

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

CLEI ANTONIO DE SOUZA JUNIOR

**Análise de integração das informações entre
Tickets e Histórias de Usuário**

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

Orientador: Prof^a. Dr^a. Karin Becker

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

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

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

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

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

RESUMO

Este trabalho realiza a análise do problema de integração entre Tickets e Histórias de Usuário, do ponto de vista de uma analista que deve tratar desta integração e consistir as informações entre os dois artefatos. Para isso, primeiramente é feita a definição da problemática e o instanciamento de um exemplo, permitindo assim a definição de critérios que serão utilizados para a análise das abordagens expostas para endereçamento do problema, que são: utilização de ferramentas de controle de tickets/projetos que permitam a manutenção das duas pontas sendo integradas em uma plataforma (Trello, JIRA e Trac).

Palavras-chave: Ticket. história de usuário. integração entre ticket X história.

ABSTRACT

This dissertation realizes an analysis of the integration problem between tickets and user stories, from an analyst point of view. This analyst must make the integration of the information between the two artifacts and consists this information. To this end, first the problem is defined, an instance is declared, allowing the definition of the criteria used to analyze the exposed approaches, which are: the utilization of tickets/projects tools, which allow the maintenance of the two artifacts in a single platform (Trello, JIRA and Trac).

Keywords: tickets,user story,ticket X user story integration.

LISTA DE FIGURAS

Figura 2.1	Exemplo de quadro Kanban.....	17
Figura 2.2	Scrum.....	19
Figura 2.3	Exemplo de cartão do Trello.....	21
Figura 2.4	Exemplo de quadro do Trello	22
Figura 2.5	Exemplo de quadro do Trello para Tickets.....	23
Figura 2.6	Exemplo de quadro do Trello para Scrum	25
Figura 2.7	Exemplo de Sprint e história de Usuário no Jira	26
Figura 2.8	Interface de Abertura dos Tickets	27
Figura 2.9	Interface de Acompanhamento de Tickets.....	27
Figura 2.10	Detalhe de um ticket na visão do analista.....	28
Figura 2.11	Exemplo de uma Sprint do Jira.....	29
Figura 2.12	Interface de Abertura de Tickets do Trac.....	30
Figura 2.13	Exemplo de uma Fila de Tickets no Trac	31
Figura 2.14	Exemplo de um Backlog de Produto no Agilo	32
Figura 2.15	Exemplo de uma história no Agilo	33
Figura 3.1	Ciclo do ticket.....	36
Figura 3.2	Exemplo: parte 1.....	37
Figura 3.3	Exemplo, parte 2.....	38
Figura 3.4	Exemplo, parte 3.....	39
Figura 3.5	Exemplo completo	40
Figura 4.1	Quadros de Tickets e do Time	47
Figura 4.2	Preenchimento do Cartão do Ticket T1	47
Figura 4.3	Criação do cartão para a história através da funcionalidade de Cópia	48
Figura 4.4	Correspondências entre campos a serem sincronizados pelo Unito	49
Figura 4.5	Sincronização de Listas no Unito	49
Figura 4.6	Quadros após o início da Primeira Sprint	50
Figura 4.7	Quadros após a finalização da Primeira Sprint	51
Figura 4.8	Recebimento do primeiro ticket.....	53
Figura 4.9	Triagem da Primeira Demanda e Criação da História	54
Figura 4.10	Triagem da Segunda Demanda e Criação das Histórias	54
Figura 4.11	Início da Sprint	55
Figura 4.12	Convenção para demandas iguais a outras existentes.....	56
Figura 4.13	Situação após a conclusão da primeira Sprint	57
Figura 4.14	Funcionalidade de Link de Ocorrência no Jira	58
Figura 4.15	História 1 criada e ligada ao Ticket 1	59
Figura 4.16	Histórias 2 e 3 criadas e ligadas ao Ticket 2.....	60
Figura 4.17	Histórias 2 e 3 ligadas ao Ticket 3	60
Figura 4.18	Funcionalidade de Criação de História Associada	63
Figura 4.19	História criada já no Backlog de Produto	63
Figura 4.20	Ticket 2 com as histórias associadas.....	64
Figura 4.21	Associação de Histórias ao Ticket 3	64

LISTA DE TABELAS

Tabela 3.1	Tabela Básica de Integração Ticket X História	43
Tabela 3.2	Tabela Estendida de Integração Ticket X História	43
Tabela 4.1	Resumo das Avaliações	66

LISTA DE ABREVIATURAS E SIGLAS

WIP Work in Progress

NA A abordagem não atende este critério.

AP A abordagem atende parcialmente este critério.

A A abordagem atende o critério.

CRM Customer Relationship Manager

SLA Service Level Agreement

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Motivação	12
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 Sistemas de Tickets	14
2.2 Métodos Ágeis	16
2.3 Histórias de Usuário	16
2.4 Kanban	17
2.5 Scrum	18
2.6 Ferramentas com Suporte a Métodos Ágeis e Tickets	20
2.7 Trello	20
2.7.1 Trello como ferramenta de Tickets	22
2.7.2 Trello como ferramenta de controle de projeto Ágil.....	24
2.8 JIRA	25
2.8.1 JIRA como ferramenta de Tickets	26
2.8.2 JIRA como ferramenta de controle de projeto Ágeis.....	28
2.9 Trac	29
2.9.1 Trac como ferramenta de Tickets.....	29
2.9.2 Trac como ferramenta de controle de projeto Scrum.....	31
3 DESCRIÇÃO DO PROBLEMA	34
3.1 Exemplo	37
3.2 Critérios Derivados	41
3.3 Integração neste Cenário Utilizando Planilhas	42
4 AVALIAÇÃO DAS ABORDAGENS	45
4.1 Integração dos artefatos na Ferramenta Trello	45
4.1.1 Soluções por meio de extensões	45
4.1.1.1 Unito	45
4.1.1.2 Exemplo com o Unito	46
4.1.1.3 Avaliação seguindo os critérios derivados - Trello com Unito	51
4.1.1.4 Placker.....	52
4.1.1.5 Exemplo - Trello com Placker	53
4.1.1.6 Avaliação seguindo os critérios derivados - Trello com Placker	57
4.2 Integração dos artefatos na Ferramenta JIRA	58
4.2.1 Aplicação do Exemplo no JIRA	59
4.2.2 Avaliação seguindo os critérios derivados	61
4.3 Integração dos artefatos na Ferramenta Trac	62
4.3.1 Aplicação do Exemplo no Trac com Agilo.....	62
4.3.2 Avaliação seguindo os critérios derivados	65
4.4 Consolidação das análises	66
5 CONCLUSÃO	67
REFERÊNCIAS	68

1 INTRODUÇÃO

Um tíquete é "um registro único eletrônico feito por um usuário autorizado de uma solicitação de serviço iniciada através de uma central de atendimento, uma interface Web ou outros meios" (KARLYN, 2013). Neste tipo de estrutura, o solicitante (geralmente o cliente), utiliza um formulário estruturado para descrever a solicitação, a qual será recebida pela equipe responsável para atendimento.

Para a abertura destes tickets, geralmente são utilizados sistemas de tickets, que são responsáveis pelo recebimento e armazenamento destas demandas, como coloca a equipe do Zendesk: "Em serviço ao cliente, um sistema de tickets ou sistema de suporte de tickets recebe as demandas de suporte dos clientes e gera automaticamente um ticket de serviço" (ZENDESKTEAM, 2017). Além disso, os sistemas de tickets também são utilizados para o gerenciamento das demandas e o monitoramento do andamento de cada item, bem como é a ferramenta para os clientes informarem a criticidade dos itens, solicitando prioridade no atendimento quando necessário.

Em resumo, um ticket é destinado ao registro da solicitação do cliente, que pode se tratar de uma demanda de qualquer tamanho. No ticket, o cliente coloca os detalhes do problema que vem enfrentando ou a descrição da melhoria que necessita ser desenvolvida no sistema para atender um cenário de utilização do mesmo.

A utilização de tickets para o recebimento de demandas de clientes é uma prática adotada em diversas empresas, devido a vantagens que serão detalhadas no Capítulo 2.

Para empresas que trabalham com o desenvolvimento de software, é comum também a utilização de métodos ágeis, tais como Scrum (SCRUM.ORG, 2017) e Kanban (CAROLI, 2017), por exemplo, para organizar e acompanhar o desenvolvimento de seus produtos, releases e demandas, bem como para acompanhar o andamento dos times de desenvolvimento ao longo do tempo. Estes métodos utilizam artefatos diferentes dos tickets e possuem diferenças como: a organização dos itens, a definição dos status e a finalidade de cada artefato.

Os métodos ágeis utilizam histórias de usuário para organização dos itens a serem desenvolvidos, as são caracterizadas por 3 pontos principais (COHN, 2009):

- Uma descrição sucinta escrita, utilizada para fins de planejamento e lembrete.
- Conversas sobre a história que servem para compreender e acordar detalhes da história.
- Testes que registram detalhes e podem ser utilizados para saber quando uma história

está pronta (RUBIN, 2013).

Desta forma, tickets e histórias de usuário diferem quanto a finalidade, visto que o ticket registra a demanda, enquanto a história de usuário corresponde a funcionalidade que será desenvolvida. Primeiramente esta é o ponto de partida para uma conversa e depois de detalhada, deve ser confirmada através de seus critérios de aceitação testáveis.

Outra diferença entre tickets e histórias refere-se aos seus estados ao longo dos ciclos de desenvolvimento, que serão detalhados no Capítulo 3.

Por último, o ticket serve para a comunicação com o cliente e/ou áreas internas da empresa, que podem ser formadas por pessoas sem conhecimento de desenvolvimento de software, o que leva a diferenças nos termos utilizados em relação aos artefatos exclusivos da equipe de desenvolvimento.

Assim, como se tratam de paradigmas diferentes, geralmente as ferramentas que disponibilizam o suporte ao atendimento de demandas em tickets, como o ZenDesk¹, FreshDesk², Siebel³ e Remedy⁴, geralmente não disponibilizam funcionalidades para o acompanhamento do desenvolvimento utilizando métodos ágeis.

Por outro lado, as ferramentas que dão apoio ao gerenciamento de desenvolvimento ágil, como CA Agile Central⁵, Icescrum⁶, Taiga⁷ e Scrumwise⁸ não oferecem um sistema de tickets integrado.

No entanto, como o ticket é o registro da demanda realizada pelo cliente e a história de usuário é o artefato que servirá de base para o desenvolvimento da solução para esta demanda, é importante que ambos artefatos sejam conectados, de forma a termos visibilidade sobre o andamento de cada um em suas respectivas plataformas de controle.

Esta conexão contribui para, por exemplo, reduzir o risco de que sejam iniciados projetos em duplicidade, eliminando assim uma fonte de desperdício. Também através da associação de ambos artefatos, o controle das demandas é facilitado e possivelmente o aumento da eficácia das soluções desenvolvidas.

Desta forma, para empresas que optam pela utilização de tickets como método padrão de recebimento de demanda do cliente e um método ágil para o desenvolvimento de soluções de atendam estas demandas, é necessário, para cada ticket válido recebido, a

¹zendesk.com.br

²freshdesk.com.br

³www.oracle.com/us/products/applications/siebel/

⁴www.bmcsoftware.com.br/it-solutions/remedy-itsm

⁵www.ca.com/Agil/Central

⁶icescrum.com

⁷taiga.io

⁸scrumwise.com

criação das histórias de usuário que serão utilizadas como base para o desenvolvimento da melhoria que solucionará a situação reportada no ticket. Após a criação destas histórias, elas precisam ser conectadas com o ticket, como descrito anteriormente, para fins de acompanhamento até a finalização da demanda.

Atualmente, existem soluções de integração entre certas ferramentas, como por exemplo a integração entre o zenDesk e o CA Agile Central através de plataformas dedicadas a esta integração, tais como as ferramentas disponibilizadas pela OpsHub⁹. Estas integrações demandam recursos adicionais para manutenção e aplicação das plataformas, além de exigir a contratação de um serviço adicional para integração de ferramentas que já são pagas. Como estas ferramentas precisam de configurações adicionais, ainda há o risco de que, se a configuração não for feita corretamente, os artefatos fiquem dessincronizados.

Ao ser utilizada esta abordagem de duas ferramentas integradas por uma terceira ferramenta, teremos diversos usuários que terão acesso as duas ferramentas, como usuários diferentes, tendo que trocar de ferramenta a cada vez que necessitar de uma informação que não é sincronizada. Além disso, como várias destas ferramentas têm o custo baseado no número de usuários, então os funcionários que tiverem acesso a ambas as ferramentas gerarão custos duplicados, o que pode ser um problema à medida que a empresa crescer.

Desta forma, o ideal é uma plataforma única, em que cada usuário possa navegar entre os paradigmas de forma fácil e nenhum funcionário precise ser representado por mais de um usuário, facilitando o uso e reduzindo o custo da solução.

Este trabalho, realiza a apresentação do problema da integração de tickets e histórias e da sincronização das informações entre estes itens em detalhes. Primeiramente, é considerado o controle dos artefatos sem qualquer estrutura de apoio e depois com o apoio de planilhas, mostrando os problemas de dessincronização enfrentados em cada um destes casos.

Posteriormente a análise de como esta integração pode ser realizada utilizando 3 ferramentas que podem atender os dois paradigmas e sua integração (i.e. recebimento de demandas através de um sistema de tickets e o desenvolvimento de software utilizando Métodos Ágeis) em uma única plataforma. São elas:

- Trello;
- JIRA;

⁹www.opshub.com/integrations/

- Trac.

Para cada plataforma, é demonstrado como o problema apresentado seria resolvido, bem como as vantagens e desvantagens.

O restante deste texto está estruturado como segue. No Capítulo 2 é feita a fundamentação teórica, detalhando o que é um sistema de tickets, o que são Métodos Ágeis, Histórias de Usuário e os dois Métodos Ágeis que utilizaremos neste trabalho (Scrum e Kanban). Posteriormente, é falado sobre as ferramentas escolhidas para avaliação (Trello, JIRA e Trac).

No Capítulo 3, será demonstrado o detalhamento do problema sendo considerado, da integração dos tickets utilizados para envio das demandas pelos clientes com as histórias de usuário utilizadas para registro das funcionalidades que serão desenvolvidas para atender a demanda recebida, demonstrando um exemplo do cenário exposto.

No Capítulo 4, é feita a avaliação de cada uma das ferramentas frente aos critérios derivados, primeiramente descrevendo como o exemplo demonstrado no Capítulo 3 pode ser atendido na ferramenta e a avaliação com base nos critérios derivados da descrição do problema. No Capítulo 5, é feita a conclusão da análise, resumindo o resultado das avaliações.

1.1 Motivação

Este trabalho é motivado pela experiência de trabalhar por cerca de 3 anos em uma equipe que utiliza tickets para receber as demandas originadas de diversas áreas, que representam as necessidades expostas pelos clientes e com base nessas necessidades, desenvolver funcionalidades em um sistema já existente e consolidado, expressando estas funcionalidades por meio de histórias de usuário, que são utilizadas para descrever o que o time de desenvolvimento fará ao longo das sprints, visto que o método utilizado é o Scrum.

Como os sistemas utilizados não possuem a integração necessária, os problemas descritos neste trabalho são enfrentados diariamente, com o controle do relacionamento entre os tickets e as histórias de usuário sendo realizado manualmente, através de processos criados para suprir as necessidades da equipe. Estes processos foram utilizados como subsídios para os critérios utilizados na avaliação das ferramentas.

2 FUNDAMENTAÇÃO TEÓRICA

Conforme citado no Capítulo 1, tickets e histórias de usuário são diferentes em diversos aspectos, como por exemplo os status que cada artefato pode assumir: nos métodos ágeis, cada item primeiramente pode estar em um estado de "Não priorizado", onde a funcionalidade a ser desenvolvida foi registrada, mas ainda não foi priorizada frente aos demais itens que devem ser desenvolvidos.

Quando é utilizado o Scrum, após a priorização do item, a história é alocada no backlog do Produto, que servirá de subsídio para determinar o backlog de cada *sprint*. No início de cada *sprint*, o item passa pelas diversas etapas do desenvolvimento, que podem ser descritas como tarefas, como por exemplo a Codificação, Teste Unitário e Certificação. Para fins deste trabalho, estas etapas do desenvolvimento serão tratadas como um status só, que será chamado de "Em desenvolvimento". Após a finalização da *sprint* haverá um conjunto de histórias finalizadas, que poderão ser liberadas no ambiente de produção quando se julgar oportuno. Ao realizar esta liberação, podemos declarar o item como finalizado.

Já no Kanban, não há a definição de ciclos, e cada item segue pelas diversas etapas de desenvolvimento de forma contínua, sendo que a quantidade de itens em andamento é determinada para cada fase, utilizando a definição de WIP. Igualmente, existe um estágio de processamento final onde o item é declarado finalizado.

Por outro lado, o ticket possui um conjunto de status menor, que pode ser resumido em:

- Aberto: o ticket foi aberto pelo cliente e aguarda atendimento.
- Em andamento: o ticket está sendo atendido por uma equipe, sendo que para este status podem ser definidos subestados para descrever estágios do andamento do item.
- Cancelado: o ticket foi cancelado e não deve mais ser desenvolvido, seja por iniciativa do cliente (desistência da demanda, por exemplo) ou por iniciativa da equipe de atendimento (inviabilidade da demanda, por exemplo)
- Finalizado: a demanda foi atendida e o ticket encerrado.

Por isso, o ticket permanecerá como "Aberto" enquanto as histórias correspondentes estejam no backlog do produto, sendo atualizado para "Em andamento" a partir do momento em que as histórias são passadas para o backlog da Sprint e uma vez a história

alocada a um ciclo de desenvolvimento percorrerá todas as fases de desenvolvimento até a entrega final do ciclo. Uma vez aceitas as funcionalidades entregues, o solicitante da demanda original deve ser informado da conclusão de seu pedido. Além disso, como os status assumidos pelas histórias de usuário e tickets são diferentes, tais diferenças precisam ser levadas em consideração na sincronização dos itens, estabelecendo um relacionamento que determine quando um ticket deve ter o status alterado devido a finalização do desenvolvimento da demanda.

Nas Seções 2.1, 2.3, 2.4 e 2.5 serão detalhadas cada uma destas estruturas, que serão posteriormente utilizadas na descrição do problema e análise das ferramentas selecionadas (Trello, Jira e Trac).

2.1 Sistemas de Tickets

Um ticket, como explicado anteriormente, é um artefato no qual um cliente encaminha os detalhes do problema/necessidade encontrado, descrevendo o que aconteceu e o que seria o resultado esperado, bem como anexando evidências que auxiliem na análise e desenvolvimento da solução. A equipe de atendimento, por sua vez, registra no ticket o andamento do atendimento, eventuais dúvidas sobre a solicitação e a previsão de atendimento. O prazo deve estar de acordo com o o SLA (Service Level Agreement - Nível de Serviço Acordado), que por sua vez provê "tanto uma declaração do que é feito quanto da qualidade do serviço prestado, de forma a justificar o custo e os recursos utilizados" (HILES, 2016).

Esta estrutura é utilizada por diversas empresas, devido às seguintes vantagens que o processo traz (TIMO, 2017; GIOVANNINI; MEANA, 1995; BIESSE, 2015):

- Rastreabilidade das demandas: com uma ferramenta de tickets, o andamento de todas as demandas podem ser acompanhadas por seus respectivos solicitantes, bem como pelos responsáveis pelo atendimento das demandas. Estes podem inclusive utilizar relatórios gerenciais e dashboards disponibilizados pela ferramenta para obter uma visão geral do andamento dos itens, podendo verificar gargalos e possíveis pontos de falha do processo de atendimento das demandas existentes.
- Possibilidade da utilização de um processo único para diversas áreas da empresa: conforme a empresa cresce, podem surgir diversas áreas especializadas, porém, para garantir a padronização do atendimento, é necessário utilizar um processo que

seja aderente a qualquer tipo de demanda. Como os tickets tendem a ser bastante flexíveis, os mesmos podem ser utilizados para este fim.

- Redução do risco de uma demanda não ser atendida, devido a problemas de comunicação: ao utilizar uma ferramenta de controle de tickets para o recebimento de demandas, há a garantia de que as demandas serão registradas e direcionadas para atendimento, de forma que os responsáveis pelo prosseguimento do mesmo recebam o item. Também se eliminam problemas de perda dos dados (que pode ocorrer caso o contato seja feito via e-mail) ou esquecimento (que pode ocorrer caso o contato seja feito por telefone), o que aumenta a satisfação do cliente.
- Facilidade no treinamento dos funcionários para utilização do processo: como as ferramentas focam no recebimento e atendimento de tickets, as mesmas não precisam ter diversos campos obrigatórios, facilitando assim a aprendizagem do processo por novos integrantes, sejam solicitantes ou atendentes.

Sistemas de tickets são: "o ponto central no qual o problema ou situação é reportado e por consequência gerenciado e controlado" (ARIZA, 2012). Desta forma, são ferramentas que permitem a organização das demandas recebidas através dos tickets para o respectivo atendimento.

Em um sistema de tickets, temos algumas funcionalidades típicas, entre as quais pode-se destacar:

- Distinção entre tipos de usuário: a ferramenta deve permitir a separação entre tipos de usuário, separando os usuários que podem abrir novos tickets (nesta categoria se encaixam os clientes) e os usuários que atendem tais tickets (nesta categoria se encaixam os analistas, entre outros).
- Descrição e Atualização dos Tickets: a ferramenta deve permitir a criação de tickets, com o registro de um título e da descrição da demanda, onde é realizado o detalhamento da necessidade que originou a criação deste ticket. A qualquer momento, podem ser agregadas novas informações que permitam resolver as demandas.
- Filas de atendimento: a ferramenta deve disponibilizar a funcionalidade da criação de filas de atendimento, para as quais o cliente deve enviar o ticket para atendimento e o analista deve retirar para analisar e encaminhar para desenvolvimento se necessário de acordo com critérios de priorização acordados na empresa ou com o cliente.
- Controle de status: a ferramenta deve permitir o controle de status de cada ticket,

conforme os status descritos no Capítulo 1

2.2 Métodos Ágeis

Como uma alternativa ao modelo tradicional de desenvolvimento de software, os métodos ágeis buscam mudar o desenvolvimento de forma que a eficácia da produção de software seja maior, com entregas mais frequentes e melhor adaptabilidade às mudanças do negócio. Todavia, "ser ágil terá implicações para a empresa que irão bem além do departamento de desenvolvimento de software" (COHN, 2011).

Métodos ágeis são utilizados para organização do desenvolvimento de software e cada um deles possui suas próprias normas e características, porém há "práticas básicas que todos os métodos compartilham, como iterações fechadas e curtas com o refinamento de planos e objetivos de forma adaptativa e evolucionária" (LARMAN, 2004). Além disto, todos têm em comum o fato de serem consideravelmente mais leves que os métodos tradicionais em termos de documentação e estruturas de controle e buscarem capacidade de adaptação a mudanças no decorrer do desenvolvimento.

Na Seção 2.3 falaremos mais sobre como são formadas as histórias de usuário, que representam requisitos. Já nas Seções 2.4 e 2.5 são detalhados dois métodos ágeis que serão considerados neste trabalho, a saber Scrum e Kanban.

2.3 Histórias de Usuário

Histórias de usuário são a estrutura utilizada para descrição das funcionalidades que serão desenvolvidas, podendo representar uma melhoria em uma funcionalidade existente (por exemplo, uma nova opção de cadastro), ou uma nova funcionalidade. As histórias de usuário são uma das bases do desenvolvimento ágil, sendo normalmente registradas em cartões.

Uma história de usuário inicia como uma ideia que é registrada sucintamente em cartão, a qual evolui até alcançar o estágio em que está madura o suficiente para desenvolvimento. Este processo pode ser resumido como: "O cartão contém o texto da história, enquanto os detalhes são trabalhados durante a Conversa sobre o item e são registrados na Confirmação" (COHN, 2009).

Ao longo deste trabalho, utilizaremos a história como unidade que registra uma

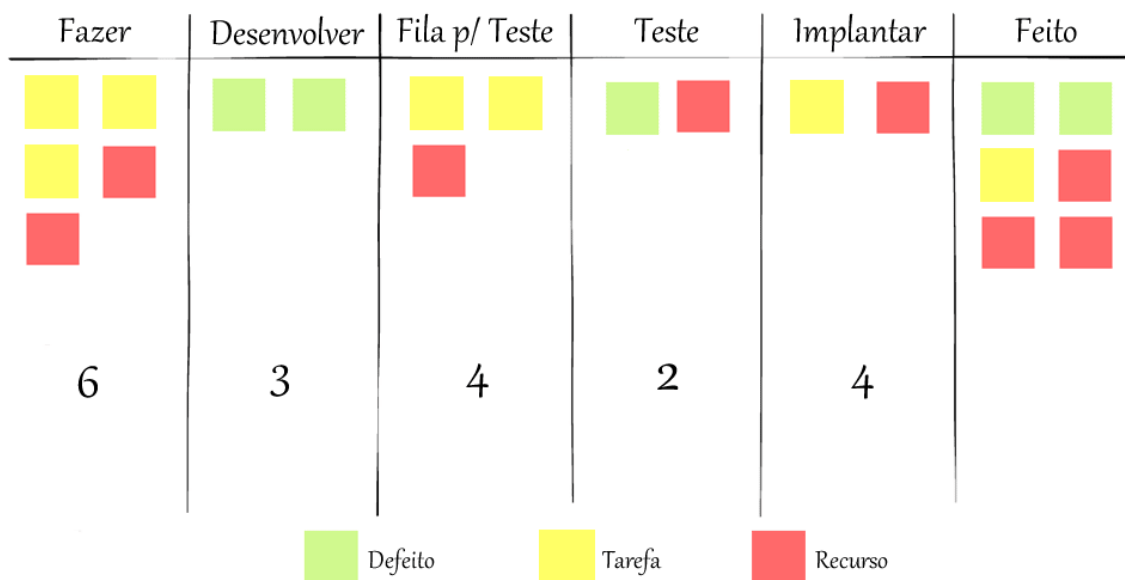
melhoria ou nova funcionalidade a ser desenvolvida, que irá passar pelos diversos estágios de desenvolvimento. Esta é originada da análise de uma necessidade registrada em um ticket.

2.4 Kanban

O Kanban é um método ágil baseado no Kanban de produção, em que há a busca contínua por melhoria, mas sem forçar mudanças bruscas na organização que o adota, de forma a reduzir os atritos. O Kanban tem "um conjunto emocionante de princípios desenhados para criar um ambiente de trabalho produtivo e saudável" (STEPHENS, 2015).

Este método é baseado na utilização de um quadro, também conhecido como *Kanban Board*, em que cada item de trabalho corresponde a um cartão a ser alocado em uma das etapas de andamento do ciclo de desenvolvimento. Esta utilização é voltada à visualização do fluxo de trabalho como um todo, de forma a possibilitar o gerenciamento deste fluxo, e a coleta de métricas (e.g. vazão, tempo médio de processamento). A Figura 2.1 mostra um exemplo de Kanban.

Figura 2.1: Exemplo de quadro Kanban



Fonte: <https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/>

No Kanban, primeiramente é necessário representar um processo de produção no quadro, onde cada etapa deste processo é representada por uma coluna do quadro. Na Figura 2.1, por exemplo, temos as fases *Desenvolver* e *Teste* como algumas das etapas do

processo representado. Já no caso da utilização do quadro para o atendimento de tickets, cada status do ticket, como descrito no Capítulo 1 deve ser representado como uma coluna do quadro.

No Kanban, cada cartão representa um item de trabalho, o qual pode ser por exemplo uma história de usuário, que foi detalhada na Seção 2.3, que irá passar pelas diversas fases de desenvolvimento. Depois de finalizados, os itens ficam na última coluna, destinada à representação da conclusão do processo de transformação de itens.

Com o quadro montado e as fases descritas, podemos então definir o limite de WIP (*Work in Progress* ou Trabalho em Andamento), que é a quantidade de trabalho que pode estar em andamento simultaneamente em cada etapa do processo. Com isto, o número de cartões em uma fase não deve exceder o limite de WIP determinado.

Com a adoção de algumas convenções, o quadro de Kanban também pode ser adaptado como base de outros métodos ágeis, como por exemplo o Scrum, que será explicado na Seção 2.5.

Desta forma, dentre as funcionalidades típicas do Kanban, podemos destacar as seguintes funcionalidades:

- Quadro de Kanban: o quadro de Kanban representa o processo, que no caso dos tickets representam o processo de atendimento e no caso de desenvolvimento, o ciclo de desenvolvimento.
- Colunas: no quadro de Kanban, cada coluna representa uma etapa do processo.
- Cartões: cada cartão registra um item que passa pelo fluxo definido no processo, que pode ser um ticket ou uma história de usuário.

2.5 Scrum

O Scrum também é um método ágil, porém tem um conjunto de normas mais detalhado que o Kanban, com diversas práticas que devem ser seguidas. "Scrum e Kanban são altamente adaptativos, mas falando de forma comparativa, Scrum é mais prescritivo que o Kanban. Scrum fornece mais restrições e, portanto, deixa menos opções abertas. Por exemplo, Scrum prescreve o uso de iterações definidas, Kanban não" (KNIBERG; SKARIN, 2010).

No Scrum, são descritos 3 papéis: "Product Owner (define a visão de produto e as prioridades), Time (implementa o produto) e Scrum Master (remove impedimentos e

provê liderança de processo)" (KNIBERG; SKARIN, 2010).

Os times implementam o produto com base em histórias de usuário, que devem ser inicialmente alocadas em um *backlog* de produto, que é a estrutura onde são armazenados os itens que aguardam desenvolvimento. É no backlog que os itens devem ser priorizados e organizados, para que à medida que o time tenha espaço para o desenvolvimento destes itens, os itens no topo do *backlog* sejam os primeiros a serem alocados para desenvolvimento.

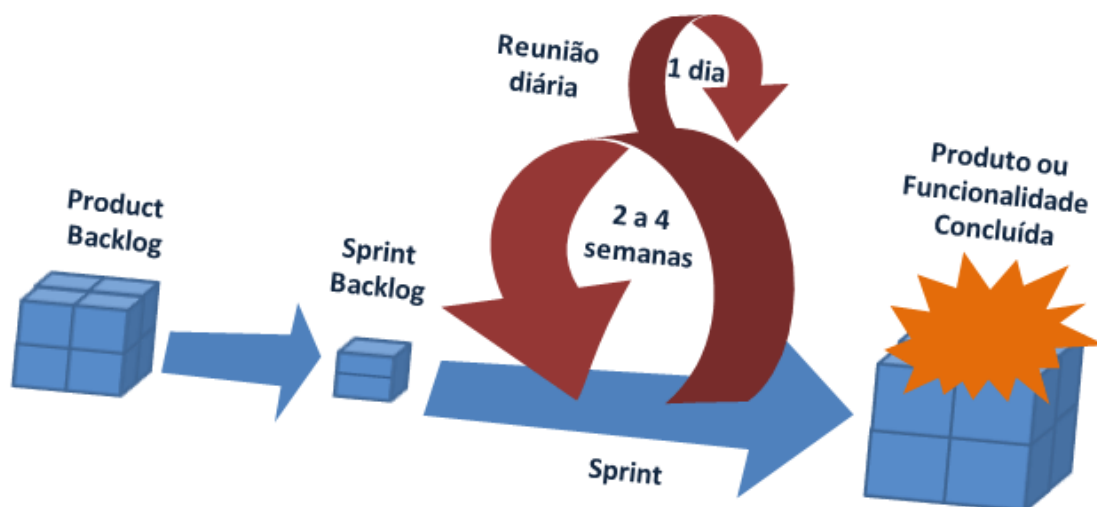
O desenvolvimento de software ocorre em ciclos de tempo determinado, também conhecidos como *Sprints*. Após a priorização dos itens no *backlog* do Produto, os mais prioritários são movidos para o *backlog* da *Sprint*.

Com base nas prioridades definidas pelo PO, no início de cada ciclo, o time de desenvolvimento estima e aloca os itens mais prioritários do *backlog* de *Sprint* para desenvolvimento, de acordo com o objetivo do sprint, a velocidade registrada do time (KNIBERG; SKARIN, 2010) e outros critérios.

Uma *Sprint* é uma unidade de não mudança, e uma vez feito o planejamento do *Sprint*, não devem ocorrer mudanças em seu escopo. As Histórias a serem abordadas em um *Sprint* são transformadas em tarefas, e compõem o backlog de *Sprint*.

Ao final de cada ciclo, espera-se que os itens com que o time se comprometeu no início do ciclo estejam finalizados, para que sejam liberados em ambiente de produção em momento oportuno. Então, o processo se repete, com o time iniciando o próximo ciclo. A Figura 2.2 exemplifica o funcionamento do processo.

Figura 2.2: Scrum



Fonte: <http://www.mindmaster.com.br/scrum/>

2.6 Ferramentas com Suporte a Métodos Ágeis e Tickets

Como descrito no Capítulo 1, no mercado existem várias ferramentas que se apresentam como ferramentas de suporte a Tickets, bem como existem ferramentas para suporte a Métodos Ágeis. Porém, ao utilizar estas ferramentas para apoiar os respectivos processos de atendimento/desenvolvimento, é necessária a utilização de uma terceira ferramenta de integração.

Neste trabalho, utilizaremos para análise 3 ferramentas de mercado mais genéricas que podem ser utilizadas para ambos os propósitos, são elas: Trello, JIRA e Trac. Estas 3 ferramentas foram escolhidas primeiramente pois são ferramentas que atendem os paradigmas de atendimento de tickets e de suporte ao uso de métodos ágeis em uma plataforma única. Além disso, cada uma destas ferramentas representa um tipo de abordagem, visto que o Trello é uma ferramenta de *Kanban* sendo adaptada, o Jira é uma ferramenta de suporte a *Kanban* e *Scrum* que possui um módulo para atendimento a tickets, e o Trac é uma ferramenta de tickets que possui extensão para dar apoio a métodos ágeis, como *Scrum*.

2.7 Trello

O Trello é uma ferramenta gratuita de organização de itens, que utiliza a estrutura do *Kanban*, explicada na Seção 2.4, disponibilizando para isto uma interface simples. De acordo com a empresa (TRELLOTEAM, 2017), "é uma lista de listas, cheia de cartões...". Com esta estrutura, o Trello busca ser adaptável a qualquer necessidade de gerenciamento de projetos, onde o usuário deve definir como organizar seus itens e qual entidade cada cartão representa.

O Trello possui 3 entidades principais, o quadro, as listas e os cartões. O quadro corresponde a todo o ciclo de processamento/desenvolvimento, onde todas as atividades, etapas e informações estarão representadas dentro de um ambiente único de gerenciamento. Já as listas corresponder às etapas nas quais um item pode estar no ciclo do projeto. Os cartões por sua vez, correspondem aos itens que a serem desenvolvidos/em desenvolvimento.

Em um cartão é possível registrar uma série de informações, permitindo a utilização dos cartões para diversos fins. Ele disponibiliza campos de descrição e comentários, bem como a classificação dos cartões por meio de cores e a utilização de checklists para a especificação de tarefas ou para a quebra do item em partes. Na Figura 2.3, temos um exemplo de um cartão sendo utilizado como um ticket, com as informações da demanda do cliente no campo de descrição e a funcionalidade de etiqueta sendo utilizada para facilmente identificar quem foi o cliente originador desta demanda. O campo de Comentários pode ser utilizado para a troca de informações e dúvidas entre o analista e o cliente.

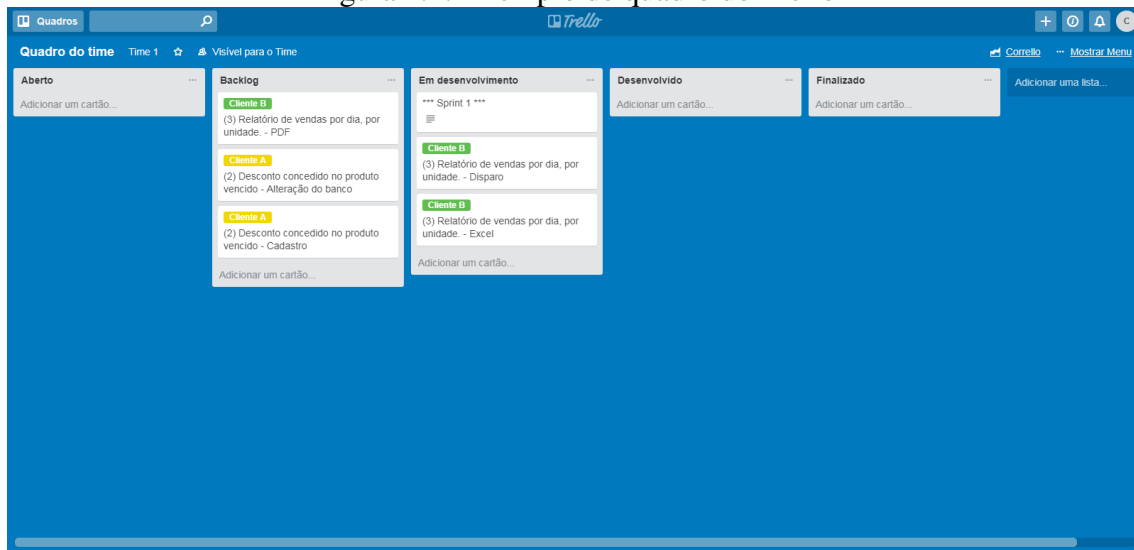
Figura 2.3: Exemplo de cartão do Trello



Fonte: Imagem criada pelo autor

Para exemplificar a utilização das 3 entidades principais, temos na Figura 2.4 o exemplo de um quadro com suas respectivas estruturas

Figura 2.4: Exemplo de quadro do Trello



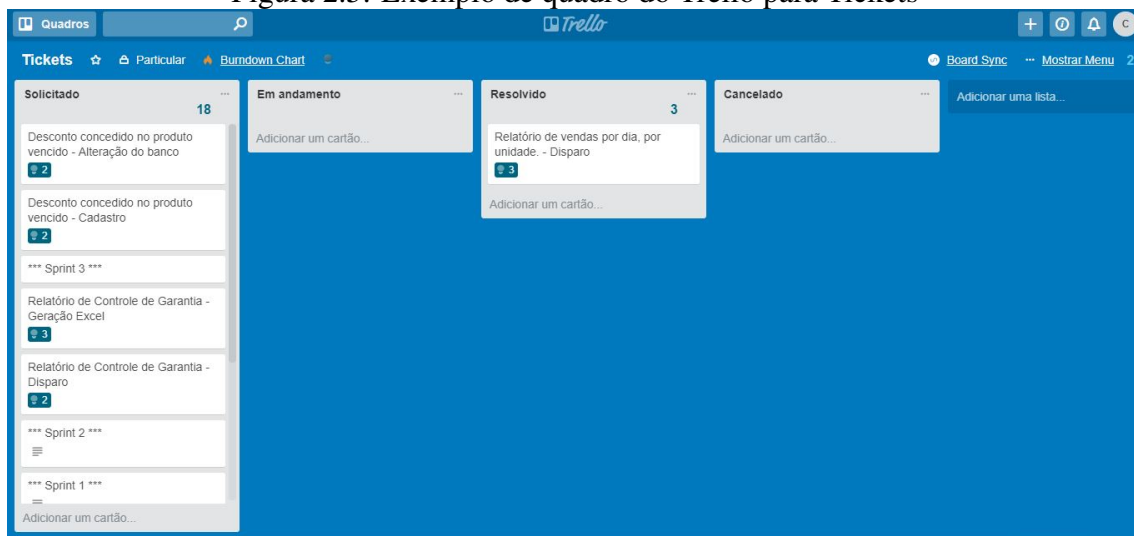
Fonte: Imagem criada pelo autor

Devido a sua característica de estar baseada em *kanban boards*, o Trello não é uma ferramenta de tickets nativa e o mesmo pode-se dizer sobre o controle de projetos de desenvolvimento ágil. Desta forma, necessitamos de algumas convenções e extensões para podermos utilizar esta ferramenta para estas finalidades, que serão detalhadas a seguir.

2.7.1 Trello como ferramenta de Tickets

Para o atendimento de tickets, podemos utilizar um quadro para registros dos tickets, conforme demonstrado Na Figura 2.5.

Figura 2.5: Exemplo de quadro do Trello para Tickets



Fonte: Imagem criada pelo autor

Neste board, os tickets serão representados por cartões, e as listas representarão os status em que o ticket poderá estar. As informações da solicitação deverão estar no cartão, bem como qualquer anexo que auxilie na análise da demanda.

Desta forma, as funcionalidades descritas na Seção 2.1 precisam ser definidas através de convenções, sendo que a fila de atendimento deve ser definida como uma coluna, preferencialmente a primeira, enquanto o título do cartão deve ser utilizado como o ticket, onde o título do cartão é o título do ticket e a descrição do cartão é a descrição do ticket.

Para a definição de troca de status utilizamos a movimentação dos cartões entre as colunas, enquanto a distinção de usuários pode ser feita com o apoio da funcionalidade de "Membros do Time" disponibilizada pela ferramenta. Para atender o público em geral (clientes), o quadro pode ser público. Já os analistas devem ser definidos membros do time, permitindo o controle de acesso as alterações nas estruturas do quadro e visualização das notas internas. Se forem necessárias funcionalidades adicionais de segurança, pode-se migrar para a versão paga da ferramenta, que disponibiliza outras funcionalidades de distinção de usuários e times. Neste caso, o cliente deverá criar os cartões para cada ticket na coluna que representa a fila de atendimento, enquanto o analista move o cartão ao longo das demais etapas.

2.7.2 Trello como ferramenta de controle de projeto Ágil

Para o controle de projetos Scrum, precisamos utilizar uma série de convenções, bem como o apoio de extensões, como por exemplo o Scrum for Trello¹. Esta necessidade acontece, pois, o Trello é uma ferramenta de Kanban, como mencionado na Seção 2.7, e desta forma estas convenções são necessárias para adaptar as funcionalidades disponibilizadas ao desenvolvimento cíclico.

Para cada história, que foi explicada na Seção 2.3, deve ser criado um cartão a ser colocado no quadro. Se necessário, podemos utilizar uma lista inicial para estes itens recém-criados, visto que ainda não foram priorizados para o Backlog de Produto.

Após o processo de priorização, estes cartões deverão ser movidos para o backlog de Produto e posteriormente de Sprint, de acordo com a utilização pelo time, que serão representados também por duas listas no quadro, com a movimentação de uma lista para a outra representando a alteração da história de um Backlog para o outro.

Para registrar a ordenação, pode-se utilizar a ordenação dos cartões nas listas que representam os Backlogs de Produto e de Time, mantendo os itens mais prioritários no topo. Ao realizar a estimativa, deve-se utilizar uma convenção para registrar a pontuação de cada cartão, a qual deve estar de acordo com a convenção entendida pela extensão utilizada. No Scrum for Trello, por exemplo, deve-se colocar a pontuação entre parênteses no início do nome da história.

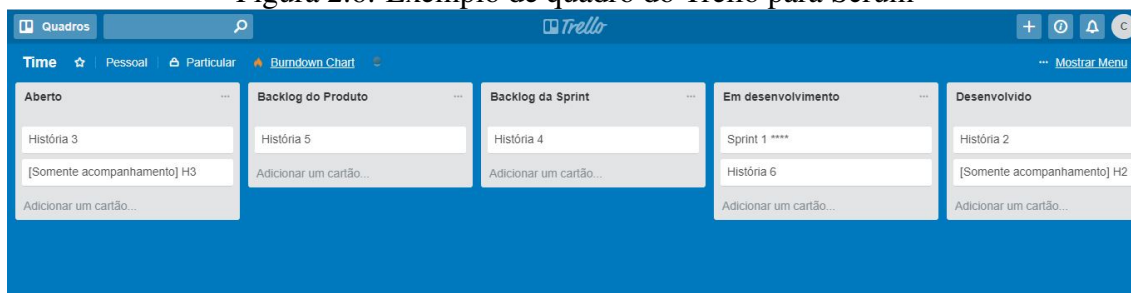
Para identificar uma sprint, também deve-se utilizar uma lista (coluna), com os cartões sendo as histórias alocadas na sprint, sendo que esta alocação é representada pela movimentação do cartão da lista de Backlog da Sprint para a lista que representa a Sprint.

Por último, para gerenciar as histórias que já foram desenvolvidas, pode-se criar duas outras listas, uma para armazenar os cartões das histórias finalizadas aguardando liberação em ambiente de produção e outra para os itens que já estão liberados, ou utilizar a funcionalidade de arquivamento do Trello para os cartões que representam as histórias de usuário entregues.

A Figura 2.6 demonstra um exemplo de como poderia ser formatado o quadro seguindo estas regras.

¹scrumfortrello.com

Figura 2.6: Exemplo de quadro do Trello para Scrum



Fonte: Imagem criada pelo autor

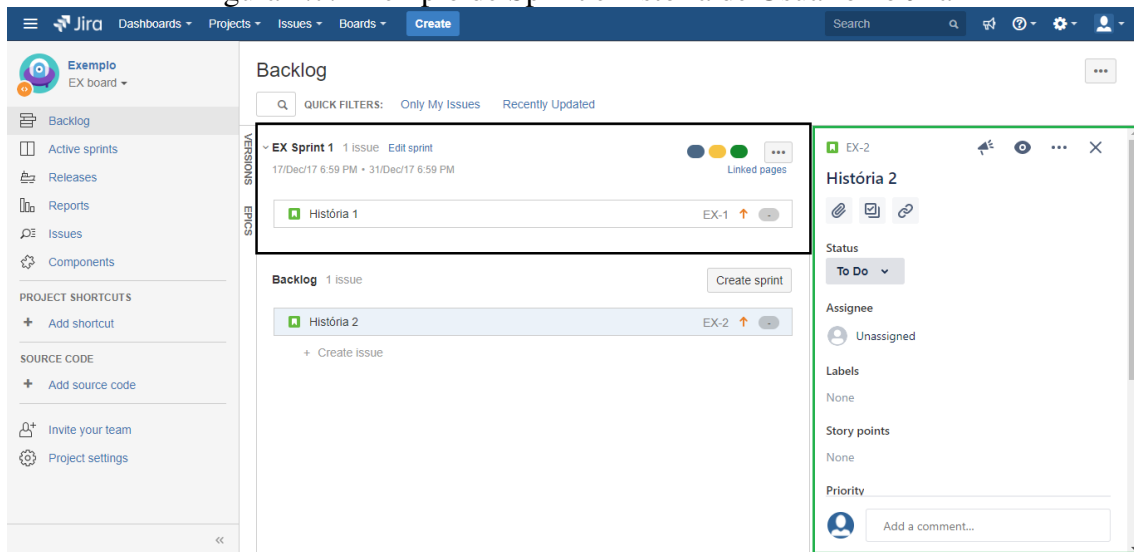
2.8 JIRA

O Jira ² é um sistema desenvolvido pela Atlassian para controle de projetos de desenvolvimento e rastreamento de problemas que forem identificados do software. Assim como outras ferramentas do ramo, o Jira possui um módulo de gerenciamento de projetos robusto, oferecendo funcionalidades importantes para a utilização de métodos como o Scrum e o Kanban.

O Jira trabalha com a noção de "Projetos" que são diferentes instâncias para tratar cada um dos paradigmas necessários. Para o uso deste trabalho, utilizaremos apenas dois Projetos, um para a parte de desenvolvimento, que utilizará o módulo do Jira de Métodos Ágeis, que utiliza como base histórias de usuário que são alocadas em um "Backlog", sendo que a partir deste Backlog, pode-se criar Sprints para o time, alocando as histórias. A Figura 2.7 demonstra essa estrutura, com a área destacada em preto representando as histórias que estão em uma Sprint em andamento, e abaixo desta área o Backlog, sendo que na área destacada em verde temos o detalhe de uma história de usuário. O outro Projeto a ser utilizado fará uso do módulo de Service Desk, que oferece as funcionalidades de um sistema de tickets.

²<https://br.atlassian.com/software/jira>

Figura 2.7: Exemplo de Sprint e história de Usuário no Jira



Fonte: Imagem criada pelo autor

Nas Seções 2.8.1 e 2.8.2, faremos o detalhamento das funcionalidades para atender cada um dos paradigmas.

2.8.1 JIRA como ferramenta de Tickets

Para a entrada de tickets, é disponibilizado o JIRA Service Desk, que serve como ferramenta de tickets, sendo composto de duas interfaces principais, a interface do solicitante e a interface do atendente/analista. Na interface do solicitante, a interface é simplificada, disponibilizando apenas os campos necessários para a informação que precisa ser comunicada, conforme demonstrado pela Figura 2.8.

Figura 2.8: Interface de Abertura dos Tickets

Fazer uma solicitação URL para clientes: <https://cleisouzajr2.atlassian.net/servicedesk/customer/portal/1/create/1>

Abrir esta requisição em nome de

Summary

Description (opcional)

Attachment (opcional)

Fonte: Imagem criada pelo autor

Depois de aberta, o solicitante pode acompanhar o andamento de suas solicitações através da funcionalidade “Meus Chamados”, conforme demonstrado pela Figura 2.9, onde aparece a lista dos tickets abertos pelo solicitante.

Figura 2.9: Interface de Acompanhamento de Tickets

Fazer uma solicitação URL para clientes: <https://cleisouzajr2.atlassian.net/servicedesk/customer/user/requests>

Help Center Solicitações 2

Centro de Ajuda Solicitações

Chamados abertos Criadas por mim Qualquer tipo de solicitação A solicitação contém...

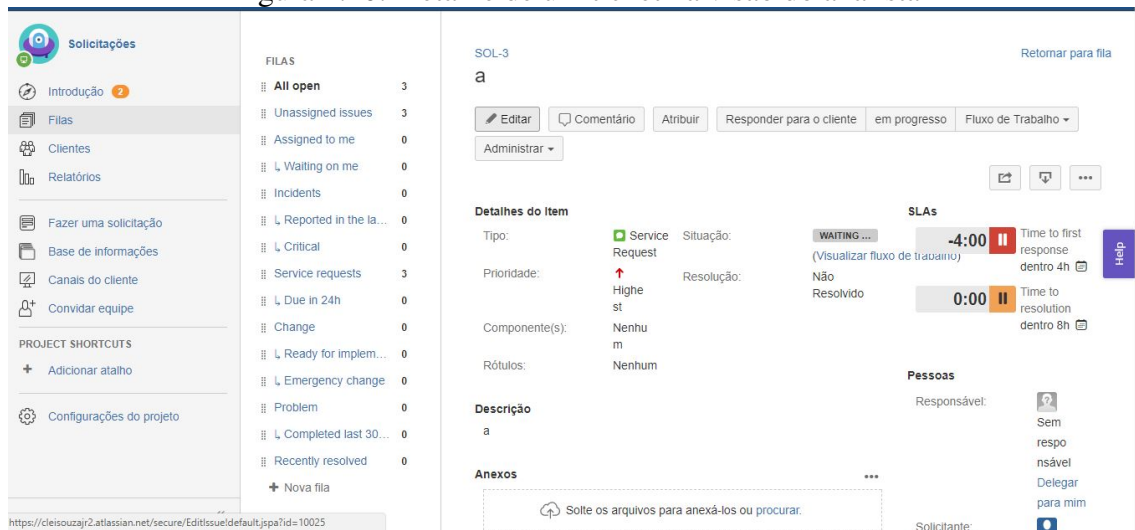
Digite	Referência	Sumário	Service desk	Estado	Solicitante
	SOL-3	a	Solicitações	WAITING FOR SUPPORT	Cleli A. Souza Jr

Fonte: Imagem criada pelo autor

Neste caso, cada ticket tem os seus estados conforme definidos anteriormente, como por exemplo Aguardando e Em andamento. É possível também ao solicitante cancelar ou solicitar prioridade no atendimento de sua demanda.

Já para o analista, a interface é bem mais complexa, com todos os campos pertinentes para o atendimento dos itens e gerenciamento das diferentes demandas. A Figura 2.10 mostra listas onde podem estar organizados os tickets, e o detalhe de um ticket em aberto.

Figura 2.10: Detalhe de um ticket na visão do analista



Fonte: Imagem criada pelo autor

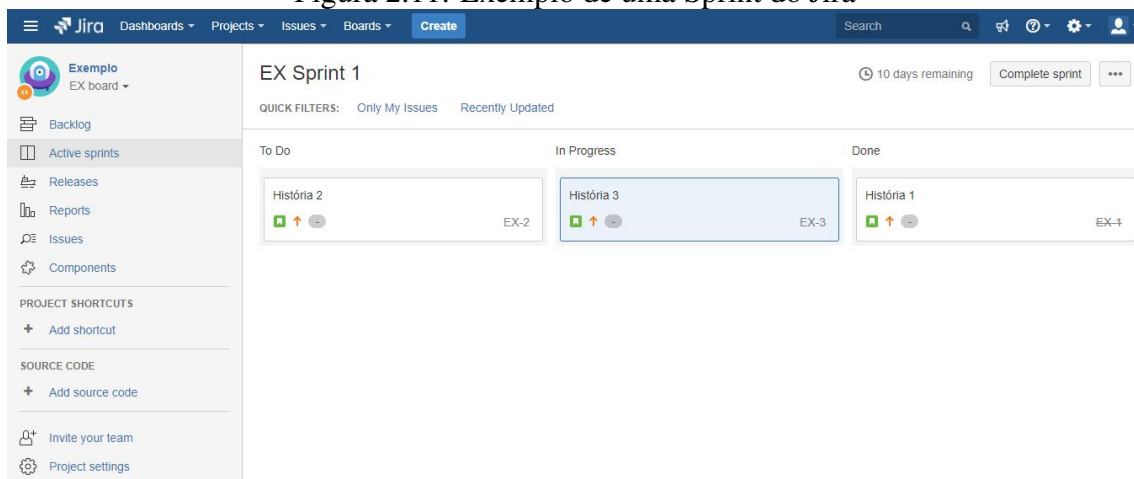
Desta forma, o analista pode verificar todos os dados referentes à demanda e ao solicitante, bem como atualizar o andamento do item e alterar a prioridade e o status conforme for necessário.

Assim, a ferramenta JIRA Service Desk, possui todas as funcionalidades dispostas na Seção 2.1, disponibilizando o a distinção de usuários entre cliente e analista, bem como a utilização das filas e status de cada ticket.

2.8.2 JIRA como ferramenta de controle de projeto Ágeis

Para o controle de projetos Scrum, o JIRA disponibiliza outro módulo, voltado para projetos de Scrum, que possibilita o controle dos itens ao longo das sprints, bem como no backlog, porém não disponibiliza a distinção entre os Backlogs de Produto e Sprint. Na Figura 2.11, temos o exemplo de uma sprint no Jira, demonstrando as três situações que uma história pode assumir durante esta Sprint (Não iniciado, em andamento e concluída). Ao final da Sprint, todos os itens deverão estar como Concluídos ou devem ser movidos para a próxima Sprint, visto que o Jira armazena as Sprints anteriores para fins de histórico.

Figura 2.11: Exemplo de uma Sprint do Jira



Fonte: Imagem criada pelo autor

O JIRA disponibiliza as funcionalidades principais de Kanban (explicadas na Seção 2.4) e as funcionalidades de suporte à Scrum (descritas na Seção 2.5), com exceção da separação entre Backlog de Produto e Backlog de Sprint. Neste trabalho, utilizaremos somente as funcionalidades de Scrum na análise da integração, visto que em ambos os casos a integração entre histórias de usuário e tickets é feita da mesma forma.

2.9 Trac

O Trac é uma ferramenta *Open Source* de controle de tickets, que possibilita a equipe de desenvolvimento receber as demandas dos clientes de forma simples, visto que esta é a proposta da ferramenta.

Devido ao Trac ser uma ferramenta aberta, há diversas extensões para a mesma que possibilitam funcionalidades adicionais, como o suporte a métodos ágeis, visto que a ferramenta não oferece a funcionalidade de suporte a Scrum ou Kanban de forma nativa.

2.9.1 Trac como ferramenta de Tickets


Como exposto na Seção 2.9, o Trac é uma ferramenta de Sistemas de Tickets, que disponibiliza a interface de criação do ticket pelo cliente, conforme demonstrado na Figura 2.12

Figura 2.12: Interface de Abertura de Tickets do Trac

Edgewall Trac | Trac Demo | Create a new ticket - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://trac-demo/trac/trac.cgi/newticket/

 **trac**
The link. No longer missing.

[Login](#) | [Help/Guide](#) | [About Trac](#)

Wiki | Browser | Timeline | Reports | Search | **New Ticket**

New ticket:

Reporter: Priority:

Component:

Version: Milestone:

Severity: Assign to:

Cc:

URL:

Summary:


Description:

Trac is really great. It's simple and straight-forward approach to issue-tracking works great for us.

However, it seems to missing a vital feature, a nice and shiny kitchen sink. Someone else also pointed this out in bug #27 earlier.

I've committed a patch in changeset [42] to fix it, please take a look at it and let me know what you think.

<http://www.bobsoftwareworkshop.com/>

 **trac**
The link. No longer missing.

Powered by Trac 0.1
By Edgewall Research & Development.

Visit the Trac open source project at
<http://trac.edgewall.com/>

Fonte: <https://trac.edgewall.org/wiki/ScreenShots>

Já para o analista, é disponibilizada uma interface para a administração da fila de tickets, bem como o atendimento de cada ticket, permitindo ao analista realizar o atendimento do mesmo. A Figura 2.13 demonstra esta interface, mostrando um exemplo de lista de tickets abertos.

Figura 2.13: Exemplo de uma Fila de Tickets no Trac

The screenshot shows the Trac web interface for the marketing team. The browser title is "{1} Active Tickets - marketing-team - Mozilla Firefox". The URL is "https://fedorahosted.org/marketing-team/report/1". The page is logged in as Chris Roberts. The main content area displays a list of active tickets, sorted by priority. The table below is a representation of the data shown in the screenshot.

Ticket	Summary	Component	Version	Milestone	Type	Owner	Status	Created
#136	Fedora Flyer for Virtualization and Cloud	Press Kit			task		new	09/06/13
#133	Update Firefox bookmarks for F20 release	Release deliverables		Fedora 20	task	chrisroberts	assigned	09/05/13
#138	Marketing goodies for Ambassadors	Release deliverables		Fedora 20	task	ankursinha	assigned	09/17/13
#139	Fedora 20 release flyer for home users	Release deliverables		Fedora 20	task	ankursinha	assigned	09/17/13
#140	Fedora 20 release flyer for developers	Release deliverables		Fedora 20	task		new	09/17/13
#141	Update generic Fedora flyer	Release deliverables		Fedora 20	task		new	09/21/13
#142	Create F20 screenshots	Release deliverables		Fedora 20	task	chrisroberts	new	10/28/13
#131	Get infra to scrape easyfix tasks from marketing trac also	other		Future releases	enhancement		new	09/03/13
#134	History of Fedora talking point	Feature Stories		Future releases	task	chrisroberts	new	09/05/13
#135	FPL Retrospective for 10th Anniversary of Fedora	Feature Stories		Future releases	task		new	09/06/13
#137	Fedora Flyer for Server Topics	other		Future releases	task		new	09/06/13
#83	Visual "where are we in the release cycle" indicator	other		F14: Pre-alpha	enhancement		new	11/08/09
#11	Prepare blurbs for "Fedora fun projects" rotation on fp.o	Fedora insight		F14: Pre-alpha	task	duffebag	assigned	08/10/09
#12	Prepare short stories for "I use Fedora" rotations on fp.o/FI	Fedora insight		F14: Pre-alpha	task		new	08/10/09
#39	Gather market research requirements from other teams	Market research		F14: Pre-alpha	task	rbergero	new	09/16/09
#41	Draft a marketing research timeline for the remainder of F13	Market research		F14: Pre-alpha	task	rbergero	new	09/16/09
#42	Watch Planet Fedora for potential FI articles	Fedora insight		F14: Pre-alpha	task	jonrob	new	09/16/09
#63	Generic Fedora slogan for fedoraproject.org front page	Branding		F14: Pre-alpha	task	pfriends	assigned	09/16/09
#106	Marketing Video Editing Guide	other		F14: Pre-alpha	task		new	03/23/10
#112	Write a "how to use the Marketing Trac queue" SOP	other		Fedora 20	task	chrisroberts	new	04/27/10
#122	Docs project feature story	Feature Stories		Future releases	task	jzb	new	06/24/10

Fonte: [https://fedoraproject.org/wiki/File: TracTickets1.png](https://fedoraproject.org/wiki/File:TracTickets1.png)

2.9.2 Trac como ferramenta de controle de projeto Scrum

O Trac possui diversas extensões para contemplar as funcionalidades para suporte a Scrum (Seção 2.5), dentre estas extensões, destacaremos o Agilo for Trac ³.

Nesta extensão, há o suporte para os Backlogs de Produto e de Sprint como entidades separadas, permitindo ao analista alocar as histórias no Backlog de Produto e conforme a priorização, alocar no Backlog de Sprint, por meio da alteração da Sprint da história de "Vazio" para a Sprint desejada, visto que a ferramenta trabalha com um Backlog de Sprint para cada Sprint. Na Figura 2.14 é demonstrado um exemplo do Backlog de Produto com alguns itens, que podem ser a qualquer momento movidos para o Backlog de uma Sprint.

³www.agilofortrac.com/

Figura 2.14: Exemplo de um Backlog de Produto no Agilo

The screenshot shows the Agile Backlog tool interface. The main content area displays a 'Product Backlog' with 8 items. The items are listed in a table with the following columns: ID, Summary, Business Value Points, Roif, Priority, and User Story Points. The items are as follows:

ID	Summary	Business Value Points	Roif	Priority	User Story Points
#36	Test Requirement	500	n.a.		n.a.
#37	Referenced mandatory story			Mandatory	
#40	An exciter story for the example			Mandatory	
#38	A linear story			Linear	
#39	Another linear story			Linear	
#31	Test story			Mandatory	
#32	A linear story			Linear	
#33	Another linear story			Linear	
Totals:		500	n.a.	n.a.	n.a.

The interface also includes a sidebar with navigation options like 'Wiki', 'Timeline', 'Roadmap', 'Browse Source', 'View Tickets', 'Search', 'Admin', 'Dashboard', and 'Teams'. The bottom section contains buttons for 'Backlog' (Save Order & Changes, Default Sorting, Tickets, Delete Selected, Remove Selected) and 'Planning' (Calculate Story Points/Time, Confirm Commitment).

Fonte: Imagem criada pelo autor

Depois de movida para o Backlog da Sprint, o time ao iniciar a Sprint deve ir atualizando o Status das histórias, até a finalização desta história, que então deverá ser marcada como "Resolvida", visto que a ferramenta utiliza as funcionalidades do sistema de ticket para endereçar também as histórias. Na Figura 2.15 temos a demonstração do detalhe de uma história, com as funcionalidades de atribuição da Sprint e de marcar a história como "resolvida" destacadas.

Figura 2.15: Exemplo de uma história no Agilo

The screenshot shows a JIRA Agile ticket page for 'História 1'. The page is divided into several sections:

- Header:** Includes navigation icons (Wiki, Timeline, Roadmap, Browse Source, View Tickets, Whiteboard, Teams, Dashboard) and user information (Logged in as admin, Preferences, Admin, Sign out, HELP).
- Left Sidebar:** Contains navigation links for Actions (New Requirement, New Task, New User Story, New Bug), Backlogs (Product Backlog, Sprint Backlog For Sprint: Sprint 1), Tickets (Active Tickets, Active Tickets By Version, Active Tickets By Milestone, Accepted, Active Tickets By Owner, All Tickets By Milestone (including Closed), My Tickets, Active Tickets, Mine First), and Wiki (Index By Title, WikiStart, WikiRestructuredTextLinks, WikiRestructuredText, WikiProcessors, WikiPageNames).
- Main Content Area:**
 - Summary:** História 1
 - Reporter:** Analista
 - Description:** A large text area with a rich text editor toolbar (B, I, A, etc.) and a note: "You may use WikiFormatting here."
 - Type:** User Story
 - User Story Points:** A dropdown menu.
 - Importance:** A dropdown menu.
 - Keywords:** A dropdown menu with "Sprint" selected and "Sprint 1" as an option.
- Action Section:** A list of actions with radio buttons:
 - leave as new The ticket will remain with no owner.
 - resolve as fixed The resolution will be set. Next status will be 'closed'.
 - reassign to admin The owner will be changed from (none) to admin. Next status will be 'assigned'.
 - accept The owner will be changed from (none) to admin. Next status will be 'accepted'.

Fonte: Imagem criada pelo autor

3 DESCRIÇÃO DO PROBLEMA

Conforme descrito anteriormente, dado um ambiente em que haja uma ferramenta para o recebimento de tickets e outra para o controle dos projetos de desenvolvimento, é necessário uma forma de integração entre as partes, com a conversão das demandas recebidas através dos tickets em histórias de usuário e a sincronização de informações entre os registros em ambas as ferramentas.

Neste cenário, podemos citar 3 personas que fazem parte do ciclo da demanda:

- **Solicitante:** é o usuário final que solicita a demanda de desenvolvimento, descrevendo o problema ou situação que enfrenta e o levou a solicitar o desenvolvimento. Normalmente corresponde a um cliente, que pode ser interno (um membro de outra área da empresa) ou externo (diretamente o cliente da empresa).
- **Analista:** responsável pela triagem dos tickets recebidos e controle das demandas, bem como o direcionamento das mesmas para o time de desenvolvimento.
- **Time de desenvolvimento:** grupo de pessoas que realiza o desenvolvimento das demandas recebidas, utilizando a metodologia escolhida.

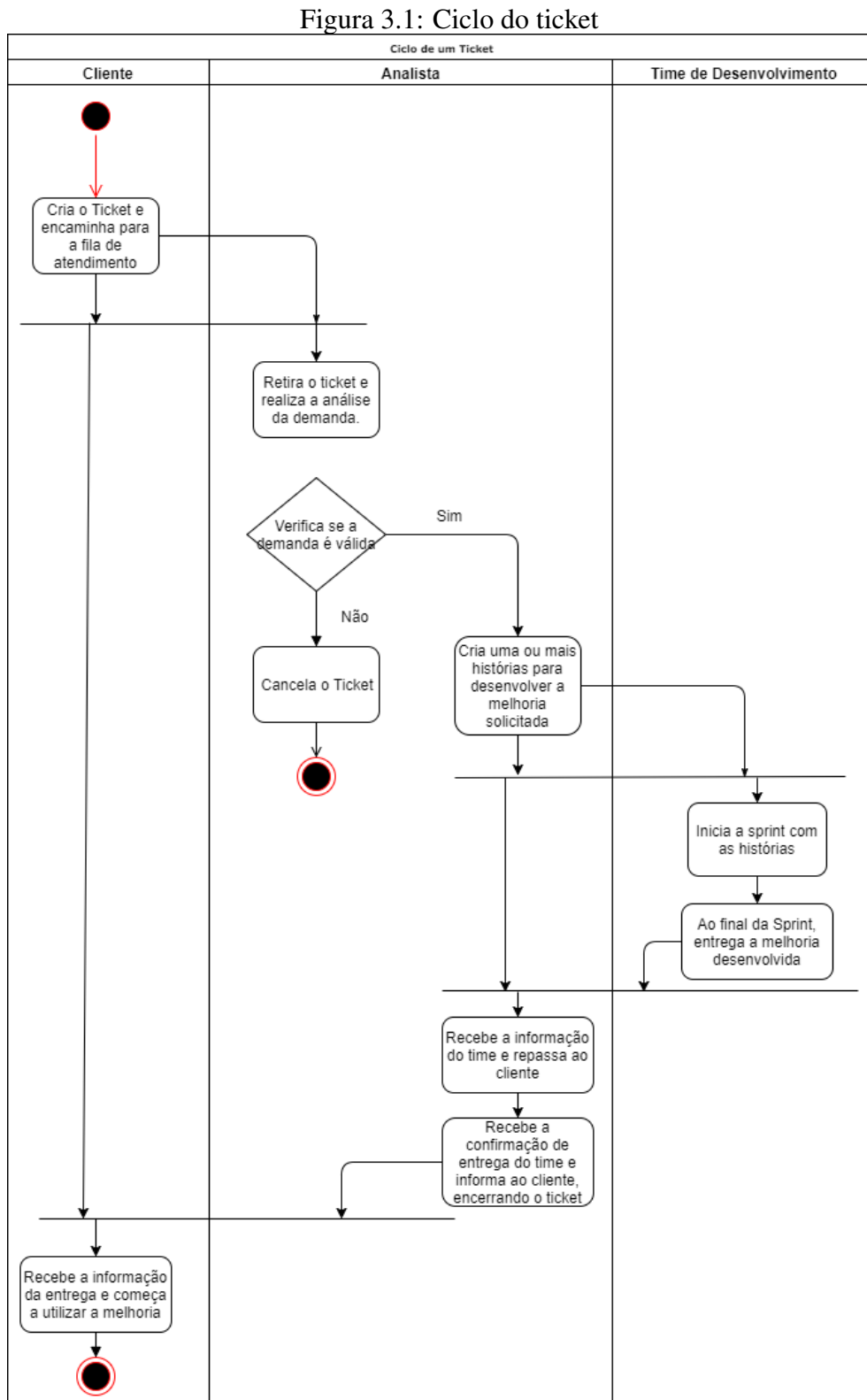
Neste trabalho, buscaremos demonstrar como pode ser realizada a integração das histórias de usuário e dos tickets, apoiando as atividades do analista, que tem as seguintes responsabilidades:

- **Controle da fila de tickets**
 - **Triagem dos novos tickets, para posterior análise das demandas:** ao receber uma nova demanda, o analista deve realizar a triagem deste item, verificando se o ticket demonstra uma demanda válida e se as informações para a análise do item são suficientes.
 - **Monitoramento dos tickets abertos:** com uma forma a buscar que uma demanda fique parada por muito tempo ou que as prioridades não sejam atendidas, o analista deve monitorar os tickets abertos, verificando há quanto tempo cada item está aberto e se há itens que não estão em andamento.
 - **Encerramento dos tickets atendidos:** após a entrega do desenvolvimento da solução de cada demanda, o analista deverá encerrar o ticket encaminhado pelo solicitante.
- **Comunicação com outras áreas da empresa e clientes**

- Contato com os solicitantes: o analista deve manter contato com os solicitantes, para que seja possível a solução de dúvidas que surgirem do time de desenvolvimento durante o processo de criação da solução para a demanda, bem como se for necessário realizar alinhamentos sobre a solução planejada.
- Esclarecimentos de dúvidas dos solicitantes: caso o cliente solicite através do ticket informações como o prazo de entrega e informações sobre o andamento da demanda, o analista necessita realizar a comunicação caso as informações disponíveis no ticket não sejam suficientes para esclarecimento do cliente.
- Demonstração da solução desenvolvida: após o atendimento da demanda, o analista deve notificar o solicitante, explicando a solução realizada e como utilizá-la.
- Acompanhamento do time de desenvolvimento
 - Conversão dos tickets em artefatos de desenvolvimento: após analisar um ticket, o analista associar a demanda a história de usuário que será trabalhada pelo time, o que pode ser realizado de duas formas:
 - Criação de uma nova história: se a demanda não é semelhante a nenhuma das demandas existentes, o analista deve criar uma nova história de usuário, que posteriormente será priorizada e alocada no backlog.
 - Associação a uma história existente: Caso uma demanda seja semelhante ou até mesmo igual a uma demanda já existente, verificar se a nova demanda deve ser unida as histórias existentes.
 - Acompanhamento dos projetos em andamento: o analista deve acompanhar junto ao time de desenvolvimento o andamento dos itens sendo desenvolvidos, coletando informações pertinentes (como previsões de entrega, por exemplo) e verificando se não há bloqueios e/ou riscos que possam impactar a entrega prevista.
 - Priorização dos itens criados: o analista deve realizar a priorização dos itens para o time de desenvolvimento, de acordo com a informações repassadas através dos tickets, bem como políticas da empresa.

Para exercer estas atividades, o analista precisa organizar os tickets e os artefatos, bem como estabelecer uma regra de sincronização das informações entre o ticket e suas histórias, seja controlando de forma manual ou com o apoio de ferramentas de controle.

Na Figura 3.1, temos um diagrama demonstrando o ciclo de um Ticket desde a chegada da demanda através do Ticket até a finalização do atendimento do Ticket, com a entrega da funcionalidade desenvolvida:



3.1 Exemplo

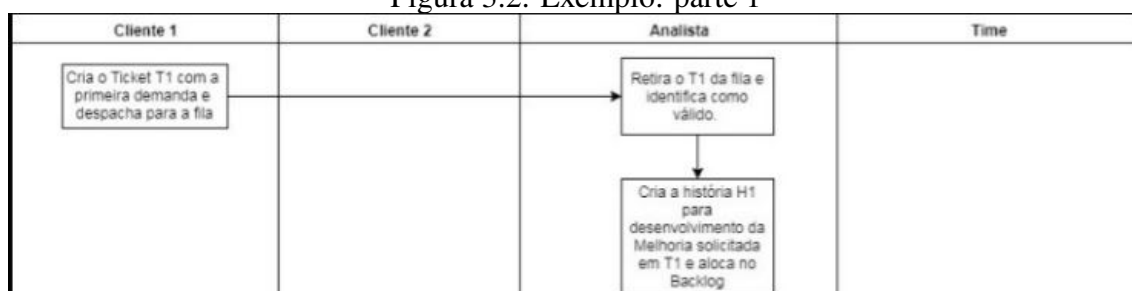
Primeiramente apresentaremos um exemplo de demandas em um sistema fictício, com algumas solicitações de diferentes clientes, que será utilizado ao longo do trabalho. Para este exemplo, vamos utilizar o seguinte cenário inicial: Não haverá backlog anterior já previamente organizado e priorizado. Haverá 1 analista responsável pela triagem das demandas de 2 clientes diferentes.

Na sequência, temos as demandas e como elas serão endereçadas:

- A primeira demanda chega do cliente 1 através do ticket 1, que solicita o desenvolvimento de um novo relatório. Este cliente cria o ticket e o encaminha para a fila de tickets.
- Recebendo este ticket, o analista realiza a análise da demanda e verifica que a demanda é uma melhoria válida para o produto, pois trará benefícios aos clientes permitindo a extração de dados relevantes.
- Então, o analista cria a história H1 e aloca a mesma no Backlog.
- Para acompanhamento, o analista registra no ticket que a demanda está sendo atendida pela história HU1 com seus respectivos status e na história que a mesma atende o ticket T1, através de notas internas.

Na Figura 3.2 temos a demonstração destas ações, onde o cliente direciona o ticket para a fila de tickets, o analista retirou o mesmo para análise e acompanhamento e direcionou a história de usuário para o Backlog.

Figura 3.2: Exemplo: parte 1

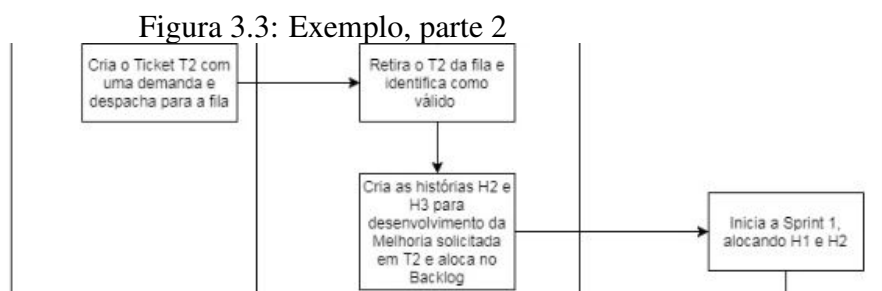


Fonte: Imagem criada pelo autor

- Em seguida, o cliente 2 encaminha outra demanda para a fila através do ticket T2, solicitando um novo cadastro acompanhado de um relatório para exibição dos dados cadastrados.

- Recebendo este ticket, o analista realiza a análise da demanda e verifica que a demanda é uma melhoria válida para o produto, pois permitirá o registro de informações pertinentes que faltam no produto atual.
- Então, o analista cria as histórias H2 e H3 e aloca ambas no Backlog, priorizando H2 em relação a H3, visto que há uma relação de dependência entre as histórias.
- Para acompanhamento, o analista registra no ticket T2 que a demanda está sendo atendida pelas histórias H2 e H3 com seus respectivos status, e nestas mesmas histórias que estas atendem o ticket T2, através de notas internas.
- Neste momento, o time inicia a primeira Sprint, alocando as histórias H1 e H2 para desenvolvimento, visto que são os itens mais prioritários.

Na Figura 3.3, demonstramos estas etapas, onde a segunda demanda é recebida e a sprint iniciada.



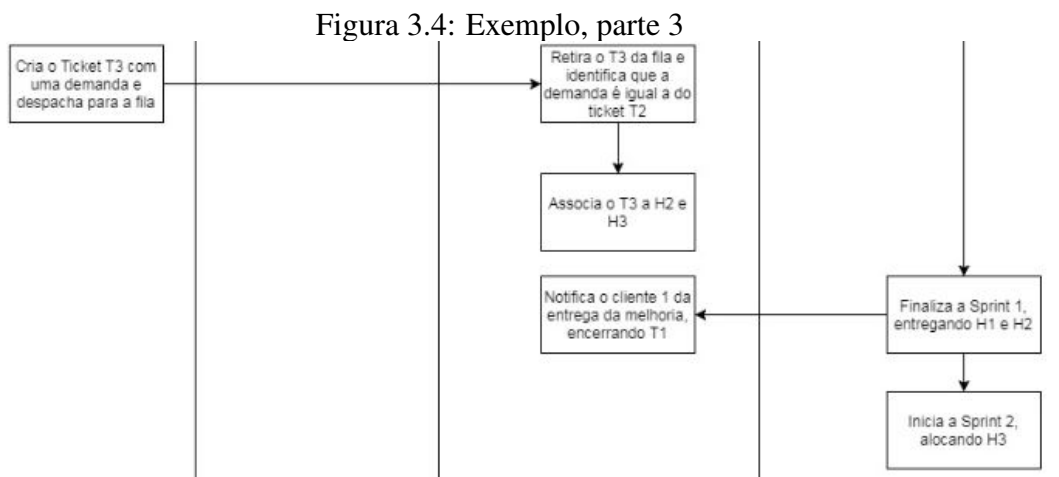
Fonte: Imagem criada pelo autor

- O cliente 1 solicita uma posição quanto ao andamento do desenvolvimento da demanda 1, então o analista deve acessar o ticket 1 para identificar que o mesmo está sendo atendido pela história H1 e verifica que a mesma está em desenvolvimento com previsão de entrega no final da Sprint, encaminhando esta informação ao cliente.
- A terceira demanda chega através do ticket T3 também criado pelo cliente 1, que o analista retira da fila e após análise, identifica como sendo a mesma demanda do cliente 2 registrada no ticket T2.
- Desta forma, o analista associa o ticket T3 às histórias HU2 e HU3, registrando nas histórias que elas também atendem o ticket T3, e no ticket que ele será atendido por estas histórias.
- O time de desenvolvimento fica em dúvida sobre um critério da história H2, então o analista deve contatar o cliente para esclarecimento da dúvida, o que leva a acessar

a história H1 para identificar que a mesma atende o ticket T1 e então enviar a dúvida ao cliente.

- Em seguida, a Sprint do time é finalizada, entregando as histórias HU1 e HU2. A próxima Sprint é iniciada, alocada a história HU3. Como a HU1 foi entregue, o analista deve encerrar o ticket T1, comunicando o cliente da entrega e informando as orientações para utilização da melhoria.

Na Figura 3.4 temos a inclusão das ações, onde temos a chegada da demanda do ticket T3 e a finalização da Sprint.



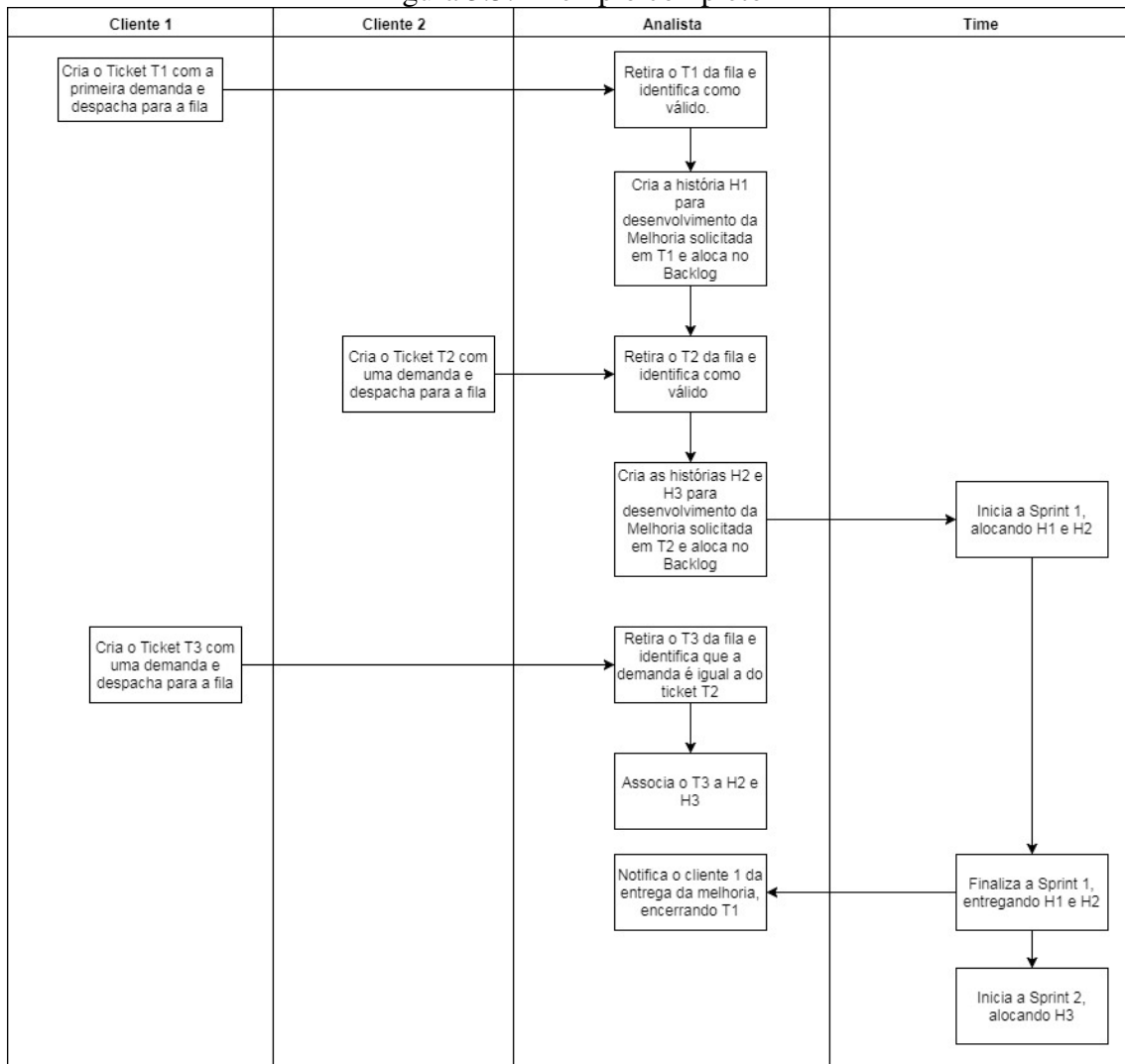
Fonte: Imagem criada pelo autor

Desta forma, temos um cenário de exemplo da integração entre os itens, demonstrando as 3 situações que podem ocorrer entre um ticket e um conjunto de histórias, sendo elas:

- a integração de 1 ticket com 1 história (que chamaremos de integração 1 para 1);
- a integração de 1 ticket com N histórias (que chamaremos de integração 1 para N);
- integração de 1 ticket com histórias pré-existentes.

Na figura 3.5 temos o exemplo completo, demonstrando todo o processo.

Figura 3.5: Exemplo completo



Fonte: Imagem criada pelo autor

Neste exemplo, não temos a utilização de nenhuma estrutura de armazenamento da conversão entre tickets e projetos, o que traz os seguintes problemas:

- Para facilitar a pesquisa, o analista acaba tendo de realizar trabalho duplicado, registrando tanto no ticket quais histórias que estão atendendo-o, quanto nas histórias qual o ticket originário;
- As informações ficam vulneráveis a erros de cadastro, pois se o analista registrar em algum dos lados a informação errada, pode-se acabar prestando uma informação errada ao cliente.
- As informações ficam vulneráveis também à deleção indevida, devido a estarem disponíveis para a edição de qualquer pessoa.

3.2 Critérios Derivados

A partir das atividades e do exemplo exposto derivamos os seguintes critérios para avaliação das ferramentas propostas:

- Conversão de tickets em histórias: após triagem e análise do ticket pelo analista, a ferramenta deve possibilitar o estabelecimento de um processo de conversão dos tickets em histórias. Deve, portanto, disponibilizar uma forma, idealmente automática, de que a partir de um ticket encaminhado pelo cliente, sejam criadas 1 ou mais histórias para desenvolvimento do item. Estas histórias devem poder serem alocadas para um time em específico, de forma que seja possível organizá-las no backlog do time, para posterior alocação nas sprints.
- Associação de um ticket com histórias existentes: uma situação possível é que mais de um solicitante faça a mesma demanda, a qual chegará por tickets diferentes. Para evitar a ocorrência de histórias duplicadas, deve ser possível a associação de novos tickets a histórias existentes.
- Sincronização de Status: a ferramenta deve realizar a sincronização do status das demandas entre os tickets e suas respectivas histórias em função do andamento da resolução da demanda durante o ciclo de desenvolvimento. Para esta sincronização, podemos definir as seguintes regras:
 - Quando a demanda é recebida pelo analista, temos o primeiro ponto, onde ambos os sistemas possuem o status 1 (intitulado "Solicitado" no ticket e "Não definida" na história).
 - Ao passo que a demanda começa a ser analisada, o ticket deve ter seu status alterado para "em andamento" e a história é movida para "em análise", estabelecendo um ponto de sincronização.
 - Nos demais estágios de desenvolvimento, a demanda continua em andamento, e para o solicitante não é uma informação pertinente em qual fase do processo de desenvolvimento o item de fato está. Por este motivo, o ticket permanece como "Em andamento".
 - No momento em que o projeto é liberado, o ticket deve ser encerrado, marcando o atendimento da demanda e a finalização do item, último estágio de sincronização entre os artefatos.
 - Se em qualquer momento do processo a demanda for cancelada (não impor-

tando o motivo deste cancelamento), o ticket deve ser marcado como "cancelado" e a história deve ser encerrada ou, idealmente, excluída, visto que nenhum esforço de desenvolvimento adicional é necessário para o desenvolvimento da solução desta demanda.

- Interligação de informações entre ticket e histórias: a ferramenta deve possibilitar o fácil acesso a informações pertinentes do ticket através da história e vice-versa, agilizando o processo de consulta. Estas informações podem ser os dados de contato do solicitante (que se encontra no ticket) ou o nome da nova funcionalidade que será disponibilizada (informação contida na história), por exemplo.
- Fechamento do ticket: ao concluir a demanda, deve ser possível o encerramento automático do ticket, ou ao menos facilidades que permitam, partindo do fato de que as histórias foram finalizadas e o artefato liberado, seja comunicado a finalização do projeto para o solicitante.

3.3 Integração neste Cenário Utilizando Planilhas

Uma primeira abordagem possível para facilitar o controle da interligação dos artefatos é a utilização de uma tabela em Excel, que deverá contar, pelo menos com as seguintes informações:

- Código do ticket.
- Código das histórias.

A Tabela 3.1 mostra as informações correspondentes. Nesta tabela, temos apenas as informações básicas para, dado um ticket, sabermos quais as histórias relacionadas e vice-versa, não disponibilizando qualquer outra informação diretamente na tabela. Desta forma, a manutenção da tabela é facilitada, reduzindo também o risco de inconsistências. Porém como poucas informações são armazenadas, o analista continua precisando acessar a outra ferramenta não importando qual informação necessita.

Tabela 3.1: Tabela Básica de Integração Ticket X História

Código do Ticket	Código da História
T1	HU1
T2	HU2
T2	HU3
T3	HU2
T3	HU3

Uma forma de endereçar este problema da falta de informações na tabela, facilitando o acesso às informações necessárias para o dia-a-dia, é adicionar na planilha também as informações mais importantes, tais como:

- Cliente solicitante.
- Informação de contato do solicitante.
- Status da demanda.
- Histórias que atendem a demanda.
- Prioridade atual do item.
- Se foi solicitado urgência na demanda.
- Previsão de Entrega.

A Tabela 3.2 ilustra esta segunda opção com duplicação de informações. Desta forma, temos um conjunto básico de informações diretamente na tabela, o que apesar de aumentar o esforço de manutenção, reduz o esforço de consulta dos itens. Isto elimina em diversos momentos a necessidade de consulta das duas ferramentas para o acesso a informação, sob o risco de inconsistência à medida que as informações originais forem alteradas.

Tabela 3.2: Tabela Estendida de Integração Ticket X História

Código do Ticket	Cliente solicitante	Contato do cliente	Status da História	Código da História	Urgente	Previsão
T1	Cliente 1	b@gmail.com	Finalizada	HU1	Não	Finalizado
T2	Cliente 2	c@gmail.com	Finalizada	HU2, HU3	Não	Finalizado
T3	Cliente 1	b@gmail.com	Finalizada	HU2, HU3	Não	Finalizado

Assim, podemos concluir que esta alternativa não atende nenhum dos critérios expostos, visto que o analista permanece tendo de realizar a manutenção de todas as informações entre os elementos e a planilha. Em consequência, permanece o risco de dessincronização de informações e a necessidade de trabalho manual de registro. Todavia, esta abordagem reduz a necessidade de registro de informações nos tickets e histórias,

visto que a informação passa a ficar em um lugar (a planilha) ao invés de 2 (ticket e história).

4 AVALIAÇÃO DAS ABORDAGENS

Dado o exemplo demonstrado na Seção 3.1 e os critérios apresentados na Seção 3.2, será realizada a análise das 4 abordagens citadas anteriormente, primeiramente buscando simular os cenários apresentados em cada uma das ferramentas, utilizando as ferramentas na forma gratuita, ou em caso de ferramentas que não dispõem de uma modalidade gratuita, utilizando o período de testes disponibilizado.

Após a simulação das situações na ferramentas, então será realizada a avaliação conforme os critérios estabelecidos, utilizando uma escala de 3 níveis (Atende o critério, Atende parcialmente o critério e Não atende o critério). Com base na avaliação de cada ferramenta, por último, as avaliações serão consolidadas para comparação.

4.1 Integração dos artefatos na Ferramenta Trello

Dado que o Trello permite a utilização do controle de tickets e de projetos ágeis por meio de quadros separados, então temos o problema de integração das informações entre estes dois quadros. Nativamente, o Trello não oferece esta funcionalidade de integração, e desta forma, é necessário que esta funcionalidade seja suprida de outra maneira, como por exemplo com o uso de extensões ou de convenções adicionais nos quadros.

4.1.1 Soluções por meio de extensões

Ao utilizar extensões para o Trello, passamos a ter a funcionalidade de integração entre os boards, com cada extensão propondo um comportamento específico. Uma desvantagem desta abordagem é que a maioria das extensões são pagas, eliminando a vantagem do controle gratuito disponibilizado pelo Trello. Exemplos dessas extensões são o Unito e o Placker, que serão analisadas a seguir.

4.1.1.1 Unito

Esta ferramenta possibilita a sincronização de cartões em diferentes quadros, realizando automaticamente a troca de informações entre os campos dos cartões e mantendo a consistência, caso a sincronização automática esteja habilitada.

Nesta ferramenta, é possível informar quais campos nos cartões devem ser sincro-

nizados, bem como a correspondência entre as colunas dos quadros. Isto permite que as alterações nos status do cartão da história sejam refletidas no ticket de forma adequada e vice-versa.

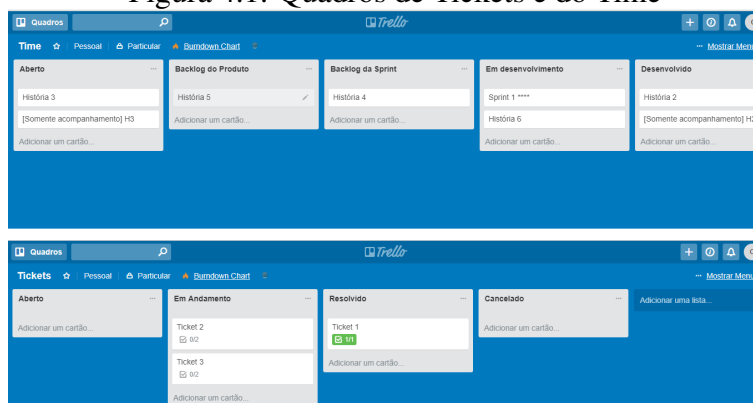
Além de permitir que as informações pertinentes sejam sincronizadas entre os quadros, possibilita que a privacidade dos dados que não precisam ser compartilhados com o outro grupo seja mantida. O ponto fraco desta abordagem é que a sincronização ocorre apenas entre pares de tickets iguais, ou seja, para cada história que tivermos no quadro de histórias, deve ser criada uma cópia do cartão que representa o ticket, mostrando ao cliente esta divisão e ficando com várias cópias do mesmo ticket no quadro de tickets. Neste caso, o cliente deverá ser informado que foram criadas estas cópias e que o ticket só estará resolvido quando a última cópia do ticket for movida para resolvido.

Na Seção 4.1.1.2 iremos demonstrar a aplicação do Exemplo demonstrado na Seção 3 nesta abordagem, para posteriormente realizar a avaliação da abordagem na Seção 4.1.1.3.

4.1.1.2 Exemplo com o Unito

Para a avaliação desta abordagem vamos aplicar o exemplo montado no Capítulo 3, executando cada uma das etapas na ferramenta. Para esta demonstração do exemplo, teremos dois quadros, o primeiro quadro chamaremos de "Quadro de Tickets" que servirá como sistema de tickets, onde o cliente criará cartões que representarão cada um dos tickets. O segundo quadro servirá para o acompanhamento do andamento das histórias de usuário, que será chamado de "Quadro do Time", visto que é utilizado para o acompanhamento das atividades de desenvolvimento e da alocação dos itens nos Backlogs de Produto e Sprint. Na Figura 4.1 mostramos um exemplo destes dois quadros, que serão integrados pela ferramenta Unito.

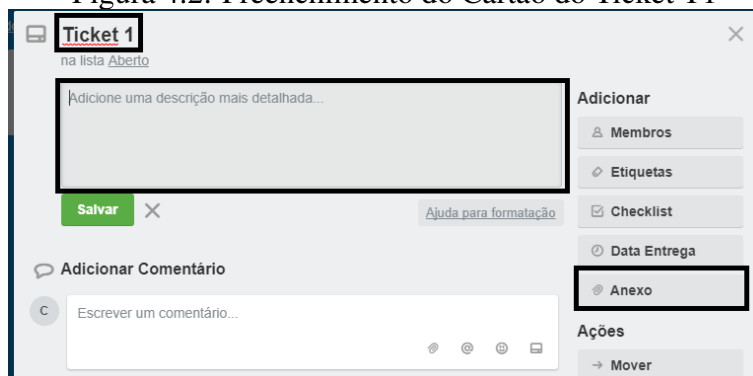
Figura 4.1: Quadros de Tickets e do Time



Fonte: Imagem criada pelo autor

Primeiramente o cliente deve criar o ticket no quadro de tickets, tarefa que pode ser realizada pela inclusão de um cartão na lista de tickets em aberto, descrevendo a demanda no campo de descrição. Na Figura 4.2 temos a demonstração do Ticket a ser criado, destacando os campos que o cliente irá preencher para explicar sua demanda.

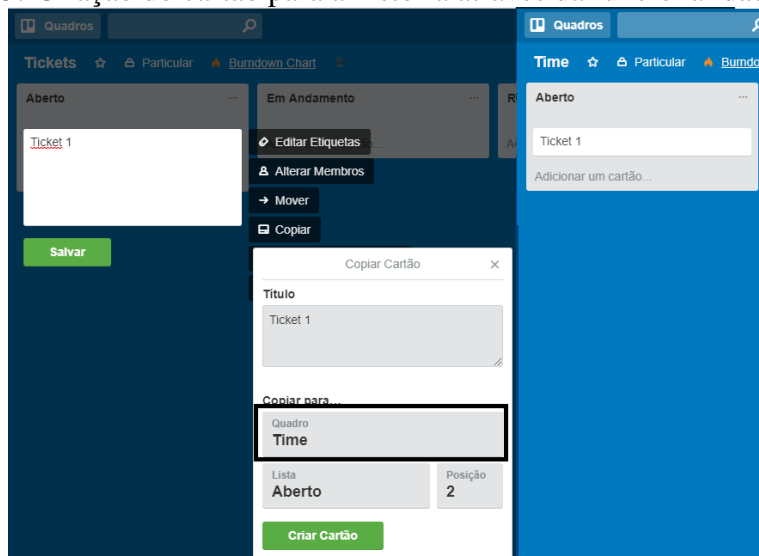
Figura 4.2: Preenchimento do Cartão do Ticket T1



Fonte: Imagem criada pelo autor

Após a análise inicial, o analista cria a história H1 no quadro do Time, na lista que representa os itens Abertos e movendo para a lista que representa o *Backlog* do Produto após a priorização do item. Para isso, o analista pode utilizar a funcionalidade de cópia do cartão, já movendo para o segundo quadro. Na Figura 4.3 podemos visualizar este processo, destacando as funcionalidades que devem ser utilizadas para a criação do novo cartão e movimentação dele para o quadro do Time, na coluna Aberto.

Figura 4.3: Criação do cartão para a história através da funcionalidade de Cópia

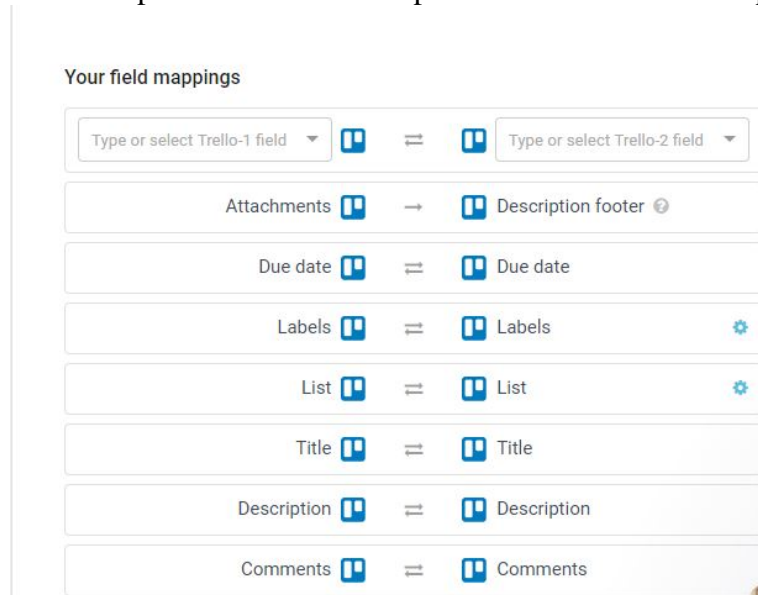


Fonte: Imagem criada pelo autor

Para o correto funcionamento do Unito para a sincronização das informações e status, precisamos configurar quais campos serão sincronizados, isto pode ser realizado pela tela de configuração do Unito. Nas Figuras 4.4 e 4.5 demonstramos como essa configuração pode ser realizada, primeiramente mostrando a seleção de quais campos serão sincronizados e depois a configuração da correspondência entre os status, que é realizada pela associação das listas entre os dois quadros.

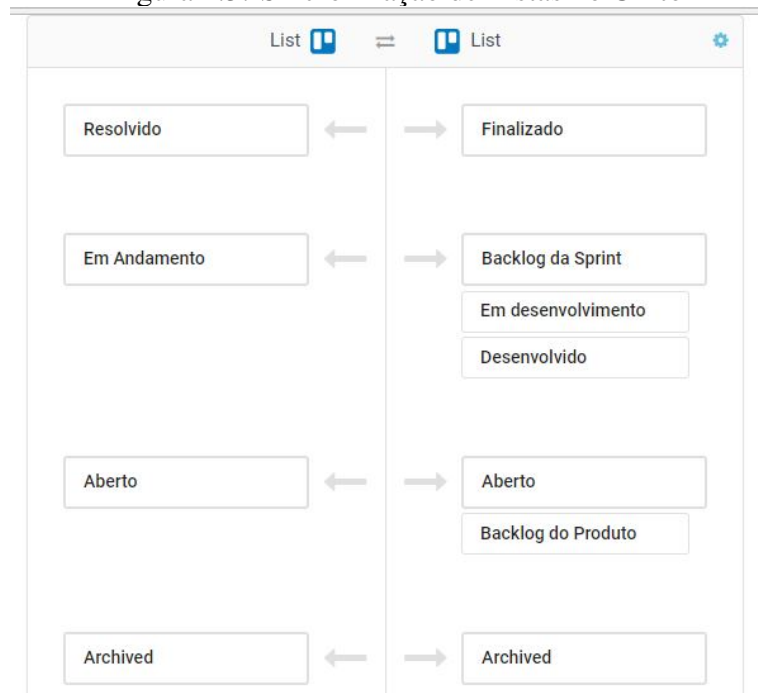
Nesta configuração, o Unito utiliza as setas para indicar o sentido da sincronização, então quando há apenas a seta em um sentido, isto descreve que a sincronização é apenas naquele sentido, como é mostrado na figura, por exemplo, pela ligação entre o campo "Anexos" do cartão que representa o ticket e o campo "Rodapé da Descrição" do cartão que representa a história, onde apenas atualizações no ticket serão refletidas na história, mas a atualização no sentido contrário não ocorrerá. Já a utilização das setas nos dois sentidos, representa que toda atualização realizada em um cartão será refletida no outro cartão, em ambos sentidos. Para o correto funcionamento desta sincronização, o nome dos cartões nos dois quadros precisa ser o mesmo, o que consiste em uma desvantagem para a utilização desta abordagem.

Figura 4.4: Correspondências entre campos a serem sincronizados pelo Unito



Fonte: Imagem criada pelo autor

Figura 4.5: Sincronização de Listas no Unito



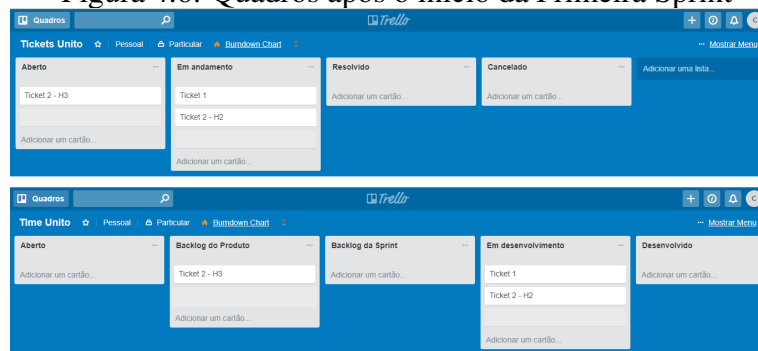
Fonte: Imagem criada pelo autor

Em seguida, o cliente 2 encaminha a segunda demanda criando o segundo ticket, que como descrito no exemplo, seria atendido em duas histórias, o que não é atendido por esta ferramenta, visto que a mesma realiza a sincronização entre dois cartões. Desta forma, para continuar a demonstração do exemplo, teremos de adotar uma convenção. Escolheremos a convenção de criar cópias do cartão do ticket para associar aos cartões

que representam as histórias. Assim, será necessário a criação de uma cópia do Ticket 2 para utilizar na ligação com o cartão da história H3.

O time aloca as histórias H1 e H2 para a Sprint, enquanto a história H3 permanece no Backlog do Produto, o que é registrado movendo os cartões das histórias H2 e H3 para a lista em desenvolvimento. Na Figura 4.6, demonstramos como ficam os quadros após estas etapas.

Figura 4.6: Quadros após o início da Primeira Sprint



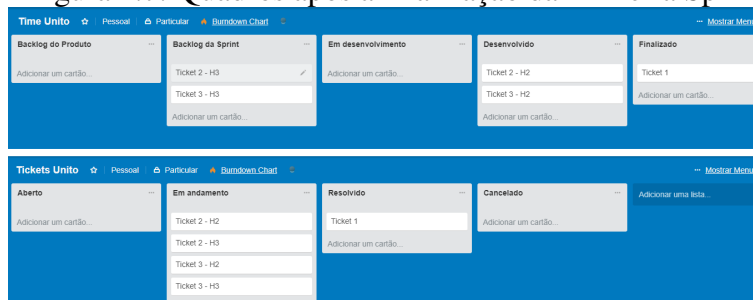
Fonte: Imagem criada pelo autor

A terceira demanda recebida no exemplo é uma demanda igual a demanda recebida no segundo ticket, o que levaria para a associação de um ticket a outra(s) história(s), o que também não é atendido pela ferramenta. Desta forma, para continuar a demonstração do exemplo, iremos adotar a seguinte convenção: será mantido um cartão para o ticket 3 em ambos os quadros, que será movido em conjunto com o cartão que corresponde ao ticket 2, controle que terá de ser feito pelo analista, criando as histórias para acompanhamento e realizando a movimentação das mesmas quando as histórias originais forem movimentadas pelo Time.

Para atender as comunicações descritas no exemplo, a ferramenta disponibiliza a funcionalidade do campo de Comentários e com a sincronização ligada, no momento que o cliente envia a sua dúvida no ticket, a mesma é demonstrada na história facilitando a comunicação entre cliente, analista e time. Este funcionamento também se repete quando uma dúvida do time é registrada no cartão da história, já sendo sincronizada para o cartão do ticket.

Por último, temos a finalização da primeira *sprint*, com a entrega da demanda do ticket 1 e da primeira parte da funcionalidade que atende os tickets 2 e 3. Assim, os tickets 2 e 3 permanecem em andamento, visto que H3 ainda não foi desenvolvida. Na Figura 4.7 temos os quadros após estas etapas, demonstrando o estado final dos quadros no exemplo.

Figura 4.7: Quadros após a finalização da Primeira Sprint



Fonte: Imagem criada pelo autor

Após a demonstração do exemplo nesta abordagem, iremos realizar a avaliação da abordagem frente aos critérios descritos na Seção 3.2.

4.1.1.3 Avaliação seguindo os critérios derivados - Trello com Unito

Esta abordagem atende alguns dos critérios, conforme a seguir:

- **Conversão de tickets em histórias:** esta abordagem atende parcialmente este critério, visto que possibilita a utilização de uma regra de criação das histórias a partir dos tickets e forma semi-automatizada por meio da funcionalidade de cópia, mas somente se há uma relação de um para um entre ticket e história de usuário. Caso haja a necessidade de criar mais de uma história de usuário, também é necessária a criação de cópias dos cartões de tickets conforme as histórias que serão utilizadas.
- **Sincronização de status:** a sincronização dos status entre os cartões é viável e permite inclusive a utilização de status diferentes entre os quadros, como esperado. Novamente, este critério só atendido para situações de 1 para 1. Para casos em que seja necessário mais de uma história, se feita a quebra do ticket também, a sincronização é mantida entre os pares de cartões, mas não há sincronização entre os grupos de cartões.
- **Interligação de informações entre ticket e história:** ao manter todas as informações desejadas em sincronização entre os dois cartões, esta abordagem permite que as informações sejam interligadas e elimina a necessidade de acesso dos dois cartões para obtenção das informações desejadas. Com a funcionalidade de comentários, podemos também interligar as comunicações realizadas entre o analista e o cliente, permitindo inclusive que o time tenha acesso a estas informações diretamente. Neste caso, também temos a limitação da relação 1 para 1.
- **Fechamento do ticket:** como o fechamento do ticket é registrado pela movimen-

tação do mesmo para o status de "Finalizado" no quadro e a ferramenta possibilita a atualização automática deste status, bem como de comentários necessários para a utilização da solução desenvolvida, este critério é atendido. Todavia, caso seja necessária a quebra da demanda em diversos cartões, não há forma para controle da conclusão da demanda como um todo, apenas do encerramento de cada uma das partes.

- **Associação do ticket com histórias existentes:** Esta abordagem não atende este critério, visto que dado que precisamos de cartões para todos os tickets em ambos os quadros, não podemos integrar um cartão de ticket em um cartão de história.

4.1.1.4 Placker

O Placker¹ é outra extensão disponível para o Trello, que se propõe a oferecer diversas funcionalidades que facilitem o gerenciamento de times utilizando os quadros e demais gráficos que a ferramenta disponibiliza. Ela também complementa as informações que podem ser colocadas em cada quadro, adicionando, por exemplo, campos para registrar o tempo utilizado no desenvolvimento do item.

Para utilizar esta ferramenta, deve-se primeiramente, instalar o *Power-Up* do Placker no Trello, de forma que a ferramenta possa acessar as informações dos quadros configurados. Para esta análise, vinculamos tanto o quadro dos tickets quanto o quadro utilizado para registro das histórias de usuário no time.

O Placker é uma ferramenta paga, e disponibiliza, além de sua integração com o Trello, uma interface própria, que é sincronizada de forma automática com o Trello, permitindo assim o usuário trabalhar em qualquer uma das duas instâncias.

O Placker utiliza realiza a integração entre os quadros de forma diferente em relação ao Unito, a transformação do Ticket em histórias com base nos *Checklists* (funcionalidade padrão do Trello) e na funcionalidade do Placker de criar novos cartões a partir destes *Checklists* (funcionalidade não presente no Unito). Desta forma, o analista poderá criar os itens no *Checklist* do cartão do Ticket e "arrastar" estes itens para o quadro do Time, onde a ferramenta criará novos cartões representando estas histórias e criando o relacionamento entre o item e o cartão (representado através das anotações "Espelha"). Nesta ligação, os campos não são sincronizados entre os cartões, permitindo apenas o acesso do Ticket a partir da história.

Na Seção 4.1.1.5 temos a demonstração da aplicação do exemplo nesta aborda-

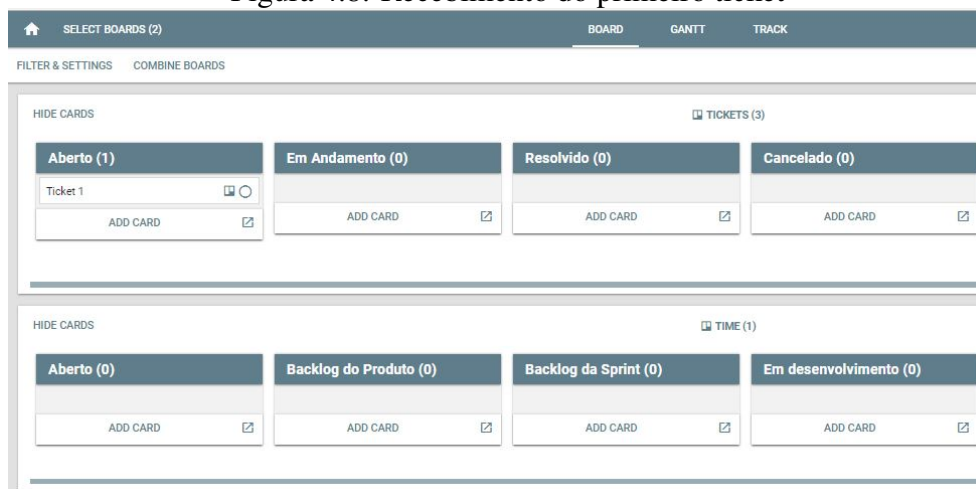
¹<http://placker.com>

gem, para depois realizar a avaliação desta abordagem na Seção 4.1.1.6.

4.1.1.5 Exemplo - Trello com Placker

Como esta abordagem também é baseada no Trello, a criação dos tickets pelos clientes pela utilização de cartões no quadro que representa o sistema de tickets é igual aos passos demonstrados no início da Seção 4.1.1.2, com o cliente criando o cartão e alocando na lista de "Abertos". Como o Placker possui uma interface própria, que permite visualizar os dois quadros que utilizamos neste trabalho de uma só vez, o analista pode utilizar esta interface para realizar suas atividades, eliminando a necessidade de mudança entre os quadros. Na Figura 4.8 temos a demonstração da primeira etapa do exemplo, com a chegada do primeiro chamado.

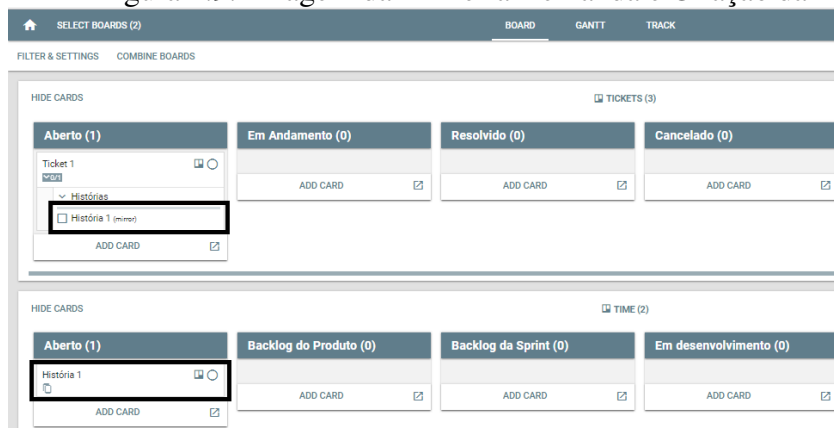
Figura 4.8: Recebimento do primeiro ticket



Fonte: Imagem criada pelo autor

O analista então deve realizar a criação dos cartões que representarão as histórias, que deve ser realizado através das funcionalidades de *Checklists* e da movimentação destes *Checklists* para o quadro utilizado para acompanhamento do desenvolvimento. Na Figura 4.9 temos o resultado destas ações, com o cartão original ficando com o *Checklist* e o novo cartão tendo o nome da história.

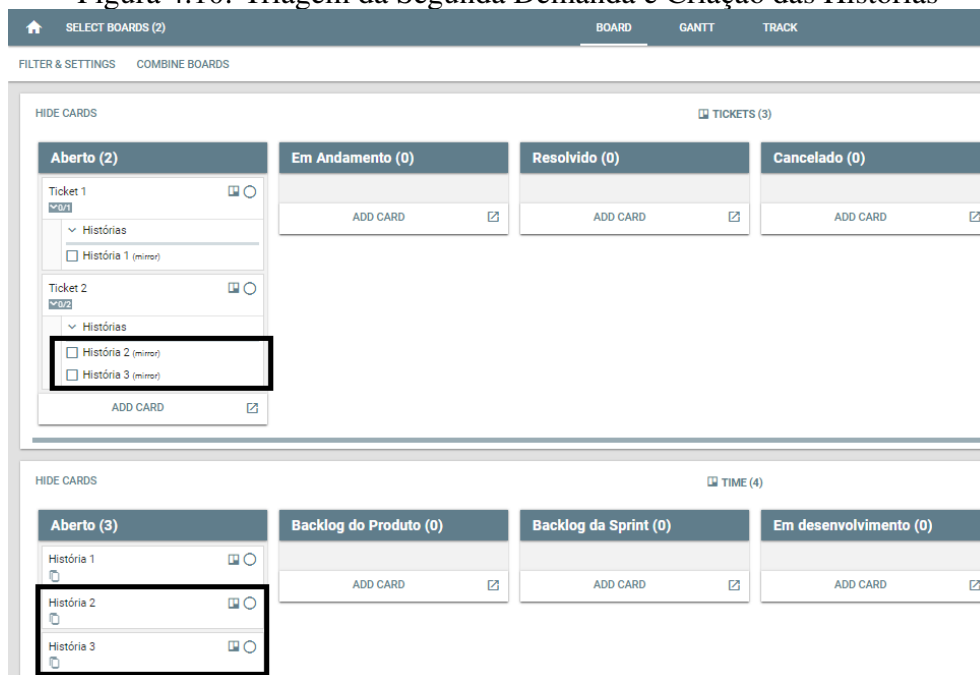
Figura 4.9: Triagem da Primeira Demanda e Criação da História



Fonte: Imagem criada pelo autor

Depois é recebida a segunda demanda, através do cartão que representa o ticket 2. Para este, o analista irá então criar duas histórias, repetindo o procedimento realizado para a primeira demanda, ou seja, criando um item de *Checklist* para cada história no cartão original e movendo estes itens para o quadro de acompanhamento do desenvolvimento, criando os cartões que representarão as histórias. A Figura 4.10 demonstra esta etapa do exemplo, com os novos cartões criados nos dois quadros.

Figura 4.10: Triagem da Segunda Demanda e Criação das Histórias

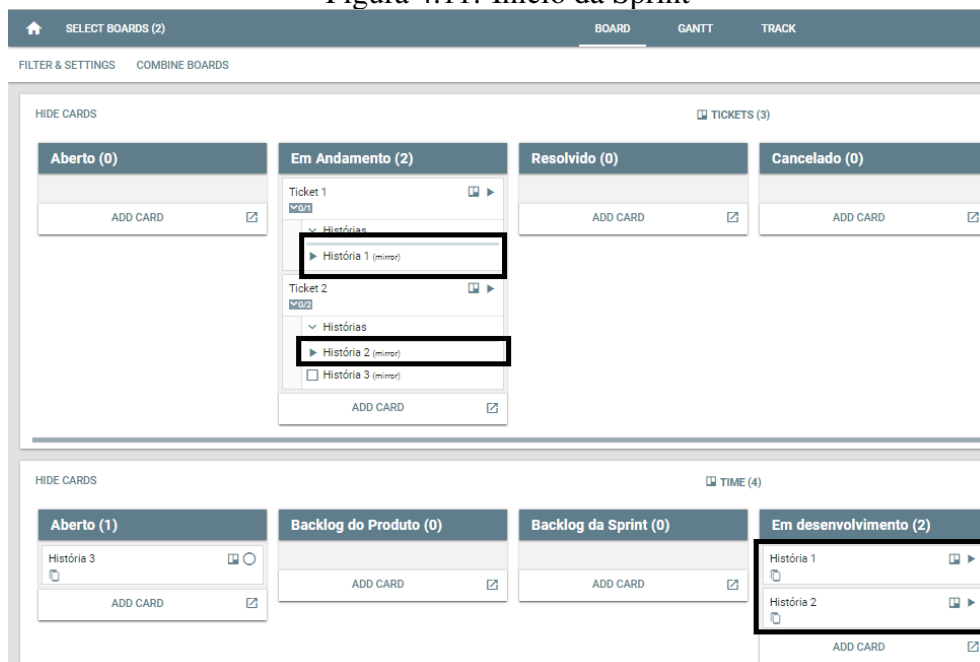


Fonte: Imagem criada pelo autor

A próxima etapa do exemplo é o início da primeira *sprint*, com a alocação das histórias na *sprint*, que é demonstrado na Figura 4.11. Como a ferramenta não realiza a

alteração do Status dos Tickets de forma automática com base na alteração das histórias, precisamos realizar a atualização dos tickets de forma manual, com base nos Status dos itens do *Checklist* associado a cada cartão.

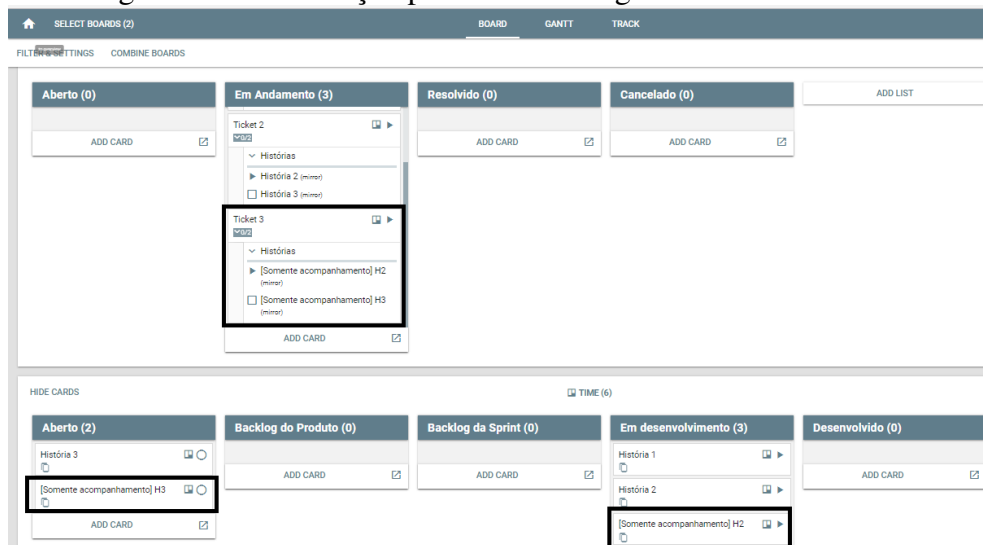
Figura 4.11: Início da Sprint



Fonte: Imagem criada pelo autor

Na chegada da terceira demanda, que é igual à segunda demanda, temos uma limitação desta abordagem, visto que não é possível associar um cartão a outros existentes anteriormente. Desta forma, para continuar a demonstração do exemplo, iremos adotar a convenção de criar novas histórias apenas para registro, identificando como "Somente acompanhamento", que irão avançar em conjunto com as histórias originais, por meio da movimentação manual realizada pelo analista. Na Figura 4.12 temos a demonstração desta convenção, com as novas histórias.

Figura 4.12: Convenção para demandas iguais a outras existentes

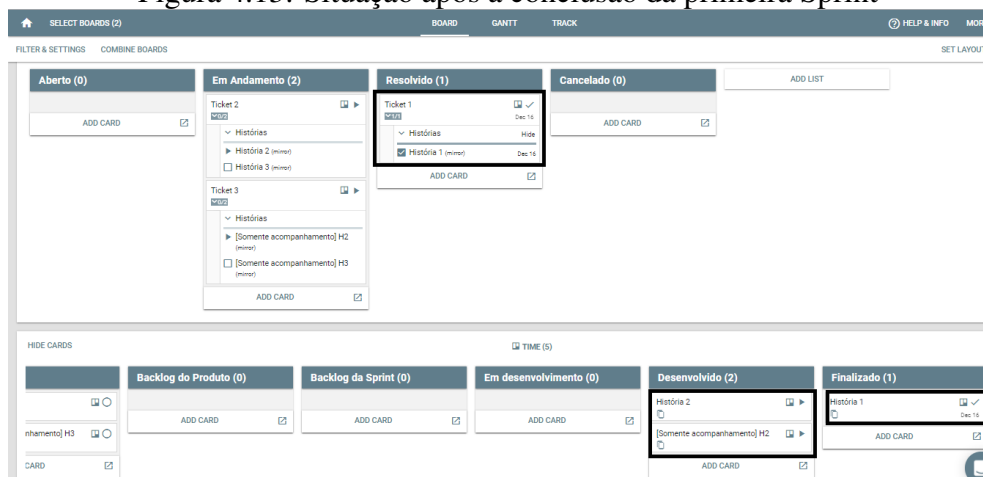


Fonte: Imagem criada pelo autor

Quando o cliente solicita o posicionamento da data de entrega, será realizado de forma semelhante ao Trello com o Unito, com o cliente registrando o questionamento como um Comentário no cartão do Ticket. A diferença é que este questionamento não é refletido na história, necessitando a ação do analista de acessar cada um dos cartões das histórias para obter a informação necessária, por meio das ligações estabelecidas. No questionamento do time, o processo também é manual, com o analista identificando o Ticket em que deve ser enviado o questionamento, pois não há a ligação neste sentido na ferramenta.

Por último, temos a finalização da Sprint, com a entrega das histórias H1 e H2, e o encerramento do ticket 1, que também deve ser manual, visto que a ferramenta não disponibiliza a funcionalidade de movimentação automática do cartão com base na conclusão do *Checklist*. O analista também deve atualizar a história criada para acompanhamento, tarefa que não é auxiliada pela ferramenta. Na Figura 4.13 demonstramos o resultado desta etapa, com o ticket 1 sendo movido para "Resolvido", enquanto os tickets 2 e 3 continuam em andamento, visto que a história H3 não foi ainda desenvolvida.

Figura 4.13: Situação após a conclusão da primeira Sprint



Fonte: Imagem criada pelo autor

Concluída a demonstração do exemplo nesta abordagem, vamos para a avaliação desta abordagem seguindo os critérios definidos na Seção 3.2.

4.1.1.6 Avaliação seguindo os critérios derivados - Trello com Placker

Com base no cenário exposto acima, podemos concluir que com a adoção da extensão Placker, o Trello atende alguns dos critérios definidos, conforme detalhado a seguir.

- **Conversão de tickets em histórias:** esta abordagem atende a este critério, pois possibilita a criação das histórias de forma automática, de acordo com os itens do *checklist* definidos no ticket, que podem ser um ou mais itens.
- **Sincronização de status:** esta abordagem não atende este critério, visto que apesar de facilitar o controle dos itens através da conclusão dos itens no *checklist*, não há a possibilidade de sincronização do avanço do status entre os cartões entre os dois quadros.
- **Interligação de informações entre ticket e história:** esta abordagem atende parcialmente este critério, uma vez que é possível acessar as informações do ticket a partir da história, mas não é possível o caminho de volta, que seria acessar as histórias a partir do ticket.
- **Fechamento do ticket:** esta abordagem não atende este critério, já que a ferramenta não encerra automaticamente o ticket (mover para a lista de resolvido) na conclusão de todos os itens do *checklist*.
- **Associação do ticket com histórias existentes:** esta abordagem não atende este

critério, visto que não possibilita a associação de outros tickets a histórias já existentes.

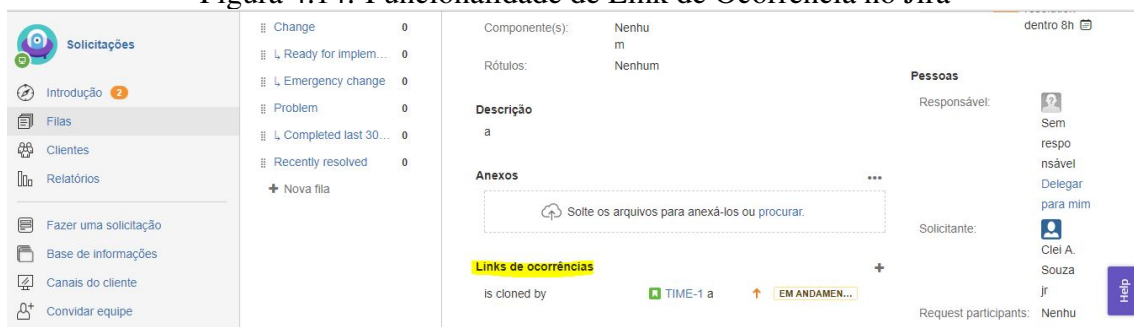
4.2 Integração dos artefatos na Ferramenta JIRA

Como são diferentes módulos da ferramenta, os tickets e histórias de usuário devem ser mantidas em seus respectivos módulos, necessitando assim da comunicação entre os mesmos.

Como demonstrado na Seção 2.8.1, ao receber um ticket vindo do Service Desk, o analista irá analisar a demanda e após esta análise realizar a criação da história de usuário no módulo de Scrum, conforme demonstrado na Seção 2.8.2.

Para esta transformação do Ticket em história, a ferramenta não disponibiliza nenhum auxílio. Depois de criado dos artefatos criados manualmente, precisamos ligá-los, de forma a permitir a consulta de informações entre os mesmos, por meio da utilização da funcionalidade de Link de Ocorrências. Esta funcionalidade realiza a ligação entre dois artefatos, indicando qual a a relação entre os mesmos, com esta conexão estabelecida, é possível acessar um artefato a partir do outro. Esta funcionalidade é demonstrada pela Figura 4.14

Figura 4.14: Funcionalidade de Link de Ocorrência no Jira



Fonte: Imagem criada pelo autor

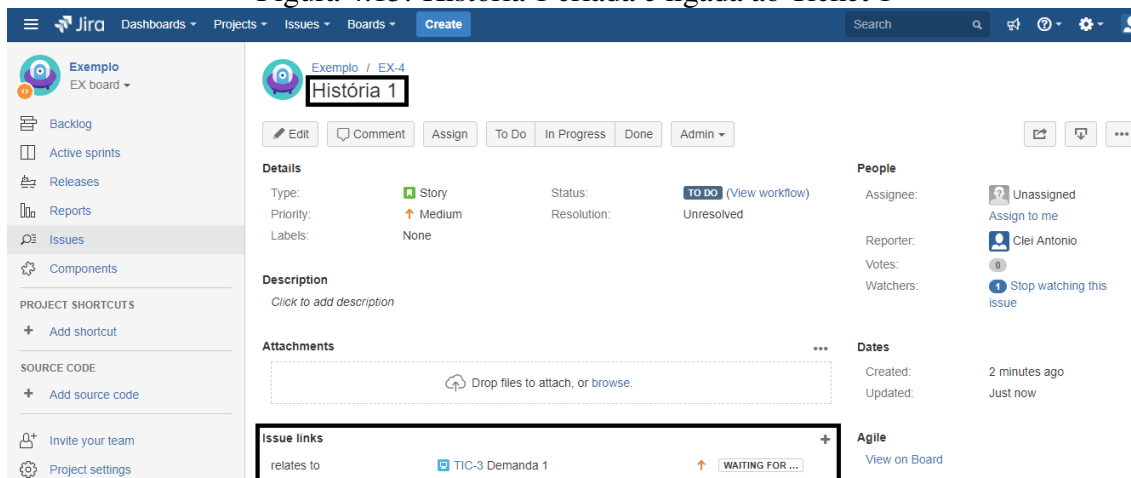
Na Seção 4.2.1 faremos uso desta funcionalidade, visto que aplicaremos o exemplo descrito na Seção 3 nesta abordagem.

4.2.1 Aplicação do Exemplo no JIRA

Primeiramente, o cliente realiza a abertura do ticket 1 para registrar a sua primeira demanda. Como demonstrado na Seção 2.8.1, esta criação deve ser realizada através da interface de abertura de tickets do Service Desk.

Depois da análise, o analista deverá criar a história H1, ligada ao Ticket T1. Para tal, o analista deverá acessar o Projeto de desenvolvimento e realizar a criação da história de forma independente, ligando depois a História ao Ticket com o uso do Link de Ocorrência. Na Figura 4.15 temos o resultado desta etapa, com a história criada e ligada ao ticket. Um ponto de atenção nesta forma é que caso haja o esquecimento de gravar a ligação ou a ligação for feita com o ticket errado, pode-se ficar com inconsistências que se não identificadas rapidamente, podem causar transtornos e desencontro de informações.

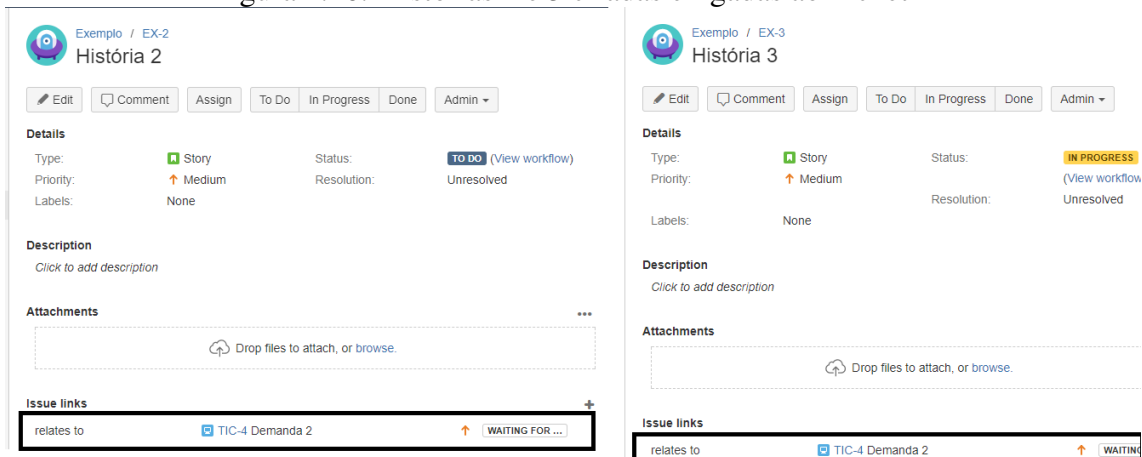
Figura 4.15: História 1 criada e ligada ao Ticket 1



Fonte: Imagem criada pelo autor

Depois, é recebida a segunda demanda, vinda do cliente 2, que será por sua vez ligada a duas histórias. Para isto, deve-se repetir o processo feito para a primeira demanda para cada uma das histórias, criando-as no Projeto de desenvolvimento e ligando-as ao ticket 2. Na Figura 4.16 são mostradas as histórias lado-a-lado, com a ligação para o ticket.

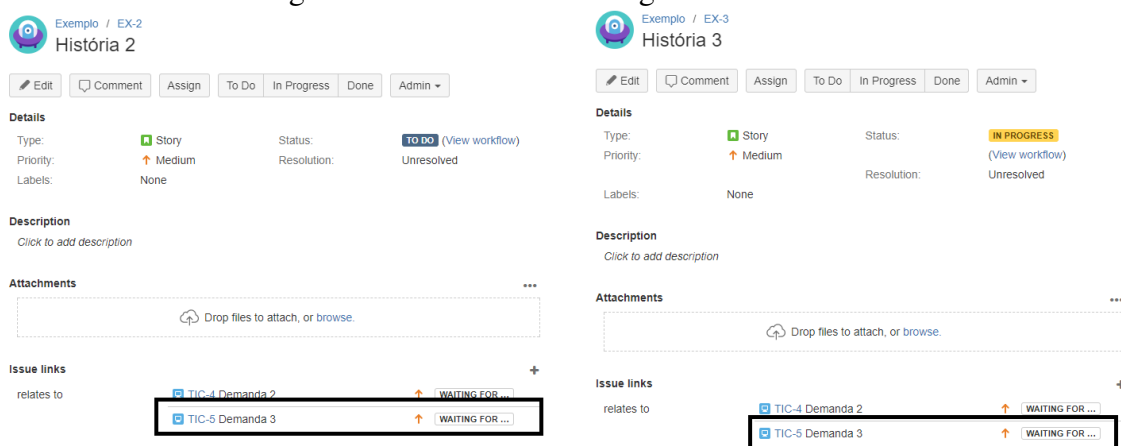
Figura 4.16: Histórias 2 e 3 criadas e ligadas ao Ticket 2



Fonte: Imagem criada pelo autor

Com o início da Sprint, as histórias são iniciadas, sendo que esta atualização de status que não é refletida no ticket. Para este fim, o analista deve acessar os tickets T1 e T2 e alterar o seu status para "Em andamento". Na chegada da terceira demanda através do ticket T3, o analista deve realizar a ligação das histórias H2 e H3 a este novo ticket, visto que a demanda é igual. Na Figura 4.17, é demonstrado o resultado desta ligação, com as histórias agora estando ligadas a dois tickets diferentes.

Figura 4.17: Histórias 2 e 3 ligadas ao Ticket 3



Fonte: Imagem criada pelo autor

Quando o cliente realiza o questionamento sobre a data de entrega do item através do Ticket, o analista deverá acessar o mesmo para verificar qual é o questionamento e irá acessar as histórias relacionadas por meio do uso do link de ocorrência, de forma a ver o Status das histórias e a sua previsão de entrega. No questionamento do time, o analista

acessar a história e por sua vez utilizará o link de ocorrência na história para acessar o Ticket e encaminhar o questionamento, copiando a resposta manualmente para a história se necessário.

Ao final da *sprint* são entregues as histórias H1 e H2, que são marcadas como concluídas. Porém esta alteração de status também não é refletida no ticket, que portanto também não são encerrados de forma automática. Cabe ao analista verificar os tickets que estão ligados a cada uma das histórias e julgar se os mesmos devem ser encerrados, efetuando este encerramento de forma manual.

Após apresentado o exemplo nesta abordagem, faremos a avaliação da mesma conforme os critérios descritos na Seção 3.2.

4.2.2 Avaliação seguindo os critérios derivados

- **Conversão de tickets em histórias:** a ferramenta não atende este critério, visto que não automatiza o processo de conversão de tickets em histórias de usuário, nem oferece qualquer tipo de auxílio. Cabe ao analista criar as histórias de usuário e depois vinculá-las corretamente.
- **Sincronização de Status:** a ferramenta não atende este critério, visto que à medida que as histórias progridem, não há a atualização do status do ticket para registrar o andamento/finalização da demanda, já que requer a ação manual do analista no ticket.
- **Interligação de informações entre ticket e histórias:** a ferramenta atende a este critério, uma vez que através dos links, o analista consegue navegar entre os elementos e buscar as informações necessárias em cada elemento.
- **Fechamento do ticket:** a ferramenta não atende a este critério, visto que não há como o analista saber se todas as histórias do ticket foram atendidas sem consultar cada ticket e por consequência, efetuar manualmente o encerramento dos itens que estão finalizados.
- **Associação de ticket com histórias existentes:** a ferramenta atende este critério, visto que ao receber uma demanda que já esteja sendo endereçada por uma história existente, basta realizar o link do ticket para a história.

4.3 Integração dos artefatos na Ferramenta Trac

Como descrito na Seção 2.6, o Trac não possui suporte nativo para o acompanhamento do desenvolvimento, precisando da utilização de uma extensão para tal tarefa. Para este trabalho, selecionamos a extensão Agilo for Trac² que disponibiliza as funcionalidades faltantes para atendimento do Scrum, tais como Backlog de Produto e Backlog de Sprint, criação/acompanhamento de Sprints e registro das histórias de usuário com base nos tickets criados.

Como esta abordagem é construída utilizando com base em uma ferramenta de tickets, todos os artefatos (tickets e histórias) são construídos utilizando tickets, estabelecendo a diferença entre eles através dos tipos estabelecidos pela extensão, que são requisitos e histórias. Os tickets criados no Trac são sincronizados para o Agilo como requisitos, que também podem ser criados diretamente via Agilo se o usuário tiver permissão para tal tarefa.

Na Seção 4.3.1 iremos demonstrar a aplicação do Exemplo demonstrado na Seção 3 nesta abordagem, para posteriormente realizar a avaliação da abordagem na Seção 4.3.2.

4.3.1 Aplicação do Exemplo no Trac com Agilo

Primeiramente, o cliente 1 encaminha a primeira demanda através do Ticket 1, que nesta abordagem, pode ser encaminhado pela criação do ticket no Trac, que será sincronizado para o Agilo.

O analista então deve realizar a triagem da demanda e a criação da História 1. Esta ação pode ser realizada pela própria visão do ticket que o analista utiliza, através da funcionalidade "Criar uma nova história referenciada", esta ação resulta na criação da história de usuário e automaticamente associa a mesma ao ticket que a originou. Na Figura 4.18 temos a demonstração da visão do ticket do analista, destacando esta funcionalidade. Na Figura 4.19, temos o resultado desta ação mostrando o Backlog do Produto com a História 1, que está "abaixo" do Ticket 1, demonstrando esta conexão entre os dois elementos.

²<http://www.agilofortrac.com>

Figura 4.18: Funcionalidade de Criação de História Associada

Requirement #12 (assigned)

Opened 34 minutes ago
Last modified 33 minutes ago

Ticket 1

Reported by:	Cliente 1	Owned by:	admin
Milestone:	n.a.	Keywords:	n.a.
Business Value Points:	n.a.		
Total Story Points:	n.a.	Roif:	n.a.
Mandatory Story Points:	n.a.		

Description

References

Reference: #

Associate New: Create a new referenced 'User Story'

Fonte: Imagem criada pelo autor

Figura 4.19: História criada já no Backlog de Produto

Product Backlog

ID	Summary	Business V...	Roif	Importance	User Story ...
12	Ticket 1				
14	História 1				
2	Totals				

Fonte: Imagem criada pelo autor

Em seguida, o cliente 2 encaminha a segunda demanda, através do ticket 2, para a qual serão criadas as histórias H2 e H3, tarefa que pode ser realizada de forma semelhante a primeira etapa, repetindo o processo para cada história que precisa ser criada. Na Figura 4.20 temos o resultado destas ações, com o Ticket 2 ficando com estas duas histórias associadas.

Figura 4.20: Ticket 2 com as histórias associadas

Opened 50 seconds ago

Ticket 2

Reported by: [Cliente 2](#) Owned by:

Milestone: n.a. Keywords: n.a.

Business Value Points: n.a.

Total Story Points: n.a. Role: n.a.

Mandatory Story Points: n.a.

Description

References

Reference: (#) [create link](#)

Associate New: [Create a new referenced 'User Story'](#)

References:

- [User Story \(#17\): História 3](#) [delete link](#)
- [User Story \(#16\): História 2](#) [delete link](#)

Fonte: Imagem criada pelo autor

Como esta ferramenta não realiza a distinção entre os Status Aberto e Em andamento nem realiza a sincronização de Status entre as histórias e os Tickets, estes permanecem no Status de Aberto quando as histórias H1 e H2 são iniciadas na Sprint 1. Para acompanhamento do andamento, o analista deverá entrar no ticket e acessar suas histórias associadas ou entrar na história da sprint e acessar os tickets associados.

Com a chegada da demanda através do ticket T3, que é identificada como uma demanda igual à do ticket T2, basta ao analista criar as referências as histórias já existentes, através da funcionalidade de referenciar uma história a um Ticket, precisando apenas dos códigos sequenciais das histórias que a ferramenta cria. Na Figura 4.21 temos a demonstração deste processo, com a história H2 já associada e a história H3 sendo associada.

Figura 4.21: Associação de Histórias ao Ticket 3

Opened 14 seconds ago

Ticket 3

Reported by: [Cliente 1](#) Owned by:

Milestone: n.a. Keywords: n.a.

Business Value Points: n.a.

Total Story Points: n.a. Role: n.a.

Mandatory Story Points: n.a.

Description

References

Reference: (#) [create link](#)

Associate New: [Create a new referenced 'User Story'](#)

References:

- [User Story \(#16\): História 2](#) [delete link](#)

Fonte: Imagem criada pelo autor

No momento em que o cliente realiza o questionamento sobre a data de entrega do item, o analista deverá acessar o Ticket para verificar qual é o questionamento e irá acessar as histórias relacionadas por meio do link "Referencia", de forma a ver o Status das histórias e a sua previsão de entrega. No questionamento do time, o analista acessará a história e por sua vez utilizará o link "Referenciado por" para acessar o Ticket e encaminhar o questionamento, sendo que ao receber a resposta, deverá copiar manualmente o retorno para a história se necessário.

Como a funcionalidade não possibilita o encerramento automático do ticket na conclusão das histórias, ao final da *sprint* 1, o analista deve entrar no Ticket 1 manualmente para efetuar a alteração do Status para Resolvido, informando a funcionalidade desenvolvida para atendimento da demanda.

Com a demonstração do exemplo nesta abordagem, vamos realizar a avaliação desta abordagem frente aos critérios expostos na Seção 3.2

4.3.2 Avaliação seguindo os critérios derivados

- Conversão de tickets em histórias: esta abordagem possibilita a criação das histórias de usuário associando automaticamente ao ticket que originou a demanda, sendo 1 ou mais histórias que devem ser criadas, atendendo este critério.
- Sincronização de status: esta abordagem não atende este critério, visto que a alteração de Status de um artefato não reflete no outro, necessitando da alteração manual em ambos os lados.
- Interligação de informações entre ticket e história: esta abordagem atende este critério, visto que possibilita a comunicação entre ticket e histórias através dos links de "Referencia" e "É referenciado por" que ficam disponíveis nos artefatos, permitindo a fácil obtenção dos dados necessários.
- Fechamento do ticket: esta abordagem não atende este critério, visto que não possui funcionalidade de encerramento automático do ticket na entrega da funcionalidade resultante ao cliente.
- Associação do ticket com histórias existentes: é possível associar histórias existentes a um ticket novo, utilizando a funcionalidade de referenciar as histórias no novo ticket. Desta forma, esta abordagem atende este critério.

4.4 Consolidação das análises

A tabela 4.1 consolida as análises realizadas, mostrando quais critérios cada abordagem atende.

Tabela 4.1: Resumo das Avaliações

	Critério 1	Critério 2	Critério 3	Critério 4	Critério 5
Trello com Unito	AP	A	AP	AP	NA
Trello com Placker	A	A	NA	AP	NA
JIRA	NA	NA	A	NA	A
Trac	A	NA	A	NA	A

Onde as siglas representam:

- NA A abordagem não atende este critério.
- AP A abordagem atende parcialmente este critério.
- A A abordagem atende o critério.

A partir desta tabela, podemos concluir que nenhuma das abordagens analisadas atende todos os critérios derivados das atividades do analista. Assim, mesmo com a utilização de uma destas abordagens, algumas das atividades permanecerão manuais, com os problemas comuns da intervenção manual, como possíveis erros de preenchimento e esquecimentos, sendo necessária a avaliação de em quais pontos estes riscos são mais administráveis para o time ou companhia buscando este tipo de integração.

5 CONCLUSÃO

Neste trabalho, foi efetuado primeiramente a demonstração do problema da integração das demandas recebidas pelo cliente através de Tickets e que originam histórias de usuários utilizadas para o desenvolvimento de funcionalidades para atender tais demandas, do ponto de vista do analista, que é o responsável justamente pela intermediação entre os grupos dos clientes e de desenvolvimento, bem como é responsável pela definição das funcionalidades que serão desenvolvidas.

Para este problema, foi montado um exemplo de situação que pode ocorrer, com os diversos cenários de relacionamento entre os tickets e as histórias, derivando disto um conjunto de critérios para a avaliação de ferramentas de mercado que podem ser utilizadas para atender os dois paradigmas do problema, seja com o apoio de convenções ou com o apoio de extensões da ferramenta original.

Para cada uma das ferramentas selecionadas, foi aplicado o exemplo exposto demonstrando como cada situação seria atendida pela ferramenta, de forma a demonstrar se a mesma iria resolver os problemas enfrentados pelo analista quando são utilizadas ferramentas independentes ou uma ferramenta única que não realiza a integração entre os dois paradigmas. Após a aplicação do exemplo, realizamos a avaliação de cada ferramenta utilizando os critérios definidos e unificamos estes resultados.

Infelizmente, nenhuma das ferramentas escolhidas, apesar de cada uma utilizar uma abordagem diferente para tentar resolver o problema, atende todos os critérios definidos, sempre falhando em atender uma ou mais situações desta integração. Desta forma, mesmo utilizando uma ferramenta como estas, que atende os dois paradigmas (recebimento de demandas através de tickets e o desenvolvimento das funcionalidades para atender estas demandas com o uso de histórias de usuário) ainda serão necessárias algumas ações ou controles manuais.

REFERÊNCIAS

ARIZA, I. **Help Desk System**. [S.l.]: UMP, 2012.

BIESSE, F. **A Guide to Computer User Support for Help Desk and Support Specialists**. Boston, Massachusetts, USA: Cengage Learning, 2015.

CAROLI, P. **Sobre o Kanban**. 2017. Available from Internet: <<http://www.caroli.org/sobre-o-kanban/>>.

COHN, M. **User Stories Applied**. Upper Saddle River, New Jersey, USA: Addison-Wesley, 2009.

COHN, M. **Desenvolvimento de Software com Scrum**. São Paulo, Brasil: Bookman Editora, 2011.

GIOVANNINI, F.; MEANA, V. R. di. **Help Desk Applications at Telesoft**. Torino, Italy: Telesoft SpA, 1995.

HILES, A. **Service Level Agreements - Aligning IT Services to Business Needs**. Brookfield, Connecticut, USA: Rothstein Publishing, 2016.

KARLYN, M. R. O. e M. A. **A Guide to IT Contracting**. Boca Raton, Florida, USA: CRC Press, 2013.

KNIBERG, H.; SKARIN, M. **Kanban and Scrum - making the most of both**. [S.l.]: C4Media, 2010.

LARMAN, C. **Agile and Iterative Development - A Manager's Guide**. Upper Saddle River, New Jersey, USA: Addison-Wesley, 2004.

RUBIN, K. S. **Essential Scrum**. Upper Saddle River, New Jersey, USA: Addison-Wesley, 2013.

SCRUM.ORG. **What is Scrum?** 2017. Available from Internet: <<https://www.scrum.org/resources/what-is-scrum>>.

STEPHENS, R. **Beginning Software Engineering**. [S.l.]: Wiley, 2015. ISBN 9781118969175.

TIMO. **The advantages of a ticket system**. 2017. Available from Internet: <<https://www.timo24.de/blog/die-vorteile-eines-ticket-systems-46/?lang=en>>.

TRELLOTEAM. **Página Inicial do Trello**. 2017. Available from Internet: <<https://trello.com/>>.

ZENDESKTEAM. **Consistent Customer Service: the Ticketing System**. 2017. Available from Internet: <<https://www.zendesk.com/help-desk-software/features/ticketing-system/>>.