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

FERNANDA LANÇÃO MORAIS MARTINS

MÉTODOS ÁGEIS NA CONCEPÇÃO E MANUTENÇÃO DE SISTEMAS LIGADOS A INFRAESTRUTURA

Trabalho de Graduação.

Profa. Dra. Karin Becker
Orientador

Porto Alegre, Dezembro de 2013.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitora de Graduação: Prof. Sérgio Roberto Kieling Franco
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber
Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço aos diversos amigos que me deram força para seguir com esse estudo através de palavras ou até referências bibliográficas.

A minha família por torcer por mim durante todo o curso e, mais recentemente, pela paciência com a autora deste trabalho.

Ao Fernando Prestes Laimer por me dar a idéia deste trabalho durante sua palestra sobre o Sistema Toyota de Produção.

Ao setor de sistemas e redes da ThinkAgile, em especial ao chefe deste, Rafael Nunes, por receber minhas idéias de portas abertas, me ajudar e permitir que o setor seja cobaia deste estudo.

Aos criadores da ferramenta Mingle que se provou, sem dúvida, a melhor ferramenta para o uso de Kanban e, também, a Giordana Sfreddo pelos seus scripts de manipulação de dados no mesmo.

Ao Gabriel Oliveira, parceiro incansável nesta jornada e por me ensinar os truques do Excel.

A minha orientadora e a UFRGS pela oportunidade dada.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 6
LISTA DE FIGURAS 7
LISTA DE TABELAS 8
RESUMO 9
ABSTRACT 10
1 INTRODUÇÃO 11
2 O CONTEXTO DO SETOR DE SISTEMAS E REDES DA EMPRESA THINKAGILE 14
2.1 As Tarefas 14
2.2 Características Observadas 15
2.2.1 O sistema de chamados 15
2.3 Necessidades 16
3 KANBAN 18
3.1 Histórico 18
3.2 Organização e Conceitos 19
3.3 Motivação 21
4 A CULTURA ÁGIL 23
4.1 O Manifesto Ágil 23
4.2 Métodos Ágeis 25
4.2.1 SCRUM 25
4.2.2 XP 27
4.3 Práticas Ágeis 28
4.3.1 Reunião Dária 28
4.3.2 Retrospectiva 30
4.3.3 Reuniões de Planejamento do Sprint 31
4.3.4 Programação Pareada 31
4.3.5 Considerações 32
5 INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY 33
5.1 Visão Geral 33
5.2 Operação de Serviço 34
5.2.1 Processo de Gestão de Eventos 35
5.2.2 Processo de Gestão de Incidentes 36
5.2.3 Processo de Gestão de Problemas 37
5.3 Considerações 39
6 CONSTRUÇÃO E ADOÇÃO DO KANBAN 40
6.1 Organização do Kanban 40

6.1.1 Priorização 41
6.1.2 Investigação 42
6.1.3 Resolução 43
6.1.4 Fechamento 43
6.2 Adoção do Kanban 43
6.2.1 Propriedades dos Cartões 43
6.2.2 Ferramenta de Visualização 44
6.2.3 Escolha da prioridade dos cartões 45
6.2.4 Gestão de Solicitações de Clientes 46
6.2.5 Atomicidade de um cartão 47
6.2.6 WIP 48
6.3 Complementando o Kanban com práticas ágeis 49
6.3.1 Reuniões Diárias 49
6.3.2 Programação Pareada 50
6.3.3 Retrospectiva 51
6.3.4 Planejamento do Backlog 52
7 RESULTADOS DA THINKAGILE 53
7.1 Captura dos Dados 53
7.2 Lead Time & WIP 54
7.3 Raias Unplanned Vs Raias Planned 58
7.4 Conclusões sobre resultados 61
8 CONCLUSÃO 63
9 BIBLIOGRAFIA 65
APÊNDICE A 68
APÊNDICE B 69

LISTA DE ABREVIATURAS E SIGLAS

ITIL	Information Technology Infrastructure Library
S&R	Setor de Sistemas e Redes da empresa ThinkAgile
SLA	Service Level Agreement
STP	Sistema Toyota de Produção
TI	Tecnologia da informação
WIP	Work-in-progress ou tarefas simultâneas

LISTA DE FIGURAS

Figura 3.2.1: Kanban Simples (KNIBERG; SKARIN, 2010) 19
Figura 3.2.2: Wip e Lead Time no Kanban (CAROLI, 2013)..... 21
Figura 4.2.1.1: Scrum 26
Figura 4.3.2.1: Passos de uma Retrospectiva (DERBY; LARSEN, 2006) 30
Figura 5.1.1: Ciclo de Vida do Serviço (CARTLIDGE et al., 2007) 34
Figura 5.2.1.1: Gestão de Eventos baseado no livro de Van Bon (2007)..... 35
Figura 5.2.2.1: Gestão de Incidentes (VAN BON et al., 2007)..... 37
Figura 5.2.3.1: Gestão de Problemas (VAN BON et al., 2007) 38
Figura 6.1.2.1: Divisão de cartões no Kanban..... 42
Figura 6.2.2.1: Propriedades de um cartão 45
Figura 6.2.4.1: Gestão de Solicitações 47
Figura 7.2.1: Média de cartões por semana nos estágios indicados 54
Figura 7.2.2: Lead Time Médio por Semana..... 56
Figura 7.2.3: Lead Time Vs Número de Cartões..... 57
Figura 7.3.1: Planned Vs Unplanned..... 59
Figura 7.3.2: Relação das raias com o Lead Time..... 60

LISTA DE TABELAS

Tabela 6.1.1: Kanban do setor de S&R 41
Tabela 7.1.1: Tabela parcial exportada do Kanban 53
Tabela 7.2.1: Comparação detalhada Lead Time e WIP 58
Tabela 7.3.1: Detalhamento entre Lead Time e Raias..... 62

RESUMO

Gerenciar uma equipe que cria e administra componentes de infraestrutura e ainda prestar suporte aos usuários impõe alta disciplina ao gerenciar de requisições devido a diversificação de tarefas e cenários que cada atividade propõe.

Muitas empresas optam por utilizar uma ferramenta de gerenciamento de requisições, porém é observado que, ao atender empresas que utilizam a cultura Ágil, essa solução se mostra insuficiente.

Entende-se por cultura Ágil, uma cultura, principalmente utilizada em empresas de desenvolvimento de software, onde os valores e princípios explicitados no Manifesto Ágil são amplamente utilizados e afetam o comportamento de usuários e situações.

Esse trabalho propõe a utilização de Kanban , aliado a uma série de práticas ágeis, como forma de solucionar problemas e melhorar a coordenação de uma equipe de infraestrutura frente ao contexto apresentado.

Palavras-Chave: Metodologia, Ágil, ITIL, Kanban, gerenciamento, redes, sistemas, suporte.

ABSTRACT

Manage a team that creates and manages infrastructure and also provide support to users requires high discipline in managing requests due to the different tasks and scenarios that each activity proposes.

Many companies choose to use a request management tool, however it is noted that, while attending companies inside an Agile culture, this solution is insufficient.

It is understood by Agile culture, a culture, mainly used in software development companies, where the values and principles explained in the Agile Manifesto are widely used and affect the behavior of users and situations.

This paper proposes the use of Kanban, allied with a number of agile practices, as a way to solve problems and improve the coordination of a infrastructure team inside the context presented.

Keywords: Methodology, Agile, ITIL, Kanban, management, networks, systems, support.

1 INTRODUÇÃO

Abordagens ágeis, de acordo com o site Agile Methodology, são utilizadas no desenvolvimento de software para ajudar empresas a responder a imprevisibilidade. O XP (JEFFRIES, 1999) e Scrum (COHN, 1998a) são exemplos conhecidos de abordagens e propõem um conjunto de práticas que podem ser utilizadas em conjunto, de acordo com o método, ou em isolado. Entre as práticas frequentemente adotadas estão (MELO; SANTOS; CORBUCCI; et al., 2012):

- Uso de diferentes recursos visuais para planejar e monitorar itens de trabalho;
- Reuniões diárias para relatar o andamento do trabalho realizado;
- Reuniões de retrospectiva para analisar o trabalho que já foi concluído, analisar quais foram os pontos positivos e negativos deste e incorporar sucessos e melhorias no futuro;
- Reuniões para Priorização de Tarefas;
- Programação em Par.

Existe um consenso de que a adoção de práticas ágeis muito têm contribuído para o desenvolvimento de software de qualidade, com níveis de satisfação do cliente quanto ao retorno do investimento, prazos, custos, e qualidade do software (MELO et al., 2012)

Entre os recursos visuais, está o Kanban (KNIBERG; SKARIN, 2010), que tem por objetivo organizar o trabalho em estágios de processamento, cada qual com seu limite de tarefas simultâneas (WIP) e permitir a visualização dos itens de trabalho sendo processados. Com esta ferramenta, é possível estabelecer um fluxo de trabalho no qual não há desperdícios nem gargalos, o trabalho é cadenciado, e pode-se estabelecer o tempo médio de processamento dos itens.

Em desenvolvimento de software, as vantagens do Kanban frequentemente citadas são:

- Visualização de tarefas por estágio de desenvolvimento;
- Cálculo de vazão de tarefas;
- Limitar o WIP para obter um passo constante sem sobrecarregar os membros do time.

Assim, muitas empresas têm adotado métodos ágeis para projetos de desenvolvimento de software e manutenção de software. Contudo, menos se sabe quanto à adoção deste método em outros tipos de atividades relacionados aos ambientes de Tecnologia da Informação (TI).

As atividades ligadas à gestão de infraestrutura e redes é um exemplo. As principais atividades ligadas a infraestrutura e redes, de acordo com a Usenix[+] são:

- Instalar e manter servidores, redes e outros sistemas;
- Responder a indisponibilidades de serviços e outros problemas;
- Suporte a usuários.

O Information Technology Infrastructure Library (ITIL) é um conjunto de boas práticas a serem aplicadas na gestão de serviços de tecnologia da informação (CARTLIDGE; HANNA; RUDD; et al., 2007). Neste trabalho serão utilizados alguns conceitos do ITIL na construção e organização do Kanban.

O presente trabalho descreve a experiência de adoção de Kanban e outras práticas ágeis em uma empresa, neste trabalho denominada de ThinkAgile. Esta empresa possui larga experiência mundial no desenvolvimento de projetos de software de acordo com princípios ágeis, mas estas práticas não eram estendidas ao setor de redes e sistemas.

De forma resumida, os problemas do setor eram falta de visibilidade em suas tarefas, falta de foco devido a interrupções constantes de suporte e falta de confiabilidade por parte de outros setores da empresa. Como forma de solucionar esses problemas, o setor resolveu utilizar Kanban e outras práticas ágeis. Este esforço tem sido empírico, e realizado sem nenhuma fundamentação teórica.

Os objetivos deste trabalho são:

- Analisar este esforço, posicionando-os face a um conjunto de requisitos para gestão de serviços de tecnologia derivados do ITIL;
- Propor soluções dentre as práticas ágeis adequadas ao setor e seus requisitos;
- Analisar de forma crítica a sua adoção.

O restante do trabalho está organizado como segue:

- No capítulo 2, será mostrado em profundidade o contexto e os problemas enfrentados pela ThinkAgile;
- No capítulo 3, será apresentada a motivação para o uso de Kanban como ferramenta visual;
- No capítulo 4, será abordada, brevemente, a cultura ágil e algumas práticas ágeis utilizadas neste trabalho;
- No capítulo 5, serão mostrados alguns conceitos e processos do ITIL;

[+] https://www.usenix.org/system/files/lisa/books/usenix_22_jobs3rd_core.pdf

- No capítulo 6, será discutido sobre a organização e adoção do Kanban;
- No capítulo 7, serão apresentados resultados obtidos na ThinkAgile com a adoção do Kanban;
- No capítulo 8, será feita uma conclusão sobre este trabalho.

2 O CONTEXTO DO SETOR DE SISTEMAS E REDES DA EMPRESA THINKAGILE

Para entender a motivação para o uso Kanban e outras práticas ágeis na administração de sistemas e redes, é preciso entender um pouco os problemas que um administrador de infraestrutura pode ter no seu dia-a-dia e como a cultura da ThinkAgile motivou um comportamento diferente ao lidar com serviços. Por esse motivo foi realizado um estudo de caso com a ThinkAgile, mais especificamente, com o setor de Sistemas e Redes (S&R) desta empresa.

2.1 As Tarefas

A equipe S&R da empresa ThinkAgile atende a times de desenvolvimento software e ao setor administrativo da empresa e possui as seguintes tarefas:

- Manter os servidores existentes;
- Projetar novos servidores;
- Manter equipamentos de infraestrutura em funcionamento;
- Provisionar máquinas com o sistema operacional e alguns softwares da empresa para os seus funcionários;
- Prover suporte de infraestrutura e aplicações a todos os funcionários da empresa;
- Auxiliar na abertura de novos escritórios da empresa.

Embora esse seja um estudo de caso de uma empresa pode-se observar que todas as tarefas podem vir a ser comuns ao dia-a-dia de qualquer administrador de sistemas conforme documento na Usenix[+].

Muitas vezes as empresas optam por dividir as tarefas em pequenos grupos: uma equipe de suporte, de administradores de infraestrutura, administradores de servidores focados em um tipo de sistemas e etc. No caso da empresa ThinkAgile todas as tarefas são realizadas pelo mesmo setor.

[+] https://www.usenix.org/system/files/lisa/books/usenix_22_jobs3rd_core.pdf

2.2 Características Observadas

Durante a realização das tarefas mencionadas foi identificado que:

- O S&R tem que lidar com muitas interrupções de suporte e em alguns casos estas são tão sérias (como a falha de um servidor) que o setor têm que abandonar as tarefas que está fazendo e atacar diretamente a interrupção;

- Dar suporte as equipes de desenvolvimento é a principal prioridade do time, mas isso gerava muitas requisições sobrecarregando o time. Por vezes muitas requisições demoravam para serem atendidas o que gerava uma certa frustação por outros setores da empresa;

- O trabalho do S&R não era contabilizado e o setor não conseguia mostrar a seus superiores a quantificação do trabalho realizado, em realização ou pendente;

- O time era separado geograficamente e possuía problemas de comunicação dentro do time de S&R.

2.2.1 O sistema de chamados

Uma das formas mais comuns de garantir o bom atendimento em atividades de prestação de serviços e suporte é controlá-los através de sistemas de chamados. De forma genérica e resumida, em um sistema de chamados:

- Um usuário que necessita de um serviço registra sua demanda especificando detalhes de sua chamada, incluindo prioridades e datas. Esta demanda gera direta ou indiretamente um chamado;

- Uma vez no sistema, o chamado passa por algum processo de triagem, possivelmente (semi) automatizado, que o redireciona para a equipe que melhor consegue resolver o problema;

- Uma vez direcionado a uma equipe ou pessoa, uma investigação sobre a demanda é realizada, uma solução proposta é implementada, dentro das prioridades da demanda e da capacidade da equipe de atendimento.

É importante destacar que um chamado pode ser criado tanto por clientes que necessitam de serviços, como pelos próprios administradores de S&R como forma de controlar seu fluxo de trabalho.

A empresa ThinkAgile possui vários escritórios pelo mundo, cada qual com seu setor de S&R. Para que as demandas sejam melhor controladas, foi adotado um sistema de chamados. Dessa forma o funcionário de S&R que estivesse no Japão poderia auxiliar usuários que estivessem na Índia e os vários setores de S&R poderiam se ajudar.

Contudo, é cultura da empresa priorizar a comunicação face-a-face. Logo os usuários iam pessoalmente conversar com o setor de S&R e, portanto, nunca usavam o

sistema. O administrador de sistemas poderia abrir um chamado ele mesmo, porém isso implicava em mais um trabalho que o administrador teria que fazer.

A empresa ThinkAgile usa métodos ágeis para as equipes de desenvolvimento e isso implica em vários princípios que serão ressaltados ao longo desse trabalho. Entre eles, espera-se que o setor de S&R seja ágil ao atender seus pedidos. Esperar uma requisição passar por todos os processos descritos acima para então gerar uma solução não era uma opção culturalmente aceita pela empresa.

O sistema de chamados adotado pelo S&R não apresenta uma visualização satisfatória de requisições, de seu andamento, e de sua priorização. Assim, os usuários que haviam realizado as demandas, mais uma vez, vinham conversar face-a-face com o time de S&R local para que este resolvesse o problema em questão.

Ainda, o próprio time de S&R não conseguia visualizar muito bem seus chamados pois a ferramenta não é muito intuitiva o que fazia com que os próprios membros do time não a usassem com frequência.

Grande parte das demandas de suporte (cerca de 90%) são locais, o que significa que o time global não pode ajudar o time local e os chamados, neste caso, são uma mera formalidade.

Os problemas de visualização do sistema de chamados poderiam ser resolvidos com a aquisição de uma nova ferramenta, porém os problemas mais impactantes para o não uso de qualquer ferramenta de chamados reside no fato de que a ThinkAgile é uma empresa que possui uma cultura ágil, como apontado anteriormente. Será visto mais à frente, no capítulo 4, do que se trata essa cultura e como ela foi uma grande motivação para a realização desse trabalho.

2.3 Necessidades

De posse das características observadas, o time de S&R pensou em soluções que atendessem aos seguintes requisitos:

- Era preciso conscientizar os usuários de que, se sua demanda não fosse crítica, era necessário esperar sua resolução. Muitas vezes o usuário esperava que seu problema fosse atendido prontamente ou em um espaço curto de tempo, o que o fazia questionar constantemente o setor, gerando mais interrupções;

- Como o S&R fazia pouco uso da ferramenta de chamado, devido a problemas já mencionados, era preciso uma ferramenta ou processo que mostrasse aos usuários o andamento de sua demanda;

- O S&R precisava visualizar suas próprias tarefas. Era necessário encontrar uma ferramenta que possibilitasse que cada membro do time soubesse o que o outro estava fazendo. Dessa forma o trabalho não seria duplicado e o problema de comunicação poderia ser apaziguado;

- O S&R precisava de uma ferramenta ou processo para controle de fluxo das tarefas. Diante de tantas interrupções de suporte, era comum ao time perder o foco. O time ficava tão concentrado em resolver solicitações de

usuário que não conseguia evoluir nas tarefas de longo prazo. Com o controle de fluxo, esperava-se conseguir manter um foco maior nestas;

- O time precisava de uma ferramenta que contabilizasse tarefas. Dessa forma seria possível medir quantas tarefas estavam sendo concluídas em um período de tempo, verificar em quais tipos de tarefa estavam gastando mais tempo e etc. Com esses dados seria possível criar estimativas, justificar novas contratações, se preparar para meses em que as solicitações de usuários são maiores e etc.

3 KANBAN

Neste capítulo será apresentado o Kanban mostrando seu contexto histórico, conceitos, organização e motivações para seu uso dentro da administração de infraestrutura.

3.1 Histórico

O Kanban tem origem no Sistema Toyota de Produção (STP). O STP tem como um dos seus principais objetivos evitar desperdícios (Muda), classificando-os como Superprodução, Espera, Transporte Desnecessário, Processamento Impróprio, Estoque Desnecessário, Movimentos Desnecessários e Defeitos. Ainda, propõe uma série de princípios para que estes sejam evitados, dentre os quais alguns interessantes a este trabalho são citados por Liker (2004):

1. Baseie suas decisões em uma filosofia, mesmo em detrimento de metas financeiras de curto prazo. As pessoas precisam de um propósito para achar motivação e estabelecer metas;

2. Crie um fluxo de processo contínuo para trazer problemas para a superfície. Processos são redesenhados para evitar desperdício;

3. Use sistemas para evitar superprodução. Produza apenas o material necessário depois que a operação subsequente sinaliza uma necessidade para isso;

4. Nivele sua carga de trabalho;

5. Construa uma cultura que pare para resolver os problemas, construindo-os desde o início com qualidade. Esse princípio é exemplificado quando há, por exemplo, um problema na linha de produção e, ao perceber isso, o funcionário puxa um cordão e pára a linha de produção. A partir daquele momento o foco é resolver o problema o mais rápido possível e depois pensar em como aquele problema pode ser prevenido;

6. Padronize tarefas e processos;

7. Use controles visuais, dessa forma não haverá problemas escondidos;

8. Use somente tecnologias confiáveis e testadas com as pessoas e processos;

9. Crie líderes que entendam o trabalho, vivam a filosofia e que ensinem aos outros;

10. Desenvolva pessoas e times excepcionais que seguem a filosofia da empresa;

11. Respeite sua rede de parceiros, e fornecedores desafiando-os e ajudando-os a melhorar.

A Toyota, famosa na fabricação de carros, utiliza um sistema sobre demanda chamado de pull. Este sistema tem como objeto dar ao cliente o que ele precisa, quando

ele precisa, na quantidade em que ele quer (LIKER, 2004) e é utilizado para satisfazer o item 3 acima.

A fábrica da Toyota precisava de uma maneira de sinalizar que sua linha de produção tinha usado todas as peças e precisava de mais peças. Foi então que Ohno (criador do STP) usou cartões, caixas e carrinhos vazios chamados de kanban. (LIKER, 2004). Kanban significa sinal de qualquer tipo, logo, toda vez que uma caixa vazia aparecia na Toyota era um sinal de que era preciso repor as peças.

É importante lembrar que mesmo possuindo esse sistema de demanda, a Toyota necessitava de estoque para algumas peças. Era preciso manter algum estoque já que a fabricação de peças na Toyota dependia de um processo e obtê-las não era de um dia para o outro, mas também era preciso evitar o estoque desnecessário (LIKER, 2004). O estoque mínimo também pode ser chamado de buffer.

Quando os funcionários da Toyota usavam as peças eles colocavam um cartão numa caixa. Um pessoa vinha de tempos em tempos, pegava essa caixa e reabastecia as peças baseado no número de cartões deixados pelos funcionários. Da mesma forma, assim que esta pessoa retirava peças do estoque era sinalizado ao fornecedor das peças que o estoque precisava de mais peças e assim por diante.

3.2 Organização e Conceitos

Em desenvolvimento de software, o Kanban é representado através de um mural que possui cartões, mas estes não significam peças, mas sim itens de trabalho. Ele também possui colunas que representam estágios de trabalho. Esses estágios são flexíveis e devem ser definidos de acordo com a necessidade em questão. Os cartões são movidos da esquerda para direita entre os estágios conforme a Figura 3.2.1.

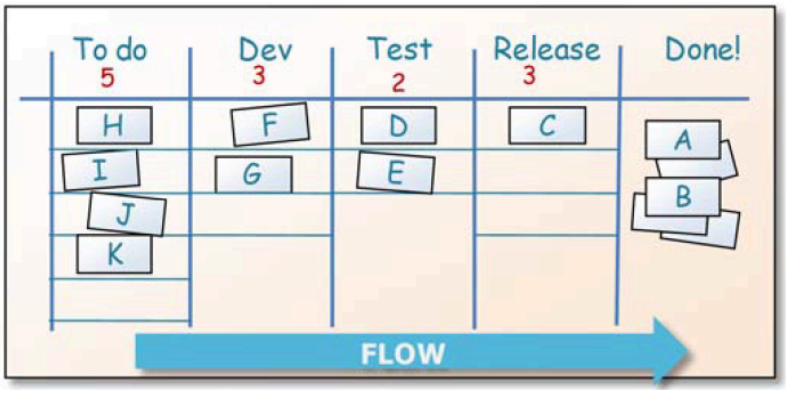


Figura 3.2.1: Kanban Simples (KNIBERG; SKARIN, 2010)

Na Figura 3.2.1, os cartões estão representados por letras. Ao invés de letras, é possível colocar dados relativos ao cartões como seu nome, descrição e prioridade, mas este é um mural simplificado.

Os estágios de trabalho, na Figura 3.2.1, estão representados pelas colunas Todo (para fazer), Dev (em desenvolvimento), Test (em teste), Release (em lançamento da versão) e Done (terminado). Cada empresa pode criar seus próprios estágios com o objetivo de representar melhor seu fluxo de trabalho.

Em vermelho, na Figura 3.2.1, logo abaixo do nome das colunas, tem-se o número máximo de cartões (ou tarefas simultâneas) que são executadas por coluna. Chamamos esse número de limite do WIP (work-in-progress). Quando houverem 3 (três) cartões no estágio de Desenvolvimento, nenhum outro cartão poderá entrar nesse estágio e assim por diante.

Ao limitar o número de tarefas simultâneas é possível criar um equilíbrio entre a demanda e o quanto o time pode produzir. Fazendo isso, será possível criar uma passo sustentável, no desenvolvimento do software, e todos os indivíduos poderão ter um equilíbrio entre trabalho e a vida pessoal (ANDERSON, 2010).

De acordo com Anderson (2010), controlar o WIP pode ter um impacto significativo sobre a qualidade ao impedir que os membros do time fiquem sobrecarregados de trabalho. Ele afirma, de acordo com sua experiência em projetos de desenvolvimento, que os defeitos no projeto irão aumentar conforme o WIP aumenta. Isso implica que, ao realizar uma ou pouquíssimas tarefas por vez, o desenvolvedor pode se focar exclusivamente na(s) tarefa(s) correntes.

O limite do WIP também encoraja que um membro do time, ao ver que não pode mais trabalhar em cartões, auxilie no andamento de outras tarefas que não as suas. Isso aumenta a colaboração e a confiança no time. Outro benefício é que, ao estabelecer um WIP, é dada mais atenção à priorização de tarefas, pois se sabe que ao puxar um cartão isso implica em estar mais próximo do limite estabelecido.

É importante que o WIP máximo não seja muito pequeno. Se isso ocorrer, membros do time ficarão ociosos muito rápido. Se o WIP máximo for muito grande isso fará com que eles façam muitas tarefas ao mesmo tempo, o que não permite que os benefícios do mesmo sejam alcançados.

Anderson (2010) diz que existem algumas pesquisas e observações empíricas que sugerem que o limite do WIP deve ser de dois itens por pessoa, porém estes não levam em consideração a maturidade do time ou a ocorrência de problemas externos. Conclui-se então que o WIP deve ser estabelecido através de observações empíricas no comportamento e fluxo de trabalho de cada time.

Além de limitar o WIP, o Kanban também possui outras propriedades como o Lead-Time e o Throughput. De acordo com Anderson (2010), o Throughtput, ou vazão, é a quantidade de cartões - ou qualquer outra indicação de valor – que foram entregues em um período de tempo.

O Lead Time é o tempo que um cartão entra no mural até sua saída. No mural acima o Lead Time é o tempo em que um cartão entra no Backlog até ir para Done.

Um Lead Time pequeno, significa a entrega de funcionalidades mais rapidamente, já um Lead-Time maior significa o contrário. Nessa comparação é importante também considerar o tamanho do cartão. Não se pode comparar o Lead-Time de cartões (ou tarefas) com complexidades distintas, conforme apontado por Caroli (2013).

Anderson (2010), ao observar o comportamento de dois times na Motorola, percebeu uma relação entre WIP e o Lead Time: quanto maior o Lead Time médio, pior é a qualidade. Como se sabe que o WIP está relacionado à qualidade, logo quanto maior o WIP, pior a qualidade. Portanto entende-se que quanto maior o WIP, maior é o Lead Time.

Na Figura 3.2.2, pode-se visualizar o WIP e o Lead Time no Kanban. Perceba que nesse mural junto aos cartões foi indicado o dono da tarefa em laranja:

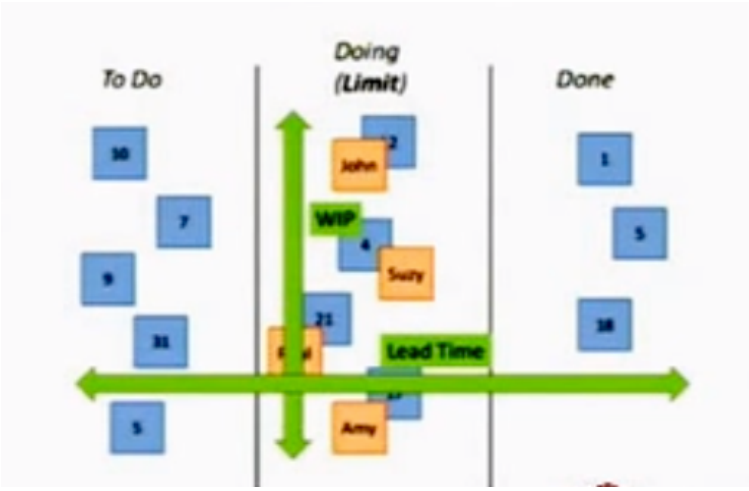


Figura 3.2.1: Wip e Lead Time no Kanban (CAROLI, 2013)

Existem várias maneiras de medir o Lead Time (CAROLI, 2013):

- Através da fórmula: WIP = Throughput x Lead Time;
- Anotar a data de entrada e saída no cartão;
- Através da realização de gráficos como o Control Chart (ACHOUIANTZ, 2013) e o Cumulative Flow Diagram (CAROLI, 2013) que não serão abordados neste trabalho.

Pode-se analisar o limite do WIP e o Lead Time através do tempo e, caso algum comportamento se mostre fora do padrão, pode-se tentar modificar o Kanban como forma de obter mais vantagens e maximizar o uso do Kanban conforme será mostrado no capítulo 7.

3.3 Motivação

Foram vistos os problemas do S&R e a tentativa da ThinkAgile de usar o método tradicional de suporte através de um sistema de chamados e como isso foi insuficiente. Neste capítulo foi descrito como funciona o Kanban e suas principais propriedades.

Os motivos que levaram o time a utilizar o Kanban no setor de S&R foram:

- Visualização do fluxo de trabalho: Possibilitar a visualização das tarefas entre os membros do time com o objetivo de manter o foco e saber o que os seus colegas do time estão fazendo. Além disso os clientes do S&R também podem visualizar o Kanban do time para saber o andamento de suas tarefas;

- Limitar o WIP: Como forma de não sobrecarregar os membros do time, ajudar a manter o foco, especialmente, evitando realizar tarefas de suporte que não tinham prioridade;
- Medir e gerenciar o fluxo: Controlar tarefas através do uso de métricas como Lead Time e Throughput;
- Flexibilidade: O Kanban é uma ferramenta flexível que não impõe mudanças radicais ao time, ao contrário de outros métodos iterativos, que necessita de mais disciplina para ser adotado (KNIBERG; SKARIN, 2010) como veremos adiante.

Uma vez construído, o Kanban é uma ferramenta muito simples e ajudou muito o time de S&R, porém foi preciso muito trabalho para construir um Kanban que refletisse o fluxo de trabalho do time. A construção do mural é fundamental para que o Kanban cumpra os seus objetivos. Era preciso que as colunas do Kanban refletissem exatamente o trabalho do time de S&R e, devido a gama de tarefas que o time possui, esse processo se deu de forma gradual e lenta.

Antes de analisarmos o Kanban resultante é importante entender a cultura ágil que motivou, ainda mais, essa adoção.

4 A CULTURA ÁGIL

A grande motivação que impulsou a criação deste trabalho tem origem na metodologia Ágil. Foi decidido trabalhar com métodos ágeis pois a empresa na qual o caso de estudos foi realizado é uma empresa com vários valores e filosofia que condiz com a metodologia.

Os valores e a filosofia ágil são tão impactantes que criaram uma cultura ágil dentro da ThinkAgile. Essa cultura, como foi descrito nos capítulos anteriores, impede que soluções tradicionais, como o sistema de chamados, sejam suficientes na gestão da infraestrutura.

Será visto neste capítulo do que se trata a filosofia ágil e seus valores como também algumas práticas como Reuniões Diárias e Programação em Par que são, em suma, práticas Ágeis. Essas práticas foram fundamentais para a manutenção do Kanban como será visto adiante.

4.1 O Manifesto Ágil

Os métodos ágeis tiveram sua origem no manifesto ágil, que se opunha ao desenvolvimento de software centrado em processos pesados. O Manifesto Ágil propõem os seguintes valores:

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

Fazendo uma comparação com o que foi dito sobre o sistema de chamados nos capítulos anteriores, podemos notar, no primeiro princípio, que o uso de uma ferramenta tradicional de chamados não parece condizente com uma cultura ágil.

Os chamados passam essencialmente por processos de atendimento e esta é uma contradição com o primeiro princípio ágil. Um chamado também demanda bastante documentação. Ao abrir um chamado em um sistema simples é necessário: uma descrição do problema, um andamento, uma prioridade e uma categoria. Todos esses detalhes de documentação vão contra o segundo valor.

Ao alterar um chamado, uni-lo ou dividi-lo com outro pode ser trabalhoso e não muito intuitivo para quem tenta responder rápido a mudanças como o quarto princípio gostaria.

A ideia é que esse paradigma engessado de processo, documentação e pouca interação seja quebrado e o Kanban seja usado para facilitar a interação com o usuário, com pouca ou nenhuma documentação e processos mais leves.

O manifesto ágil inclui, ainda, um conjunto de 12 princípios. Seu significado no desenvolvimento e manutenção de sistemas tem sido muito debatidos, mas possuem menor entendimento em outros setores relacionados a TI. Abaixo, será apresentado como estes princípios, explicitados em (BECK, 2001) podem ser aplicados a administração de sistemas, foco deste trabalho.

É importante destacar que as referências a clientes nesse capítulo podem ser correlacionadas diretamente com todos os níveis de usuários ou clientes que podem vir a ser atendidos por uma equipe de TI, desde o mais simples até o mais experiente. Tenha em mente que o nível de solicitações também pode variar: desde um simples suporte a problemas, até a concepção de sistemas.

A. "Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.". Um paralelo pode ser feito com administração de sistemas ao tentar entregar valor aos clientes, seja ele através de um atendimento de suporte até um projeto mais complexo. Entende-se por valor a entrega do serviço ou a resolução de um problema no menor tempo possível, com a máxima qualidade e com interação do usuário do início ao fim de sua solicitação.

B. "Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.". Também pode-se fazer uma correlação com administração de sistemas ao adequar mudanças na construção de um serviço, por exemplo.

C. "Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo". Também pode-se usar esse princípio ao entregar um serviço.

D. "Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho". O fator confiança neste princípio é de vital importância para que o Kanban contribua positivamente para a administração de sistemas pois cada um deve ter a liberdade para trabalhar em qualquer cartão que se sinta a vontade e com motivação.

E. "O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face-a-face.". Novamente aqui podemos observar como a interação, especialmente face-a-face, foi determinante nesse trabalho como apontado nos capítulos anteriores.

F. “Software funcionando é a medida primária de progresso.”. Manter um serviço funcional também pode ser visto uma medida de progresso para o administrador de sistemas. Quanto mais o sistema é estável, mais feliz o cliente fica.

G. “Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.”.
Esse princípio condiz com o que o setor S&R deseja: manter um foco e ritmo em suas tarefas conforme apontado nos capítulos anteriores.

H. “Simplicidade -a arte de maximizar a quantidade de trabalho não realizado - é essencial.”.
Podemos fazer um paralelo com a automatização de sistemas e com processos para o atendimento de suporte. Com ambas pode-se atingir a simplicidade ao lidar com o trabalho do dia-a-dia. Porém não iremos tratar disso neste trabalho.

I. “As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis”
Equipes auto organizáveis tendem a ter mais autonomia para realizar tarefas. O Kanban ajudou neste princípios ao trazer mais autonomia ao time.

J. “Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.”
Também será visto como esse princípio será importante juntamente ao Kanban na administração de sistemas com o uso de reuniões de Retrospectiva.

4.2 Métodos Ágeis

Os princípios fundamentam vários métodos ágeis, dentre os quais os mais populares são o Scrum e XP.

4.2.1 SCRUM

O Scrum, de acordo com Cohn (1998), é um framework utilizado em desenvolvimento de software que se baseia em times auto organizáveis. Isto significa que todos os problemas são resolvidos por todos os membros do time ao invés da centralização de decisões em um chefe.

O processo pode ser observado na Figura 4.2.1.1 e ao longo dessa seção serão abordado os elementos dentro do Scrum.

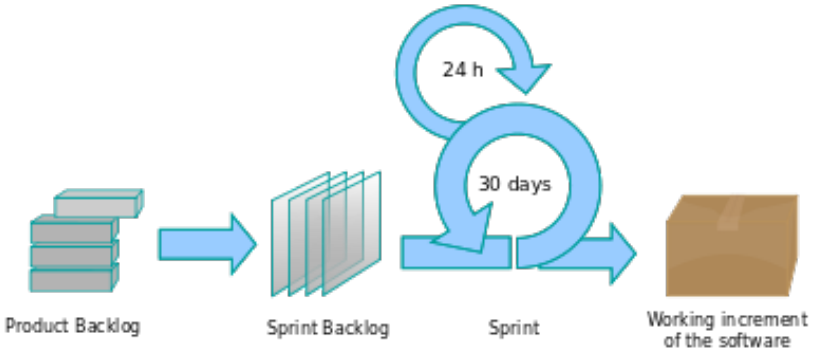


Figura 4.2.1.1: Scrum

Um projeto Scrum é realizado através de uma série de Sprints. Um Sprint, unidade básica do Scrum, não pode durar mais que um mês e é um esforço dentro de uma caixa de tempo (ou seja, restrito a uma duração específica) de comprimento constante (COHN, 1998a).

No início do Sprint, membros do times se comprometem a desenvolver algumas das funcionalidades que são listadas no Backlog do Produto. Ao final do Sprint, essas funcionalidades devem ser codificadas, testadas e integradas ao produto ou sistema. Uma revisão é, então, conduzida na qual o time demonstra as novas funcionalidades ao dono do produto e outros interessados que contribuem com feedbacks que podem influenciar o próximo Sprint. O ciclo então se repete até o lançamento do produto (Release).

Cohn (1998), ainda, afirma que o Scrum possui os seguintes artefatos :

- O Backlog do Produto é uma lista completa de funcionalidades que pode mudar de tamanho ao longo do projeto. Geralmente é organizada por prioridade;

- O Backlog do Sprint é a lista de funcionalidades que deverá ser desenvolvida em um único Sprint;

- Gráficos de Burndown do Sprint e da Release. Gráficos de burndown mostram a quantidade de trabalho restante seja em um Sprint ou em uma Release. Eles são ferramentas muito eficazes para determinar rapidamente se um sprint ou Release está programado para ter todo o trabalho planejado concluído até a data desejada.

O Scrum também possui algumas atividades (COHN, 1998a):

- Reunião de Planejamento do Sprint onde o dono produto define quais funcionalidades são prioritárias e o time define quais tarefas são factíveis de serem terminadas no Sprint. Dessa reunião é gerado o Backlog do Sprint;

- Reunião Diária é frequentada por todos os membros da equipe, incluindo o proprietário do produto. Esta reunião tem duração de não mais que 15 minutos. Durante esse tempo, os membros da equipe compartilham no que trabalharam no dia anterior , no que irão trabalhar e identificam eventuais impedimentos para o progresso.

- No final de um Sprint, a equipe conduz uma revisão sprint. Durante a Revisão da Sprint, a equipe demonstra as funcionalidades adicionadas durante o sprint. O objetivo deste encontro é obter feedback do proprietário do produto, quaisquer usuários ou outras partes interessadas que foram convidadas para a revisão. Este feedback pode resultar em alterações em funcionalidade recém-entregues.

- Outra atividade realizada no final de cada sprint é a Retrospectiva. O encontro é uma oportunidade para refletir sobre o sprint que está terminando e identificar oportunidades de melhoria no novo sprint.

Vários benefícios do Scrum são apontados por um grupo de consultoria em Ágil chamado de Braintrust como:

A. Promover transparência;

B. Permitir aos clientes a mudança na priorização e requisitos mais rapidamente;

C. Promover um melhor relacionamento entre a equipe e o cliente.

O Scrum também pode ajudar a (KNIBERG, 2007):

1. Evitar atrasos de entrega de tarefas ou produto;

2. Trazer mais qualidade ao produto;

3. Reagir e responder mais rapidamente a mudanças.

4.2.2 XP

O XP (Extreme Programming) é descrito por Jeffries (1999) como uma disciplina de desenvolvimento de software com base nos seguintes valores: simplicidade, comunicação, feedback e coragem. Ele funciona, trazendo toda a equipe, na presença de práticas simples, com feedback suficiente para permitir que a equipe veja onde está e ajuste as práticas à sua situação específica.

Existem 13 práticas no XP (JEFFRIES, 1999): Time Coeso, Jogo de Planejamento, Fases Pequenas, Testes de Aceitação, Design Simples, Programação Pareada, Desenvolvimento Orientado a Testes, Refatoração, Entrega Contínua, Propriedade Coletiva, Padronização de Código, Metáfora e Passo Sustentável.

Algumas semelhanças com Scrum podem ser observadas como:

- Time Coeso diz que o time e o dono do produto devem estar sempre presentes no desenvolvimento do mesmo;

- Jogo de Planejamento define o planejamento de interações semelhante ao de Sprints como é feito pelo Scrum. Outra semelhança com Scrum é que ele define o planejamento de releases e o uso de priorização das funcionalidades;

- Fases Pequenas encorajam a entrega incremental de software;

- O Passo sustentável, semelhante ao Scrum, incentiva o uso de uma velocidade constante e saudável para o time durante a construção do software.

A seguir, serão explicadas mais a fundo algumas práticas que podem ser vantajosas na área de administração de sistemas e redes.

4.3 Práticas Ágeis

Neste trabalho algumas práticas específicas advindas das metodologias ágeis citadas foram utilizadas com o propósito de auxiliar a manutenção do Kanban.

4.3.1 Reunião Dária

A reunião diária foi originalmente proposta no Scrum, e posteriormente adotada como um dos 13 princípios do XP. De acordo com Martin Fowler (2011) as reuniões diárias do Scrum têm cinco objetivos:

- Começar bem o dia: Energizar o time, esclarecer os objetivos e o que precisa ser feito para que esses sejam alcançados;

- Melhorias: As reuniões servem para expor os problemas que o time está enfrentando. Também servem para compartilhar conhecimento. Ambas as práticas trazem melhorias para o time. É importante destacar que os problemas, se complexos, devem ser resolvidos em uma reunião separada para que a reunião diária não se estenda.

- Foco: As reuniões encorajam a manter o foco sobre as tarefas para que objetivos sejam alcançados

- Time: As reuniões incentivam a comunicação no time e que cada um ajude o outro em busca de soluções para obstáculos comuns.

- Status: O time compartilha informações valiosas sobre o progresso de trabalho e fatos interessantes.

Os objetivos observados são muito semelhantes com o que o time de S&R precisa que é manter o foco da time e aumentar comunicação.

A reunião deve, principalmente, responder as seguintes questões (FOWLER, 2011):

- O que você concluiu ontem?

- O que você irá fazer hoje?
- Quais obstáculos estão impedindo seu progresso?

Essa estrutura não precisa necessariamente ser seguida, o mais importante é a informação compartilhada. Conforme o time amadurece, essas perguntas são respondidas naturalmente.

Também não existe ordem sobre quem começa a dar sua contribuição na reunião. Geralmente as pessoas se baseiam em algum quadro de progresso ou existe um facilitador que controla quem deve começar a falar.

Quando há um facilitador, é importante que este seja trocado periodicamente pois a ideia é que os times sejam autônomos e não dependam de um "chefe" ou pessoa para que a reunião aconteça.

Problemas comuns que podem acontecer em uma reunião diária e devem ser evitados (FOWLER, 2011) são:

A. Quando pessoas dependem de um líder para ser facilitador como dito anteriormente;

B. Pessoas começam a chegar atrasadas na reunião;

C. Depender da reunião para começar o dia: Esse problema tende a ocorre com reuniões que não ocorrem de manhã. É importante que as pessoas não usem a reunião como ponto de partida para começar a trabalhar;

D. Socialização: A reunião deve possuir um foco no que está sendo trabalhado e não deve ser usada para bate papo;

E. Falta de preparação: Todos os participantes devem lembrar o que fizeram de modo a poder contribuir mais efetivamente na reunião que é de curta duração. Por esse motivo é bom usar um quadro de tarefas para o time poder se lembrar mais rapidamente de seus itens de trabalho;

F. Resolução de Problemas: A reunião não deve ser usada para resolver problemas complexos. Se necessário realize uma reunião para que um problema abordado na reunião diária seja resolvido;

G. Contar histórias: Não entre em detalhes sobre seu trabalho. Apenas compartilhe o necessário;

H. Obstáculos não são levantados: Um dos principais pontos é que a reunião sirva para levantar problemas. É importante que a pessoas sejam encorajadas a compartilhar seus problemas. Se vários problemas são levantados durante as reunião não se esqueça ou ignore eles.

4.3.2 Retrospectiva

Como foi dito anteriormente, a Retrospectiva é uma reunião utilizada em Scrum que serve para identificar o que pode ser melhorado de uma iteração para a outra. Isso possibilita que problemas sejam identificados e tratados.

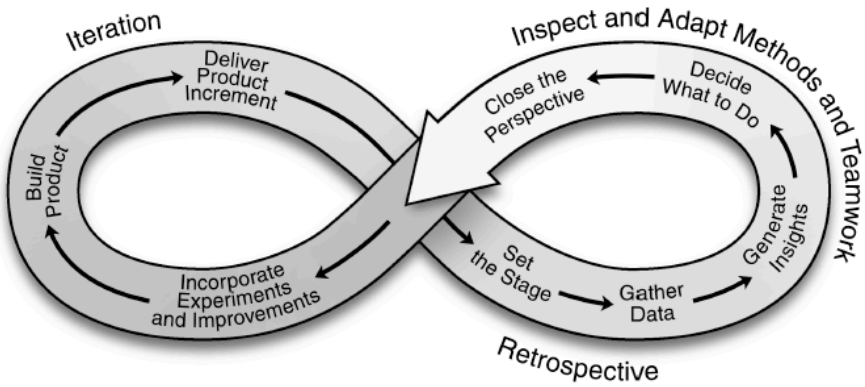


Figura 4.3.2.1: Passos de uma Retrospectiva (DERBY; LARSEN, 2006)

Derby e Larsen (2006) descrevem em seu livro a estrutura de uma reunião de retrospectiva: Set The Stage, Gather Data, Generate Insights, Decide What To Do e Close the Retrospective. Esses passos podem ser observados na Figura 4.3.2.1.

4.3.2.1 Set The Stage

O líder da retrospectiva, geralmente o líder do projeto, estabelece o objetivo principal da reunião e é responsável por criar um ambiente confortável para que as pessoas possam discutir problemas.

Antes da reunião também é preciso definir algumas regras como, por exemplo, não atender o telefone durante a reunião para que não existam interrupções. Não é preciso estabelecer regras para cada reunião, tendo o time discutido sobre isso uma vez já é o bastante.

No início da reunião, é preciso dizer seu objetivo, como este será abordado e quanto tempo irá durar a reunião. Também é preciso que todas as pessoas falem ou se apresentem e, assim, façam um check-in na meeting.

4.3.2.2 Gather Data

Gather Data é a próxima fase. Nela são coletados os dados de cada participante sobre os assuntos abordados na reunião. Isso pode ser feito da seguinte forma: cada um anota suas opiniões em um post-it, por exemplo, e o grudam em um quadro. Posteriormente, temos a fase chamada de Generate Insights onde são procurados padrões nos mesmos agrupando-os de acordo.

4.3.2.3 Decide What To Do

Decide What To Do é a fase na qual é necessário definir o que vai ser feito nas próximas iterações para melhorar os itens identificados no passo anterior. Se o time sugerir muitas mudanças, algumas são escolhidas e outras são colocadas em um backlog onde serão escolhidas em iterações futuras.

4.3.2.4 Close The Retrospective

Por fim temos a fase de Close The Retrospective onde ocorre o fechamento da reunião mostrando o que foi aprendido.

4.3.3 Reuniões de Planejamento do Sprint

Como já foi visto, a Reunião de Planejamento do Sprint é onde o dono produto define quais funcionalidades são prioritárias e o time define quais tarefas são factíveis de serem terminadas no Sprint.

De acordo com Cohn (1998) em uma reunião de Planejamento do Sprint estão presentes os membros do time, o dono do produto e, podem participar também, outros interessados.

Durante a reunião, o dono do produto descreve o qual a funcionalidade do software é mais importante e o time faz as perguntas necessárias para entender mais detalhes sobre a mesma.

O dono do produto não precisa descrever cada cartão do Backlog do Produto. Uma sugestão é que ele venha para reunião preparado para falar de uma quantidade de cartões que considera importante para serem abordados nos próximos dois Sprints, ou os 10 cartões que considera mais importante. (COHN, 1998b). Como resultado deve-se obter: O objetivo do Sprint e o Backlog do Sprint.

O objetivo do Sprint é uma frase, criada pelo time e o dono do produto, simbolizando o que o time deve atingir neste Sprint. Esse objetivo será, mais tarde, utilizado na reunião de Retrospectiva.

O Backlog Do Sprint é uma lista de funcionalidades do software que será desenvolvida no Sprint mais uma lista de tarefas necessárias para entregar essas funcionalidades(COHN, 1998b). É importante que o time diga quanto trabalho ele pode se comprometer a fazer no Sprint e que se faça estimativa sobre as tarefas que serão abordadas.

Em uma discussão criada no site Arstechnica[+] indica que é recomendado que a reunião dure de 1 a 2 horas para um Sprint de uma semana.

Kniberg (2007) diz em seu livro que um planejamento exato da reunião deve ser realizado dispondo as horas disponíveis e o que será feito dentro de cada hora(s) como forma de prever a duração exata da reunião. Caso a reunião atrase, é sugerido que dê a todos duas opções: Ou é marcada outra reunião, ou a reunião é terminada no tempo sugerido. A probabilidade é que a segunda opção seja sempre escolhida.

4.3.4 Programação Pareada

Programação Pareada ou Programação Em Par tem origem no XP e é constituída de dois desenvolvedores, sentados lado a lado, na mesma máquina trabalhando em um mesmo código. Essa prática garante que todo o código em produção seja revisado por pelo menos um desenvolvedor e isso resulta em melhor qualidade. (JEFFRIES, 1999)

Programação em par é um diálogo entre duas pessoas desenvolvendo juntas e entendo como programar melhor. Ela permite que o conhecimento seja compartilhado e

---

[+] http://arstechnica.com/information-technology/2013/10/how-can-you-make-sprint-planning-fun/

ambos os pares aprendam juntos. Mesmo quando uma pessoa é menos experiente que a outra, a tendência é que a situação melhore com o tempo e, ao final, ambas se tornem mais experientes no negócio (BECK, 2000).

Jeffries (1999) afirma que pesquisas indicam que usar programação em par produz códigos melhores quase no mesmo tempo que uma pessoa desenvolvendo sozinha.

Beck (2000) afirma que, mesmo que essa prática demore mais tempo para produzir código, a qualidade é superior pois enquanto uma pessoa digita, a outra está pensando sobre o problema em um âmbito mais estratégico, fazendo perguntas do tipo: Onde essa linha de pensamento está indo? Essa é a melhor estratégia?

Ele também afirma que, quando se está sozinho, a chance de ignorar boas práticas como refatoração, testes, por exemplo, é fácil, especialmente sobre pressão. Com o seu par assistindo, fica mais difícil de ignorar detalhes pois ambos os desenvolvedores precisam entrar em um consenso.

Não há uma regra específica que diz quanto tempo as pessoas podem trabalhar juntas e com quanta frequência os pares podem ser trocados. A ideia é que isso ocorra de forma dinâmica com quem se tem mais afinidade e dependendo dos conhecimentos de cada um.

4.3.5 Considerações

As práticas mencionadas muito contribuem para o Kanban. As reuniões diárias foram eficazes para aprender sobre como utilizar o Kanban, na fase inicial de implementação, ajudar a manter o foco nos cartões sendo trabalhados e permitindo a detecção de gargalos no Kanban.

As reuniões de retrospectiva permitem que sejam analisados os cartões terminados e que novos cartões sejam criados como forma de corrigir ou prevenir problemas que foram ressaltados durante a reunião. Elas também possibilitam que sejam identificados problemas no Kanban ou problemas na movimentação ou criação dos cartões.

A programação em par é aplicada quando um bloqueio ou gargalo é detectado no Kanban e permite que este seja resolvido mais rapidamente.

A reunião de planejamento ajuda na priorização de tarefas do Backlog do Kanban.

Detalhes da implementação dessas práticas serão abordadas no capítulo 6.

5 INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY

Para construção do Kanban foi necessário, primeiramente, definir as atividades do setor de sistemas da empresa. Posteriormente, as colunas do Kanban foram criadas com base em cada estágio das atividades. Para definição de atividades foi utilizado parte do Information Technology Infrastructure Library (ITIL), um framework em utilização a bastante tempo no mercado de tecnologia da informação (TI).

Neste capítulo será dada uma visão geral sobre o ITIL e alguns processos interessantes a esse trabalho serão descritos.

5.1 Visão Geral

O ITIL é um framework que descreve as melhores práticas para gerenciar um serviço. O objetivo desse framework é melhorar o serviço entregue para o negócio e o consumidor.

O desafio dos administradores de sistema é trabalhar em parceria com a empresa ou negócio de forma a entregar serviços de qualidade. Esses serviços podem estar dentro de um sistema (ou servidor) e é papel do administrador mantê-los da melhor forma possível.

Se um administrador mantém os serviços de uma empresa nada mais lógico que definir o que é um serviço: "Um serviço é um meio facilitador de entregar um objetivo para os consumidores sem agregar riscos ou custos." (CARTLIDGE et al., 2007). Como exemplo, podemos citar como objetivo quando um vendedor deseja gastar mais tempo com seus consumidores e como facilitador um serviço que permite que ele se relacione remotamente com estes (CARTLIDGE et al., 2007).

Para entender o que um serviço precisa prover, para garantir que ele realmente facilite a vida do consumidor e reflita seus objetivos, é necessário entender o valor do serviço e gerenciar seus custos e risco. Para isso, o ITIL criou um ciclo de vida do serviço que tem como objetivo entendê-lo. Esse ciclo não se aplica só a serviços, mas também a processos e a componentes de infraestrutura.

O ciclo de vida de um serviço inclui as seguintes fases: Estratégias, Projeto, Transição, Operação e Melhoria Contínua conforme a Figura 5.1.1

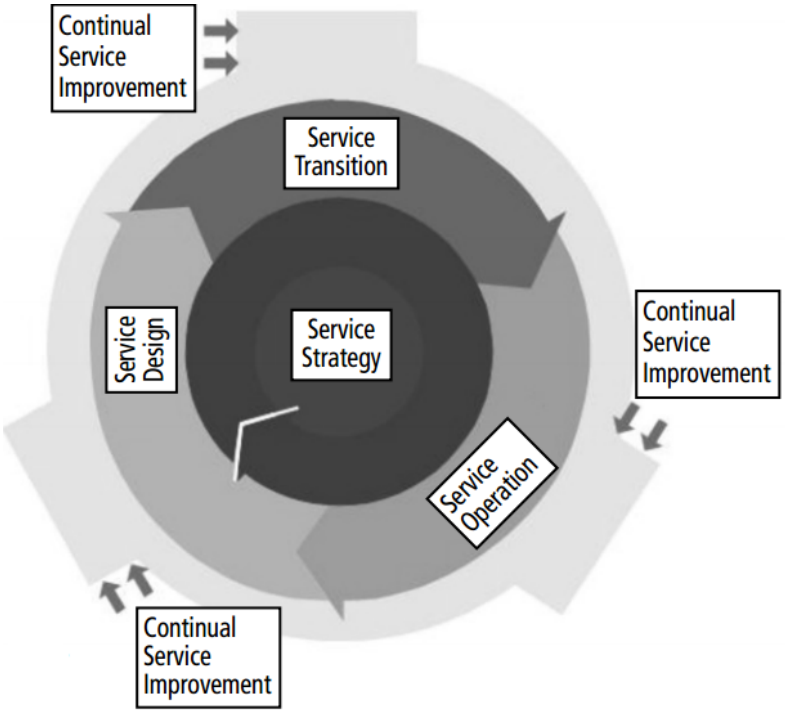


Figura 5.1.1: Ciclo de Vida do Serviço (CARTLIDGE et al., 2007)

Na Figura 5.1.1 tem-se a Estratégia de Serviço e Design de serviço cujo objetivo é auxiliar na concepção do mesmo. A Transição de Serviço representa o momento em que ele passa para produção e a Operação de Serviço mantém o serviço. Posteriormente temos a fase de Melhoria Contínua do Serviço que tem como objetivo melhorá-lo constantemente de forma a melhorar seu valor para o negócio.

A versão mais atual do ITIL descreve o ciclo de vida de um serviço em cinco livros. No presente trabalho, abordaremos o livro Operação de Serviço (Service Operation), o qual trata das principais atividades do dia-a-dia de um administrador de sistemas que é justamente manter um serviço. No setor de S&R manter serviços e dar suporte ao serviço são as principais tarefas do time e o Kanban tem de estar preparando para, principalmente, cuidar dessa fase.

5.2 Operação de Serviço

O objetivo dessa fase é entregar o serviço para usuários e gerenciar as aplicações, tecnologias e infraestrutura para suportar a entrega desses serviços. De acordo com (CARTLIDGE et al., 2007), é somente nessa fase que o serviço irá entregar valor para o negócio. A entrega de valor é uma das características mais marcantes dos métodos ágeis (BECK et al., 2001), razão pela qual selecionamos alguns processos do ITIL da fase de operação para fundamentar a realização das atividades do Setor S&R neste trabalho:

- Processo de Gestão de Eventos: Gerencia eventos que ocorrem dentro de um sistema. Um evento pode ser desde demandas de usuários até problemas mais complexos, como a falha de um link. É nessa fase que um evento será classificado e priorizado quanto a sua severidade e tipo;

- Processo de Gestão de Incidentes: Incidentes são interrupções não planejadas e este processo gerencia os mesmos;

- Processo de Gestão de Problemas: Gerencia problemas que são identificados durante a resolução de incidentes.

5.2.1 Processo de Gestão de Eventos

"Um evento é uma mudança de estado que tem significância para o gerenciamento do serviço" (CARTLIDGE et al., 2007, p.29). Um evento pode significar que algo não está funcionando corretamente, levando a um incidente. Eventos também podem indicar atividades normais ou a necessidade de uma intervenção de rotina, como trocar uma fita de backup.

Abaixo podemos observar um fluxograma da gestão de eventos:

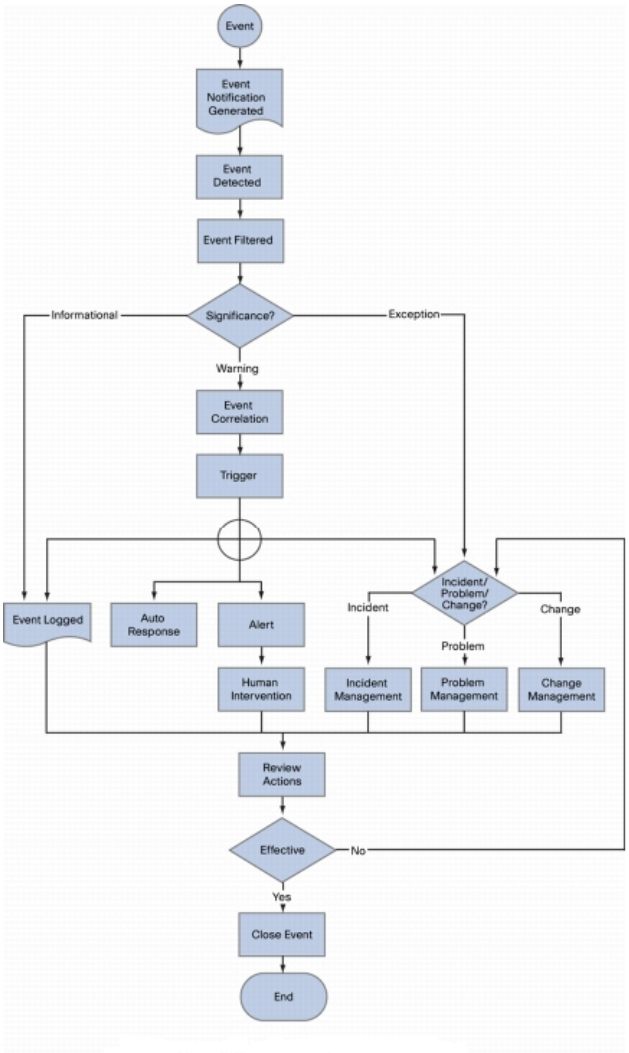


Figura 5.2.1.1: Gestão de Eventos baseado no livro de Van Bon (2007)

Como pode ser observado na Figura 5.2.1.1, um evento é gerado por um administrador de sistemas ou por alguma ferramenta de monitoramento. Logo após é feita uma análise de severidade (ou significância) do evento e este é caracterizado como informativo, excepcional (crítico) ou alerta.

Um evento informativo é, por exemplo, um usuário entrando em um sistema ao que ele tem acesso. Neste caso só é preciso guardar esta informação em algum lugar para uma futura auditoria.

Um alerta (warning) é, por exemplo, quando o sistema de monitoramento detecta que um disco está quase cheio e que medidas deverão ser tomadas. Um alerta é um estado anterior ao estado crítico, e ele antecipa e realça situações que podem vir a ser tornar grandes problemas se não forem tratados de antemão.

Um evento crítico significa que um serviço está se comportando de forma anormal e portanto está impactando no negócio e não cumprindo o Service Level Agreement (SLA). Terão de ser tomadas ações usando os processos de gestão de incidentes, problema ou mudanças como forma de resolver o evento.

Depois que um evento é caracterizado quanto a sua severidade ele é definido como um Incidente, Problema ou Mudança. Esse trabalho não tratará a Gestão de Mudanças.

5.2.2 Processo de Gestão de Incidentes

"Um incidente é uma interrupção não planejada ou a redução de qualidade em um serviço de TI. Falha de configuração de um item que ainda não impactou o serviço também é um incidente." (CARTLIDGE et al., 2007). O objetivo do processo de gestão de incidentes é restaurar um serviço, o mais rápido possível, afim de minimizar o impacto para o negócio.

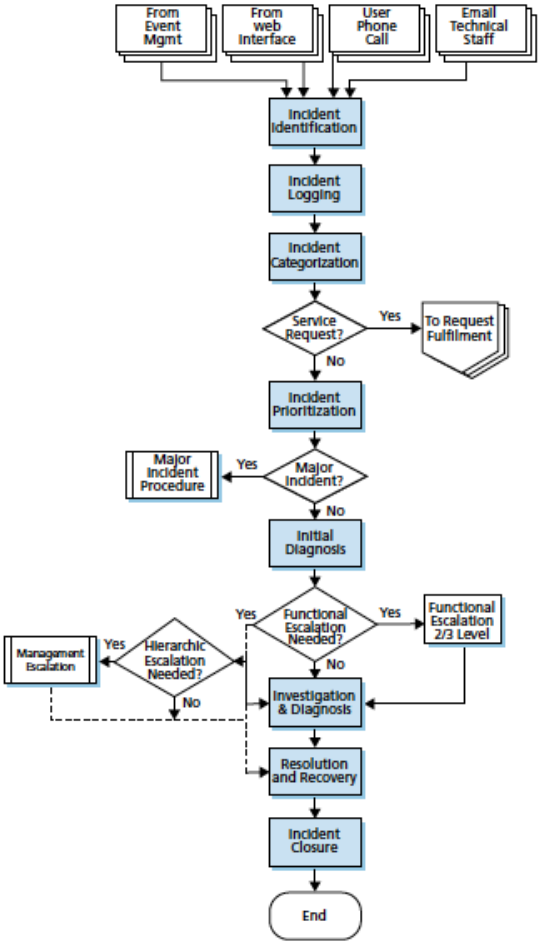


Figura 5.2.2.1: Gestão de Incidentes (VAN BON et al., 2007)

Incidentes são detectados pela Gestão de Eventos ou através de solicitações de usuário de diversas formas.

Depois que um incidente é caracterizado, é feita uma priorização para medir o impacto do incidente para o serviço. Depois é feito um diagnóstico inicial para ver se o incidente já é conhecido, ou seja, se já existem procedimentos de resolução ou atendimento. Caso não exista, o incidente terá que ser investigado mais a fundo.

Depois que o incidente foi investigado, diagnosticado e sua resolução foi testada, o suporte deve garantir que o usuário está satisfeito antes de fechar o incidente (CARTLIDGE et al., 2007). Uma ferramenta que gerencia incidentes é fundamental para gravação e gerenciamento destes.

Durante o processo de Gestão de Incidentes, um problema pode ser detectado e terá de ser resolvido pelo processo de Gestão de Problemas. Caso uma mudança seja necessária durante a resolução de um incidente, essa deverá ser abordada pela Gestão de Mudanças.

5.2.3 Processo de Gestão de Problemas

O objetivo desse processo é prevenir problemas decorrentes de incidentes, eliminar incidentes recorrentes e minimizar o impacto de incidentes que não podem ser

evitados (CARTLIDGE et al., 2007). De acordo com Herold (2007), geralmente um problema é uma situação indesejável que pode ser tornar a causa raiz de um incidente.

A gestão de problemas inclui diagnosticar as causas de um incidente, determinando e garantindo que sua resolução seja implementada. Ela também mantém informações sobre um problema e as formas de resolvê-lo.

Problemas são categorizados de maneira similar a eventos, mas o objetivo aqui é entender as causas, documentar soluções e solicitar mudanças de forma que o problema seja permanentemente resolvido.

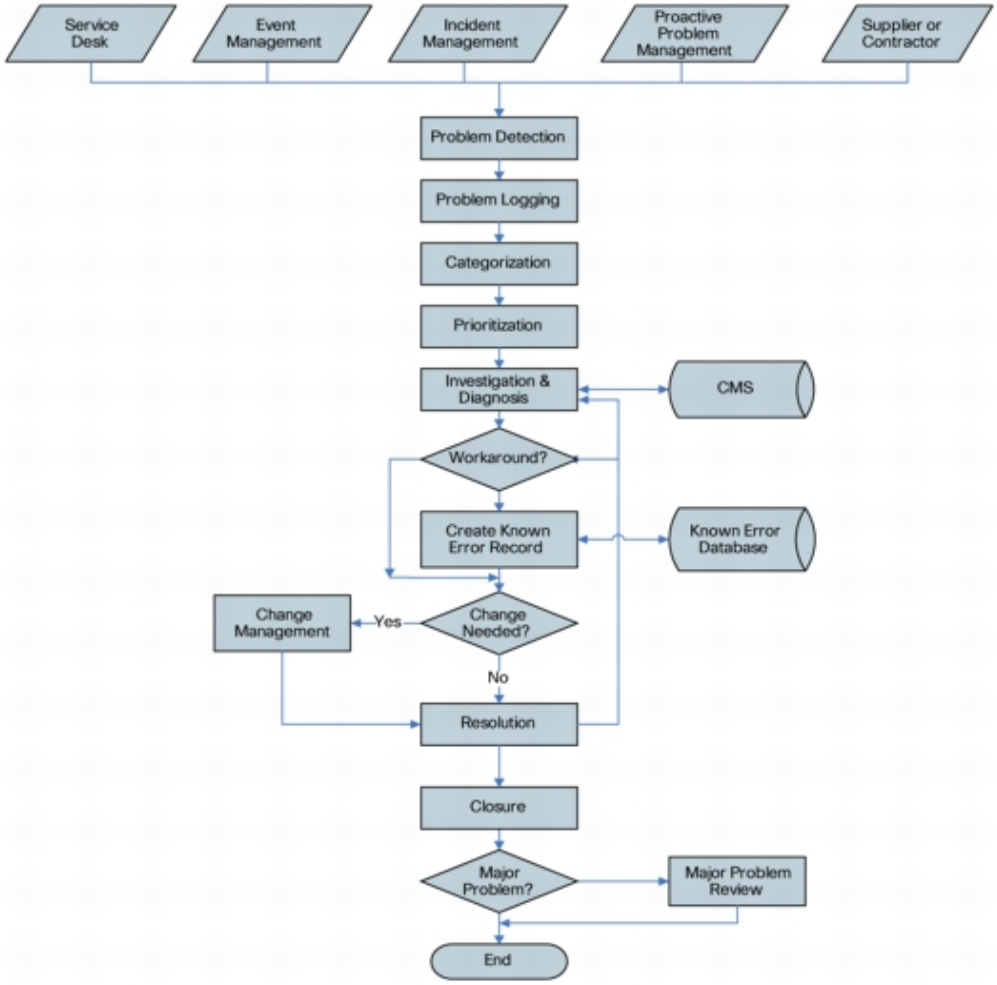


Figura 5.2.3.1: Gestão de Problemas (VAN BON et al., 2007)

De acordo com Van Bon (2007), depois que um problema é diagnosticado, é verificado se o problema é originado de alguma configuração inconsistente através de um Sistema de Gerenciamento de Configurações.

É realizada uma solução temporária como forma de resolver o problema o mais rápido possível. O erro gerado pelo problema é gravado em alguma base de dados para que se possa recuperar do problema mais rapidamente da próxima vez. Todas as soluções são documentadas para melhorar a eficiência e efetividade não só da Gestão de Problemas como da Gestão de Incidentes.

Posteriormente é verificado se alguma mudança é necessária para resolver o problema permanentemente. Se sim, então o processo de Gestão de Mudança deve ser ativado.

Depois que uma resolução foi de fato aplicada o problema é encerrado. Se o problema for muito significativo, como por exemplo a queda de um link de internet, após a resolução do problema é feita uma revisão do problema e um estudo é realizado para que se possa verificar o que pode ser melhorado no processo de forma a resolver o problema de internet mais rápido da próxima vez.

5.3 Considerações

Apesar do S&R não utilizar esse vocabulário e essa categorização no dia a dia, podemos notar que as tarefas tratadas pelo time são passíveis de serem administradas através dos processos de gestão citados anteriormente.

A queda de um link de internet pode ser tratada, por exemplo, pela Gestão de Problemas. Seguindo o fluxo da Figura 5.2.3.1 podemos exemplificar resumidamente da seguinte forma:

1. Os efeitos da queda de um link são identificados pela Gestão de Eventos e este o classifica como Problema;

2. O problema é investigado e é observado que a falha no link se deve a um problema no provedor de internet. Como "workaround" a equipe resolve mudar o tráfego da empresa para outro provedor em funcionamento;

3. O problema é enfim resolvido pelo provedor de internet.

Da mesma forma podemos utilizar a Gestão de Incidentes para resolver o problema de um usuário:

- Usuário informa que não consegue acesso ao sistema. A Gestão de Eventos detecta o incidente;

- Esse incidente é então priorizado, investigado e resolvido.

O ITIL propõe um fluxo para lidar com essas situações diárias de S&R. Kanban é um forma de visualizar esse fluxo de forma mais amigável. Logo surgiu a idéia de usar o ITIL na construção do Kanban.

A caracterização de Eventos oriundas do Processo de Gestão de Eventos chama atenção pois, de acordo com cada tipo de evento, são especificadas etapas de trabalho que por sua vez serão as colunas do Kanban.

A Gestão de Incidentes possui as seguintes etapas que podem vir a ser usadas no Kanban: priorização, diagnóstico inicial, investigação, resolução e fechamento. Na Gestão de Problema, por sua vez, possui as seguintes etapas e que estas poderiam ser utilizadas no Kanban: priorização, investigação, investigação, aplicação de solução temporária, resolução, fechamento e revisão.

Será visto no próximo capítulo como essas etapas auxiliaram na construção do Kanban.

6 CONSTRUÇÃO E ADOÇÃO DO KANBAN

Neste capítulo será descrito como se deu a construção e organização do Kanban usando alguns processos do ITIL e como foi seu processo de adoção juntamente com o uso de práticas ágeis.

6.1 Organização do Kanban

No capítulo anterior foi observado que os processos de Gestão de Incidentes e Problemas do ITIL possuem algumas etapas de trabalho. Já que o time de S&R trata tanto problema como incidentes, as etapas em comum foram identificadas: priorização, investigação, resolução e fechamento.

O Kanban, ilustrado na Tabela 6.1.1, possui estágios de processamento para todas estas etapas, mais precisamente:

A. Priorização: Long Term Backlog e Short Term Backlog;

B. Investigação: Investigation e Buffer Working;

C. Resolução: Working e Follow Up;

D. Done.

Esses estágios estão organizados em dois tipos de raias chamadas de Planned e Unplanned. Planned representa o trabalho planejado, ou seja, demandas não urgentes de clientes e tarefas rotineiras do setor. Unplanned representa os incidentes, ou seja, o trabalho não planejado, constituído de tarefas urgentes oriundas de requisições de clientes, ou falhas de serviços, ou configurações vitais para o negócio. Estão nesta raia tarefas críticas que devem ser tratadas de imediato.

Na Tabela 6.1.1 está um esquema do Kanban resultante de algumas das etapas mencionadas. As etapas em branco nunca são utilizadas.

	LongTerm Backlog	ShortTerm Backlog	Investigation	Buffer Working	Working	Follow-Up	Done
UNPLANNED							
PLANNED							

Tabela 6.1.1: Kanban do setor de S&R

Antes de explicar sobre o Kanban é importante que seja compreendido que este não foi criado de uma hora para outra. Para chegar ao mural representado na Tabela 6.1.1 foi preciso modificar o mesmo quatro vezes até que o fluxo de trabalho do time esteja bem representado. Neste trabalho não serão demonstrados os Kanban anteriores, mas é importante ter a noção de que criar um fluxo que represente o trabalho diário requer uma reflexão profunda sobre o mesmo.

6.1.1 Priorização

No ITIL podemos notar que a priorização de tarefas aparece com frequência nos processos de gestão. Para lidar com a priorização no Kanban definiu-se:

- As colunas de Short Term e Long Term Backlog;
- A posição do cartão dentro da coluna de Short Term Backlog e Buffer Working;
- Raias Planned e Unplanned;
- Cor: Os cartões podem ser coloridos conforme sua prioridade.

Os cartões mais prioritários geralmente ficam no topo da coluna Backlog. Entretanto, quando o time de S&R começou a implementar o Kanban, ele já possuía uma quantidade enorme de cartões por fazer, o que tornou difícil realizar a priorização utilizando somente seu posicionamento ou cor como artifício.

Por essa razão a coluna de Backlog foi separada em duas: Backlog de Longo Prazo (Long Term Backlog) e Backlog de Curto Prazo (Short Term Backlog). Na

primeira são colocados os cartões com menor prioridade e na segunda os de maior prioridade e as solicitações de clientes, pois estas devem ser atendidas em um tempo razoável (curto) para o mesmo. Na seção 6.2 será abordado como se dá o processo de priorização de cartões dentro das colunas de Backlog.

As raias Planned (Planejado) e Unplanned (Não planejado) podem representar prioridade. Quando um cartão é criado e executado no tempo planejado pelo time, seu fluxo é realizado na raia Planned passando por todos os estágios necessários. Esse cartão não tem prioridade urgente pois não impacta o negócio caso não seja tratado. Já os incidentes como, por exemplo, a queda de um link de internet ou uma solicitação de usuário que o impede de trabalhar está representado por um cartão pela raia Unplanned. Quando surge um cartão em Unplanned, este é criado e colocado direto em Working sem passar pelos estágios anteriores e desconsiderando o WIP, logo sua resolução é de altíssima prioridade.

6.1.2 Investigação

Na Tabela 6.1.1 também podemos notar a presença da etapa de investigação (Investigation). Ao movimentar um cartão do Backlog, se sua solução é conhecida ele é então colocado em Trabalhando (Working), caso contrário é colocado em Investigation.

Na etapa de Investigation, um cartão pode ser divido em vários. Por exemplo, quando um usuário solicita a criação de um servidor este é colocado em investigação. Após uma investigação inicial, este cartão é divido em tarefas menores. Com isso, é possível que mais de uma pessoa trabalhe na criação de um servidor ao pegar tarefas independentes que irão resultar na criação do mesmo.

Os cartões gerados de um cartão em Investigation vão para o Buffer Working. É importante destacar que quando um funcionário do S&R está livre, ele irá primeiro pegar um cartão do Buffer Working antes do Backlog e por isso a priorização da coluna de Buffer Working também é importante.

A divisão de um cartão em Investigation pode ser representada pela Figura 6.1.2.1.

O time de S&R criou uma maneira de sinalizar os cartões que são originados de um mesmo cartão ao colocar uma palavra chave entre colchetes que simboliza o cartão original como o exemplo da Figura 6.1.2.1:

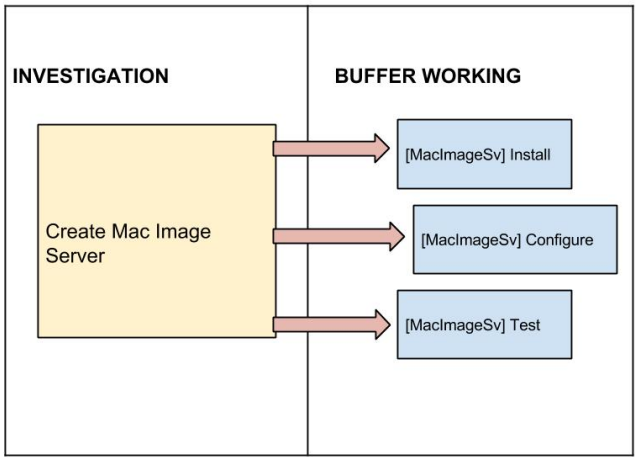


Figura 6.1.2.2: Divisão de cartões no Kanban

Na Figura 6.1.2.1 é possível ver que, durante a etapa de Investigation, o cartão de instalação de um servidor foi divido em três: Instalação, Configuração e Teste. Os três novos cartões possuem uma sinalização entre colchetes dizendo que os mesmos fazem parte da criação de um mesmo servidor. O cartão original (em amarelo) desaparece ao ser dividido em vários.

A coluna de Buffer Working também serve para detectar gargalos. Se um cartão está nessa coluna por muito tempo significa que existem tarefas mais urgentes sendo requisitadas, como por exemplo, cartões em Unplanned.

6.1.3 Resolução

A coluna Working representa o trabalho sendo realizado ou a tentativa de resolução de uma solicitação. Logo após a coluna Working, pode ser observada outra coluna chamada de Follow-Up. Esta representa um cartão que está sendo tratado por terceiros.

A coluna de Follow-Up representa tarefas que estão sendo realizadas por empresas terceiras que auxiliam o S&R. Neste estágio o administrador de sistema precisa checar periodicamente com a empresa terceira qual a evolução do cartão. Por exemplo, quando o setor precisa comprar equipamentos de rede, cria-se um cartão para comprar os equipamentos e, após a compra, o cartão é movido para Follow-Up até a chegada dos mesmos.

As vezes, após o Follow-Up, é necessário que o time de S&R trabalhe no cartão. Se o trabalho, após o Follow-Up, for de longa duração é preciso criar outro cartão no Kanban, em Unplanned ou Backlog dependendo da urgência, e mover o existente para Done. Se o trabalho for de curta duração pode-se terminar a tarefa e mover o cartão de Follow-Up para Done.

6.1.4 Fechamento

A última coluna é reservada para cartões terminados. Essa coluna é importante para realizar reuniões de Retrospectiva conforme será visto na seção 6.2.

6.2 Adoção do Kanban

Ao observar o Kanban, é possível ter uma visão intuitiva do fluxo de trabalho. Será visto, em seguida, situações que podem ocorrer durante a utilização do mesmo.

6.2.1 Propriedades dos Cartões

Cada novo cartão é caracterizado pelas seguintes informações:

- Um nome;
- Uma descrição breve com informações como, por exemplo, escritório o qual a tarefa pertence;
- Data de início;
- Dono do cartão: Nome da pessoa que irá trabalhar no cartão. Essa propriedade só é preenchida quando o cartão sai do Short Term Backlog ou do Buffer Working;

- Uma cor do cartão em si usada para representar os tipos de tarefas abaixo. Essas caracterizações pertencem a ThinkAgile, mas nada impede que sejam criados tipos de tarefas conforme a necessidade:
    - Projeto: Cartões que representam tarefas mais complexas como por exemplo a abertura de um novo escritório ou um servidor que será utilizado em grande escala;
    - BAU (Business As Usual): Nomenclatura usada pelo S&R da ThinnkAgile. Esses cartões representam solicitações de usuários incluindo suporte;
    - Aperfeiçoamento: Cartões que representam tarefas que propõem melhorias na infraestrutura existente.

Durante a realização de uma tarefa, em Working, pode ser que aconteça uma situação de bloqueio. Por exemplo, supondo que um usuário solicitou a instalação de um aparelho de telefonia VoIP e foi descoberto, durante o atendimento, que a fiação da mesa está danificada, o cartão de instalação do telefone é bloqueado e é criado um novo cartão para tratar do problema de fiação.

Outra situação de bloqueio é quando um funcionário de S&R está tratando uma solicitação e ocorre uma interrupção que o faz trabalhar em um cartão em Unplanned e, consequente, bloquear o cartão (Planned) em que estava trabalhando.

Para representar tarefas bloqueadas foi pensado em colocar um outro cartão vermelho, sem informações, sobre o cartão.

6.2.2 Ferramenta de Visualização

Como o time de S&R é separado geograficamente, decidiu-se montar o Kanban utilizando uma ferramenta online chamada Mingle[+]. Uma imagem do Kanban, na ferramenta, está representada nos Apêndices A e B. Estes representam duas partes de um mesmo Kanban e representam a realidade do setor de S&R da ThinkAgile.

O Mingle foi escolhido como ferramenta pois ela possui funcionalidades como a geração de estatísticas, várias maneiras de customização e visualização de diferentes perspectivas sobre os cartões e, principalmente, foi a única ferramenta que possibilitou o uso de raias para o Kanban.

Infelizmente não foi possível utilizar a funcionalidade de estatísticas do Mingle até a realização deste trabalho. Os gráficos deste trabalhos foram gerados manualmente conforme será detalhado no capítulo 7.

Os cartões relacionados a suporte possuem o nome do cliente de S&R no título e este pode acessar o Kanban a qualquer momento via internet para saber o andamento de suas solicitações. Todas as colunas do Kanban são mostradas para os clientes com exceção da coluna de Long Term Backlog, pois esta não possui informações úteis ao cliente já que sua solicitação sempre estará na coluna Short Term Backlog ou, em andamento, nas outras colunas do Kanban.

[+] http://www.thoughtworks.com/products/mingle-agile-project-management

O Kanban também está exposto em uma TV de 46” na frente da mesa do setor e serve de modelo para outros setores da empresa. Vale a pena destacar que foi mais efetivo colocá-lo na TV do que disponibilizar o link do site, pois este é facilmente esquecido pelos usuários.

No Mingle é possível fazer vários Kanbans diferentes onde só é alterada a cor dos cartões de forma a criar vários cenários. É possível colorir os cartões de acordo com vários cenários:

- Bloqueio: Representado nos Apêndices A e B;
- Tipos de tarefa conforme citado anteriormente;
- Por escritório: Se o time de S&R atender filiais diferentes, cada uma possui uma cor diferente;
- Por prioridade: Crítico (alta prioridade), Essencial (média prioridade) e Bom de se ter (baixa prioridade);
- Por complexidade: pequena, média ou grande.

Para simbolizar o bloqueio foi criado um esquema de cores nos cartões: quando um cartão está vermelho significa que aquele cartão foi bloqueado e quando está verde significa que está desbloqueado. Um cartão é bloqueado ao ser interrompido por um incidente em Unplanned ou por algum outro funcionário da ThinkAgile, fora do setor de S&R, que precisa dar continuidade a tarefa. Em ambas a situações não faz sentido colocá-lo em FollowUp, pois esta coluna é somente para tarefas que dependem de empresas terceiras.

A coloração de cartões permitiu ao time ter uma visão geral de quais os tipos de tarefas mais comuns, qual escritório demanda mais tarefas e etc.

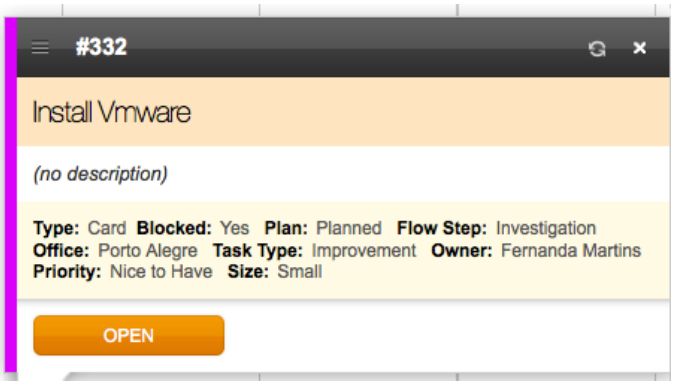


Figura 6.2.2.1: Propriedades de um cartão

A Figura 6.2.2.1 mostra as propriedades de um cartão na ferramenta. Os cartões possuem as mesma propriedades descritas anteriormente e na Figura 6.2.2.1 o cartão está colorido de rosa que representa que cartão é do tipo Aperfeiçoamento.

6.2.3 Escolha da prioridade dos cartões

Qualquer administrador do S&R pode adicionar cartões ao Kanban. Quando um cartão, no dia-a-dia, é adicionado ao mural, este é primeiramente alocado ao Short Term

Backlog. Dentro da coluna, o funcionário escolhe sua prioridade colocando o cartão no topo se é prioridade ou abaixo se não é.

A ideia é que isso acontecesse até a reunião de Planejamento do Backlog quando seria definido pelo time qual a real prioridade do cartão e se este deveria ser movido para Long Term Backlog.

Um proposta foi feita na seção 6.3.4 de como a reunião deve ser realizada, entretanto, até a realização deste trabalho, não foi possível colocar esta reunião em prática. Quem prioriza os cartões é o chefe do time de S&R. Depois que ele faz a priorização, ele solicita ao time que revisem o que foi feito e opinem acerca da priorização caso necessário.

Durante a realização deste trabalho foi percebido que, ao mover o cartão de Short Term Backlog para Long Term Backlog, este é movido no fluxo contrário ao do Kanban (da direita para a esquerda) o que não pode acontecer. A equipe de S&R irá arrumar essa situação mas até a finalização deste trabalho não foi possível estudar as implicações disso.

6.2.4 Gestão de Solicitações de Clientes

Ao atender outros funcionários da ThinkAgile o time de S&R precisou responder as seguintes perguntas:

- Quais as solicitações seriam consideradas urgentes tendo de ser atendidas prontamente e, portanto, passíveis de entrar na raia Unplanned?
- Como evitar que o administrador de sistemas perca o foco ao atender solicitações de suporte não importando o quão demoradas elas sejam?
- Se o administrador de sistemas atendesse um cliente por um período pequeno de, por exemplo, 3 minutos, prestando suporte a ele, isso seria demonstrado no Kanban?

Para responder essas perguntas foi criado o processo descrito pela Figura 6.2.4.1:

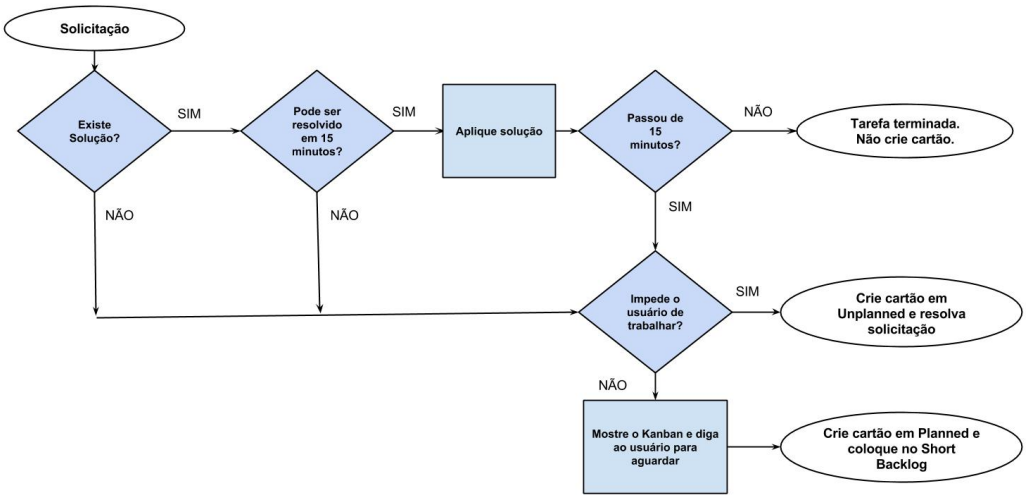


Figura 6.2.4.1: Gestão de Solicitações

A Figura 6.2.4.1 demostra como essas solicitações devem ser gerenciadas pelos administradores do S&R. Ficou combinado que o mesmo iria tratar interrupções da seguinte forma: O suporte iria analisar a requisição do usuário, caso saiba uma solução, iria tentar aplicá-la durante 15 minutos. Se desse certo, não colocaria um cartão no Kanban, senão iria criar um cartão no Short Tem Backlog no raia Planned e solicitaria que o usuário aguarde atendimento.

Caso o usuário insista em sua solicitação, o administrador mostraria o Kanban para o mesmo e explicaria que, assim que terminasse seu cartão atual, iria atender ao usuário. Mostrar o Kanban para o usuário tem sido muito efetivo no setor de S&R e isso deu mais confiança para o time negar demandas. Em alguns casos, o suporte nem mesmo atende o usuário por 15 minutos pois este consegue, através da experiência, saber de antemão que a solicitação ultrapassará o tempo.

O usuário, por sua vez, não se sente mal por ter sua solicitação negada pois compreende, ao olhar o Kanban, que o suporte tem outras tarefas a desempenhar. Como, na maioria das vezes, o suporte tenta resolver o problema do usuário, ao ter sua solicitação negada após 15 minutos, o usuário entende que o suporte tentou resolver o problema, não deu certo, e portanto seu problema é maior do que se pensava e terá de ir para o Short Term Backlog.

Obviamente espera-se que cada usuário tenha as condições necessárias para trabalhar. Caso problema em questão impeça o usuário de desempenhar suas funções por completo, o suporte criará um cartão em Unplanned e se dedicaria a resolver o problema do usuário quanto tempo fosse necessário.

O fluxograma da Figura 6.2.4.1 pode parecer pouco efetivo pois pode-se imaginar que o S&R continua sofrendo interrupções de 15 minutos. Entretanto, o suporte, ao saber que tem 15 minutos para resolver o problema, analisa a situação e já responde ao usuário de sua decisão. Ao estabelecer limites, isso evita que o suporte perca muito tempo atendendo uma solicitação de usuário e perca o foco.

6.2.5 Atomicidade de um cartão

É muito importante que os cartões sejam criados de forma equilibrada onde cada um representa apenas uma ação a ser tomada. Dessa forma é possível evitar situações onde um cartão representa uma tarefa grande (complexa) a ser realizada ou, caso contrário, uma tarefa muito pequena.

Um dos grandes benefícios do Kanban é trazer ao seu usuário a noção de que seu trabalho está em constante evolução. Criar um cartão que representa uma tarefa muito complexa anula essa noção e traz frustação à pessoa que está trabalhando no mesmo, pois não se consegue ter a sensação de que o cartão está evoluindo. Caso seja necessário criar um cartão que representa uma tarefa mais complexa, deve-se dividir o cartão na etapa de Investigation.

Caso o cartão seja muito pequeno, será observado que ele irá andar rapidamente pelo mural. Embora isso pareça bom, pode-se perder muito mais tempo para movimentar o cartão no Kanban do que trabalhando nele.

Conforme a seção 6.2.4, foi observado que um cartão deve ser criado se sua resolução demorar mais que 15 minutos. Também foi visto que, quando não se sabe sobre a resolução de um cartão, este deve ir para a coluna Investigation. Após a realização de uma investigação para saber como o cartão pode ser resolvido, a seguinte

pergunta deve ser feita para saber se o mesmo deve ser dividido: Levarei mais de um dia para terminar o cartão?

Caso a resposta a essa pergunta seja positiva, pode-se concluir que o cartão não é pequeno e, portanto, pode ser dividido. Caso a resposta seja negativa, o cartão não precisa ser dividido.

Quando se sabe sobre a resolução de um cartão, antes de cria-lo, é preciso pensar na mesma pergunta acima e dividir os cartões no Backlog. Caso seja necessário uma reflexão profunda neste momento, é aconselhável que crie apenas um cartão e o coloque em Investigation.

As perguntas acima são baseadas apenas na experiência de trabalho do setor de S&R da ThinkAgile, mas se mostraram muito úteis. Outro fator que ajudou na decisão foi discuti-lo durante as reuniões diárias, como será visto nas próximas páginas.

A importância de criar cartões com um tamanho razoável também pode ser observada no estágio de Follow-Up: caso o cartão seja muito complexo, a probabilidade do time de S&R ter que trabalhar muito tempo no cartão, antes de move-lo para Done, é muito maior.

6.2.6 WIP

No Kanban demonstrado, as colunas que representam trabalho sendo, efetivamente, realizado são: Investigation, Working e Follow-Up.

O estágio de Follow-Up demanda um esforço menor ao trabalhar no cartão já que é necessário apenas fazer um acompanhamento no mesmo.

O Investigation e Working são estágios onde o esforço é maior. Como nem todos os cartões passam por Investigation, o S&R da ThinkAgile resolveu simplificar e considerar um limite de WIP para ambos os estágios.

Dois limites, baseados no esforço de trabalho foram, então, definidos pelo time de S&R da ThinkAgile:

- Work Wip: É o limite máximo de cartões da coluna de Investigation somada com a coluna Working independente da raia em que estão.

- Follow Wip: É o limite máximo de cartões em Follow-Up independente da raia em que estão.

A estabelecer o Work Wip, foi levado em conta que cada membro do time deve ter seu foco exclusivamente no cartão que está sendo trabalhado. Portanto:

$$Work\ Wip = Número\ de\ Membros\ do\ Time$$

Como Work Wip é a soma das colunas Investigation e Working pode-se concluir:

$$NumMembrosTime \leq NumCartoesInvestigation + NumCartoesWorking$$

Para acompanhar um cartão não é necessário foco total como no caso anterior, logo foi decidido que cada membro do time poderia acompanhar duas tarefas em Follow Up, em um dia, portanto:

$$Follow\ Wip = 2\ x\ Número\ de\ Membros\ do\ Time$$

Pode-se então concluir:

$$2\ x\ NumMembrosTimes \leq Número\ de\ Cartões\ em\ FollowUp$$

Essas fórmulas foram baseadas na experiência empírica do time de S&R da ThinkAgile, mas tem-se obtido resultados positivos quando combinado às práticas ágeis. Também foi possível obter um passo constante e saudável para o time de S&R.

Foi observado que, às vezes, um mesmo administrador de S&R possuía 4 cartões em Follow-Up, logo foi definido que o mesmo não iria movimentar cartões para Working. Em pouquíssimos momentos ocorreu o contrário, ou seja, um mesmo administrador possuir dois cartões em Working. Neste caso, o time se preocupava em ajudar o administrador em questão.

6.3 Complementando o Kanban com práticas ágeis

Nesta seção será detalhado como as práticas ágeis auxiliaram na adoção do Kanban.

6.3.1 Reuniões Diárias

A reunião diária acontece de manhã e o primeiro passo é escolher uma pessoa para facilitar a reunião e este faz as seguintes perguntas:

- Alguma observação a fazer sobre os cartões na coluna de Follow-Up?
- Alguma observação a fazer sobre os cartões nas colunas Investigation ou Working?
- Alguma observação a fazer sobre outros cartões fora dessas colunas?
- Alguém vai pegar alguma tarefa? Se sim, qual tarefa acham que a pessoa em questão deve pegar?

Se há um número pequeno de cartões a ser comentado, o time tem a liberdade de falar sobre cada cartão nas colunas de Follow-Up, Investigation e Working. Se não, o time se foca em responder as duas primeiras perguntas e as outras são discutidas ao longo do dia.

A reunião diária é a prática ágil mais efetiva na manutenção do Kanban. Seus benefícios são:

1. Detectar gargalos ou bloqueios de forma rápida: Ao identificar que um administrador não consegue evoluir em um cartão, o time ou alguns

funcionários fazem outra reunião para tentar ajudar o mesmo a sair do bloqueio ou realizam a Programação em Par;

2. Aumento de comunicação dentro do time;

3. Compartilhamento de informações e avisos para o time. Esse compartilhamento se dá de forma rápida geralmente indicando aos membros do time onde eles devem ir para mais detalhes sobre a informação dada;

4. No início da implementação do Kanban, as reuniões também serviram para tirar dúvidas referentes à movimentação dos cartões.

As reuniões diárias auxiliaram na implementação do Kanban da seguinte forma:

1. Ao definir qual cartão pegar, ou sua movimentação dentro do Kanban, é possível falar rapidamente sobre a atomicidade do cartão conforme apontado 6.2.4. Esses comentários são feitos de forma rápida e caso membros do time estivessem em dúvida conversavam fora da reunião sobre o problema;

2. Com a reunião diária é possível controlar o WIP. Ao observar que um mesmo administrador está com muitos cartões impedindo, por exemplo, que outras pessoas peguem mais cartões, uma pessoa é acionada para ajudá-lo.

O time precisou fazer uma pequena modificação nas reuniões diárias: ao invés de durarem 15 minutos as reuniões passaram a durar 30. Isso aconteceu pois as vezes o time sofre com incidentes urgentes como a falha de um link ou de um servidor e quando isso ocorre mais cartões são abordados durante a reunião.

6.3.2 Programação Pareada

Foi difícil tornar essa prática regular no dia-a-dia pois o setor de S&R da ThinkAgile possui poucas pessoas. Por esse motivo o time teve a chance de utilizar a Programação Pareada somente em situações em que um cartão parecia não progredir.

Muitas vezes administradores de S&R possuem conhecimento específico em uma determinada tecnologia. Uma pessoa sem conhecimento em uma tecnologia pode vir a trabalhar em um cartão que depende daquela tecnologia e, consequentemente, ficar bloqueada. Ao identificar o bloqueio, durante as reuniões diárias, o especialista pode utilizar a programação pareada com esta pessoa.

Além do uso de programação em par em caso de bloqueio, pode-se utilizar esta prática quando existe um cartão de alta complexidade como, por exemplo, a criação de um servidor. A pessoa que está trabalhando neste cartão pode não ter todo o contexto necessário para levantar todos os requisitos na construção desse servidor e outra pessoa pode ajudá-la, se necessário.

Também pode-se parear quando duas pessoas possuem conhecimento no cartão, dessa forma é permitido que cada pessoa traga seu conhecimento e contexto sobre a tarefa.

Todas as situações acima possibilitam os seguintes benefícios:

- Ajuda a nivelar time, ou seja, a compartilhar conhecimento de forma que, ao final, duas pessoas tenham conhecimento sobre uma tecnologia específica;

- Ajuda a resolver ou desbloquear um cartão mais rapidamente;
- Melhora a comunicação e aproximação entre os membros do time envolvidos;
- Traz mais qualidade ao trabalho realizado pois duas pessoas puderam pensar sobre o mesmo problema de forma diferente.

A prática foi tão positiva para o ThinkAgile da S&R que foi construída uma estação de trabalho somente com o propósito de realizar a prática já que esta só é feita em situação de bloqueio. Quando uma pessoa está precisando de ajuda, ela conecta sua máquina na estação para trabalhar com seu colega.

Foi possível, também, realizar a programação pareada remotamente, (ou seja, de maneira remota) ao utilizar softwares específicos para isso como o Team Viewer e o Screen Hero[+]. Isso possibilita que times separados geograficamente fiquem mais próximos ao trabalharem sobre o mesmo problema.

6.3.3 Retrospectiva

O setor de S&R resolveu apenas usar o nome da reunião, porém criou a sua própria versão. Um encontro foi realizado para que os cartões que estão na coluna Done do Kanban sejam revisados. A idéia era que os cartões mais problemáticos deveriam ser discutidos e, se a discussão se prolongar para um cartão, é feita uma reunião em separado para o mesmo.

Um exemplo ocorreu quando um cabo foi desconectado de um equipamento o que ocasionou a falha de um link de internet. Ao rever essa tarefa na reunião de Retrospectiva, foi decidido que seria realizada outra reunião para refletir em como essa remoção do cabo poderia ter sido evitada. Durante a reunião sobre o cabo, foram criados vários outros cartões no Kanban o que resultou em uma melhoria na redundância do equipamento e cabos associados.

Embora o exemplo anterior tenha sido positivo, a reunião de Retrospectiva não deu certo pois existiam muitos cartões a serem discutidos. Como forma de resolver esse problema, algumas ideias da estrutura mencionada no capítulo 4 podem ser utilizadas como as fases de Gather Data e What To Do além de, é claro, diminuir o número de cartões abordados na reunião.

Não é possível realizar uma reunião exatamente com a estrutura mencionada no capítulo 4 devido a diversidade de tarefas tratadas pela time. Não é claro qual seria o objetivo para uma reunião.

A proposta é realizar a reunião da seguinte forma:

- Pense nos cartões: Todos iriam, durante a reunião, ter alguns segundos para pensar sobre os cartões da coluna Done. Aqueles que acham que algo pode ser melhorado levantam a mão e indicam quais cartões gostariam de discutir sem apresentar nenhum detalhe. Uma marquinha ou sinalização é, então, feita em cada cartão de acordo com o número de pessoas que levantou a mão.

[+] http://www.teamviewer.com/pt/index.aspx e http://screenhero.com/

- Selecione cartões: Depois de olhados os cartões marcados seriam separados. Se mais da metade do time reclamou sobre um cartão, para este seria feito uma reunião em separado, se não o time discutiria sobre o cartão na própria retrospectiva.

Infelizmente não foi possível realizar mais uma reunião utilizando esses moldes, porém mesmo após a primeira reunião houve um consenso de que discutir sobre os cartões é importante para compartilhar as lições aprendidas e evitar que problemas se repitam novamente.

6.3.4 Planejamento do Backlog

Conforme falado anteriormente não foi possível realizar a reunião de Planejamento do Backlog até o término deste trabalho.

A idéia é que a reunião de Planejamento de Backlog seja baseada na reunião de Planejamento do Sprint e seja feita da seguinte forma:

Como a maioria dos cartões de S&R são criados pelo próprio time, este poderia olhar os cartões de ambos os Backlogs: cada funcionário escolheria, antecipadamente, quais cartões acha mais importante e estes seriam discutidos na reunião. Como resultado da reunião seriam definidos os cartões a serem terminados em um período de 2 semanas que seriam movidos para o Short Term Backlog e o resto permaneceria no Long Term Backlog.

O time não se preocuparia em separar os cartões em tarefas menores pois muitas vezes o time não sabe como resolver o cartão e a separação do cartão em tarefas menores é realizada na etapa Investigation. Também não é preciso definir um objetivo já que a diversidade de cartões é muito grande e definir um objetivo comum para os cartões escolhidos seria muito complicado.

7 RESULTADOS DA THINKAGILE

Como foi dito no capítulo 2, o time de S&R da ThinkAgile não possuía uma maneira de contabilizar suas tarefas. Com a implementação do Kanban foi possível gerar dados sobre as tarefas do time. Nesta seção serão demonstrados alguns gráficos resultantes do Kanban.

7.1 Captura dos Dados

Como não foi possível utilizar a geração de dados automática do Mingle, os dados foram exportados para uma tabela que contém todos os cartões do Kanban no período de julho de 2013 até novembro de 2013 com o formato da Tabela 7.1.1:

Card	Name	Plan	To_Short_Backlog	To_investigation	.....	To_Done
#1	A	Unplanned				8/1/2013
#2	B	Planned	7/2/2013	7/4/2013		8/2/2013
#3	C	Planned	7/4/2013			

Tabela 7.1.1: Tabela parcial exportada do Kanban

A Tabela 7.1.1 é apenas uma simulação da tabela original.

Cada cartão está em uma linha da tabela. As colunas da tabela possuem o identificador do cartão, seu nome (Name), qual a sua raia (Plan) e todas as datas nas quais o cartão foi movido para as colunas do Kanban apresentado no capítulo 6. Algumas colunas foram suprimidas da tabela com a finalidade de simplificar a visualização. Algumas colunas omitidas são To_Buffer_Working, To_Working e To_FollowUp.

O cartão B, por exemplo, está na raia Planned e foi criado no dia 7/2/2013 e alocado ao Short Term Backlog, posteriormente este foi movido para Investigation e assim por diante até sua finalização no dia 8/2/2013.

Através da Tabela 7.1.1 foi possível gerar os gráficos que serão demostrados na seções seguinte.

7.2 Lead Time & WIP

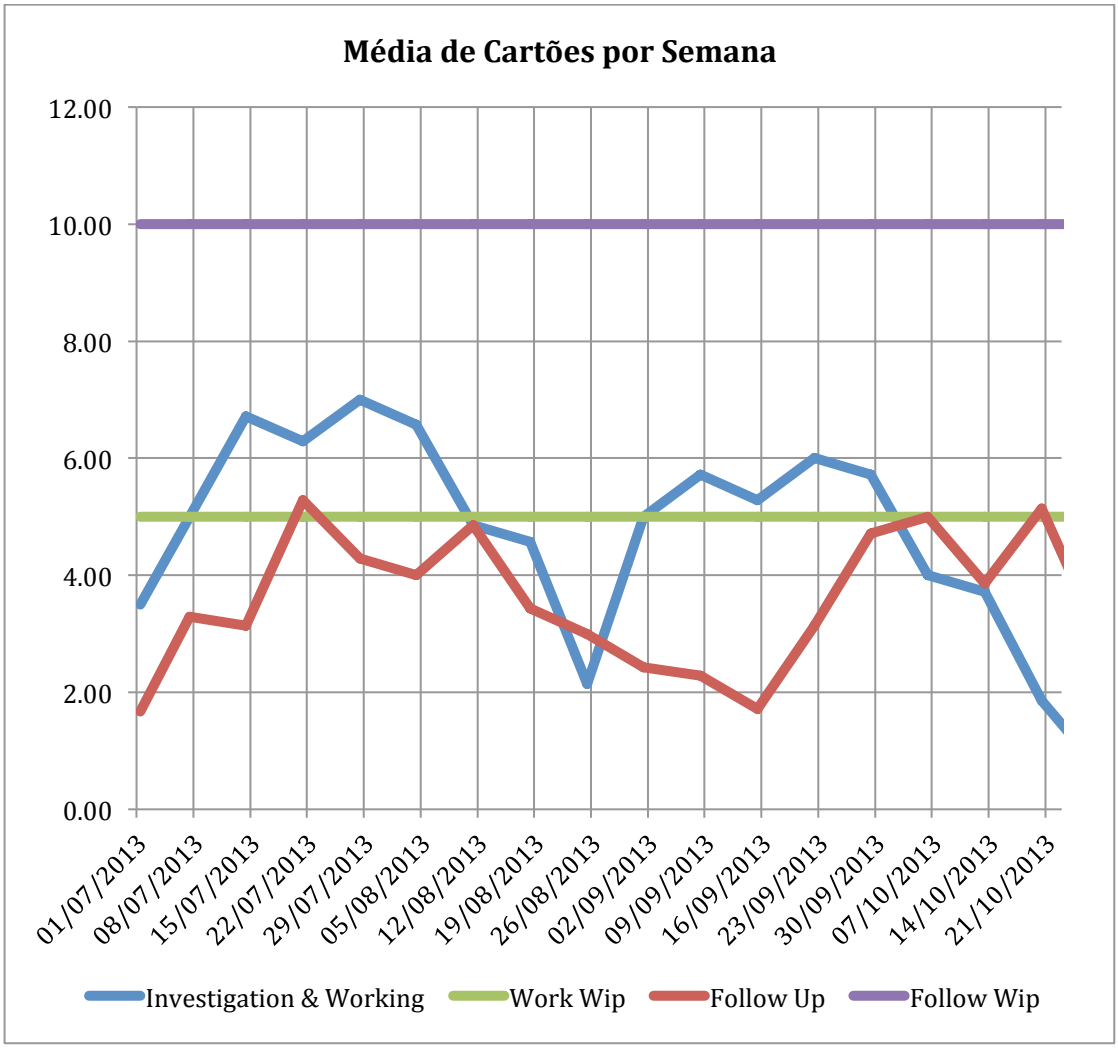


Figura 7.2.1: Média de cartões por semana nos estágios indicados

A Figura 7.2.1 representa a média do número de cartões sendo trabalhados ou em acompanhamento por semana. Nesta seção, trabalhados compreende cartões nos estágios Working ou Investigation e, em acompanhamento, os cartões em Follow Up.

Para gerar esse gráfico foram utilizadas métricas, como segue:

- Working->Done: o número de cartões que foram movidos para Working para Done naquele dia;
- FollowUp->Done: o número de cartões que foram movidos de Follow Up para Done naquele dia;
- Investigation->BufferWorking: o número de cartões que foram movidos de Investigation para Buffer Working naquele dia.
- Investigation->Working: o número de cartões que foram movidos de Investigation para Working naquele dia.

O WIP de Investigation e Working (Investigation&Working no gráfico) e o de Follow Up (com o mesmo nome no gráfico) de um dia, são obtidos pelas fórmulas:

$$Investigation \,\&\, Working = \\ (Working \rightarrow Done - FollowUp \rightarrow Done) + \\ (Investigation \rightarrow BufferWorking + Investigation \rightarrow Working)$$

Como no Kanban alguns cartões vão de Working para FollowUp antes de ir para Done, foi decidido retirar esses cartões da contagem. Portanto, queria-se obter a quantidade de cartões em Working que foram para Done mas não passaram por Follow Up. Isso é simbolizando pela subtração de FollowUp -> Done de Working -> Done.

Dessa forma, foi possível ter uma métrica realista do WIP em Investigation e Working (série em azul no gráfico) podendo comprar esta com o Work Wip (série em verde no gráfico) estabelecido no capítulo 6. O mesmo é feito com ao contabilizar os cartões em Follow Up (série em vermelho), com a fórmula abaixo, e comparar o número obtido com o Follow Wip (série em roxo) estabelecido no capítulo 6.

$$FollowUp = FollowUp \rightarrow Done$$

Cada ponto no gráfico representa a média das métricas citadas acima na semana, representada pela data do primeiro dia dessa semana.

A linha Investigation e Working demonstra um crescimento no início da implementação do Kanban chegando a 7 cartões nesses estágios, ao mesmo tempo. Isso pode ter relação com o fato do que time estar aprendendo a usar o Kanban neste período.

É observada uma queda nesse número e novamente uma elevação. Suspeita-se de que, no fim de Agosto, houveram desfalques no time de S&R da ThinkAgile causando esse comportamento.

A elevação no mês de Agosto mostra que o time conseguiu manter uma média de 5 a 6 tarefas simultâneas e, no fim de Outubro, houve uma diminuição no número de cartões.

A equipe de S&R da ThinkAgile possui 5 administradores, logo o limite Work Wip, representado pelas colunas Investigation e Working, é 5. Pode-se observar que no mês de setembro e outubro foi possível manter essa média com esforço, porém, no início da implementação do Kanban foi mais difícil e no fim essa média diminuiu.

Já no Follow Up podemos observar que o limite Follow Wip estabelecido em 10, talvez tenha sido demais para o time já que esta métrica se manteve no máximo com 5.

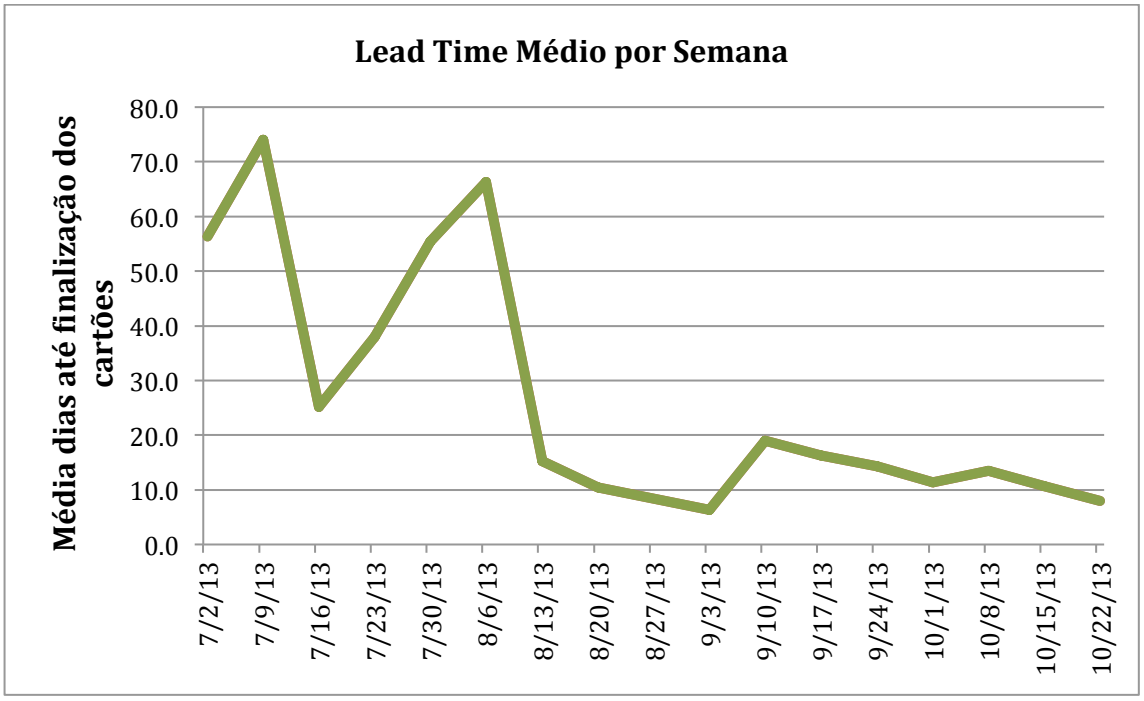


Figura 7.2.2: Lead Time Médio por Semana

A Figura 7.2.2 representa o Lead Time Médio Por Semana. Para obter essa média são considerados os Lead Times de todos os cartões em Done de cada semana. O Lead Time de cada tarefa representa o número de dias desde a entrada do cartão até sua saída do Kanban.

Para os cartões na raia Unplanned foram contados os dias no intervalo To_Working até To_Done, pois os cartões entram no Kanban no estágio Working. Já para os cartões em Planned foram contabilizados os dias no intervalo To_Short_Backlog até To_Done.

Inicialmente é percebido um Lead Time médio por semana de até 70 dias para terminar um cartão, mas com o tempo ele vai diminuindo até ocorrer uma queda brusca de Lead Time a partir da segunda semana de Agosto. Não sabemos explicar o porquê desta queda, mas novamente é sugerida a adaptação do time ao Kanban.

O mais interessante foi comparar o gráfico da Figura 7.2.2 com o número de cartões em Investigation e Working conforme Figura 7.2.3:

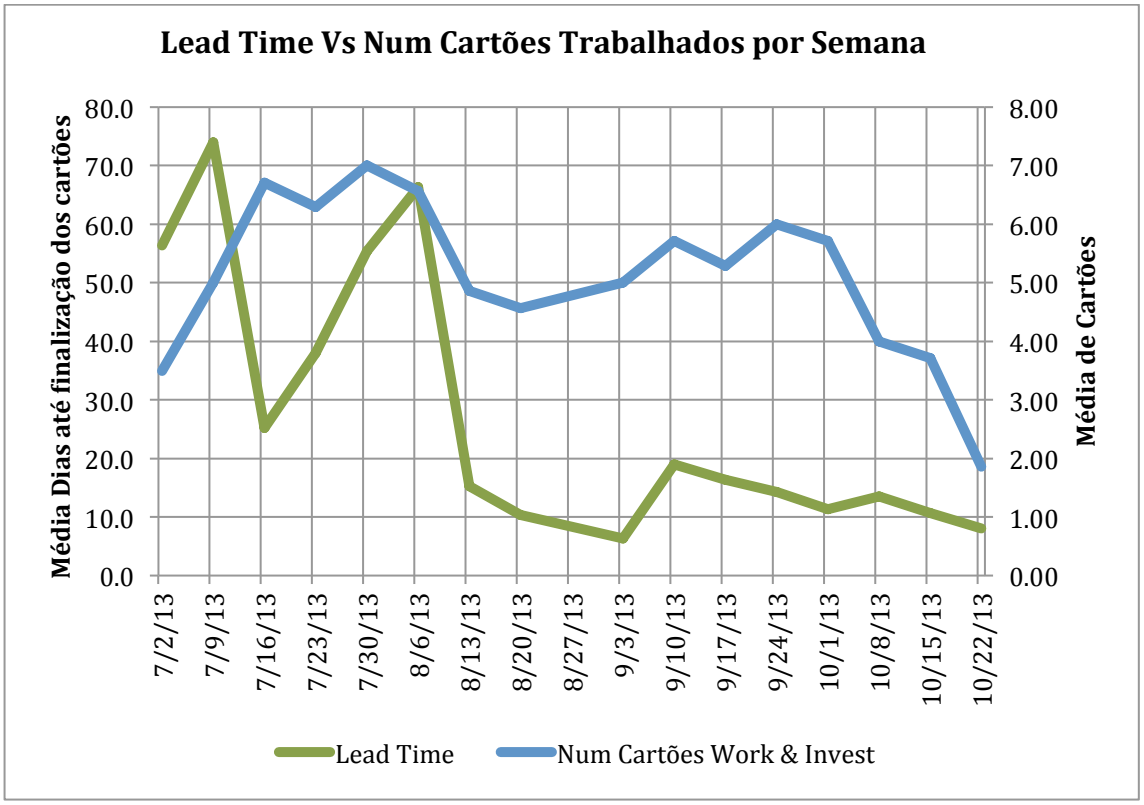


Figura 7.2.3: Lead Time Vs Número de Cartões

Na Figura 7.2.3 foi feita a junção do número de cartões nos estágios Working e Investigation, relacionados ao Work Wip, com o Lead Time.

É interessante observar como o número de tarefas em progresso é diretamente proporcional ao Lead Time conforme explicitado no capítulo 3. Para explicitar mais esse conceito foi feita a Tabela 7.2.1.

Variação Lead Time	Min Date	Número de Cartões
0	7/2/13	0
AUMENTOU	7/9/13	AUMENTOU
DIMINUIU	7/16/13	AUMENTOU
AUMENTOU	7/23/13	DIMINUIU
AUMENTOU	7/30/13	AUMENTOU
AUMENTOU	8/6/13	DIMINUIU
DIMINUIU	8/13/13	DIMINUIU
DIMINUIU	8/20/13	DIMINUIU
DIMINUIU	9/3/13	AUMENTOU
AUMENTOU	9/10/13	AUMENTOU
DIMINUIU	9/17/13	DIMINUIU
DIMINUIU	9/24/13	AUMENTOU
DIMINUIU	10/1/13	DIMINUIU
AUMENTOU	10/8/13	DIMINUIU
DIMINUIU	10/15/13	DIMINUIU
DIMINUIU	10/22/13	DIMINUIU

Tabela 7.2.1: Comparação detalhada Lead Time e WIP

Na Tabela 7.2.1, ao invés dos valores absolutos, foi mostrada a variação da série entre uma semana e a anterior usando a palavra aumentou quando ouve aumento da variação ou do número de cartões na semana e diminuiu caso contrário.

Podemos observar um comportamento errático somente em 4 semanas da tabela o que comprova que o WIP e o Lead Time são realmente proporcionais e que limitar o WIP pode ter consequências muitos positivas.

7.3 Raias Unplanned Vs Raias Planned

Sabe-se que o número de tarefas em Planned é sempre superior ao número de tarefas em Unplanned por causa da existência dos Backlogs que possuem um número elevado de cartões.

Também é conhecido que toda vez que uma tarefa é criada em Unplanned é feito um esforço para que todas as tarefas desta raia terminem o mais rápido possível. É interessante, portanto, observar o número de cartões terminados por semana para inferir quantas demandas e, principalmente, quantos incidentes são trabalhados por semana.

A Figura 7.3.1 foi gerada, com esse objetivo, ao realizar a contagem de tarefas em Unplanned e Planned terminadas (Done) por semana (representada pelo primeiro dia daquela semana). Esse gráfico não representa a média, como nos gráficos anteriores e sim, a real contagem de cartões representada pelo eixo vertical da Figura.

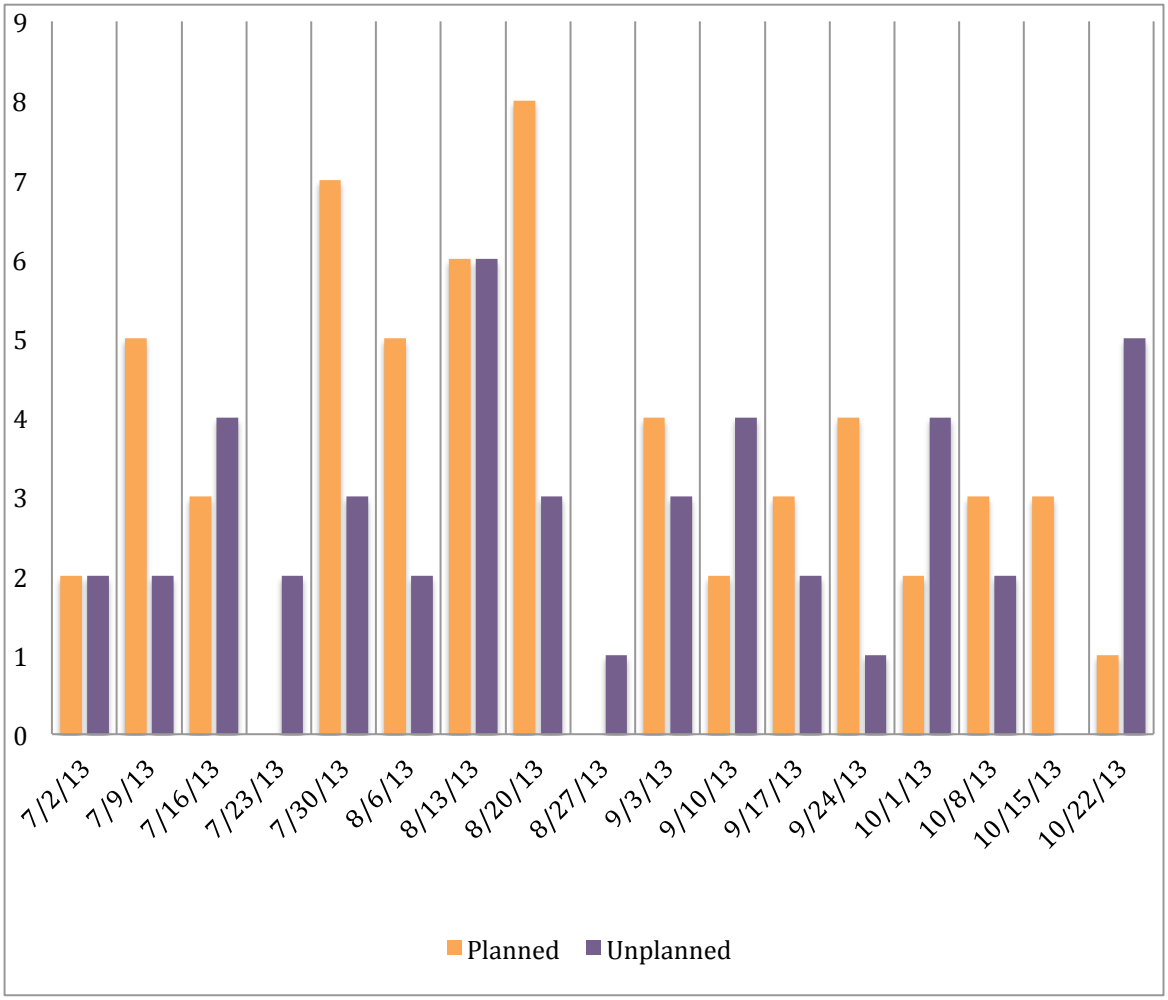


Figura 7.3.1: Planned Vs Unplanned

Na Figura 7.3.1, pode-se observar que o número de cartões em ambas as raias possui um comportamento variável.

É possível inferir através desse gráfico que pelo menos 2 incidentes acontecem por semana. Logo, talvez seja necessário aumentar o limite Work Wip de maneira que seja sempre possível acomodar os incidentes e obter uma variação para o Lead Time, já que este está relacionado ao WIP.

Ainda, é interessante saber o impacto que os incidentes possuem na média de Lead Time dos cartões e o gráfico resultante desta comparação é dado pela Figura 7.3.2.

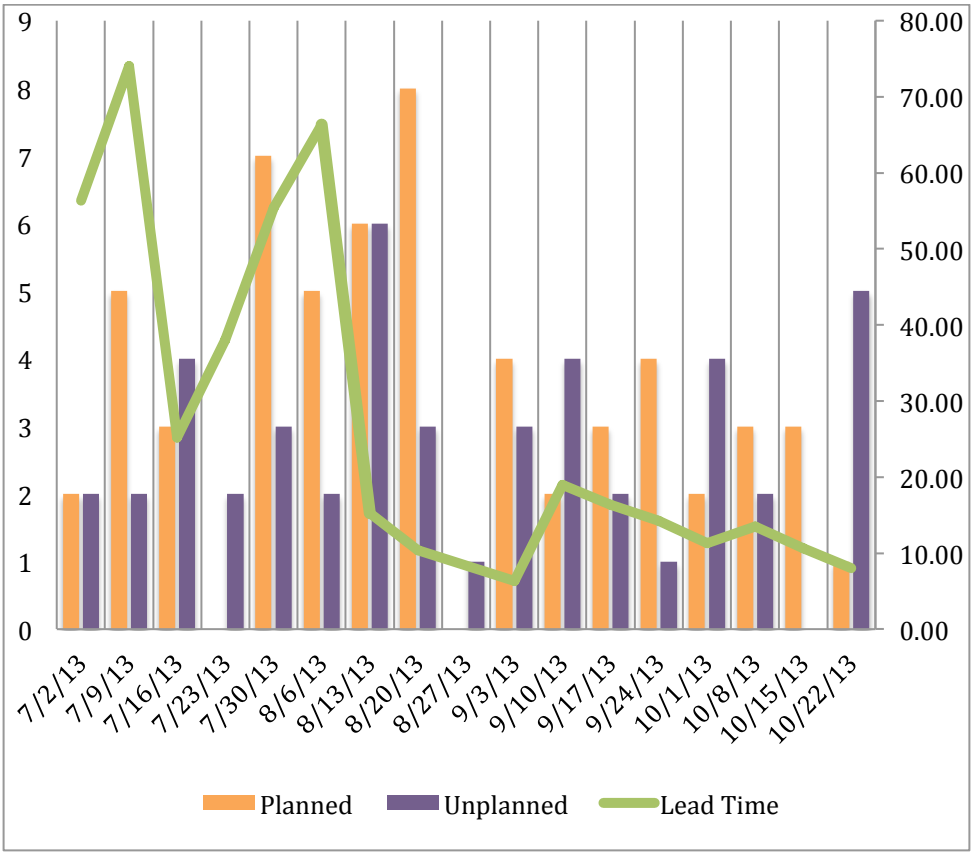


Figura 7.3.2: Relação das raias com o Lead Time

Através da Figura 7.3.2 não foi possível observar um comportamento, no gráfico, que mostre que o Lead Time está relacionado com os incidentes da semana em questão.

Mesmo nas semanas de Agosto, onde o time já teria mais familiaridade com o Kanban, é difícil inferir que o Lead Time esteja relacionado com incidentes olhando somente os gráficos, portanto, para detalhar mais a situação foi construída a Tabela 7.3.1:

Variação Lead Time	Date	Planned Vs Unplanned
0	7/2/13	0
AUMENTOU	7/9/13	PLANN
DIMINUIU	7/16/13	UNPLAN
AUMENTOU	7/23/13	UNPLAN
AUMENTOU	7/30/13	PLANN
AUMENTOU	8/6/13	PLANN
DIMINUIU	8/13/13	UNPLAN
DIMINUIU	8/20/13	PLANN
DIMINUIU	9/3/13	PLANN
AUMENTOU	9/10/13	UNPLAN
DIMINUIU	9/17/13	PLANN
DIMINUIU	9/24/13	PLANN
DIMINUIU	10/1/13	UNPLAN
AUMENTOU	10/8/13	PLANN
DIMINUIU	10/15/13	PLANN
DIMINUIU	10/22/13	UNPLAN

Tabela 7.3.1: Detalhamento entre Lead Time e Raias

Na Tabela 7.3.1 pode-se observar os seguintes dados:

- Coluna da Esquerda: Variação do Lead Time Médio em relação a semana anterior. Se a variação aumentou ou diminuiu este comportamento é descrito;
- Coluna do Meio: Primeiro dia da semana em questão;
- Coluna da Direita: Representa qual raia possui o maior número de cartões terminados. Verde se a maioria dos cartões da semana estão em Planned e em vermelho Unplanned.

Durante Julho e Agosto não é possível estabelecer nenhuma relação com a variação Lead Time e quantidade de cartões por raia. Já no fim de Agosto é possível perceber que a variação do Lead Time aumenta quando temos mais tarefas em Unplanned e diminui quando temos mais tarefas em Planned.

7.4 Conclusões sobre resultados

Os comportamentos mais erráticos se encontram nos meses de Julho em todos os aspectos abordados pelos gráficos do capitulo 7 em razão de que o time estava se adaptando ao Kaban. Não foi possível explicar o comportamento encontrado, entre a segunda e a terceira semana de Agosto, nos gráficos da seção 7.2.

O período posterior a Agosto representa uma maturidade do time em relação ao Kanban já que os números parecem ter apresentado menor variação para os gráficos da subseção 7.2.

Através dos gráficos de número de cartões por semana, pode-se dizer que o limite de 10 em Follow Wip, para o setor de S&R da ThinkAgile, pode ser diminuído, visto que os valores nos gráficos estão longe deste limite e que o time está tendo dificuldade em manter o Work Wip em 5.

O gráfico que compara Lead Time com número de cartões em WIP reforça que ambos são proporcionais e que deve-se trabalhar mais sobre o WIP como forma de garantir um Lead Time otimizado.

Os incidentes tem um impacto pequeno no Lead Time, porém eles ocorrem de maneira constante, uma ou duas vezes por semana. Talvez seja possível aumentar o WIP para acomodar os incidentes de forma que o time não tenha dificuldade em manter o limite estabelecido e possa manter um passo constante.

8 CONCLUSÃO

A adoção do Kanban na administração de sistemas ligados a infraestrutura tem contribuído positivamente para o setor ao resolver os problemas mencionados no decorrer do trabalho. Um resumo das soluções é como segue:

- A conscientização dos clientes é realizada através da Gestão de Solicitações de Clientes e, principalmente, através da exposição constante do Kanban provocando a percepção do cliente frente as demandas do setor;

- O cliente possui um local físico e online, na mesa do setor de S&R e na internet respectivamente, onde pode consultar o andamento de sua demanda de forma fácil;

- Foi possível aos administradores de S&R visualizarem suas tarefas e a de seus colegas sob vários critérios;

- Obteve-se um foco maior sobre as tarefas ao utilizar limites sobre o WIP que permitiram a conscientização do administrador sobre: quantas tarefas ele deve se comprometer, qual a prioridade delas e a complexidade das mesmas. A reunião diária também contribuiu para o foco ao incentivar os administradores de S&R a expor suas tarefas para o time. Além disso, a Gestão de Solicitações de Clientes permite que o foco na tarefa corrente não seja desviado ao prestar atendimentos de suporte;

- A programação em par juntamente com a reunião diária ajudaram o time a melhorar sua comunicação e a resolver possíveis bloqueios em tarefas;

- O time possui uma ferramenta que dá uma visão geral de quais tipos de tarefas estão tomando mais tempo, qual a complexidade e prioridade associada a elas. A contabilização exata de tarefas é realizada, automaticamente, através da ferramenta, porém não foi possível configurá-la até a finalização deste trabalho. Ao invés disso, os dados foram exportados da ferramenta para uma planilha, entretanto sua manipulação foi extremamente complexa. Apesar disso, a demonstração de gráficos sobre os dados foi importante para saber os ajustes que devem ser feitos ao Kanban de forma a melhorar o setor. Recomenda-se,

em um próximo trabalho a utilização da ferramenta para a contabilização automática de dados.

Alguns aspectos precisam ser melhor trabalhados como a reunião de planejamento do backlog para uma priorização mais consistente de tarefas, e a reunião de retrospectiva, para incentivar a reflexão do trabalho realizado e, com isso, permitir aperfeiçoamentos.

Foi detectado durante a realização deste trabalho que também é necessário ajustar o posicionamento da coluna Long Term Backlog para a direita como forma de melhorar o fluxo no Kanban. O WIP também precisa ser levemente ajustado tendo em vista o que foi detectado ao gerar os gráficos deste trabalho.

Se acostumar as metodologias ágeis propostas neste trabalho exigiu muita disciplina, mas os objetivos principais foram alcançados e os resultados foram rapidamente observados pelos clientes e pelos administradores.

9 BIBLIOGRAFIA

ACHOUIANTZ, C. Examples of Charts & Graphs used for Improvement. **Lean-Agile Software Development**. Disponível em http://leanagileprojects.blogspot.com.br/2013/04/examples-of-charts-graphs-used-for.html, 2013, October 4. Acessado em 13 de Outubro de 2013

AGILE METHODOLOGY, **Agile Methodology.** Disponível em agilemethodology.org. Acessado em 30 de Novembro de 2013.

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

BECK, K. **Extreme programming eXplained: embrace change**. Reading, MA: Addison-Wesley.[s.n]

BECK, K.; BEEDLE, M.; BENNEKUM, A. VAN; CUNNINGHAM, W.; et al. Manifesto para Desenvolvimento Ágil de Software. **Agile Manifesto**. Disponível em http://agilemanifesto.org/iso/ptbr/, 2001. Acesso em 3 de Agosto de 2013.

BECK, K.; BEEDLE, M.; BENNEKUM, A. VAN; CUNNINGHAM, W.; et al. Princípios por trás do Manifesto Ágil. **Agile Manifesto**. Disponível em http://agilemanifesto.org/iso/ptbr/principles.html, 2001. Acesso em 4 de Agosto de 2013.

BENEFITS of Scrum. .**The Braintrust Consulting Group**. Disponível em http://braintrustgroup.com/scrum/benefits-of-scrum/. Acesso em 5 de Novembro de 2013.

BURROWS, M. Kanban In A Nutshell. **Positive Incline**. Disponível em http://positiveincline.com/index.php/2010/03/kanban-in-a-nutshell/, 2010, March. Acesso em 13 de Outubro de 2013.

CAROLI, P. 7 Maneiras de Controlar o Lead Time. **Agile Tips**. Disponível em http://www.youtube.com/watch?v=9qj0v9cgnmI, 2013, January 8. Acesso em 13 de Outubro de 2013.

CARTLIDGE, A.; HANNA, A.; RUDD, C.; MACFARLANE, I.; et al. **An Introductory Overview of ITIL V3**, Desenvolvimento de Software Corporativo.. 1st ed. The UK Chapter of the itSMF, 2007.

COHN, M. What is Scrum Methodology? **Mountain Goat Software**. Disponível em http://www.mountaingoatsoftware.com/topics/scrum, 1998. Acesso em 5 de Outubro de 2013.

COHN, M. Sprint Planning Meeting. **Mountain Goat Software**. Disponível em http://www.mountaingoatsoftware.com/agile/scrum/sprint-planning-meeting, 1998. Acesso em 5 de Outubro de 2013.

CORE Job Titles. https://www.usenix.org/system/files/lisa/books/usenix_22_jobs3rd_core.pdf. Acesso em 27 de Novembro de 2013.

DERBY, E.; LARSEN, D. **Agile retrospectives: making good teams great**. Raleigh, NC: Pragmatic Bookshelf, 2006.

DESENVOLVIMENTO Ágil de Software. .**Wikipédia**. Disponível em http://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software. Acesso em 12 de Outubro de 2013.

FOWLER, M. It's Not Just Standing Up: Patterns for Daily Standup Meetings. **Martin Fowler**. Disponível em http://martinfowler.com/articles/itsNotJustStandingUp.html, 2011. Acesso em 4 de Agosto de 2013.

HEROLD, R. **Improving IT Service Support through ITIL**. 1st ed. Realtime, 2007.

HOW can you make sprint planning fun? .**Information Technology**. Disponível em http://arstechnica.com/information-technology/2013/10/how-can-you-make-sprint-planning-fun/, 2013. Acesso em 10 de Novembro de 2013.

JEFFRIES, R. E. What is Extreme Programming. **XProgamming.com**. Disponível em http://xprogramming.com/xpmag/whatisxp, 1999. Acesso em 5 de Outubro de 2013.

KNIBERG, H. **Scrum e XP Direto das Trincheiras**. 1st ed. InfoQ.

KNIBERG, H.; SKARIN, M. **Kanban & Scrum making the most of both**, Enterprise Software Development.. 1st ed. InfoQ, 2010.

LIKER, J. K. **The Toyota Way—14 Management Principles from the World's Greatest Manufacturer**. 1st ed. The MacGraw-HIll Companies, 2004.

MELO, C.; SANTOS, V.; CORBUCCI, H.; KATAYAMA, E.; et al. Estado da prática em times e organizações. **IME-USP**, p. 9, 2012.

MINGLE. .**ThoughtWorks Products**. Disponível em http://www.thoughtworks.com/products/mingle-agile-project-management. Acesso em 16 de Novembro de 2013.

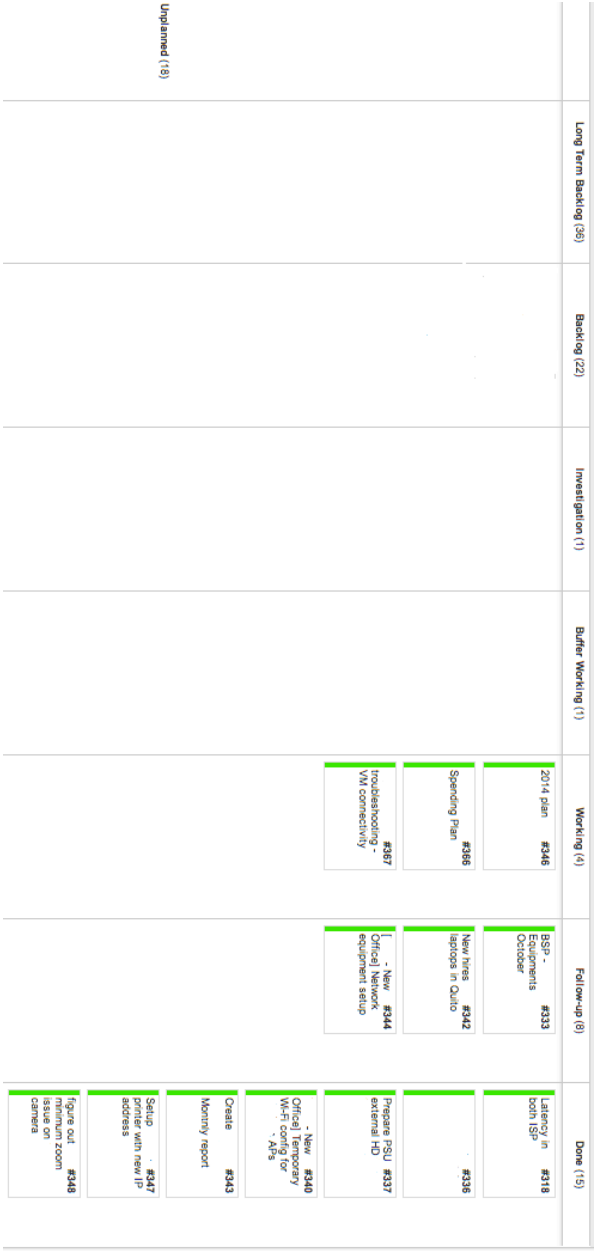
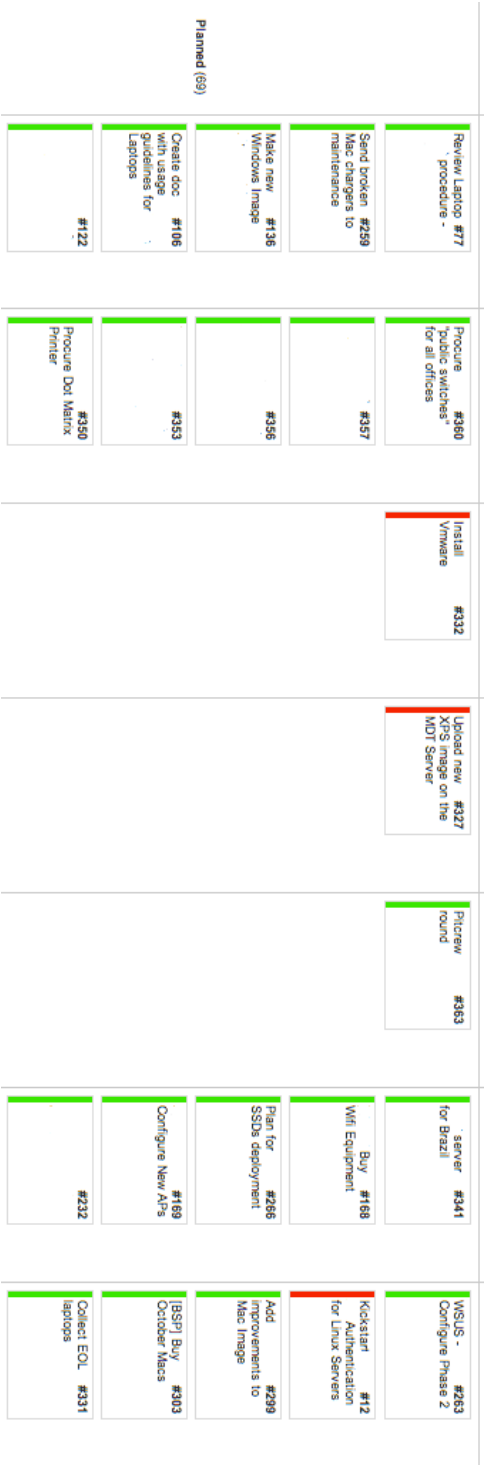
RETROSPECTIVE Plans. **Agile Retrospective Wiki**. Disponível em http://agileretrospectivewiki.org/index.php?title=Retrospective_Plans, 2013. Acesso em 5 de Outubro de 2013.

SCRUM. .**Wikipédia**. Disponível em http://pt.wikipedia.org/wiki/Scrum. Acesso em 11 de Outubro de 2013.

SKARIN, M. 10 Different Kanban Boards And Their Context. . Crisp. Disponível em http://blog.crisp.se/2011/12/05/mattiasskarin/10-kanban-boar?xt-version-1-2, 2011. Acesso em 26 de Outubro de 2013.

VAN BON, J.; DE JONG, A.; KOLTHOF, A.; PIEPER, M.; et al. **Foundations of IT Service Management based on ITIL V3**. 3rd ed. Van Haren, 2007.

APÊNDICE A: RAIAUNPLANNED NO MINGLE



APÊNDICE B: RAIA PLANNED NO MINGLE