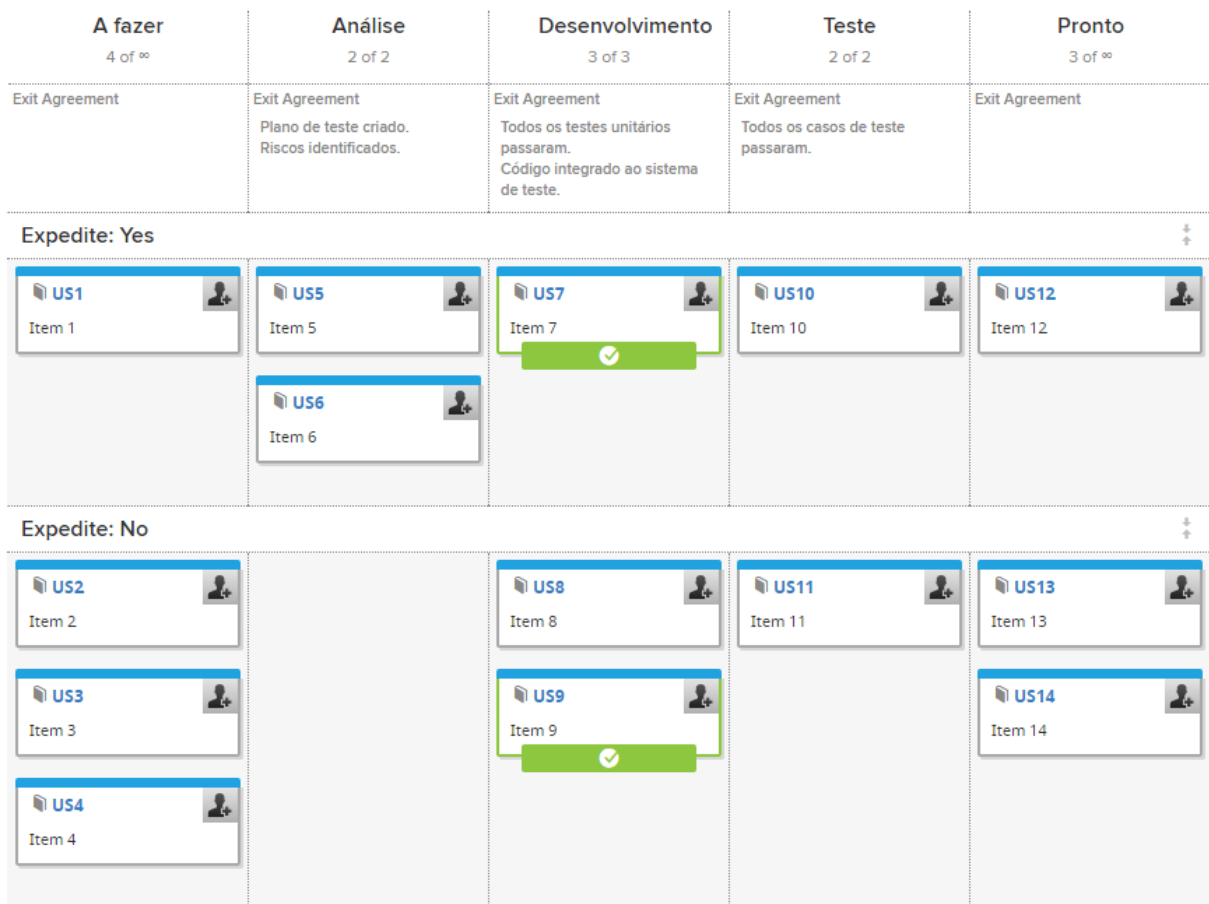
3.7.1 Reprodução do Kanban de exemplo

No CA Agile Central, é possível decompor o fluxo de trabalho em diversos estágios, bem como associar limites de WIP para cada um deles. Pode-se criar raias para o agrupamento de itens similares. Ao contrário das ferramentas anteriores, permite afixar as políticas do processo junto aos estágios do sistema.

Dentre as funcionalidades básicas de um sistema Kanban, a única que não é oferecida pela ferramenta é a subdivisão vertical de colunas. Entretanto, o CA Agile Central oferece outra possibilidade para indicar os itens que estão prontos para ser puxados para o próximo estágio do sistema: o indicador de *Ready to Pull*. Quando este indicador é ativado, uma sinalização verde é criada no item, tornando visível que o mesmo encontra-se pronto para avançar ao próximo estágio.

O resultado da reprodução do Kanban de exemplo no CA Agile Central pode ser visualizado na Figura 3.12. Como visto nesta figura, o processo pode ser decomposto em diversos estágios (colunas do quadro). É possível limitar o WIP de cada estágio e afixar as políticas do processo junto ao quadro. Neste quadro, há duas raias, denominadas *Expedite: Yes* (Urgente: Sim) e *Expedite: No* (Urgente: Não), utilizadas para agrupar itens de acordo com suas urgências. A marcação de *Ready to Pull* foi utilizada como substituta de subdivisões verticais nas colunas. Como pode-se ver, os itens 7 e 9 têm esta marcação, sinalizando que estão prontos para avançar ao estágio de Teste.

Figura 3.12 – Reprodução do Kanban de exemplo no CA Agile Central



Fonte: elaborada pelo autor.

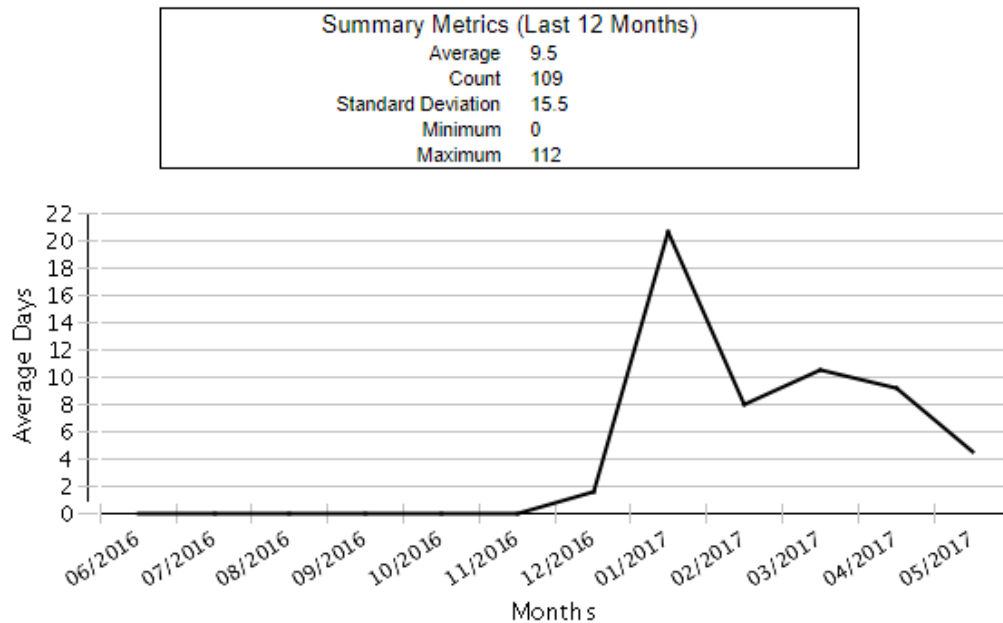
3.7.2 Métricas

O CA Agile Central oferece a geração de relatórios em relação às métricas de *lead time* e *throughput*. Em ambos os casos, o relatório indica a média mensal (*Average*) do valor nos últimos doze meses, bem como o desvio padrão (*Standard Deviation*), a contagem (*Count*) de itens concluídos e os valores mínimo (*Minimum*) e máximo (*Maximum*) da métrica nos meses do período. Ainda, o relatório conta com um gráfico, indicando a variação da métrica a cada mês. Neste gráfico, o eixo horizontal representa os meses, e o eixo vertical representa o valor médio da métrica (medido em número de itens, no caso do *throughput*, ou em dias, no caso do *lead time*).

A Figura 3.13 mostra um exemplo de relatório de *lead time* gerado pelo CA Agile Central. Neste exemplo, o período avaliado é de junho de 2016 a maio de 2017. Neste caso, a média do *lead time* no período avaliado é de 9,5 dias, com um desvio padrão de 15,5 dias. A

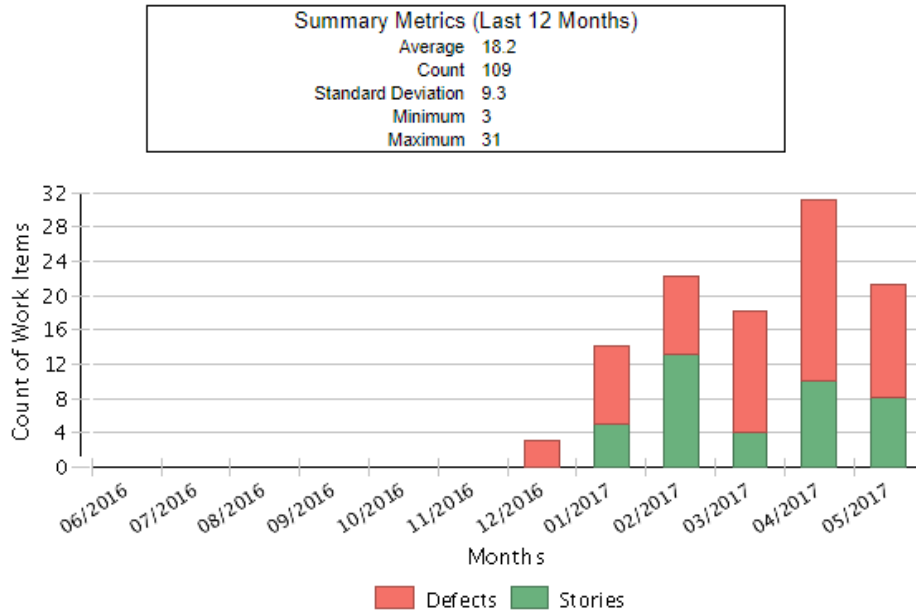
contagem de itens concluídos no período é de 109. O menor valor do *lead time* nos meses avaliados é de 0, e o maior valor é de 112.

Figura 3.13 – Relatório de *lead time* gerado pelo CA Agile Central



Fonte: elaborada pelo autor

Assim como o relatório de *lead time*, o relatório de *throughput* também apresenta as métricas de valor médio, itens concluídos no período, desvio padrão dos valores, valor mínimo e valor máximo. A Figura 3.14 mostra um exemplo de relatório de *throughput* gerado pelo CA Agile Central. No gráfico presente no relatório, cada barra, que indica o valor do *throughput* em um mês, é dividida em duas partes, de acordo com o tipo dos itens concluídos. A parte vermelha das barras indica a quantidade de defeitos concluídos em cada mês, enquanto a parte verde indica o número de histórias de usuário.

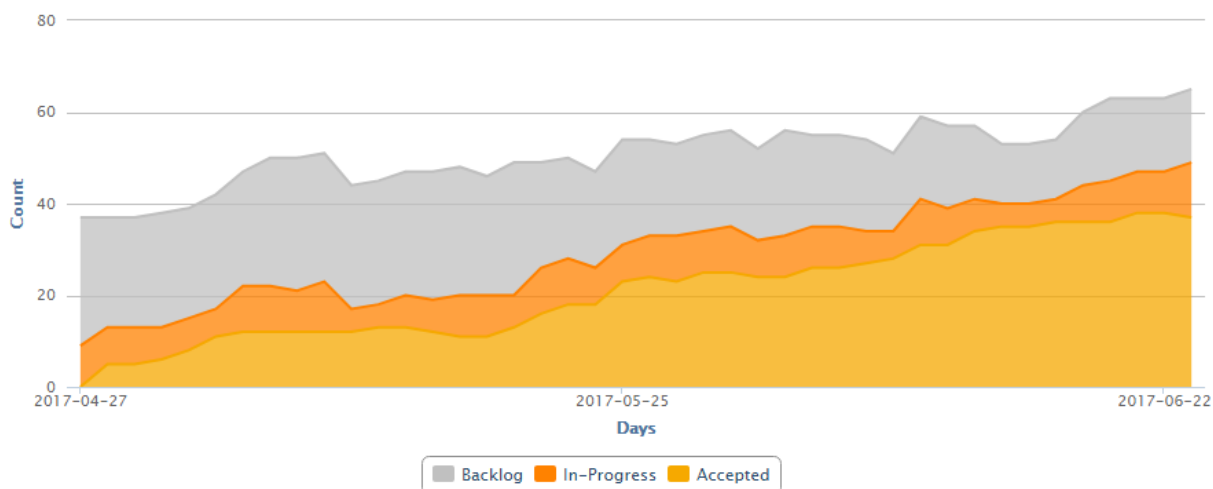
Figura 3.14 – Relatório de *throughput* gerado pelo CA Agile Central

Fonte: elaborada pelo autor.

Assim como no relatório de *lead time*, o período avaliado neste relatório é de junho de 2016 a maio de 2017. Neste caso, a média do *throughput* no período avaliado é de 18,2 itens por mês, com um desvio padrão de 9,3 dias. A contagem de itens concluídos no período é de 109. O menor valor do *throughput* nos meses avaliados é de 3, e o maior valor é de 31.

Há suporte para a geração de GFC, com um intervalo de tempo definido pelo usuário, como exemplificado pela Figura 3.15. Neste exemplo, há 3 estados no sistema: *Backlog* (a fazer), indicado no GFC pela cor cinza, *In-Progress* (em andamento), indicado pela cor laranja, e *Accepted* (aceito), indicado pela cor amarela. O GFC foi gerado em relação ao período de 27 de abril de 2017 a 22 de junho de 2017.

Figura 3.15 – GFC no CA Agile Central



Fonte: elaborada pelo autor.

3.7.3 Avaliação

O CA Agile Central cumpre todas as necessidades básicas de um sistema de Kanban, com a exceção da subdivisão vertical de colunas. Entretanto, este último requisito, que visa a indicação de itens prontos para avançar para o próximo estágio do processo, é atingido através da marcação de *Ready to Pull*. Assim, todos os requisitos de um sistema de apoio ao uso de Kanban para desenvolvimento de software são satisfeitos pela ferramenta CA Agile Central.

O CA Agile Central contempla também todas as características identificadas como básicas para uma ferramenta de suporte ao desenvolvimento de software, utilizando-se Kanban ou qualquer outro método:

- Pode-se criar diferentes projetos, cada um correspondendo a um time, distribuindo-se os itens entre os projetos;
- Os itens de trabalho podem receber uma descrição. Também, pode-se atribuir um membro do time como responsável pelo item.
- A priorização pode ser feita através da ordenação dos itens no Kanban;
- Membros do time podem interagir através de comentários na seção de Discussion (discussão) de cada item.

3.8 LeanKit

LeanKit¹³ é uma ferramenta web focada especificamente no suporte ao Kanban. É um sistema proprietário, necessitando-se pagar para ter acesso a ele. Nesta avaliação, utilizou-se uma versão de testes, que permitiu a utilização da ferramenta por um período limitado de tempo.

Nele, não há a noção de projetos, mas somente a de quadros. Assim, cada quadro representa um time de desenvolvimento. A definição de membros do time se dá a partir do controle de quais usuários têm acesso a cada quadro. Um mesmo usuário pode fazer parte de mais do que um time simultaneamente.

Os estágios do sistema são definidos pelo usuário e podem variar de acordo com o quadro. Assim como nas ferramentas avaliadas anteriormente, no LeanKit os itens também são vistos como cartões. Também de forma similar ao Jira, há duas opções para a priorização de itens: a ordenação dos mesmos no quadro ou um atributo chamado *Priority* (prioridade), que pode receber os valores *Critical* (crítico), *High* (alto), *Normal* (normal) e *Low* (baixa). Para que se mova um cartão em um quadro, este deve ser arrastado para a posição desejada.

Os tipos possíveis de itens também são definidos pelo usuário. Esta flexibilidade permite que qualquer tipo de tarefa relacionada ao desenvolvimento de software possa ser devidamente categorizada. Para cada tipo de item, associa-se uma cor, que é usada como cor de fundo dos cartões deste tipo no quadro. Há também a possibilidade de combinar a categorização pela cor dos cartões com a criação de raias no quadro.

A Figura 3.16 indica um exemplo de quadro de Kanban no LeanKit utilizando-se esta combinação. Neste exemplo, raias são utilizados para agrupar cartões de acordo com o cliente que requisitou o trabalho. Cartões que indicam melhorias recebem a cor verde, enquanto histórias de usuário recebem a cor azul, e os defeitos são coloridos de vermelho.

¹³ <https://leankit.com/>

Figura 3.16 – Exemplo de quadro de Kanban no LeanKit

A FAZER	FAZENDO	PRONTO
<p>Cliente A</p> <p>Melhoria 1</p>	<p>Cliente A</p> <p>História de usuário 1</p>	<p>Cliente A</p> <p>Defeito 1</p>
<p>Cliente B</p> <p>Melhoria 2</p>	<p>Cliente B</p> <p>História de usuário 2</p>	<p>Cliente B</p> <p>Defeito 2</p>

Fonte: elaborada pelo autor.

Além de um tipo e de uma prioridade, um cartão recebe um título (atributo chamado *Title*), bem como uma descrição (*Description*) e um identificador (*Card ID*). Também pode-se definir a data planejada de começo (*Planned Start*) e de término (*Planned Finish*) de cada item. Outros atributos disponíveis em um cartão são o tamanho (*Size*) do item, etiquetas (*Tags*), usadas na pesquisa de itens na ferramenta, a raia (*Lane*) em que o item se encontra no Kanban e um indicador de bloqueio.

Ainda, pode-se definir tarefas (*Tasks*) necessárias para a conclusão de do item, bem como inserir arquivos anexos (*Attachments*) e comentários (*Comments*). Pode-se atribuir um ou mais usuários como responsáveis por cada item (*Assigned Users*), e conectar um item com outros, que podem estar em outros quadros (através do botão *Connections*), assim permitindo desmembrar um item e distribuir as partes entre os times.

A Figura 3.17 mostra a visão detalhada de um cartão no LeanKit. Neste exemplo, o cartão tem data planejada de começo em 1 de maio de 2017, e data de término o dia 5 do mesmo mês. O item recebeu um título, uma descrição e um identificador. Ele é do tipo História de Usuário e tem prioridade Normal. O tamanho do item é 5, e ele recebeu as etiquetas “pesquisa” e “cliente”. Não há anexos, comentários, tarefas ou conexões com outros itens.

Figura 3.17 – Visão detalhada de um cartão no LeanKit

Fonte: elaborada pelo autor.

Para a extensão da ferramenta, encontrou-se apenas um complemento para o navegador Google Chrome, denominado Leankit Preview¹⁴, que possibilita que imagens anexadas a um item sejam visualizadas diretamente no navegador, sem a necessidade de fazer download das imagens. Não se descobriu nenhuma outra forma de estender as funcionalidades do LeanKit.

3.8.1 Reprodução do Kanban de exemplo

No LeanKit, o fluxo de trabalho pode ser decomposto em diversos estágios. Entre as ferramentas analisadas, é a única que permite a criação de subcolunas, de forma a indicar itens em andamento em um estágio e itens prontos para avançar para a próxima coluna. É possível definir um limite de WIP para cada coluna. Cartões podem ser agrupados por suas cores e também através da criação de raias no quadro.

A Figura 3.18 mostra o resultado da reprodução do Kanban de exemplo com o LeanKit. Neste exemplo, o agrupamento de itens é feito através de raias, denominadas

¹⁴ <https://chrome.google.com/webstore/detail/leankit-preview/lejjobpcplpbhomedjglcillhlfnllfb?hl=en>

Urgente e Não Urgente. Os limites de WIP são indicados pelo número em amarelo, mostrado ao lado do título de cada coluna.

Figura 3.18 – Reprodução do Kanban de exemplo com o LeanKit

A FAZER	ANÁLISE 2		DESENVOLVIMENTO 3		TESTE 2	PRONTO
Urgente	Em progresso	Finalizado	Em progresso	Finalizado	Urgente	Urgente
Item 1	Urgente Item 5	Urgente	Urgente	Urgente Item 7	Item 10	Item 12
Não urgente	Não urgente	Não urgente	Não urgente	Não urgente	Não urgente	Não urgente
Item 2	Item 6	Não urgente	Não urgente Item 8	Não urgente	Item 11	Item 13
Item 3	Não urgente		Item 9			Item 14
Item 4						

Fonte: elaborada pelo autor.

As políticas do processo de cada estágio podem ser descritas, mas elas não ficam diretamente visíveis no quadro. Para acessar as políticas de uma coluna, deve-se pressionar o botão cinza que fica ao lado direito do título da coluna. A Figura 3.19 mostra um exemplo de visualização das políticas de processo no LeanKit. A coluna utilizada neste exemplo foi a de Teste.

Figura 3.19 – Visualização de políticas de processo no LeanKit



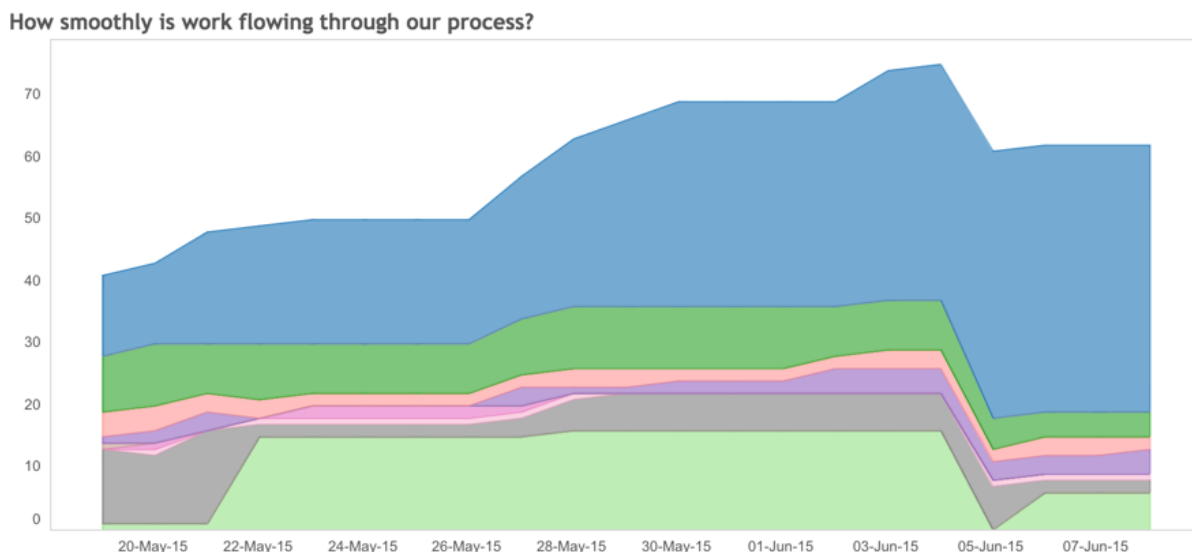
Fonte: elaborada pelo autor.

Desta forma, o LeanKit é a única das ferramentas avaliadas que cumpre todos os requisitos relacionados ao quadro de Kanban avaliados neste trabalho, sem necessitar de nenhuma forma de extensão para isto.

3.8.2 Métricas

O LeanKit tem suporte a geração de GFC, em relação a um período de tempo escolhido pelo usuário. Assim como o Jira, não há medição explícita de *lead time* e *throughput*, sendo possível apenas inferi-los a partir do GFC. A Figura 3.20 mostra um exemplo de GFC gerado pelo LeanKit.

Figura 3.20 – GFC gerado pelo LeanKit



Fonte: Reports (s.d.).

3.8.3 Avaliação

O LeanKit oferece todas as funcionalidades consideradas essenciais para uma ferramenta de apoio ao uso de Kanban: pode-se decompor o processo em várias etapas, criar raias para agrupar itens semelhantes, afixar as políticas do processo junto ao quadro, limitar o WIP de cada coluna, criar subcolunas, gerar o GFC e inferir o *lead time* e o *throughput* a partir dele.

Assim como nas outras ferramentas avaliadas, todas as características identificadas como básicas para uma ferramenta de suporte a desenvolvimento de software, independentemente do método utilizado, são contempladas pelo LeanKit:

- Pode-se criar times e quadros (equivalente a projetos) e distribuir os itens de trabalho entre estes quadros;
- Os cartões podem ser descritos e atribuídos a um ou mais membros do time de desenvolvimento;
- A priorização pode ser feita pelo atributo de prioridade dos cartões ou pela ordenação destes no quadro;
- Membros do time podem interagir através de comentários nos cartões.

3.9 Avaliação comparativa

A Tabela 3.1 apresenta a comparação entre as ferramentas Trello, Jira, CA Agile Central e LeanKit em relação aos critérios definidos na Seção 3.3.

Tabela 3.1 – Comparação entre as ferramentas avaliadas

	Trello	Jira	CA Agile Central	LeanKit
Decomposição do processo em diversas etapas	Sim	Sim	Sim	Sim
Raias para agrupamento de itens semelhantes	Não	Sim	Sim	Sim
Indicação de agrupamento de itens por coloração dos cartões	Sim	Sim	Sim	Sim
Limitação do WIP	Sim, com um complemento	Sim	Sim	Sim
Políticas do processo afixadas junto ao quadro	Não	Não	Sim	Sim
Divisão de uma coluna em subcolunas para indicação de itens prontos para avançar ao próximo estágio	Não	Não	Sim, através da marcação de <i>Ready to Pull</i>	Sim
Geração do CFG	Sim, com um complemento	Sim	Sim	Sim
Medição do lead time	Sim, com um complemento	Sim (inferido a partir do GFC)	Sim	Sim (inferido a partir do GFC)
Medição do throughput	Sim, com um complemento	Sim (inferido a partir do GFC)	Sim	Sim (inferido a partir do GFC)
Criação de diferentes projetos e times	Sim	Sim	Sim	Sim
Separação dos itens entre os projetos e times	Sim	Sim	Sim	Sim
Capacidade de descrever os itens de trabalho	Sim	Sim	Sim	Sim
Atribuição de responsável para cada item	Sim	Sim	Sim	Sim
Interação de membros do time de desenvolvimento em relação a um item de trabalho	Sim	Sim	Sim	Sim
Priorização dos itens de trabalho	Sim	Sim	Sim	Sim

Fonte: elaborada pelo autor.

4 CONCLUSÃO

O Kanban é um método bastante popular para o desenvolvimento de software. Por este motivo, um grande número de ferramentas está disponível com o intuito de apoiar o uso deste método. Entretanto, não haviam critérios claros para a avaliação e comparação destas ferramentas, que possibilitassem a escolha da ferramenta mais adequada.

Este trabalho propôs a criação de tais critérios e a avaliação de quatro das principais ferramentas disponíveis com este alvo (Trello, Jira, CA Agile Central e LeanKit) de acordo com estes requisitos. Para isso, o Kanban, enquanto método de desenvolvimento de software, foi descrito. Dele, cinco propriedades principais foram extraídas, resultando em oito critérios de avaliação. De forma a complementar estes requisitos específicos para o Kanban, outros seis critérios, relevantes para o desenvolvimento de software utilizando-se qualquer método, foram definidos.

Ao avaliar as ferramentas mencionadas anteriormente, verificou-se que apenas o CA Agile Central e o LeanKit cumprem todos os requisitos avaliados, sendo assim indicados como ferramentas completamente aderentes ao Kanban. Trello e Jira cumprem todos os requisitos gerais para o desenvolvimento de software, mas falham em alguns pontos relevantes ao Kanban.

Assim, pode-se concluir que os objetivos deste trabalho, de fazer uma avaliação das principais ferramentas disponíveis para o apoio ao uso do Kanban, definindo-se um conjunto de critérios para este fim, foram concluídos com sucesso.

Trabalhos futuros são sugeridos a fim de avaliar, utilizando-se o conjunto de critérios definidos neste trabalho, outras das ferramentas disponíveis com esta finalidade. Por estarem colocadas entre as principais ferramentas disponíveis, sugere-se que esta análise seja feita a partir das aplicações citadas na Seção 3.2 que não foram abordadas neste trabalho.

REFERÊNCIAS

ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business**. 1st ed. Sequim, Washington: Blue Hole Press, 2010.

ANDERSON, D. J.; CARMICHAEL, A. **Essential Kanban Condensed**. 1 ed. Seattle, Washington: Lean Kanban University Press, 2016.

AUGUSTINE, A. 7 Kanban Board Tools for Project Collaboration. **Lifewire**, fev. 2017. Disponível em: <<https://www.lifewire.com/kanban-board-tools-for-project-collaboration-771630>>. Acesso em: 19 mai. 2017.

BOEG, J. **Priming Kanban: A 10 step guide to optimizing flow in your software delivery system**. 2 ed. Dinamarca: Trifork A/S, 2012.

BURGER, R. 5 Standout Online Kanban Software. **Capterra**, jun. 2017. Disponível em: <<http://blog.capterra.com/5-standout-online-kanban-software/>>. Acesso em: 4 jun. 2017.

CA Agile Central Login. **CA Agile Central**. Disponível em: < <https://rally1.rallydev.com/>>. Acesso em: 27 mai. 2017.

CARDCOUNTER for Trello. **Chrome Web Store**. Disponível em: <<https://chrome.google.com/webstore/detail/cardcounter-for-trello/miejdnaildjcmaahbhmfnfgdoficmkdhi> >. Acesso em: 17 jun. 2017.

CAROLI, P. **Whisky, Sushy, Systems & Flow**. [S.l.:s.n], 2016.

CORRELO | Connect to Trello. **Correlo**. Disponível em: <<https://getcorrelo.com/Onboard/start>>. Acesso em: 15 mai. 2017.

DASHBOARDS for Kanban teams using Trello. **Correlo**. Disponível em: <<https://getcorrelo.com/trello-reports-for-kanban> >. Acesso em: 23 jun. 2017.

DRAW.IO Diagrams for Jira. **Atlassian Marketplace**. Disponível em: <<https://marketplace.atlassian.com/plugins/com.mxgraph.jira.plugins.drawio/cloud/overview>> . Acesso em: 13 jun. 2017.

EXPLORE add-ons for Jira. **Atlassian Marketplace**. Disponível em: <<https://marketplace.atlassian.com/addons/app/jira> >. Acesso em: 18 jun. 2017.

GITHUB App Plugin for Jira. **Atlassian Marketplace**. Disponível em: <<https://marketplace.atlassian.com/plugins/com.osowskit.jira.github.app/cloud/overview>>. Acesso em: 13 jun. 2017.

KANBAN TOOLS | A Listly List. **Listly**. Disponível em: <<https://list.ly/list/1nV-kanban-tools>>. Acesso em: 3 jun. 2017.

KNIBERG, H.; SKARIN, M. **Kanban and Scrum – making the most of both**. 1 ed. Estocolmo, Suécia: InfoQ, 2009.

KONRAD, A. Trello Gets Serious About Big Businesses As It Passes 1.1 Million Daily Users And Triples Sales. **Forbes**, mai. 2016. Disponível em: <<https://www.forbes.com/sites/alexkonrad/2016/05/23/trello-get-serious-about-big-businesses-as-it-passes-1-1-million-daily-users-and-triples-sales/#274598332f94>>. Acesso em: 6 jun. 2017.

LEANKIT for Business. **LeanKit**. Disponível em: <<https://leankit.com/>>. Acesso em: 3 jun. 2017.

LEANKIT PREVIEW. **Chrome Web Store**. Disponível em: <<https://chrome.google.com/webstore/detail/leankit-preview/lejjobcplpbhomedjglcillhlfhnlfb?hl=en>>. Acesso em: 23 jun. 2017.

LIKER, J. K. **The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer**. 1 ed. Nova Iorque: MacGraw-Hill, 2004.

MELO, C. D.; SANTOS, V. A.; CORBUCCI, H.; KATAYAMA, E.; GOLDMAN, A.; KON, F. **Métodos ágeis no Brasil: estado da prática em times e organizações**. São Paulo: IME-USP, Departamento de Ciência da Computação, 2012.

Q1 2015 Release Notes. **CA Agile Central Help**. Disponível em: <<https://help.rallydev.com/q1-2015-release-notes>>. Acesso em: 25 mai. 2017.

REPORTS Overview – LeanKit Help & Support. **LeanKit**. Disponível em: <<https://support.leankit.com/hc/en-us/articles/204413593-Reports-Overview>>. Acesso em: 24 jun. 2017.

RYDER, L. The Big List Of Trello Chrome Extensions. **Trello**, mar. 2017. Disponível em: <<https://blog.trello.com/big-list-of-trello-chrome-extensions>>. Acesso em: 20 jun. 2017.

SHANBHAG, A. 8 Browser Extensions You'll Love If You Use Trello. **MakeUseOf**, set. 2016. Disponível em: <<http://www.makeuseof.com/tag/8-browser-extensions-youll-love-use-trello/>>. Acesso em: 20 jun. 2017.

SKARIN, M. 10 kanban boards and their context. **Crisp**, 2010. Disponível em: <http://blog.crisp.se/mattiasskarin/files/pdf/10different_kanban_boards_and_their_context_mskarin.pdf>. Acesso em: 20 abr. 2017.

SLACK Jira Integration (Free). **Atlassian Marketplace**. Disponível em: <<https://marketplace.atlassian.com/plugins/jiraslackintegration/cloud/overview>>. Acesso em: 13 jun. 2017.

SYSTEM Dashboard – Atlassian Jira. **Atlassian Jira**. Disponível em: <<https://jira.atlassian.com/>>. Acesso em: 18 mai. 2017.

TESTRAIL for Jira Test Management. **Atlassian Marketplace**. Disponível em: <<https://marketplace.atlassian.com/plugins/com.testrail.jira.testrail-plugin/cloud/overview>>. Acesso em: 13 jun. 2017.

TOP 10 Best Online Kanban Tools for Business. **TheTopTens**. Disponível em: <<https://www.thetoptens.com/online-kanban-tools-for-business/>>. Acesso em: 13 mai. 2017.

TRELLO. **Trello**. Disponível em: < <https://trello.com/>>. Acesso em 15 mai. 2017.

VASHISHTHA, S. Agile Thinking: Stop Starting, Start Finishing. **Agile Buddha**, 2014. Disponível em: <<http://www.agilebuddha.com/agile/agile-thinking-stop-starting-start-finishing/>>. Acesso em: 6 jun. 2017.

VERSIONONE. **11th Annual State of Agile Report**. [S.l.:s.n], 2017.