



[www.devmedia.com.br](http://www.devmedia.com.br)

[versão para impressão]

Link original: <https://www.devmedia.com.br/scrum-kanban-para-manutencao-de-software/19276>

## Scrum + Kanban para Manutenção de Software

**Neste artigo veremos como a adaptação de metodologias ágeis no desenvolvimento de software pode trazer benefícios ao cliente e, principalmente, ao grupo de trabalho.**

---

### [lead]De que se trata o artigo:

Neste artigo veremos como a adaptação de metodologias ágeis no desenvolvimento de software pode trazer benefícios ao cliente e, principalmente, ao grupo de trabalho. Mostraremos um novo modelo de metodologia ágil criado a partir das boas práticas identificadas no Scrum e no Kanban e que foi definido para atender às necessidades específicas da equipe de manutenção de software.

### Para que serve:

O planejamento e o desenvolvimento de um software sob o paradigma ágil oferecem mais flexibilidade do que os modelos tradicionais para a condução de projetos de software que possuem incerteza ou instabilidade. A junção entre Scrum e Kanban traz, às atividades de manutenção de software onde existem contratos rígidos de suporte (SLA), bastante dinamismo, além de direcionar os esforços para os problemas de maior valor e importância para o cliente.

### Em que situação o tema útil:

Além de ser um modelo que permite a criação de planos flexíveis, as técnicas ágeis evitam desperdícios de tempo e esforços, pois focam em alcançar rapidamente meios de validar as características do produto em desenvolvimento e a evolução do projeto. A aplicação desses métodos na manutenção de software trouxe benefícios ao ciclo final do produto, mantendo assim, a satisfação do cliente.

### Autores: Marcelo Camera e Anselmo Junior[/lead]

Provavelmente você já tenha ouvido falar em desenvolvimento ágil de software, na utilização de Scrum e Kanban e nos benefícios da adoção das metodologias ágeis. Por outro lado, pouco se tem falado sobre a aplicação de uma dessas metodologias nas operações de manutenção de software, área em que o trabalho é regido por contratos de serviço que determinam prazos, custos e novas prioridades a cada momento.

Pensando nessa situação, seria possível aplicar alguma metodologia ágil para aumentar a produtividade e o tempo de resposta na operação de manutenção de software?

Este artigo revela nossa procura por respostas e a maneira como adotamos e adaptamos algumas das melhores práticas de duas reconhecidas metodologias ágeis, Scrum e Kanban, e como temos obtido benefícios depois que alteramos nossa maneira de trabalhar. Além disso, acreditamos que este artigo possa servir de inspiração para outras organizações que têm sua atividade voltada à operação de manutenção de software.

[subtitulo]Como é a operação de manutenção de software[/subtitulo]

Todo nosso processo tem início quando um projeto de desenvolvimento termina e o cliente aceita formalmente os produtos: software e documentação. A partir do momento da entrega, a responsabilidade pelo produto passa a ser do grupo de manutenção, que vai analisar toda e qualquer falha reportada.

Operacionalmente, quando o cliente observa um comportamento atípico do sistema abre uma ordem de serviço (OS) descrevendo essa possível falha. Nesse momento, a OS é encaminhada a um dos grupos de manutenção para análise e, se for constatada a falha, tem início o desenvolvimento da correção do software (CS).

As OS estão atreladas a um contrato de serviço (SLA) que é estabelecido entre o cliente e a organização e que define como a OS é classificada e como deverá ser tratada pelo grupo de manutenção. Essa classificação varia de acordo com a severidade da falha, sendo 1 a maior severidade e 3 a menor. As SLAs especificam também o tempo máximo de resposta para cada uma das OS, que variam também de acordo com a severidade da falha, sendo 10 dias para severidade 1, 20 dias para severidade 2 e 30 dias para severidade 3. Em função do extenso portfolio de produtos e projetos desenvolvidos, OS são abertas a todo o instante e nas mais diversas prioridades.

Além das classificações anteriores, é denominada *blacklist* a lista de todas as OS que tiveram seu prazo de resposta perdido. A partir desse dia, passamos a contabilizar o número de dias perdidos por OS, ou seja, o prazo extra gasto na resposta da OS.

Diz-se que uma OS perdeu um dia se ela permanecer na *blacklist* por um dia, com isso no final do período temos então a quantidade de dias perdidos por OS, que mostra nossa eficiência em responder as OS no prazo.

As SLAs descrevem também os indicadores de qualidade, performance, velocidade do time de manutenção e multas em caso do não cumprimento dos contratos de serviço.

A qualidade é medida a partir de CS canceladas (correções de software liberadas que apresentam algum tipo de falha ou inserem algum efeito colateral no sistema) e CS liberadas. Para medir a performance, é calculada a média de dias que uma OS permanece aberta sem ser solucionada. A velocidade é representada pelo número de dias perdidos pelas OS em um dado período de tempo.

A **Tabela 1** ilustra os números da operação de manutenção durante o terceiro trimestre de 2009.

Números da Operação de Manutenção – Trimestre 3, 2009	
Número de colaboradores do time de manutenção (corpo técnico)	50
OS abertas no período	250
OS respondidas no período	230
Tempo médio de resposta por OS (dias/OS)	26,36
Dias perdidos (dias)	1032

**Tabela 1.** Números da Operação de Manutenção, Trimestre 3-2009.

Em uma situação hipotética de operação ideal estimava-se que cada membro do time fosse capaz de analisar até duas OS paralelamente.

Apesar do time contar, no terceiro trimestre de 2009, com 50 pessoas dos mais variados perfis de senioridade, a capacidade de responder as OS não foi satisfatória (230 OS respondidas), deixando como resíduo 20 OS a serem respondidas no próximo período. Comparando-se com a situação hipotética ideal, um time de 50 pessoas deveria ser capaz de analisar até 400 OS.

A experiência de nossa organização em projetos de desenvolvimento de software que seguem uma metodologia ágil evidenciou alguns dos benefícios que a operação de manutenção procurava alcançar: flexibilidade do time, rápida resposta a mudanças e entrega do que é prioridade para o cliente.

Essa constatação nos fez acreditar que o sucesso alcançado por esses outros projetos de desenvolvimento, também poderia ser conquistado nas operações de manutenção. Para isso, fomos buscar nas mais reconhecidas metodologias ágeis, os benefícios já comprovados pelos outros times.

### Características do Scrum, Kanban e da operação de manutenção

A maioria das práticas descritas na metodologia ágil não é diretamente aplicada à realidade da operação de manutenção de software. Além disso, a adoção das práticas ágeis no ambiente de projetos de desenvolvimento havia motivado a mudança de comportamento dos times, encorajando-os a uma melhor e mais efetiva comunicação e colaboração na execução do trabalho.

[subtitulo]Scrum[/subtitulo]

Os projetos executados por nossa organização, em grande parte, fazem uso da metodologia ágil Scrum, que tem as seguintes características:

- O grupo é dividido em times pequenos, multifuncionais e autogerenciados;
- O trabalho é fragmentado em pequenos itens testáveis e entregáveis, classificados segundo sua prioridade e esforço relativo para completar cada item;

- O cronograma de projeto é dividido em pequenas iterações ou ciclos (*Sprints*), com duração de 1 a 4 semanas, com demonstrações entregáveis ao final de cada iteração;
- O planejamento das entregas é otimizado e atualizado em parceria com o cliente. O objetivo é resolver as OS que o cliente definiu como prioridade, aumentando o retorno sobre o investimento;
- O processo é atualizado após cada iteração baseado nos *feedbacks* recebidos nas retrospectivas;
- As reuniões diárias (*daily meetings*) têm o objetivo de atualizar o progresso das atividades e responde a três perguntas: O que fiz ontem? O que vou fazer hoje? O que está me impedindo de fazer?

A **Tabela 2** mostra um comparativo entre as características do Scrum e da operação de manutenção. É possível identificar muita similaridade entre as práticas do Scrum referentes à configuração do time e da fragmentação do trabalho a ser realizado. Por outro lado, existem grandes diferenças em relação à maneira de lidar com o escopo fixo a cada *sprint*, uma vez que para a operação de manutenção, o escopo eram as OS que chegavam todos os dias.

Scrum	Operação de Manutenção
Times pequenos, multifuncionais e auto gerenciados.	Times pequenos e multifuncionais.
Trabalho dividido em pequenos itens testáveis e entregáveis.	A OS é a menor unidade de trabalho que pode ser testada e entregue (CS).
Cronograma é dividido em pequenas e frequentes iterações ( <i>Sprints</i> ).	Cronograma é dividido somente para o controle estatístico de métricas.
Reuniões diárias ( <i>Daily Meetings</i> )	Reuniões técnicas semanais.
Escopo fixo por iteração, decidido durante reunião de planejamento de <i>Sprint</i> .	Escopo constantemente alterado devido à entrada de novas OS.
Plano de entregas otimizado de acordo com as prioridades estabelecidas junto ao cliente.	Todo escopo é prioridade na entrega.
Processo otimizado a cada iteração.	Processo não é otimizado.

**Tabela 2.** Comparativo entre o Scrum e a Operação de Manutenção.

[subtitulo]Kanban[/subtitulo]

A maioria das características e boas práticas do Scrum não eram completamente aderentes à realidade da operação de manutenção e por isso precisávamos de algo que complementasse as atividades sem descaracterizar a agilidade do Scrum. Esse suporte veio do Kanban.

O Kanban, como conhecemos, remonta à década de 40 e tem como principal característica o controle de processos através do controle da demanda. As características são:

- Fluxo de trabalho visualizável, sendo que cada item é representado por um cartão fixado no quadro de trabalho Kanban;
- WIP (*Work In Progress* – trabalho em execução) limitado;
- *Lead time* (tempo de execução do trabalho) é a métrica de performance, em que o *lead time* é o tempo médio para se completar um item de trabalho.

A **Tabela 3** mostra um comparativo entre algumas características do Kanban e da operação de manutenção. O Kanban trouxe o conceito de WIP (*work in progress*) limitado, preenchendo a principal lacuna deixada pelo Scrum com relação ao escopo de trabalho fixo durante um *sprint*.

Kanban	Operação de Manutenção
Fluxo de trabalho visualizável.	O fluxo de trabalho pode ser entendido, mas não era visualizável.
WIP limitado e explícito.	WIP limitado pelo número de recursos disponíveis, mas não explicitado.
<i>Lead time</i> como métrica.	<i>Lead time</i> como métrica.

**Tabela 3.** Comparação entre o Kanban e a Operação de Manutenção.

[subtítulo]Customização das metodologias ágeis à operação de manutenção[/subtítulo]

Com as principais características do Scrum e Kanban mapeadas, o próximo passo foi escolher quais delas seriam adotadas diretamente e quais seriam customizadas pela operação de manutenção. Em seguida, apresentamos a descrição de cada uma dessas boas práticas.

[subtítulo]Times de Manutenção[/subtítulo]

A operação de manutenção já apresentava características multifuncionais e era composto por um número limitado de integrantes (a operação era dividida em times responsáveis por áreas funcionais distintas). Cada time tinha um *team leader* que era responsável pelo controle e interface externa com outros times, da mesma maneira que age o *Scrum Master*.

Era necessário certificar que os procedimentos e práticas dentro de cada time funcional seriam o mais uniforme possível para que, em caso de alguma demanda ou picos de trabalho, a transferência de membros de um time para outro fosse feita de maneira rápida e eficiente.

[subtítulo]Cronograma[/subtítulo]

O cronograma de trabalho não é limitado, mas na operação existem alguns indicadores de desempenho medidos mensalmente. Devido à natureza da atividade, em que OS eram abertas a todo o instante, manter um cronograma de entregas de um dado escopo não faz sentido, uma vez que o escopo é alterado diariamente. O diferencial é que esses indicadores de desempenho podem ser mensurados a qualquer momento sem que haja necessidade de se esperar o final de um período específico. Sendo assim, os ajustes no time e nas prioridades são feitas com maior rapidez e melhores resultados.

## [subtitulo]Escopo[/subtitulo]

Diferentemente do Scrum que estabelece escopo fixo em cada *Sprint*, o trabalho da operação de manutenção é variável a cada período, sendo que o número de OS abertas determina o escopo do período. Não seria possível manter uma OS sem análise até que um determinado período termine e comece o seguinte. Se, por alguma razão, OS não forem respondidas dentro de determinado tempo, elas serão mantidas na fase de análise do período seguinte.

## [subtitulo]Planejamento de Entregas[/subtitulo]

Ao contrário do Scrum, em que a priorização das atividades é o foco do cliente, no caso das OS, existe uma priorização natural definida pela severidade do problema por ela relatado. Essa escala de prioridades é estabelecida nas SLAs, que institui que todas as OS deverão ser respondidas. Em caso de alguma necessidade específica de um cliente, as prioridades podem ser reajustadas de acordo com o cliente, sem deixar de lado a análise de impacto dessa repriorização na operação de manutenção em geral.

## [subtitulo]Otimização no Processo[/subtitulo]

A cada final de período, o time de manutenção realizaria internamente a reunião de retrospectiva do período com o objetivo de discutir sugestões de melhoria e boas práticas. Ao final de cada três períodos, os Scrum Masters se reúnem (*Scrum of Scrums*) para troca de experiências e atualização do processo se necessário. O objetivo dessa reunião é propor a melhoria contínua no processo adotado pela operação de manutenção, visando maior produtividade e redução de custos associados à manutenção de software.

## [subtitulo]Reuniões diárias (Daily Meetings)[/subtitulo]

As reuniões técnicas semanais deixarão de existir dando lugar às reuniões diárias no mesmo modelo das *daily meetings* do Scrum. As reuniões técnicas ou de tomadas de decisão aconteceriam apenas sob demanda. A idéia das *daily meetings* é a mesma do Scrum: identificar impedimentos e monitorar o progresso das atividades.

## [subtitulo]Fluxo de Trabalho Visualizável[/subtitulo]

O modelo de trabalho baseado no quadro Kanban foi customizado de acordo com as necessidades levantadas pelos times de manutenção. O objetivo é que todo o fluxo de trabalho possa ser visualizado nesse quadro e que impedimentos e possíveis gargalos sejam identificados no processo.

## [subtitulo]WIP (Work In Progress) limitado, mas não explicitado[/subtitulo]

A operação de manutenção faz uso do WIP (*work in progress*) limitado pelo número de recursos de uma dada área funcional. Em casos de pico de OS em uma dessas áreas funcionais ou de alguma outra demanda específica será possível deslocar força de trabalho de uma área para outra. A adoção do quadro de trabalho facilitará a visualização desses picos de demanda.

## [subtitulo]Implantação da metodologia ágil na operação de manutenção[/subtitulo]

O processo de implantação da metodologia ágil na operação de manutenção ocorreu de forma gradual. Inicialmente aconteceram cursos de capacitação do time na metodologia Scrum e no Kanban. Workshops foram realizados para entender a dinâmica do que denominamos operação ágil de manutenção, que na verdade, é como o trabalho de manutenção de software será feito após a adoção da metodologia ágil.

Após o treinamento e os workshops, o grupo iniciou a customização do quadro de trabalho. O objetivo era que o quadro fosse capaz de mostrar todo o fluxo de trabalho, as limitações e principalmente gargalos do processo durante a realização da operação. Para atingir esse objetivo, a participação de todo o grupo de manutenção foi fundamental.

## [subtitulo]Elementos do Quadro de Trabalho[/subtitulo]

O quadro de trabalho desenvolvido pelo grupo de manutenção é composto pelos seguintes campos:

- Novas OS – Coluna onde ficam as OS que ainda não foram analisadas;
- Time – Coluna da tabela formada pelos membros do time de manutenção;
- Em Análise – Ficam as OS que estão sendo analisadas no momento;
- Solução em Preparação – Ficam as OS que tiveram, na fase de análise, constatada a falha. Esse campo indica que uma CS está sendo preparada para solucionar o problema reportado pela OS;
- Em Teste – Local onde ficam as OS que têm CS em fase de testes e validação;
- Finalizadas – Ficam as OS respondidas. É o único campo do quadro de trabalho que é limpo ao final de cada período;
- Impedimentos – Área onde ficam as OS que, devido a algum impedimento, não tiveram progresso na análise, no desenvolvimento da solução ou nos testes da CS;
- *Blacklist* – é o que identifica atraso na resposta da OS.

## [subtitulo]Cores dos Cartões OS[/subtitulo]

À medida que as OS são recebidas são transformados em um cartão para inclusão no quadro de atividades similarmente aos cartões do Kanban. Nesse cartão serão descritas algumas informações básicas como severidade, produto onde a falha foi encontrada, número de identificação da OS e se a OS está na *blacklist*.

A severidade da OS é representada pela cor do cartão, como mostra a **Tabela 4**.

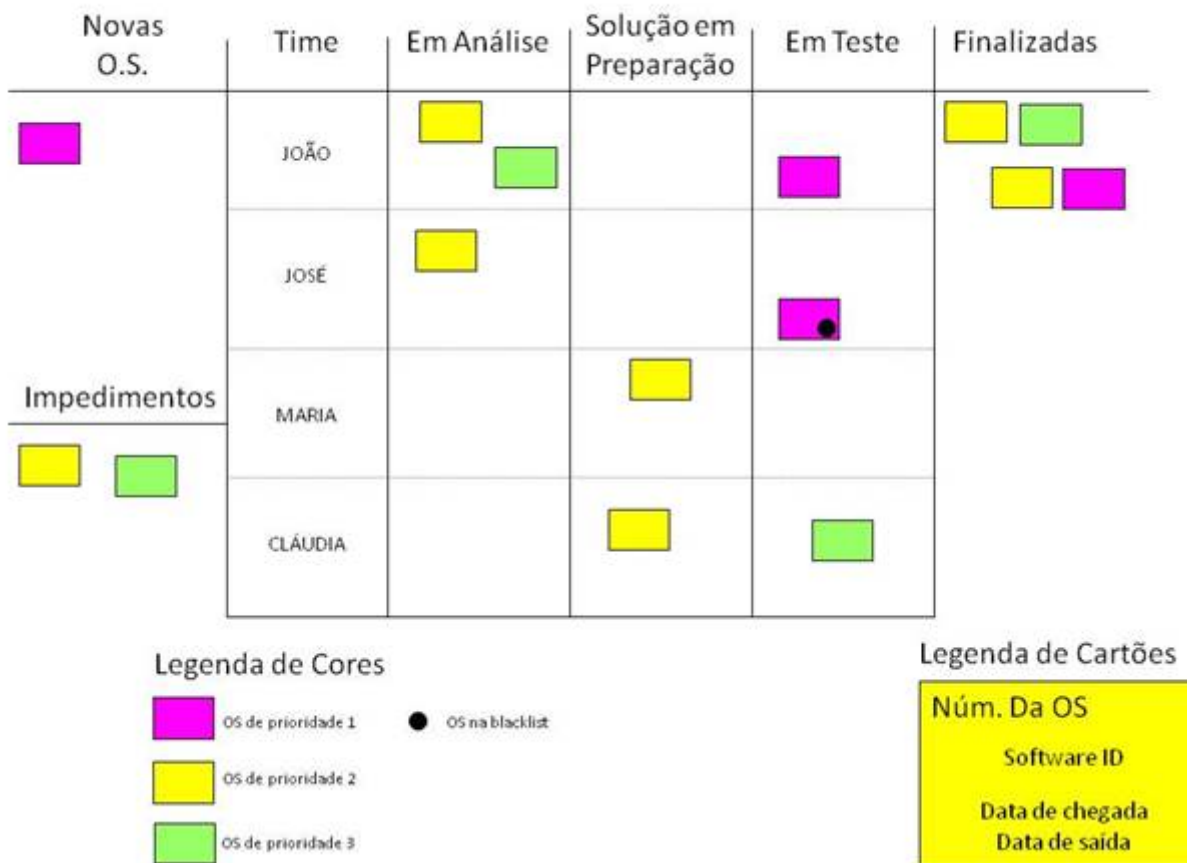
Severidade	Cor do Cartão

Severidade 1	Vermelho
Severidade 2	Amarelo
Severidade 3	Verde

**Tabela 4.** Relação Severidade x Cor do Cartão.

A **Figura 1** mostra como idealmente o quadro de trabalho da operação ágil de manutenção deverá se apresentar.

### Quadro de Trabalho adaptado à Operação de Manutenção



**Figura 1.** Quadro Kanban adaptado à Operação de Manutenção.

[subtítulo]Dinâmica do Quadro[/subtítulo]

A partir do recebimento da OS o grupo de manutenção cria um cartão que vai representar essa atividade. Inicialmente, este cartão fica fixado no campo de “Novas OS” e permanece lá até que um membro do time inicie a análise do problema reportado.

Nesse momento, o cartão com a OS vai para a coluna “Em Análise” e para a linha que identifica o membro do grupo que assumiu essa responsabilidade.

Em não constatada a falha, a OS poderá ser respondida e, então, transferida para o campo “Finalizadas”. Caso seja constatada a falha, a OS é movida para o campo “Solução em Preparação”, onde será desenvolvida a CS que solucionará o problema reportado pela OS.



Após o completo desenvolvimento da CS, inicia-se a fase de teste. Nesse momento, a OS é transferida para o campo “Em Teste”. Uma vez finalizados os testes, a CS será liberada para o cliente e a OS será respondida, possibilitando assim que o cartão seja transferido para o campo “Finalizadas”.

#### [subtitulo]Projeto Piloto[/subtitulo]

O objetivo inicial do projeto piloto era aplicar, durante dois meses, a metodologia ágil de operação de manutenção para uma avaliação e validação dos métodos. Para isso, foi escolhido um grupo separado, responsável por alguns tipos de OS e assim, comparar no final de cada período as performances antes (chamado mês-zero) e depois da implantação (mês1 e mês2). Também é importante frisar que o tamanho do time não foi alterado durante os três períodos.

Essa comparação foi baseada nos seguintes indicadores:

- Número de OS recebidas;
- Número de dias perdidos no mês: Blacklist
- Número de dias perdidos por OS por mês;
- Número de CS canceladas.

A **Tabela 5** mostra os indicadores obtidos durante o projeto piloto.

	Mês 0	Mês 1	Mês 2
Número de OS recebidas no período	16	21	25
Número de dias perdidos no período	23	29	33
Número de dias perdidos por OS no período	1,44	1,38	1,32
Número de CS liberadas no período	0	0	0

**Tabela 5.** Indicadores de performance do projeto piloto.

Baseado na **Tabela 5** algumas análises foram feitas para avaliar a efetividade do método.

#### [subtitulo]Conclusão Quantitativa[/subtitulo]

A análise quantitativa baseia-se puramente nos números, comparando a performance do grupo antes e após a implantação do método. A partir disso, observamos que:

- Foram recebidas 50% mais OS se compararmos o mês 2 com o mês 0;
- Tivemos um aumento de 30% no número de dias perdidos. No período do projeto piloto, propositalmente, não houve manobras nos times para suprir as demandas de aumento de OS, logo era esperado um aumento na mesma ordem de dias perdidos;

- O número médio de dias perdidos por OS no período diminuiu, mostrando uma melhora na performance do time;
- A qualidade, representada pelo cancelamento de CS continuou inalterada.

#### [subtitulo]Conclusão Qualitativa[/subtitulo]

A impressão do time referente às atividades desenvolvidas no dia a dia define nossa conclusão qualitativa:

- Diminuição do número de OS de maior severidade na *blacklist*;
- Aumento da performance e da velocidade de resposta das OS.

A **Tabela 6** mostra o número médio de dias perdidos por OS de acordo com a sua severidade. A SLA especifica que o grupo de manutenção tem 10 dias de prazo para responder as OS com severidade 1, 20 dias para severidade 2 e 30 dias para severidade 3. Os números mostraram que as melhorias aconteceram mês a mês. Este bom resultado é possível devido à correta priorização das OS, sendo que aquelas com maior severidade passaram a ser respondidas em prazos menores.

	Mês 0	Mês 1	Mês 2
<b>Dias perdidos por OS de severidade 1</b>	14	7	5
<b>Dias perdidos por OS de severidade 2</b>	8	10	10
<b>Dias perdidos por OS de severidade 3</b>	1	12	18

**Tabela 6.** Número de dias perdidos versus severidade da OS.

A **Tabela 7** ilustra o *lead time* (tempo médio de respostas) das OS. O resultado mostra que a implementação do método gerou a diminuição do *lead time* mesmo com o aumento do número de OS.

	Mês 0	Mês 1	Mês 2
<b>Total de OS recebidas</b>	16	21	25
<b>Lead time (dias por OS)</b>	26,36	22,53	19,58

**Tabela 7.** Lead time durante o projeto piloto.

A cada final de mês decidimos avaliar o método e suas proposições para atuar em possíveis desvios e assim definir ações corretivas. Em seguida, descrevemos algumas características positivas e negativas do projeto piloto.

#### [subtitulo]Prós Identificados no Projeto Piloto[/subtitulo]

Os pontos positivos definidos pelo grupo e pelos gestores são:

- Maior visibilidade do progresso das atividades de análise das OS;

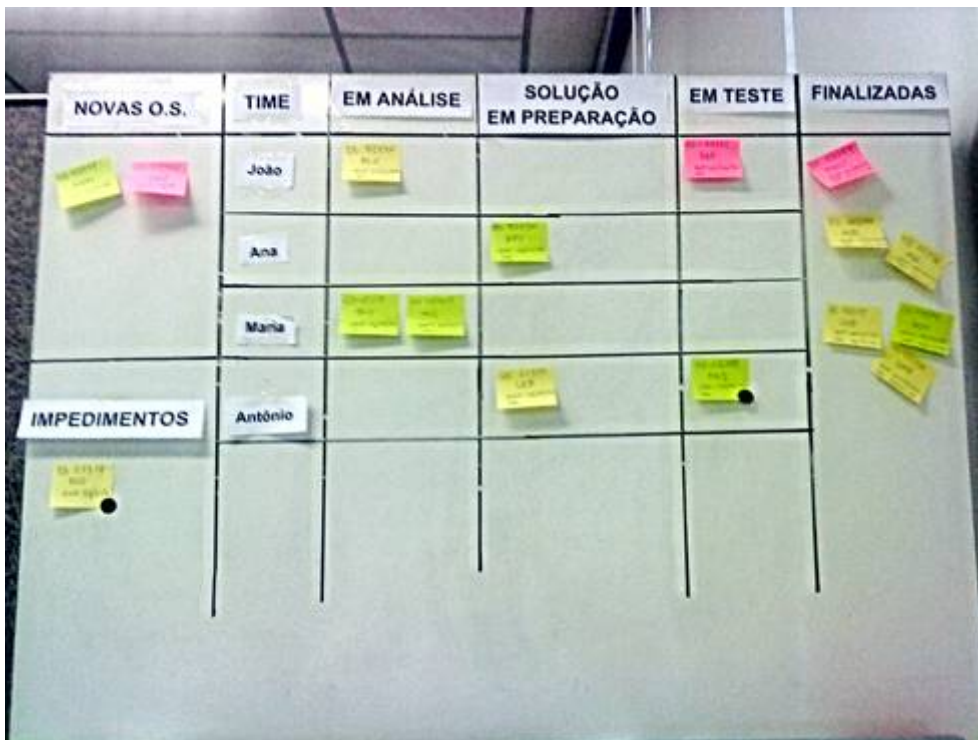
- Melhor e mais fácil distribuição de atividades dentro do time de manutenção;
- Impedimentos tratados de maneira mais rápida e direta;
- Gargalos no processo facilmente identificados;
- Melhora da performance do grupo.

Contras que necessitam de ação futura:

Itens que serão alterados para evolução do método:

- Sensação de redução no número de discussões técnicas. Anteriormente à implementação do método, existia uma reunião semanal de status em que as discussões técnicas aconteciam. Com a utilização do Scrum + Kanban essa reunião deixou de existir, forçando que as discussões acontecessem fora das reuniões diárias. Isso trouxe a percepção para alguns membros do time de que as discussões técnicas não estavam acontecendo mais;
- Apesar de menores, houve um aumento no número de reuniões. Aplicar um método que muda uma reunião semanal com maior duração para uma reunião diária rápida traz a percepção que temos mais reuniões;
- Para o time, tivemos um maior número de impedimentos. Antes, os impedimentos ficavam concentrados no desenvolvedor, gestor ou *team leader*. Com o novo método esses impedimentos aparecem para todo o time.

A **Figura 2** mostra o quadro de trabalho do projeto piloto.



**Figura 2.** Quadro de trabalho durante o projeto piloto.

[subtitulo]Resultados Atingidos[/subtitulo]

Depois de concluído o projeto piloto e tomando como base os bons resultados, o método foi distribuído para os demais grupos de manutenção. Após um ano de implementação decidimos comparar o método clássico (não ágil) e o ágil a fim de identificar ganhos com o uso dessa metodologia na manutenção de software. Essa comparação é apresentada na **Tabela 8**.

	Trimestre 3, 2009	Trimestre 3, 2010	Comentários
Número de colaboradores do time de manutenção	50	53	-
Número de OS recebidas	250	341	-
Número de OS respondidas	230	356	Algumas OS abertas anteriores ao trimestre 3 foram contadas como OS respondidas, justificando o motivo de se ter respondido mais OS do que o número de novas OS.
Tempo médio de resposta por OS (dias)	26,36	19,15	Desempenho melhorou 27%

**Tabela 8.** Comparação de performance entre dois trimestres.

A partir da implementação da metodologia ágil na operação de manutenção foi possível observar que:

- O desempenho mês a mês foi igual ou melhor do que antes dessa adoção;
- O time de manutenção mostrou-se mais comprometido com o trabalho a ser feito e mais rápido na resposta a interferências externas (como por exemplo: aumento do número de OS e necessidade de ajudar outros projetos dentro da organização);
- Maior satisfação do cliente com relação a entregas do que é considerado de valor, possibilitando um rápido retorno de investimento;
- Maior agilidade no ajuste dos times, possibilitando ações que tenham resultado imediato.

#### [subtitulo]Conclusão[/subtitulo]

Com mais de um ano de implantação do projeto piloto e da adoção da metodologia ágil na operação de manutenção foi possível observar uma melhora na performance, na motivação e principalmente em como o cliente tem percebido o trabalho do grupo de manutenção. Além disso, o grupo viu nesse método a chance de um gerenciamento colaborativo, incentivado pela relação mais próxima entre colaborador e cliente na busca de uma melhoria constante com a participação de todos.