



ENGENHARIA DE SOFTWARE I

TRABALHO DE METODOLOGIAS

SCRUM

Prof.: Me. Zanoni

Camila Alcaldes Leal

Elisangela Peterson

Sérgio Tales Potrich

Geovane de Oliveira

Yonathan Stein

Índice

1.0 Introdução

1.1 Metodologia

1.2 Manifesto Ágil

1.3 Os 12 Princípios do Manifesto Ágil

2.0 Desenvolvimento

2.1 Scrum

2.2 Papéis do Scrum

2.3 Product Owner (PO)

2.4 Scrum Master

2.5 Equipe de Desenvolvimento

2.6 Fluxo do Processo Scrum

2.7 Product Backlog

2.8 Sprints

2.9 Sprint Planning

2.10 Exemplo

2.11 Benefícios

3.0 Conclusão

3.1 Conclusão

4.0 Bibliografia

4.1 Engenharia de Software – Uma Abordagem Profissional

4.2 Guia do Scrum

4.3 Fonte quadrinhos

4.4 Um Modelo Ágil para Gestão de Projetos de Software.

1.0 Introdução

1.1 Metodologia

A adoção da gestão de projetos também é uma busca pela previsibilidade. Quando nós temos uma ideia e queremos que ela se transforme em algo real nós fazemos projetos. Para fazermos um projeto e obter qualidade nós usamos uma metodologia, que nada mais é do que princípios e regras que regulamentam o gerenciamento de projetos na organização. Existem várias metodologias de gerenciamento de projetos, mas nesse trabalho mostraremos a metodologia ágil com o framework Scrum.

1.2 Manifesto Ágil

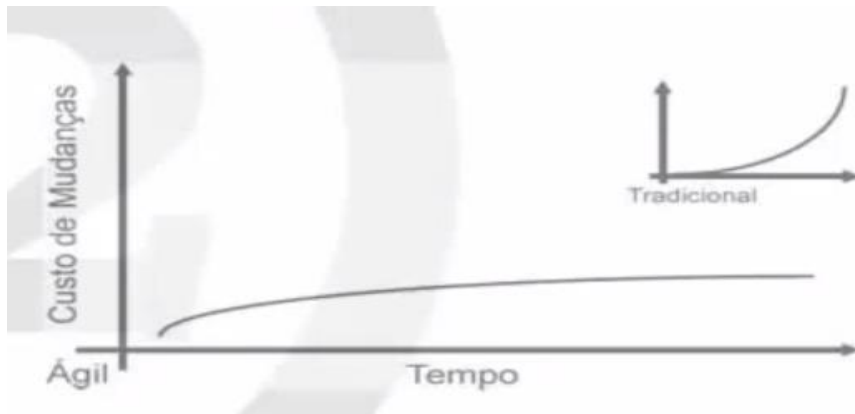
As definições modernas de desenvolvimento ágil evoluíram a partir de 1990 com uma reação contra métodos pesados. O manifesto ágil veio com algumas quebras de conceito que poderiam parecer irrealistas para muitas pessoas alguns anos, mas hoje se mostra mais forte e funcional para vários tipos de projetos. Muitas vezes uma empresa poderia achar um absurdo começar a trabalhar sem qualquer documentação, mas isso é possível quando aplicamos a metodologia ágil, ela defende a conversa com o cliente para ele dizer o que quer e chegar mais rápido ao ponto final do que ler um documento e tentar entender em que ponto ele quer chegar. Existe documentação na metodologia ágil, mas ela prega por um software em funcionamento do que uma documentação extensa, ou seja, deve-se apenas fazer a documentação necessária para se ter um software funcional.

Outro ponto é a interação da equipe, a relação entre as pessoas é mais importante e deve ser mais bem tratada do que as máquinas e os programas porque o que faz eles funcionarem são as pessoas. É muito importante também ter um contrato firmado com regras claras, porém negociar é muito mais importante do que brigar judicialmente com seu cliente, seja claro e transparente com ele, sempre dê a ele um feedback sobre o que está acontecendo no produto. Outra quebra de conceito são as mudanças, elas são inevitáveis. Não adianta lutar contra elas porque iram aparecer e precisam ser tratadas e devemos nos adequar a elas.

1.3 Os 12 Princípios do Manifesto Ágil

Existem 12 princípios do manifesto ágil e foi a partir desses princípios e do próprio manifesto que surgiu o framework Scrum e outras práticas ágeis.

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor;
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas;
3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferencia aos períodos mais curtos;
4. Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto diariamente, durante todo o curso do projeto;
5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão o seu trabalho;
6. O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara;
7. Software funcionando é a medida mais primária de progresso;
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter indefinidamente passos constantes;
9. Contínua atenção à excelência técnica e bom design, aumenta a agilidade;
10. Simplicidade: A arte de maximizar a qualidade de trabalho que não precisou ser feito;
11. As melhores arquiteturas, requisitos e designs emergem de times auto organizáveis;
12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo;



2.0 Desenvolvimento

2.1 Scrum



A imagem acima é de uma partida de rugby. Pode parecer que está havendo um caos, mas todas as pessoas estão participando sobre um bem comum, todos eles lutam por um mesmo objetivo, que é capturar a bola. Apesar de o nome Scrum ter sido tirado do jogo de rugby ele não se baseia no jogo, na verdade ele se baseia na Toyota que tem o pensamento Lean thinking.

Lean Thinking (ou Mentalidade Enxuta) é uma filosofia e estratégia de Negócios para aumentar a satisfação dos clientes através da melhor utilização dos recursos. A gestão lean procura fornecer, de forma consistente, valor aos clientes com os custos mais baixos (PROPÓSITO), identificando e sustentando melhorias

nos fluxos de valor primários e secundários (PROCESSOS), por meio do envolvimento das pessoas qualificadas, motivadas e com iniciativa (PESSOAS). O foco da implementação deve estar nas reais necessidades dos negócios e não na simples aplicação das ferramentas lean.

2.2 Papéis do Scrum

Existem três papéis no Scrum:



"Quem é o gerente no Scrum"? Não tem gerente. Pelo menos não na forma tradicional. O que temos no Scrum é uma gerência distribuída:

- O Product Owner é responsável pela macro gerência, ou seja, ele é o responsável por gerenciar "O Quê" deve ser feito.
- A Equipe de Desenvolvimento é responsável pela micro gerência, ou seja, ela é responsável por gerenciar "Como deve ser feito".
- O Scrum Master é responsável por gerenciar o processo, ou seja, garantir que os valores do Scrum estão sendo postos em prática.

2.3 Product Owner (PO)

Product Owner representa e defende os interesses do cliente. Ele é o interessado e preocupado com a preparação desse produto e com o resultado que será obtido no final do projeto.

- Materializa a visão do produto
- Estabelece os marcos do projeto (Releases)
- Garante o ROI (Retorno Sobre Investimento) do produto criado
- Dá feedback contínuo ao Time Scrum
- Captura necessidades e gerencia dos *stakeholders*

2.4 Scrum Master

O Scrum Master está a serviço do Product Owner e da Equipe de Desenvolvimento. Ele não é um chefe como pode parecer, ele serve para ajudar o time.

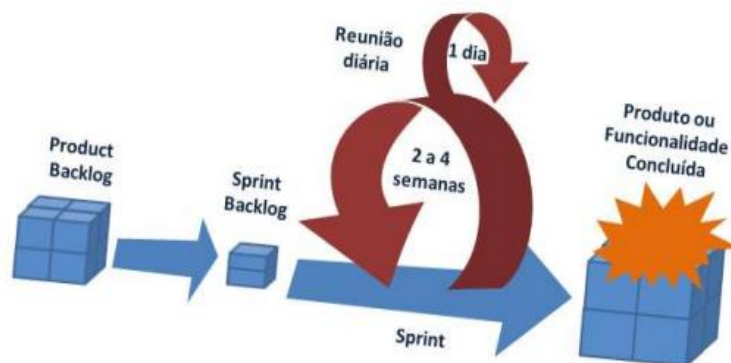
- Líder servidor
- Coach
- Melhor amigo da equipe de desenvolvimento
- Melhor amigo do Product Owner

2.5 Equipe de Desenvolvimento

Responsável pela execução direta pelos itens de processo. Ao contrário do que muitas vezes ocorre de o chefe pedir algo e dar um prazo, quem vai dar esse prazo vai ser a equipe de desenvolvimento que vai dizer quando pode ser entregue.

- Constrói o produto com qualidade
- Trabalha de forma auto organizada
- Prioriza o coletivo acima do individual

2.6 Fluxo do Processo Scrum



O Product Owner tem uma visão do que ele quer criar (o grande cubo). Como o cubo pode ser grande, por meio de uma atividade chamada Grooming, ele é

dividido em um conjunto de funcionalidades que são compilados em uma única lista priorizada chamado de Product Backlog.

Então é feita a primeira reunião de Planejamento de Sprint, para definir o Sprint Backlog, que contém todo o trabalho que será executado durante o Sprint. O Sprint tem duração média de 2 a 4 semanas e são feitas reuniões diárias de acompanhamento (Daily Scrum) do trabalho.

2.7 Product Backlog

No Scrum, sempre fazemos o trabalho mais importante primeiro. O Product Owner, com ajuda do resto da equipe Scrum e as partes interessadas, é o responsável por determinar e gerir a sequência deste trabalho e comunicando-o na forma de uma lista de prioridades conhecida como o Product Backlog.



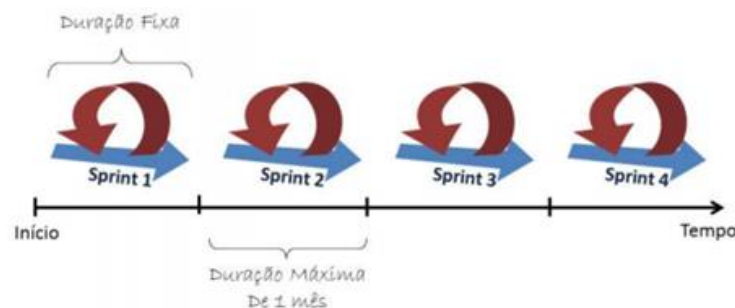
O Product Owner, em conjunto com as demais partes interessadas no produto, definem os itens do Product Backlog. Em seguida, ele garante que os itens do Backlog são colocados na sequência correta (usando fatores como valor, custo, conhecimento e risco), de modo que os itens de alto valor aparecerão no topo do backlog do produto e os itens de menor valor aparecerão em direção ao fundo.

O Product backlog é um documento que está constantemente evoluindo. Os itens podem ser adicionados, excluídos e revisto pelo Product Owner por conta de mudanças nas condições de negócios, ou conforme a compreensão da equipe Scrum sobre o produto aumenta. Em geral a atividade de criar e de refinar os itens do product backlog, estimando o tamanho e esforço de cada item, é chamada de Grooming.

Antes de finalizar a priorização, ou refinamento do produto backlog, é preciso saber o tamanho de cada item. É importante que o Product Owner saiba o custo de cada item para que possa determinar a sua prioridade de forma adequada. O Scrum não especifica como você deve medir o tamanho dos itens do backlog. Na prática, muitas equipes usam uma medida de tamanho relativo, como Story Point ou dias ideais. Essa questão sobre estimativas é um capítulo a parte e cabe um post depois para explicar somente isso.

2.8 Sprints

No Scrum, o trabalho é realizado em iterações ou ciclos de até um mês de calendário chamado de Sprints. O trabalho realizado em cada sprint deve criar algo de valor tangível para o cliente ou usuário. Sprints são timeboxed (duração fixa) para que tenham sempre um início e fim data fixa, e, geralmente, todos eles devem estar com a mesma duração.



Um novo Sprint segue imediatamente a conclusão do Sprint anterior e, via de regra, não devemos permitir nenhuma alteração de escopo ou pessoal durante um Sprint (mas por experiência própria posso afirmar que esta regra é quase sempre quebrada devido a algumas necessidades de negócio).

2.9 Sprint Planning

O product backlog pode representar muitas semanas ou até meses de trabalho, o que é muito mais do que pode ser concluído em um único e curto sprint. Para determinar quais os subconjuntos de itens do Product Backlog mais importantes para construir no próximo sprint, o product owner, junto com o time de

desenvolvimento e ScrumMaster, devem realizar o Sprint Planning (planejamento de sprint).

Durante o planejamento do sprint, a equipe de desenvolvimento e o product owner devem chegar a um acordo sobre qual o Objetivo do Sprint. Com este objetivo em mãos, eles determinam quais os itens do backlog devem ser priorizados para serem executados neste Sprint. A maioria das equipes Scrum que estão realizando Sprints de duas semanas a um mês de duração tentam completar o planejamento do sprint em cerca de 4 a 8 horas. Um sprint de uma semana não deve tomar mais do que 2 horas para planejar.

2.10 Exemplo

Scrum em quadrinhos:

DISCOVERY 1 – O PO é uma ampliação do papel do analista de negócios, pois agora ele passa a ser o representante do cliente, com foco em ROI e valor:



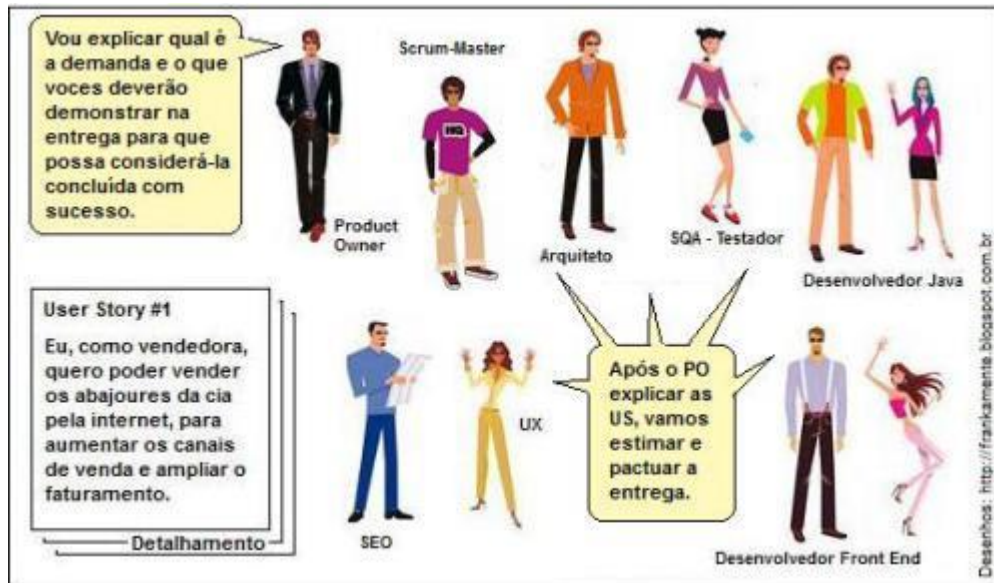
DISCOVERY 2 – O Product Owner conta com o trabalho dos papéis de UX e SEO para conceitos de comunicação, ergonomia, identidade, relevância, etc:



DISCOVERY 3 – O PO deve envolver o arquiteto ou integrante mais sênior da equipe, além do testador ou SQA, para validar e alinhar os requisitos mais técnicos e que poderiam inviabilizar a US da forma como foi descrita:



PLANNING – Após a User Story detalhada e aferida a contento, o product owner está em condições de apresentá-la a equipe, para estimativas e pacto:



CONSTRUÇÃO – O trabalho em sinergia entre os desenvolvedores e testador é fundamental para a entrega de software com qualidade e valor pactuados:



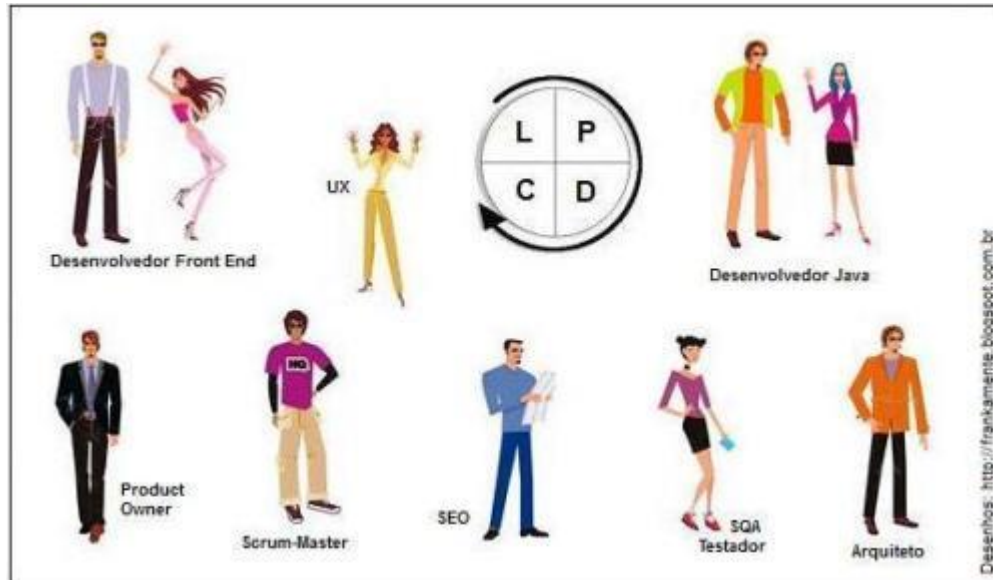
REUNIÃO DIÁRIA – A cada dia, sempre no mesmo local e horário, o time se reúne em frente ao quadro de tarefas para ajustar o rumo e corrigir desvios:



REVIEW – A cada término de um ciclo de 2 a 4 semanas, o time convida seus principais usuários e *stakeholders* para apresentar o que foi construído, o debate de 2 horas irá proporcionar maior entendimento e convergência entre os times:

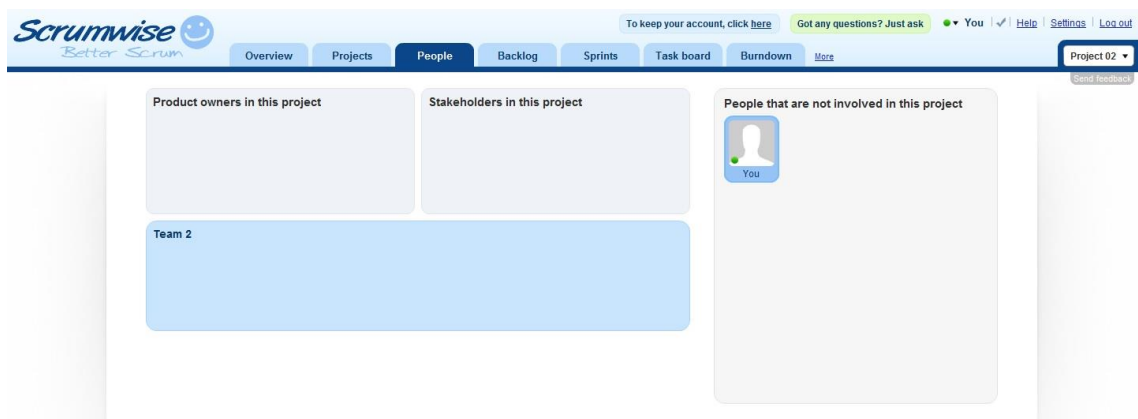
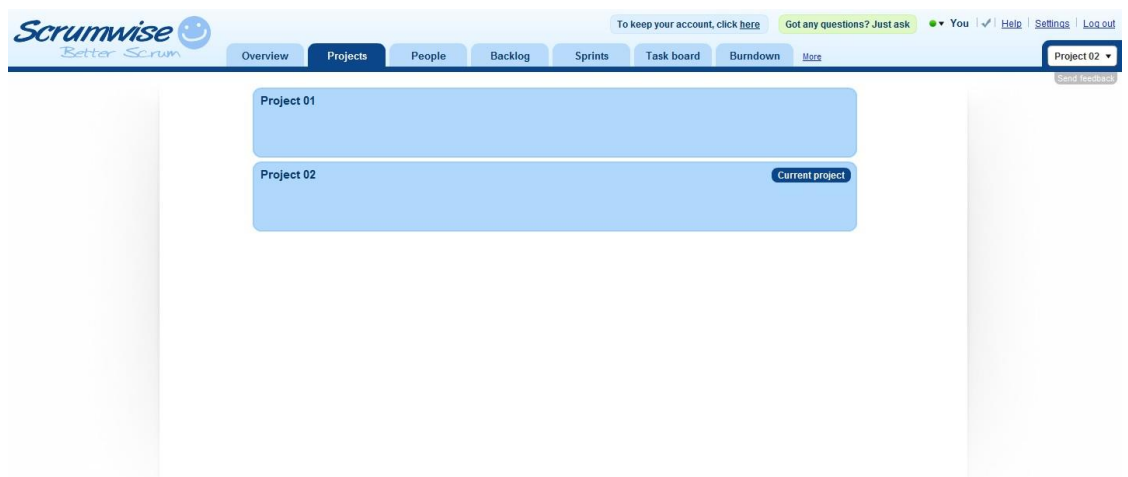


RETROSPECTIVA – A cada ciclo de 2 a 4 semanas de trabalho, a equipe se reúne por 2 horas para refletir e debater sobre como trabalharam e como melhorar, usando dinâmicas de integração e auto avaliações técnicas:






Scrum online (<https://www.scrumwise.com/>)



The image shows a login form for Scrumwise. The form is enclosed in a blue rounded rectangle on a light gray background. At the top, the Scrumwise logo is displayed, featuring the word "Scrumwise" in a blue script font, the tagline "Better Scrum" in a smaller blue font, and a blue smiley face icon. Below the logo, there are two input fields: "Email" and "Password". Under the "Password" field, there is a checkbox labeled "Remember me". At the bottom left of the form, there is a link that says "I forgot my password". At the bottom right, there is a green button with a white right-pointing arrow and the text "Log in".




Item01 

Sprint 1  

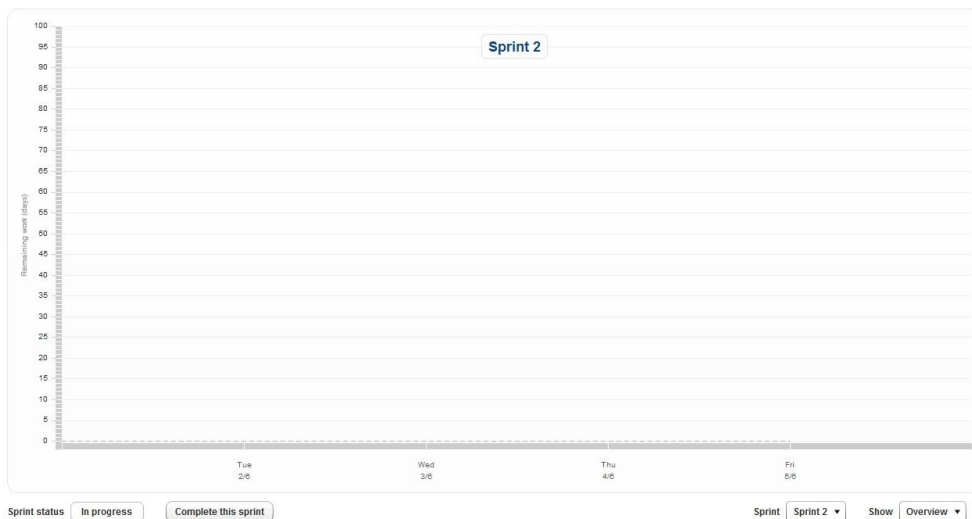
Team 2 

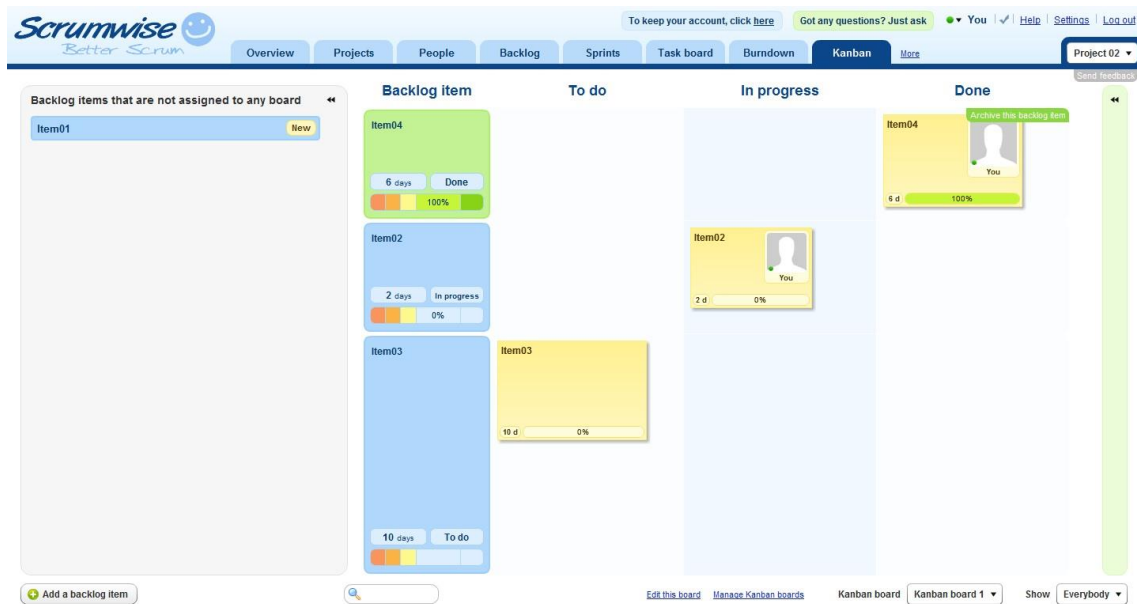
Sprint 2  

Team 2 

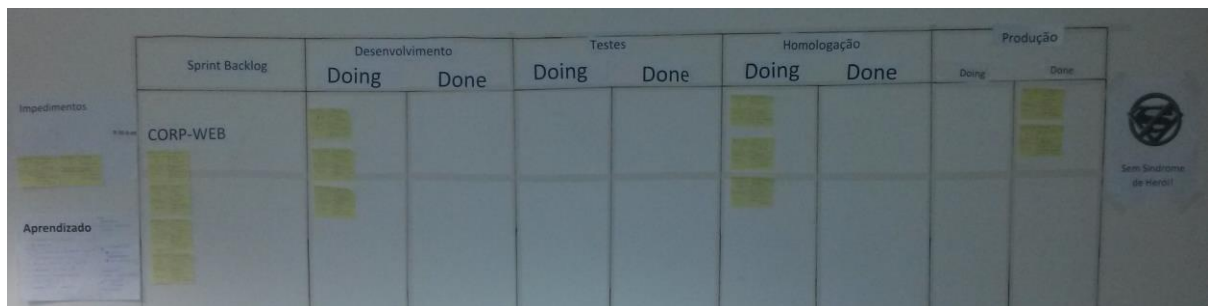
Backlog items that are not assigned to any sprint

Item01 





Exemplo prático utilizado em uma empresa Porto Alegre.



2.11 Benefícios

Esses são alguns dos benefícios obtidos com o uso da metodologia SCRUM:

- Diminuição dos riscos;
- Maior integração entre os membros das equipes;
- Rápida solução de problemas;
- Progresso medido continuamente;
- Os clientes se tornam parte da equipe de desenvolvimento;
- Entregas frequentes de funcionalidades funcionando;
- Discussões diárias de status com a equipe;
- Os profissionais de negócios e tecnologias trabalham juntos.

3.0 Conclusão

A metodologia Scrum tem como principal objetivo reduzir dificuldades como falta de planejamento, mudança constante de requisitos, escopo mal definido, falta de participação do cliente e falhas na comunicação, que são comuns em projetos de tecnologia. Além disso, o Scrum se baseia em entregas rápidas, contínuas e frequentes de softwares funcionais, constante cooperação entre equipe de trabalho e de negócio. Com o entrosamento entre a equipe de desenvolvimento e com a participação ativa dos clientes o rendimento do projeto aumenta e os requisitos e a solicitação de alteração passam a ser entendidos mais rapidamente.

4.0 Bibliografia

4.1 “Engenharia de Software – Uma Abordagem Profissional”

Autor: Roger S. Pressman

4.2 “Guia do Scrum”

Autores: Ken Schwaber e Jeff Sutherland

4.3 “Fonte quadrinhos”

Disponível em:

<https://jorgekotickaudy.wordpress.com/2012/07/17/quem-e-quem-a-revanche/>

4.4 “Um Modelo Ágil para Gestão de Projetos de Software.”

Autores: FERREIRA, D.; COSTA, F.; ALONSO, F.; ALVES, P.; NUNES

Disponível em: http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_19.pdf