

Scrum – Mais um caso de sucesso

Alvarino R. Junior

Universidade do Planalto Catarinense (UNIPLAC)
Avenida Marechal Castelo Branco, 170 – 88.509-300 – Lages – SC – Brasil

alvarinojr@gmail.com

Abstract. *With increasingly being asked and with greater complexity and shorter delivery software, it is necessary that its development is modified. Using agile methods can improve the quality and performance of the software development process. The purpose of this article is to present a case study on the deployment and use of Scrum, which has emerged as a leading agile methods in recent years, the implementation done in a software development company with a team of seven developers who used the unified process as a method of development. The results are reported various experiences of this deployment forming a successful case.*

Keywords: Software development, Agile methods, Scrum, Agile practices

Resumo. *Com o software sendo cada vez mais requisitado e com uma maior complexidade e tempo de entrega mais curto, faz-se necessário que seu desenvolvimento seja modificado. Utilizar métodos ágeis pode melhorar a qualidade e o desempenho no processo de desenvolvimento de software. O objetivo deste artigo é apresentar um estudo de caso sobre a implantação e utilização do Scrum, que tem se destacado como um dos principais métodos ágeis nos últimos anos, a implantação feita em uma empresa de desenvolvimento de software com uma equipe de sete desenvolvedores que utilizava o processo unificado como método de desenvolvimento. Como resultados são relatadas diversas experiências dessa implantação formando um caso de sucesso.*

Palavras-chaves: Desenvolvimento de software, Metodologias ágeis, Scrum, Práticas ágeis

1. Introdução

Em geral os maiores problemas enfrentados por uma empresa de desenvolvimento de software são os atrasos na entrega do projeto, produtos com baixa qualidade, aumento significativo dos custos. Tais problemas muitas vezes são causados pela falta de um método de desenvolvimento bem definido que não permite mudanças, causando assim um impacto econômico e humano enorme para a empresa. Apenas um terço aproximadamente dos projetos de software no mundo são bem sucedidos, segundo o Standish Group, no seu estudo intitulado “The Chaos Manifest”, que vem medindo ano a ano os resultados da indústria de software desde 2004. Os critérios para que os projetos sejam considerados bem sucedidos incluem serem concluídos dentro do prazo, do custo e com escopo previsto.

Os métodos ágeis estão sendo cada vez mais comentados na engenharia de software, neles as equipes buscam pelos benefícios que a agilidade oferece: software entregue mais rápido, qualidade, flexibilidade é tudo que uma empresa e seus clientes necessitam. O desenvolvimento ágil oferece ao time a capacidade de responder rapidamente as mudanças do projeto como requisitos mutáveis e não bem esclarecidos. Entretanto, implantar a agilidade em uma empresa que utiliza um método tradicional de desenvolvimento não é uma tarefa

trivial, mudar essa cultura é um grande desafio e exige bastante esforço. Segundo McBreen (2002) mudanças no processo tendem a encontrar uma maior resistência por parte dos membros da equipe, pois esses precisam mudar seus valores e atitudes. Há ainda os aspectos culturais da empresa que devem ser analisados, o predomínio do trabalho individual e a falta de comunicação do time, isso porque o desenvolvimento ágil prega a filosofia de trabalho em equipe e a comunicação entre ela, e mudar essa cultura é primordial para obter sucesso na implantação de um método ágil. Enquanto o método não estiver totalmente assimilado pelos membros da equipe é comum encontrar dificuldade em executar suas tarefas, acarretando uma queda na produtividade. Há quem diga que os métodos tradicionais são melhores, o que acontece é que em alguns casos uma ou outra metodologia irá se adaptar melhor ao contexto da empresa (NERUR et al, 2005).

Neste artigo, o objetivo é apresentar um estudo de caso sobre a implantação e utilização do Scrum. São apresentados os conceitos de métodos ágeis, seus princípios e valores bem como suas práticas e possíveis dificuldades na utilização. São descritas as experiências na implantação como, por exemplo, as reuniões diárias que permitem aos membros do time encontrar soluções diretas, pois há envolvimento de todos, tornando-os mais efetivos na realização de seu trabalho.

O Scrum foi escolhido para implantação por ser a mais utilizada no desenvolvimento de software pelas empresas que adotam os princípios ágeis. Essa escolha foi devido à adaptabilidade oferecida pela metodologia e por responder rapidamente às constantes mudanças nos projetos de software.

2. Métodos Ágeis

Diante do desafio de responder com mais rapidez às mudanças, engenheiros de software, percebendo fraquezas na engenharia de software convencional, buscaram soluções para inovar os processos de desenvolvimento de software, originando o que foi denominado de “movimento ágil”.

O manifesto ágil foi criado em 2001 e assinado pelos principais profissionais veteranos na área de desenvolvimento de softwares. Eles se uniram para discutirem uma nova forma para melhorar a velocidade no desenvolvimento de seus sistemas tendo como base as suas experiências de anos programando. Esse manifesto é uma declaração simples e concisa que busca mudar a visão tradicional de desenvolvimento de software. Além disso, ele acentua o valor dos indivíduos e interações serem mais importantes que processos e ferramentas; do software funcionando ser mais importante que documentação detalhada; da colaboração dos clientes ser mais importante que negociação de contratos; e da adaptação às mudanças ser mais importante que seguir um plano. Na tabela 1 estão descritos os valores do manifesto ágil.

Tabela 1. Valores dos métodos ágeis (Fonte: Agile Alliance, 2013)

<p>Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:</p> <p>Indivíduos e interações mais que processos e ferramentas Software em funcionamento mais que documentação abrangente Colaboração com o cliente mais que negociação de contratos</p>
--

Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita,
valorizamos mais os itens à esquerda.

Os princípios ágeis que regem o sucesso de um projeto de desenvolvimento ágil de software, seus princípios são listados abaixo:

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

Alistair Cockburn (2002) define o termo metodologia como uma série de métodos e técnicas, onde método se refere a procedimentos sistemáticos. Já Martin Fowler (2003) diz que metodologias impõem um processo disciplinado no desenvolvimento de software, com o objetivo de torná-lo mais previsível e mais eficiente. Os métodos ágeis foram criados a partir da necessidade da mudança constante de requisitos e do mau desempenho que os “processos pesados” proporcionavam ao desenvolvimento de software.

As metodologias ágeis propõem que o software seja desenvolvido de forma incremental com iterações curtas, a ideia é que a cada final de iteração seja entregue um produto pronto para ser utilizado pelo cliente. Essa entrega rápida permite ao cliente avaliar o produto logo no início e dar seu feedback sobre as funcionalidades desenvolvidas. Pode-se dizer que os métodos ágeis se caracterizam por serem incrementais, cooperativos, diretos e adaptativos. Incrementais, dadas as pequenas versões e rápidas iterações; cooperativos, por estimular a proximidade com o cliente e a interação entre os programadores; diretos, pela

simplicidade de aprendizado e de documentação; e, finalmente, adaptativos, pela habilidade de acomodar mudanças ao longo do projeto (Abrahamsson et al, 2003; Fowler, 2003).

Os benefícios da utilização de métodos ágeis envolvem a capacidade de reação a mudanças constantes, a colaboração com o cliente, processos eficientes para gerar produtos de qualidade (que atendem ao cronograma, possuem menos defeitos e que resultam em usuários mais satisfeitos), aprendizagem e melhoria contínua do projeto. Funcionários de empresas que utilizam um método ágil relatam que ficaram mais satisfeitos com o resultado do trabalho e produto final. Após a implantação do método há um aumento da velocidade de tomada de decisão dos membros da equipe que se torna auto-gerenciável, o que acaba refletindo numa maior competitividade organizacional.

Os métodos podem ser utilizados por grandes ou pequenos projetos, e são indicados geralmente para equipes pequenas. Os mais populares são:

- Extreme Programming: Baseada em cinco valores principais (comunicação, feedback, simplicidade, coragem e respeito);
- Scrum: Focado nos aspectos da gerencia de projetos de software, dividindo o desenvolvimento em iterações;
- Crystal: é uma família de metodologias aplicadas de forma diferente dependendo do tamanho da equipe. Possui três prioridades: segurança, eficiência e adaptabilidade;
- Learn: Inspirado no modelo de manufatura da Toyota, para a gerencia de software utilizando kaizen e kanban.

Segundo Paulk (2002) métodos ágeis devem ser utilizados por todos os profissionais de desenvolvimento, mas sem deixar de lado o formalismo e a disciplina, principalmente para softwares que exigem requisitos de confiabilidade.

2.1 Quando aplicar métodos ágeis

De forma a determinar a aplicabilidade de métodos ágeis específicos, uma análise mais sofisticada é necessária. O método dinâmico para o desenvolvimento de sistemas, por exemplo, provê o denominado 'filtro de aplicabilidade' para este propósito. A agilidade pode ser aplicada em empresas que:

1. As responsabilidades das pessoas não estão bem definidas, ou, nas palavras do autor, naquelas organizações onde a “descrição de cargos é ampla”;
2. Os processos são definidos por cada equipe de forma independente;
3. Inexiste um manual com políticas da organização;
4. Os processos são secundários em relação às pessoas.

2.2 Dificuldades Comumente Encontradas

Antes de implantar um método ágil é necessário fazer um levantamento sobre as dificuldades que serão encontradas durante a implantação. É preciso conhecer o ambiente que irá sofrer as mudanças. Essa mudança exige desaprender valores, premissas e comportamentos antigos antes que se possam aprender os novos. Os elementos mais importantes dessa mudança cultural são o apoio executivo e treinamento. Alguns tipos de impedimentos comumente encontrados:

- Gerentes tradicionais: medo de mudar, perder o controle, perda da autoridade;
- Arquitetos: medo de deixar a arquitetura criada por ele na mão de todos, perda de autoridade;
- Programadores: quebra da rotina, resistência ao novo, medo de novas responsabilidades;
- Testadores: medo de perder o emprego;
- DBAs: eu sou quem modelo o Banco, vocês apenas escrevem o código;
- Clientes: resistência a estar presente no desenrolar do projeto, demora para dar um *feedback*;
- Refatoração: o código não é refatorado por culpa do tempo gasto com isso;
- TDD: Testes não são criados pelos programadores por pensar que é uma perda de tempo.

O autor Alistair Cockburn (2002) sugere a realização de Workshops de Reflexão com a equipe de desenvolvimento durante o uso de um método. Nesses workshops são discutidos os prós e contras da metodologia em uso, bem como são relacionados possíveis desconfortos no seu uso ou sugeridas possíveis modificações em sua estrutura e práticas. A comunicação é fundamental para o sucesso da implantação. As dificuldades são muitas, mas as soluções também, assim é necessário fazer uma coleta de requisitos detalhada e um planejamento completo antes de começar a implantação.

3. Scrum

O Scrum é um dos modelos de desenvolvimento ágil mais utilizado atualmente, foi criado em 1996 por Ken Schwaber, Jeff Sutherland e Mike Beedle, e o nome vem de uma atividade que acontece em partidas de rugby. Tem como base o desenvolvimento iterativo e incremental, o trabalho com equipes pequenas de no máximo sete pessoas, a flexibilidade dos resultados, dos prazos, revisões frequentes, colaboração dos envolvidos, e destaca-se pela maior ênfase dada ao gerenciamento do projeto. O gerenciamento é feito a partir de iterações que tem como objetivo decompor um grande produto complexo, em pequenos subprodutos mais simples, mais rápidos e mais fáceis de serem desenvolvidos e entregues. Espera-se que ao final de cada iteração seja possível obter um produto apresentável ao cliente, esse processo é chamado de Sprint e costumam durar de 2 semanas a 1 mês, e é considerado como o principal evento do Scrum.

3.1 Porque Scrum?

A seleção do Scrum foi baseada particularmente pelos seus princípios básicos de objetividade, papéis bem definidos e facilidade de aprendizado. Além disto, uma pesquisa de 2012 (VersionOne, 2013) sobre a implantação de métodos ágeis indicava que 70% dos entrevistados utilizavam Scrum combinado a outros métodos e 40% utilizavam somente Scrum.

O Scrum é um modelo aberto e de forma alguma é previsível (Schwaber K., 2004). Ele oferece um conjunto de práticas que tem como objetivo manter o gerenciamento do

projeto visível aos usuários do modelo. A metodologia não detalha o que deve ser feito e não resolve os problemas da empresa, pois o objetivo do Scrum é dar visibilidade a estes problemas e servir como guia na resolução dos mesmos. Os caminhos a seguir e estratégias a utilizar são de responsabilidade de quem o implanta.

De forma geral podemos identificar no Scrum alguns pontos chave:

- Times auto-organizados;
- Progresso do Desenvolvimento através de Sprints. O Sprint é o ciclo de desenvolvimento do Scrum, caracterizado por ter um curto período onde a equipe foca no atingimento de uma meta específica;
- Requisitos de produtos organizados em uma lista de itens chamada "Product Backlog";
- Conceito de Tempo Fechado (Timebox). Todas as tarefas dentro do Scrum tem tempos máximos definidos para a sua execução e não devem suplantar estes tempos.

3.2 Ciclo de Desenvolvimento

No Scrum o progresso do projeto é baseado em uma série de iterações bem definidas chamadas Sprints. A recomendação é de que a duração da Sprint fique entre 2 a 4 semanas. O Scrum é baseado em entrega de produto a cada Sprint logo, é recomendável que durante o desenvolvimento do produto exista um padrão e um mesmo time-box para todas as Sprints. Isto é importante para se conseguir medir o progresso e a produtividade da equipe no projeto.

Um projeto Scrum inicia com a definição de uma lista de requisitos e funcionalidades que se deseja para o software, essa lista é chamada de Product Backlog. À medida que o produto é desenvolvido e novas funcionalidades são solicitadas, estas são incluídas no Product Backlog de modo a manter uma lista de todas as funcionalidades desejadas para o software. Cada item dessa lista recebe um valor e esse valor indica a importância dessa tarefa para o projeto.

Uma Sprint deve iniciar com uma reunião de planejamento (Sprint Planning Meeting) que é dividida em duas partes. Na primeira parte o Product Owner tem a oportunidade de atualizar a priorização dos itens do Product Backlog, formando juntamente com a equipe o que pode ser entregue no final da Sprint. Na segunda parte a equipe define o Sprint Backlog, nele está contido as funcionalidades a serem desenvolvidas durante a Sprint. A equipe fica encarregada de selecionar as tarefas do Product Backlog de acordo com as suas necessidades e tempo de duração da Sprint. Participam dessa atividade o Product Owner, Scrum Master a equipe de desenvolvimento e qualquer outro interessado.

Durante o tempo de execução da Sprint a equipe faz uma reunião diária (Daily Scrum Meeting) que deve durar no máximo 15 minutos, nessa reunião os membros da equipe respondem a três perguntas: O que eu fiz no projeto desde a última reunião? O que irei fazer até a próxima reunião? Quais são os impedimentos?. Essas reuniões servem para verificar o andamento da Sprint e os participantes são o Scrum Master e os membros da equipe.

No final da Sprint é realizada a reunião de revisão (Sprint Review Meeting), nela é apresentado o resultado da Sprint ao Product Owner. Os participantes desta reunião incluem tipicamente: o Product Owner, o Scrum Team, o Scrum Master, a diretoria e clientes. Após a Sprint Review Meeting, o Scrum Master faz a reunião de retrospectiva (Sprint Retrospective Meeting) com a equipe, onde o objetivo é identificar os pontos positivos e negativos da última

Sprint e quais ações devem ser tomadas para colocar as melhorias em prática. Segundo (AGUIAR, 2008) essas reuniões costumam durar de três a quatro horas. O fluxo de desenvolvimento do Scrum é mostrado na Figura 1.

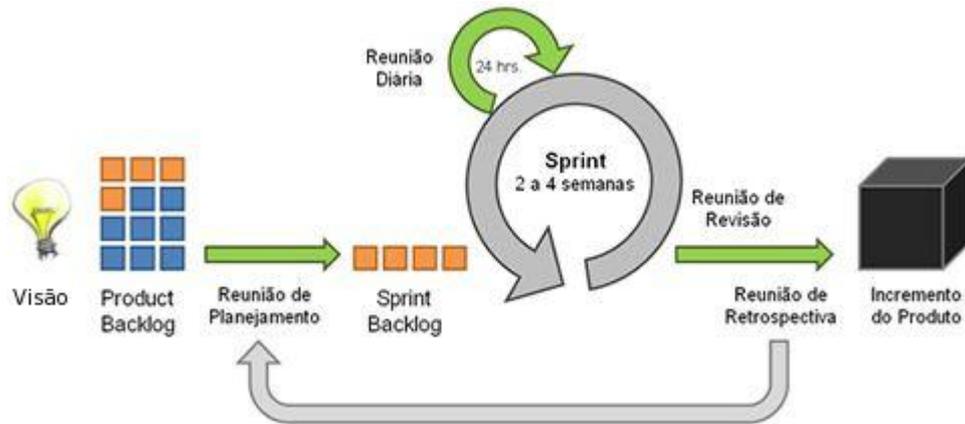


Figura 1. Fluxo do Scrum

3.3 Papéis e Responsabilidades

3.3.1 Scrum Master

O Scrum master é o líder da equipe de desenvolvimento, está em constante contato com o Product Owner, e é o responsável por gerenciar as demandas de trabalho para a equipe. Suas atribuições:

- Assegurar que a equipe de desenvolvimento funcione plenamente e seja produtiva;
- Ajudar na cooperação entre todas as funções e papéis do time;
- Remover impedimentos;
- Proteger a equipe de interferências externas;
- Assegurar-se de que a metodologia está sendo seguida.

3.3.2 Scrum Team

É a equipe de desenvolvimento, quem vai colocar a mão na massa para que o software comece a ter cara e funcionamento.

- Multi-funcional, entre 5-9 membros;
- Seleciona, entre os itens priorizados, os que irão ser executados durante a Sprint;
- Tem todo o direito de realizar o que quiser dentro da Sprint para cumprir o objetivo da iteração;
- Auto-organizado: organiza o time e o trabalho entre os membros de forma participativa;
- Ao final da Sprint, realiza o demo do produto finalizado.

3.3.3 Product Owner

Ele é o dono do produto, representa o cliente e é o responsável por determinar o que deve entrar para cada Sprint de acordo com os interesses do cliente, dentre suas atribuições podemos citar:

- Define os requisitos do produto, decide a data de release e o que deve conter nela;
- Prioriza os requisitos de acordo com o seu valor;
- Pode mudar os requisitos e prioridades a cada Sprint;
- Aceita ou rejeita o resultado de cada Sprint;
- É o responsável pela rentabilidade do produto.

3.3.4 Erros comuns com o Product Owner

É importante que o cliente esteja preparado para colaborar a equipe para o desenvolvimento do projeto. Erros comuns:

- Sobreposição de papel com o *Scrum Master*: não deixar que os anseios do cliente sobrecarreguem o time, adicionando novas funcionalidades e fazendo com que o planejamento seja perdido;
- Cliente com várias vozes: para projetos que tem divergências entre clientes, deve-se escolher um representante;
- Envolvimento Pontual: o cliente deve estar sempre disponível para tirar eventuais dúvidas sobre o projeto.

A Figura 2 apresenta um gráfico com a causa das principais falhas dos sistemas.

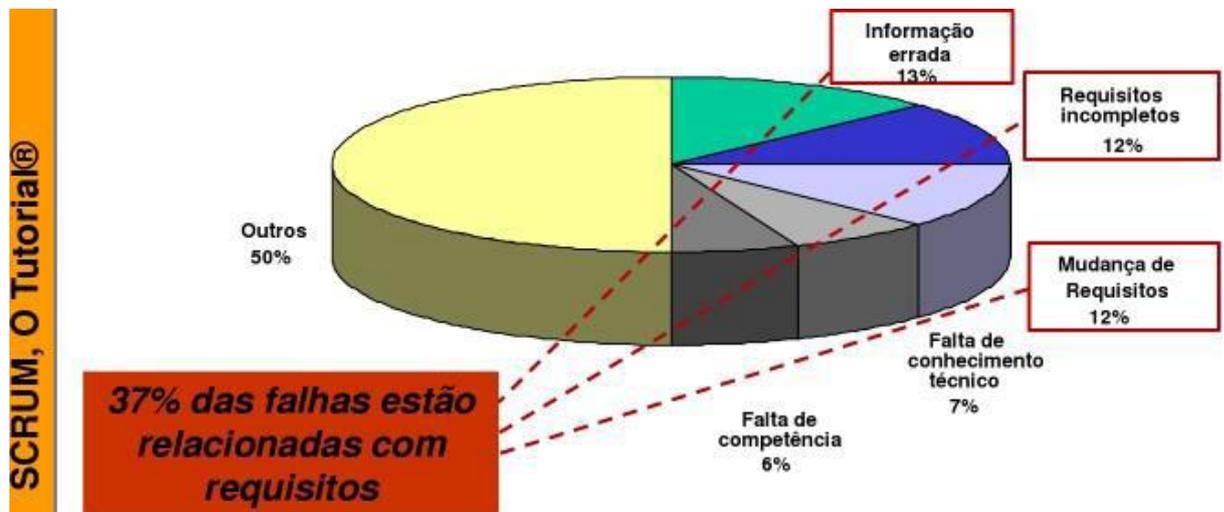


Figura 2. Falhas de sistema

3.4 Eventos

3.4.1 Reunião de Planejamento da Sprint

É o primeiro passo de um projeto Scrum. Essa reunião é onde a equipe faz a estimativa dos itens do backlog, o product owner apresenta à equipe as tarefas com maior valor para ele, e com isso a equipe juntamente com o scrum master efetua a estimativa de quanto esforço irá custar para desenvolver o item.

Quando as prioridades e prazos das funcionalidades do software são definidas por completo, o proprietário do produto sai de cena e o Scrum Master começa a trabalhar juntamente com a equipe de desenvolvimento, para fazer a quebra destas tarefas grandes em pequenas tarefas, divididas por todos os integrantes da equipe de desenvolvimento de acordo com suas especialidades. Esta reunião de planejamento geralmente dura até 4 horas e é ela quem define o Sprint Backlog.

3.4.2 Reunião Diária

Essa reunião é feita todos os dias e serve para atualizar os membros da equipe sobre o andamento da iteração. Nela os membros respondem a três perguntas:

- 1) O que foi feito desde ontem?
- 2) O que você planeja fazer para amanhã?
- 3) Você tem algum impedimento?

Para os impedimentos descritos pelos membros, deve-se tratar após a reunião com os envolvidos. Reuniões diárias melhoram as comunicações, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, destacam e promovem rápidas tomadas de decisão, e melhoram o nível de conhecimento da Equipe de Desenvolvimento. Esta é uma reunião chave para inspeção e adaptação.

3.4.3 Erros comuns em Reuniões Diárias

Alguns erros são cometidos durante a implantação do método. Esses erros por mais simples que possam ser, podem comprometer o desempenho que é esperado como resultado da implantação de um método ágil. São eles:

- Reuniões diárias a cada três dias: as reuniões devem ser diárias, pois elas fazem com que toda a equipe saiba o que cada um está fazendo. Por exemplo, se houver um problema, poderá relatar o mesmo e algum colega ajudar, assim economiza tempo e trabalho;
- Reunião com longa duração: as reuniões não devem passar dos 15 minutos, reuniões longas indicam que a comunicação entre a equipe não está indo bem.
- Na Sala de Reuniões: as reuniões devem ser feitas em frente a um quadro e de pé, para que não se prolongue muito e o quadro seja atualizado com facilidade, não devem ser feitas na sala de reuniões;
- Reunião de 2 minutos: a reunião deve durar entre 10 e 15 minutos, se estiver sendo mais rápida que isso pode indicar que ela está sendo feita apenas por protocolo;
- Detalhes e explicação: os problemas devem ser levantados sem muitas especificações técnicas, quem puder ajudar deve se manifestar e depois verificar junto ao colega a dificuldade que ele está enfrentando;

- Horário: é ideal que o horário das reuniões seja sempre o mesmo, assim todos da equipe podem se organizar para estarem disponíveis;
- Participação: a participação na reunião deve ser total e deve-se evitar falta de algum membro da equipe ou desatenção de algum deles.

3.4.4 Retrospectiva da Sprint

A Retrospectiva da Sprint é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint. Essa reunião ocorre normalmente antes da reunião de planejamento da próxima Sprint, os membros comentam o que foi bom, o que foi ruim e que sugestões ficam para a próxima iteração.

O propósito da retrospectiva da Sprint é:

- Inspecionar como a última Sprint foi, em relação às pessoas, relações, processos e ferramentas;
- Identificar e ordenar os principais itens que foram bem e as potenciais melhorias;
- Criar um plano para implementar melhorias no modo como o Time Scrum faz seu trabalho.

3.5 Artefatos

O framework Scrum é composto por várias cerimônias, que visam criar uma rotina e diminuir a quantidade de reuniões não definidas pelo framework. Os eventos são time-boxed, ou seja, possuem tempo de duração definido.

Os artefatos do Scrum foram idealizados para maximizar a transparência das informações, ou seja, para facilitar a visualização de todos os membros sobre o que está acontecendo.

Papéis, artefatos, eventos e regras do Scrum são imutáveis e embora seja possível implementar somente partes do Scrum, o resultado não é Scrum. Scrum existe somente na sua totalidade, funcionando bem como um container para outras técnicas, metodologias e práticas. (Scrum Guide, 2013)

3.5.1 Product Backlog

O Product Backlog contém uma lista de itens priorizados que incluem tudo o que precisa ser realizado e que possa ser associado com valor de negócio para a finalização do projeto, sejam requisitos funcionais ou não. É importante ressaltar que cada item no Backlog do produto deve ter um valor de negócio associado (Business Value), onde podemos medir o retorno do projeto e priorizar a realização dos itens.

Preparar o Backlog do Produto é uma atividade de tempo parcial, feita durante a Sprint, entre o Product Owner e a Equipe de Desenvolvimento. Geralmente a Equipe de Desenvolvimento tem o domínio do conhecimento necessário para realizar a preparação por si própria. Como e quando a preparação é considerada pronta é uma decisão do Time Scrum. Esta preparação usualmente não consome mais de 10% da capacidade da Equipe de desenvolvimento. A equipe de desenvolvimento é responsável por todas as estimativas. O Product Owner deve influenciar o Time, ajudando no entendimento e nas decisões conflituosas de troca, porém são as pessoas que irão realizar o trabalho que fazem a estimativa

final.

3.5.2 Planning Poker

O Planning Poker (COHN, 2006) é uma forma de estimativa em conjunto, podendo ser feita como um jogo (Figura 3). Todos os membros do time, inclusive o Product Owner, participam de forma democrática para chegar a um consenso de estimativa, para cada item do Backlog, de forma objetiva e divertida.

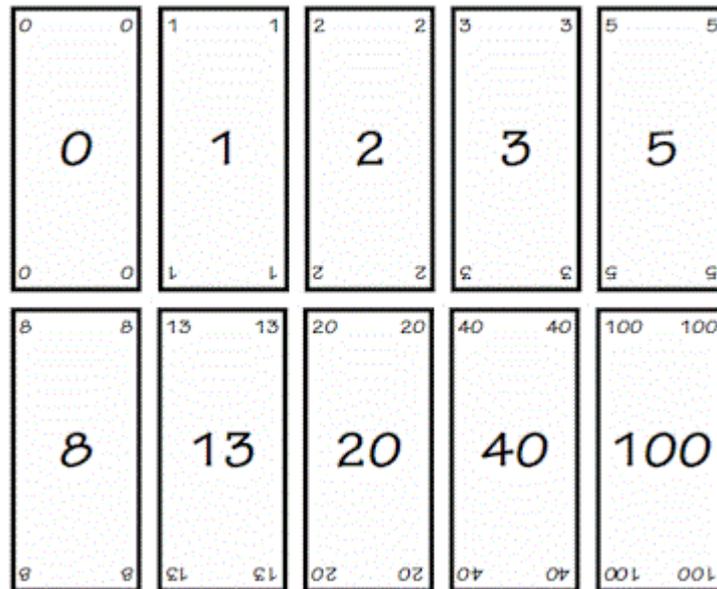


Figura 3. Cartas do planning poker

Para iniciar o jogo cada membro da equipe recebe um conjunto de cartas com valores referentes ao da sequência Fibonacci (1, 2, 3, 5, 8, 13, 21...). O scrum master seleciona um item dentre os selecionados para a Sprint, e então cada membro mostra a sua carta com o valor que acha plausível para o esforço da tarefa selecionada. Se todos os membros mostrarem o mesmo valor então o valor da carta é associado ao item, mas se os membros mostrarem divergências entre os números, então um membro que mostrou o valor mais baixo e um que mostrou o valor mais alto, defende a sua escolha e tenta convencer os demais que a sua escolha é a melhor. Depois disso mais uma rodada é feita e verifica-se se a equipe chegou em um consenso.

3.5.3 Backlog da Sprint

É um conjunto de itens do Backlog do Produto selecionados para a Sprint. O Backlog da Sprint é a previsão da Equipe de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e do trabalho necessário para entregar a funcionalidade.

O Backlog da Sprint é um plano com detalhes suficientes para que as mudanças em progresso sejam entendidas durante a reunião diária. A equipe de desenvolvimento modifica o Backlog da Sprint ao longo de toda a Sprint. Este surgimento ocorre quando a Equipe de Desenvolvimento trabalha segundo o plano e aprende mais sobre o trabalho necessário para alcançar o objetivo da Sprint.

3.5.4 Burndown Chart

O Burndown Chart é um gráfico responsável por acompanhar o andamento da Sprint, que mostra a quantidade de trabalho acumulativo restante de um Sprint, dia por dia. É baseado nas atualizações diárias da equipe, com o status das tarefas, e no número de horas restantes para a conclusão de cada tarefa. Neste gráfico, a altura indica a quantidade de tarefas do Sprint Backlog não completadas, e o comprimento são os dias. Com isto, podemos visualizar facilmente se um trabalho está tendo progresso completando suas tarefas, enquanto vemos que as colunas do gráfico vão caindo em sua altura.

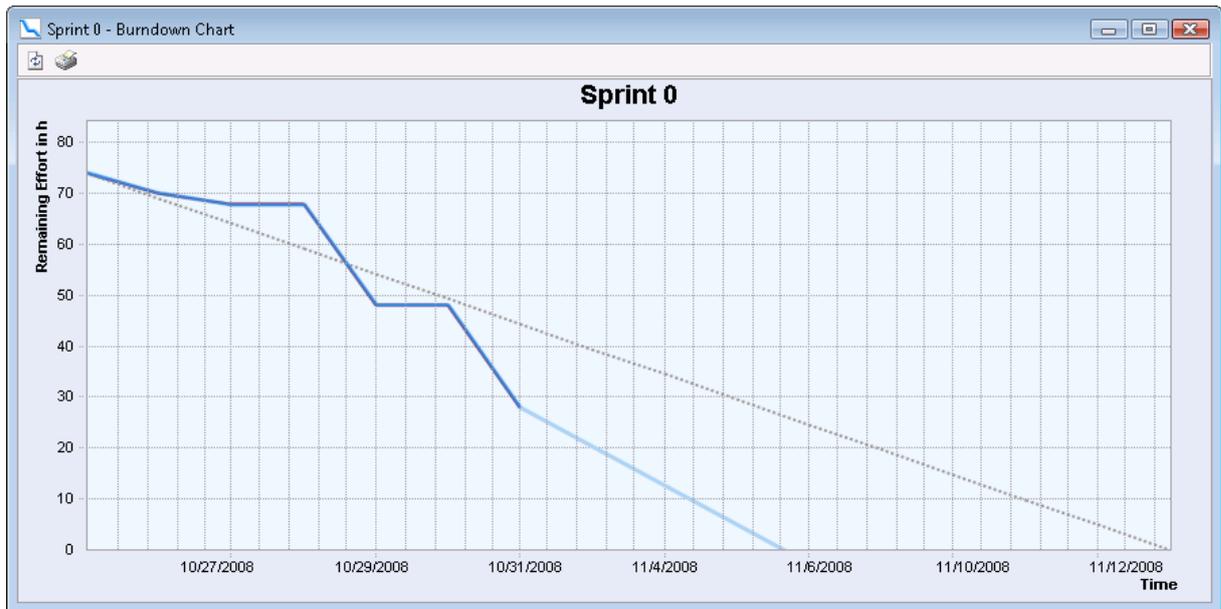


Figura 4. Gráfico burndown

3.5.5 Erros comuns com o Burndown Chart

É o gráfico utilizado por equipes scrum para medir o progresso da Sprint. Apresenta o tempo e a quantidade de trabalho restante até o final da Sprint. Seus erros comuns:

- Ausência ou abandono: não possuir gráfico desse tipo ou não o manter atualizado torna-se difícil avaliar o andamento do projeto até o seu prazo;
- Burndown para o product owner: o gráfico não deve ser utilizado pelo product owner para cobrar mais desempenho da equipe;
- Não ajustar os planos: caso alguma situação venha a mudar o tempo da Sprint, é necessário ajustar o gráfico.

4. Implantando o Scrum

4.1 A Empresa

A empresa Desktop X possui sete funcionários. O produto é um sistema de gerenciamento empresarial, onde o gerente apontou problemas com prazos de entrega, mudança de escopo e a falta de uma metodologia funcional no desenvolvimento de sistemas.

O ambiente foi analisado e foi feito um levantamento sobre todos os projetos em andamento, seus prazos e também a cultura atual da empresa. Antes da efetiva implantação da metodologia ágil Scrum, não existiam processos de gerência e de desenvolvimento de

software bem definidos/estabelecidos no Laboratório. O acompanhamento dos projetos não era feito diariamente, apenas existiam prazos de entrega entre os times e, quando este prazo estava para ser expirado, o responsável vinha à tona e cobrava os resultados. Caso houvesse a possibilidade de atraso das atividades, os membros do time tinham que cumprir horas extras para cobrir o prazo de entrega.

Assim, ficou determinado que os principais objetivos a serem alcançados com a adoção do Scrum são:

- Apresentação de um modelo solução para o gerenciamento no desenvolvimento dos sistemas para a diretoria;
- Criar indicadores de desempenho para o andamento dos projetos;
- Treinar equipes em metodologia ágil, o Scrum;
- Ganhar a confiança dos stakeholders;
- Montar uma equipe exclusiva e 100% Scrum em um projeto isolado dos outros;
- Obter o máximo de desempenho com pequenas entregas com valor de negócio;
- Obter sucesso exponencial na entrega do projeto;
- Definir um processo único, adaptado e adequado à realidade do time.

O projeto definido para essa implantação conta com uma equipe experiente com três desenvolvedores plenos, dois analistas e dois estagiários, e o software é desenvolvido utilizando a tecnologia Java. A comunicação entre os membros do time está de acordo com o previsto na metodologia Scrum.

4.2 Aplicação Scrum

A implantação começou com uma apresentação de introdução ao Scrum e suas práticas para toda a equipe. Foram abordados todos os passos que compõem a metodologia, práticas de estimativas, cerimoniais, entregas com valor de negócio para o cliente, criação dos artefatos, relatórios e etc. Foi acordado que a implantação seria inicialmente em apenas um produto da empresa, para que não houvesse um impacto tão grande e se caso houvesse o resultado positivo, ele seria replicado para os demais produtos.

Com todos os colaboradores conhecendo o Scrum, o primeiro passo foi a definição dos papéis, onde se estabeleceu o Scrum Master e o Product Owner. Definiu-se também que as entregas seriam feitas de duas em duas semanas e as reuniões diárias seriam realizadas todos os dias às 09h00min, em frente ao quadro branco. O passo seguinte foi definir um backlog para o produto, em uma reunião entre o Scrum Master e o Product Owner ficaram definidos os itens que fazem parte do mesmo. Após o término do backlog deu-se início a primeira reunião de planejamento. O Product Owner listou as tarefas com maior prioridade e a equipe fez uma estimativa, sem utilizar playing poker, porém como se trata da iteração 0 essa estimativa foi “as cegas”, pois a velocidade da equipe só vai poder ser medida após a primeira iteração.

Na iteração 1 primeiramente foi feita uma retrospectiva da iteração anterior e logo após, a entrega das atividades desenvolvidas. Novamente o Product Owner apresenta suas prioridades para a iteração. A primeira parte é o momento em que os membros da equipe

definem o que será feito. A segunda parte do planejamento é para debater como as atividades serão desenvolvidas e para a equipe de desenvolvimento listar as tarefas necessárias para implementar as atividades planejadas. A terceira parte do planejamento é para estimar as atividades, baseando-se no consenso dos membros da equipe e valores da sequência Fibonacci. No decorrer do desenvolvimento são feitas todos os dias as reuniões diárias, atualizando os envolvidos no projeto sobre o status das funcionalidades estimadas na iteração atual.

Após 3 iterações de 2 semanas a velocidade da equipe foi medida e as estimativas se mostraram cada vez mais corretas, as entregas constantes de partes do produtos foram um grande sucesso e a motivação da equipe também. A equipe a cada Sprint aprendeu a lidar com o ambiente, levando em consideração o débito técnico, a importância de não ser influenciado na pontuação das estimativas e entender que a discordância muitas vezes é o ponto inicial para a sinergia, onde a soma do todo é maior que a soma de suas partes. O resultado foi acima do esperado, principalmente pelo dono da empresa que tinha receio em ter uma mudança tão grande no processo da empresa, e por ter de deixar alguns valores culturais de lado.

4.3 Erros, problemas e dificuldades durante a implantação

O Scrum tem como objetivo mostrar os problemas da empresa, onde estes problemas são levantados como impedimentos. No princípio, a inexperiência no tratamento destes impedimentos acabaram gerando um desgaste desnecessário até que houvesse o completo entendimento do que era um impedimento real.

Um dos grandes objetivos do Scrum é aumentar a produtividade do time. O que acontece com a implantação do Scrum é que as métricas começam a aparecer rapidamente, mas a comparação entre times baseada nas complexidades entregues não é possível. É necessário um tempo de maturação de quatro a oito Sprints para conhecer a real velocidade da equipe com a qual se está trabalhando. Alguns problemas encontrados e erros cometidos:

- Falta de treinamento de todo o time;
- Falta de definição das responsabilidades de Product Owner e Scrum Master;
- Paralelismo de histórias, várias histórias inacabadas ao final da Sprint;
- Falta de Product Backlog Priorizado e Estimado;
- Não gerenciamento das reuniões no formato de tempo fechado (timebox). Reuniões diárias demorando mais de 15 minutos e os assuntos acabam fugindo das três questões que devem ser respondidas.

5. Considerações Finais

O Scrum possui um modelo de implementação simples, dele fazem parte poucas especificações e artefatos. O grande desafio está em integrá-lo as necessidade da empresa, do cliente e do time, fazendo as adaptações necessárias. Uma tarefa que não é fácil, e muitas vezes dependendo da empresa, não é indicada. Há ambientes onde a cultura tradicional é muito forte e os indivíduos não estão dispostos a mudar, tornando praticamente impossível a adoção de uma prática ágil.

A participação efetiva do cliente trouxe a redução dos riscos e custos. Além disto o grau de satisfação por participar do processo como um todo, melhorou. O modelo de entrega

nos permitiu que os projetos fossem melhor priorizados e seu tamanho discutido, eliminando funcionalidades que aparentemente não possuem valor de negócio relevante.

Com base na análise da implantação da metodologia ágil Scrum para o desenvolvimento de software, foram perceptíveis melhorias no gerenciamento e no desenvolvimento dos projetos de software, o que garantiu maior visibilidade ao seu andamento. O envolvimento e o comprometimento dos membros da equipe perante os resultados aumentaram, permitindo um trabalho mais colaborativo. Percebeu-se, também, que os membros das equipes estavam motivados e abertos às mudanças no trabalho, o que facilitou o processo de implementação/adequação da metodologia ágil Scrum. Assim, permitiu-se o amadurecimento e a busca por melhorias no processo, de modo a atender as particularidades dos demais projetos a receber o Scrum.

É importante saber que a prática do Scrum cria equipes mais motivadas, que com o passar do tempo criam laços com o produto, tendendo cada vez mais a entregar qualidade e eficiência.

Referências

- Vicente, A. (2011) “Fundamentos das Metodologias Ágeis”. In Revista Engenharia de Software, volume 4, numero 41, p. 10-24.
- Beck, K. et al. (2001). Agile Manifesto, <http://www.agilemanifesto.org>, Junho.
- Villas, M. (2010) “Métodos Ágeis de Desenvolvimento de Software”. In Revista Engenharia de Software, volume 2, numero 20, p. 5-21.
- Tolfo, C. e Wazlawick, R. (2008) ”The Influence of Organizational Culture on the Adoption of Extreme Programming”.
- Oliveira, A. Pereira, C. e Perrelli, H. (2011) “A necessidade de ser ágil”. In Revista Engenharia de Software, volume 3, numero 37, p 8-18.
- Bassi, D. (2010) “Você é Ágil Mesmo?”. In Revista MundoJ, volume 8, numero 43, p. 11-13.
- Barros, B. (2011) “Adotando Arquitetura ágil em seu processo de desenvolvimento de software”. In Revista MundoJ. volume 9, numero 50, p. 8-13.
- Santos, Rildo. (2013) “Scrum o Tutorial”, [http://pt.scribd.com/doc/43375190 /SCRUM-O-Tutorial-v1](http://pt.scribd.com/doc/43375190/SCRUM-O-Tutorial-v1), Junho.
- Scgwaber, K. and Beedle, M. Agile Software Development With Scrum, Prentice Hall, 2002.
- Kniberg, H. (2007) Scrum e XP direto das Trincheiras. InfoQ
- SANTANA, Silvina & DIZ, Henrique. Cultura e Aprendizagem Organizacional
- Bassi, D. Dificuldades na implantação de Métodos Ágeis. Disponível em: <http://ccsl.ime.usp.br/agilcoop/files/AgilCoop-Verao09-Dificuldades.pdf>. Acessado em 30 de Abril de 2013.
- Cukier, D. 9 perguntas sobre o uso de Métodos Ágeis na Locaweb. Disponível em: <http://blog.locaweb.com.br/tecnologia/9-perguntas-sobre-o-uso-de-metodos-ageis-na-locaweb/>. Acessado em 30 de Abril de 2013.
- SCHWABER, Ken. Agile Project Management with Scrum. Redmond: Microsoft Press, 2004.
- SCHWABER, K., and Beedle, M., Agile Software Development With Scrum, Prentice Hall, 2002.