



Por que SCRUM?



Isabella Fonseca

isabella@powerlogic.com.br

Atua desde 1999 como consultora em e-Business para grandes empresas utilizando Java EE. É Certified ScrumMaster e gerente da equipe de desenvolvimento e de projetos do eCompany Portal - primeira solução de EIP (Enterprise Information Portal) do país, e do eCompany Process - solução de definição e controle de Processos Corporativos integrada ao Gerenciamento de Projetos (EPM e APM), de Requisitos e de Produtos (Application Lifecycle Management). Ambos são gerenciados com SCRUM e certificados MPS.Br nível F.



Alberto Campos

alberto.campos@localiza.com.br

Está na área de TI desde 1984 atuando com gerenciamento de projetos ágeis desde 2002 para projetos de grande porte multidisciplinares incluindo Engenharia, Automação Industrial e Desenvolvimento de Software. Fora do Brasil, já implantou sistemas nos Estados Unidos, Europa e Japão.

A indústria de desenvolvimento de software evoluiu para se tornar uma das mais importantes instituições de nosso tempo criando produtos essenciais para o nosso dia a dia. Inúmeros exemplos podem ser citados. Neste ambiente intensamente competitivo, diferenciais devem ser criados para a sobrevivência. A capacidade de criar e de entregar mais rapidamente produtos de software que melhor satisfaçam as necessidades reais do cliente é um deles.

Segundo PMI Brasil 2006, os problemas mais frequentes em gerenciamento de projetos levantados são:

- Não cumprimento de prazos (72%)
- Problemas de comunicação (71%)
- Mudança de escopo (69%)
- Estimativa errada de prazo (66%)

Analisando os problemas levantados, pode-se constatar que o cerne da questão reside principalmente na comunicação – refletindo nos demais. Quando presentes, mecanismos de comunicação

ineficientes são utilizados contribuindo para a falta de compreensão e colaboração por parte dos envolvidos.

A **Figura 1** promove uma comparação entre a efetividade de comunicação e a “riqueza” do canal de comunicação. Dois mecanismos são evidenciados: os que não possuem perguntas e respostas e os que possuem. Este primeiro tipo propõe pouca interação e não permite a colaboração e troca necessária para o alcance do objetivo proposto. A utilização de papel para descrever e comunicar o problema tem importante cunho “documentacional”, além de poder ser compartilhado entre diversas pessoas.

Por outro lado, mecanismos que promovem e facilitam a comunicação, os mecanismos interativos, são compostos por perguntas e respostas - como duas pessoas falando ao telefone ou discutindo via email, são notoriamente mais eficientes. E ainda torna-se mais eficiente se tivermos a presença física na mesma sala dos diversos envolvidos. Para completar,

um quadro branco auxiliando na dinâmica das discussões, pode expandir em muito as possibilidades de entender e se fazer entendido.

Comunicação “face-to-face” é uma das formas de se resolver grande parte dos problemas citados e, metodologias ágeis, como o SCRUM, pregam incessantemente esta prática. O Manifesto Ágil, criado em 2001, instituiu valores e princípios da escola que corroboram com o descrito acima. Abaixo os quatro valores principais:

“Nós estamos descobrindo melhores formas de se desenvolver software, fazendo software e ajudando os outros a fazê-lo.

Através deste trabalho nós viemos valorizar:

Indivíduos e Interações primeiro que Processos e Ferramentas.

Software funcionando primeiro que Documentação Compreensiva.

Colaboração do Cliente primeiro que Negociação Contratual.

Resposta a Mudanças primeiro que Conformidade com Planejamento.

Isto é, embora reconheçamos que há valor nos itens à direita, nós valorizamos primeiramente os itens à esquerda.”

É importante destacar que não há ruptura entre os itens da esquerda e os da direita e sim de ênfase. A comunicação irá auxiliar em cada um dos itens da esquerda e será o grande facilitador desse processo.

O interessante deste tipo de abordagem é que não se tem a procura por um culpado. Todos estão imbuídos na busca da melhor solução para a organização, seja o analista, desenvolvedor, gerente de projeto ou o cliente. E cada um tem o seu papel claramente determinado frente ao processo de desenvolvimento. Em metodologias ágeis, uma das premissas básicas é ter o cliente sempre por perto e fazê-lo um participante ativo. É importante que ele entenda suas responsabilidades e sua grande parcela de contribuição para o sucesso do projeto.

Algumas questões diferenciam metodologias ágeis no planejamento do projeto. Nas metodologias ágeis, o planejamento é feito continuamente, durante todo o projeto, e baseado em um goal (objetivo), onde são definidas as tarefas

necessárias para se iniciar o mesmo. Este objetivo serve como um guia para todos os envolvidos e é a meta a ser perseguida. Normalmente é um parágrafo que resume, em três ou quatro frases, o foco maior que o projeto deve ter. Seu estabelecimento visa delegar à equipe do projeto um espaço de agilidade que a permita tomar decisões rápidas ao longo do projeto, norteando-o quanto a alterações de prioridades, de requisitos, práticas ou outras quaisquer.

Se todo o planejamento fosse feito no início do projeto, após as primeiras semanas, o longo planejamento feito poderia estar defasado, pois as mudanças ocorreriam e o forçariam a esta situação. Requisitos não devem ser intensamente esgotados no início do projeto, eles sim, são incrementados a cada iteração, experimentados pelos envolvidos, colocados a prova, avaliados e refinados.

É importante construir o todo aos poucos, refinando os requisitos, equalizando o entendimento sobre eles e corrigindo o rumo sempre que uma mudança provoque alterações significativas.

Planejar detalhadamente é ir contra a natureza do processo de desenvolvimento de software. As tarefas que constituem este tipo de abordagem fazem parte de um processo criativo, não linear e não palpável, fazendo com que modelos ágeis se apresentem como uma alternativa interessante. Por isso, planos menos detalhados e feitos em frequência maior se mostram mais adequados.

A forte presença de iterações curtas também contribui para o planejamento contínuo citado. Elas garantem ritmo a

todos os envolvidos e levam ao feedback real e imediato dado pelo cliente. Como resultado, auxiliam nos possíveis ajustes - que não acontecem mais tardiamente - e ajudam na entrega de software de valor. Cada uma das iterações é composta de todas as tarefas necessárias a esta entrega: levantamento de dados, análise, projeto, implementação, testes e integração a versão anterior.

Outro ponto interessante é a maneira pela qual o monitoramento e a medição do projeto são executados em metodologias ágeis. Cada tarefa definida é verificada quanto ao seu percentual de andamento (o mais usual) e também quanto à sua qualidade através de testes e entregas que são feitas com maior frequência. O progresso do projeto não é medido levando-se em consideração o plano inicial feito, como em metodologias tradicionais, ou seja, não é traçado um quadro comparativo entre o que foi planejado no início e o que foi executado até o momento. Mudanças são aceitas e algumas “linhas de bases” farão parte do projeto. Dessa forma, o atendimento ao goal definido deve ser a principal medida de progresso sendo verificado, constantemente, para garantir o máximo de retorno para o negócio.

A identificação e o gerenciamento dos riscos em metodologias ágeis são feitos em taxas diárias (através de reuniões curtas e diárias) e durante toda a iteração, o controle da qualidade dos trabalhos é avaliado. Esta tarefa pode ser executada através de testes, revisão por pares, inspeção contínua e acompanhamento pelo cliente, que seguem o mesmo processo.

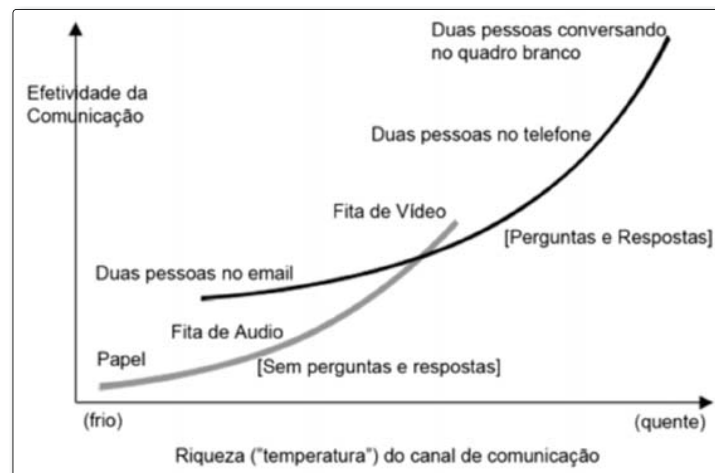


Figura 1. Temperatura da comunicação segundo Alistair Cockburn.



Impedimentos são levantados e devem ser resolvidos no prazo máximo de 24 horas para garantir que ações de resolução rápidas são executadas e para que todos conheçam os impedimentos que podem se tornar potenciais riscos para o projeto.

Além disso, mudanças no projeto são bem aceitas, pois elas existem e irão acontecer. É por isso que o comprometimento de todos os envolvidos é tão importante em um projeto. Caso o cliente queira mudar uma solicitação ou adequar-se a uma mudança de mercado, pode-se alterar o rumo rapidamente sem afetar todo o projeto pois, o planejamento é refeito a todo o momento. E ainda, caso uma iteração não corra conforme o esperado pelo cliente pode-se mudar a abordagem de levantamento de dados, os recursos envolvidos, a forma de troca de informações e corrigir o rumo a tempo do final do projeto.

Pode ser (e na maioria das vezes, é) que o plano inicial não represente mais a necessidade real de nosso cliente. Ele deixa de pedir alterações, pois o contrato já foi assinado, ele não tem mais horas adicionais para pagar, o prazo está curto e ele necessita do produto e então... ele “evita” solicitar mudanças. Pergunta: o que ele ganha? De que adianta entregar um software que o cliente não irá usar? E o quê o fornecedor do serviço ganha? Ok, entrega-se o produto, o cliente usa poucas funcionalidades, reclama das funcionalidades prioritárias que ele não recebeu e em contrapartida, recebeu várias outras que não vai utilizar. E isso pode ser considerado um projeto de sucesso?

O sucesso de um projeto de software utilizando metodologias ágeis não é somente medido se comparando projeções iniciais de prazo, custo e escopo e sim através de entrega de software

de valor ao cliente. Estes três pilares – prazo, custo e escopo – devem ter seus pesos definidos. É claro que todos nós desejaríamos que nossos projetos terminassem dentro do prazo previsto, com o custo dentro da margem determinada e com todo o escopo entregue. Todos nós já passamos por fases posteriores de manutenções e correções que consomem grande esforço de recursos e também traz desgaste no relacionamento das pessoas envolvidas.

Neste artigo, faremos uma introdução ao Scrum. Mas antes disso, para auxiliar no entendimento, utilizaremos um exemplo buscando dar uma visão mais simplista da metodologia ágil Scrum. Como diz o provérbio chinês: “O que ouço, esqueço. O que vejo, lembro. O que faço, entendo”. Portanto, segue um relato que muitas famílias já vivenciaram. Algumas correlações serão feitas no decorrer da estória – colocamos os termos Scrum entre parênteses. Mais tarde, estes conceitos serão desvendados.

Resolvemos planejar uma viagem em família. Para isso, algumas perguntas surgiram:

- Qual seria o projeto?
- Quem seriam nossos clientes?
- Como melhor atendê-los sabendo que mudanças podem ocorrer e mudar nosso rumo?
- Como e quais decisões tomar para reagir a tempo não frustrando as expectativas?
- Como saber se o projeto está alcançando o resultado esperado?
- Como aprender com a experiência vivida?

Projeto: viagem de família à praia.

Clientes: nossos filhos – Luiza, de 9 anos, Gabriel, de 7 anos e Bernardo - 13 anos (existem clientes mais exigentes?).

Custo: menor possível.

Prazo: final de semana prolongado.

Nossos filhos faziam planos para este feriadão há muito tempo e não viam a hora de chegar à praia. O carro já estava na revisão e as malas todas prontas.

Começaríamos a viagem no final da tarde de quinta-feira. Poderíamos chegar à noite, aproveitar os dias de sexta e sábado e retornar no domingo. A viagem começou e as crianças estavam eufóricas, muita gritaria, bagunça, fome, comida, pedidos para parar, ir ao banheiro, colo de mãe. E perguntas do tipo: quando vamos chegar? Já chegou? Como é lá? Problemas com o carro. Será que conseguiríamos chegar? Noite em hotel... decepção. Carro ok, continuamos a viagem, e enfim, chegamos...

Bom, tudo isto ocorreu conosco durante nossa ida e para a volta, resolvemos fazê-la a “La Scrum”.

Seguindo a metodologia, decidimos dividir a viagem de volta (Release) em etapas (Sprints), com duração prevista de duas horas ou 200 km. Cada uma com um resultado esperado (Sprint Goal), que eram acompanhadas de perto por nossos filhos. O projeto também teve seu objetivo maior definido (Release Goal ou Vision). Sendo assim, as expectativas de nossos “clientes” ficaram claramente conhecidas.

Release Goal: Chegar em casa com segurança aproveitando cada momento juntos.

Durante cada etapa, estávamos sempre verificando o andamento das atividades planejadas - o que foi feito ontem?, planejando as próximas - o que vai fazer amanhã? e removendo problemas que surgiam – que impedimentos ocorreram (Daily Scrum). Além disso, atualizávamos em nosso mapa (Agile Radiator) nossa evolução. Dessa forma, todos os envolvidos tinham visualmente o real

andamento e poderíamos identificar qualquer desvio indesejado. Tudo isso tendo como guia o Sprint Goal, também estipulado por eles.

Ao final de cada etapa parávamos para comer alguma coisa, descansar, esticar as pernas e perguntar a nossos filhos o que estavam achando de nossa viagem. Realizávamos também uma avaliação (Retrospective Meeting), analisando pontos positivos e negativos e seguíamos para a etapa seguinte (Novo Sprint).

Todos tinham suas responsabilidades e papéis definidos e nos dávamos o direito de rearranjar a próxima etapa se algum imprevisto ocorresse. Tendo sempre em mente o que foi determinado (goal), vez por outra, resolvíamos parar mais em uma agradável cidade, alterávamos o horário do jantar para curtirmos um pôr do sol em outra ou ir a uma festa típica que estava acontecendo.

Terminada a viagem, já em casa, fizemos a última retrospectiva de tudo o que ocorreu para nos prepararmos melhor para a próxima viagem, que terá uma duração maior, vários tipos de transporte e destinos e ainda levaremos nossos primos.

Esta foi somente uma estória de como podemos utilizar as práticas do Scrum em nossa vida cotidiana. Na próxima seção, explicaremos um pouco mais sobre o framework de processos.

O Scrum

O Scrum é um framework de processo ágil utilizado para gerenciar e controlar o desenvolvimento de um produto de software através de práticas iterativas e incrementais. É composto por um conjunto de boas práticas de gestão que admite ajustes rápidos, acompanhamento e visibilidade constantes e planos realísticos.

Os principais papéis do Scrum são: Product Owner, Scrum Master e Scrum Team (equipe do projeto). Não há como fazermos um mapeamento direto entre os papéis do Scrum e os papéis convencionais conhecidos. Não existe a figura única do Gerente de Projetos. Suas responsabilidades estão diluídas entre os papéis citados. Cada um conhece sua participação frente ao projeto e trabalha em conjunto para conseguir alcançar o goal definido.

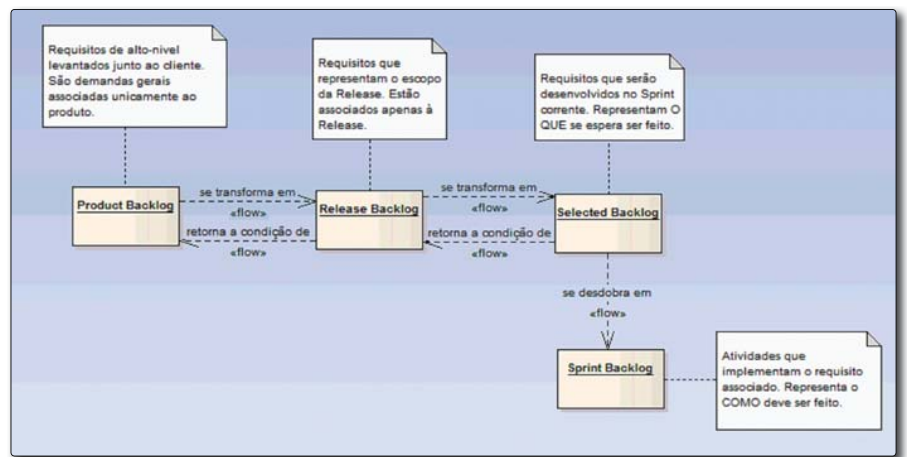


Figura 2. Ciclo de vida de um requisito e desdobramento em atividades

Seus artefatos principais são o Product Backlog e Sprint Backlog – artefatos que representam seus requisitos/atividades além de Burndown charts e impediment backlogs. Para representar o ciclo de vida de um requisito, usa-se derivações do Product Backlog, como Release Backlog e Selected Backlog. A Figura 2 exemplifica o ciclo mencionado.

Inicialmente um requisito deve fazer parte do Product Backlog, que possui descrições de necessidades de clientes de alto nível levantadas. A tarefa de manter a descrição e refinamento destes requisitos é do Product Owner. Ele é responsável por definir, para cada nova release de um produto, o objetivo (Release Goal ou Vision). Ele interage a todo momento com seu cliente final e, dessa forma, conhece suas expectativas e as do mercado. Esta lista de demandas nunca acaba e de tempos em tempos (a cada nova release), um item de Product backlog é promovido a Release backlog para integrar ao escopo de uma ou mais Releases. O Product Owner deve comunicar e discutir o novo goal com todos os envolvidos em reuniões chamadas de Release Planning.

Portanto, a primeira reunião oficial de Release que ocorre é a Release Planning. Nesta reunião, são discutidos os recursos envolvidos – tanto computacionais quanto humanos, riscos identificados, prazo acertado e requisitos previstos acordados junto ao cliente e coerentes com a demanda de mercado. Seu propósito principal é definir e comunicar as funcionalidades macro de uma Release do produto e fazer com que os envolvidos no projeto entendam e se comprometam com o Release

Goal definido. As funcionalidades não devem ser discutidas em detalhes. Isso será feito em cada Iteração durante reuniões de Sprint Planning. Ela pode ter a presença de diversos papéis: seja cliente externo, desenvolvedor, área comercial, etc.

Cada release é composta de iterações curtas, chamadas de Sprints. A Figura 3 traz um desenho das reuniões existentes durante o período de uma Release, composta por 3 Sprints. Conforme dito, a primeira reunião é a Release Planning. Cada Sprint tem reuniões de planejamento, Sprint Planning, que detalham o requisito apresentado durante o Release Planning e ainda possuem seu Sprint Goal. Semelhante ao goal do Release, tem o mesmo cunho de comunicar a todos os envolvidos o objetivo maior de Sprint corrente. É realizado de forma iterativa, através de ciclos de “tempo fechado” (Time-Boxed) em 30 dias corridos. Ainda sobre a reunião, são conhecidos os requisitos do Release Backlog que serão promovidos como Selected Backlog. Eles são discutidos em maiores detalhes e servem para elucidar O QUE se espera como resultado final. Em um Sprint, portanto, são executadas atividades de refinamento de requisitos, análise, projeto, desenvolvimento e testes pelo Scrum Team, devendo resultar em um incremento de produtos funcionando e demonstrável para o Product Owner ao final do Sprint, na reunião de Sprint Review.

O Scrum Master tem o papel de liderança muito importante para o processo. Ele deve remover todo e qualquer obstáculo que surgir durante o desenvolvimento, garantindo que o Scrum Team

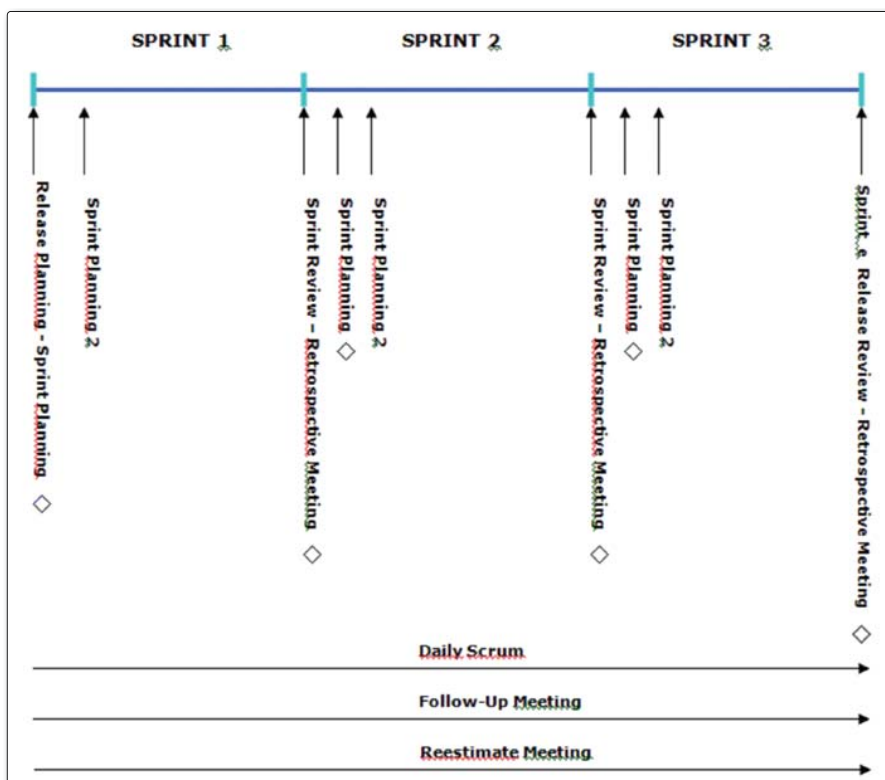


Figura 3. Reuniões Scrum

possa focar no real objetivo definido. Além disso, ele é responsável por fazer com que a equipe siga e pratique o processo e ainda por criar uma atmosfera de ajuda mútua entre a equipe (o resultado é sempre da equipe e não individual).

O Scrum Team é responsável por se organizar e determinar a melhor estratégia de se entregar as funcionalidades de maior prioridade. Interessante observar que iterações curtas garantem ritmo à equipe e facilita o entendimento do processo, além de proporcionar visibilidade ao cliente sobre o andamento do projeto.

Diariamente são executadas reuniões de acompanhamento e monitoramento do Sprint através da reunião de Daily Scrum. Esta reunião deve ter a presença do Scrum Team e Scrum Master obrigatoriamente. Deve ter a duração máxima de 15 minutos (é interessante até utilizar um cronômetro para isso) e são permitidas somente 3 perguntas:

- O que você fez hoje?
- O que fará amanhã?
- Que impedimentos surgiram e que atrapalharam sua produtividade?

Outros assuntos deverão ser discutidos em outras reuniões, como Follow-Up Meetings para detalhamentos e elucida-

ções relativas aos requisitos e Reestimate Meetings para aplicação de técnicas como o Poker Planning para estimativa de requisitos.

Há uma reunião de Sprint Planning 2 que promove o desdobramento do Selected Backlog em Sprint Backlog onde aspectos técnicos são discutidos. É a hora do COMO as coisas devem ser feitas. Neste momento, pode-se discutir sobre a arquitetura do produto, reuso de componentes, mudança em interfaces, etc.

Ao final do Sprint, é feita uma reunião final de apresentação do trabalho executado ao Product Owner chamada Sprint Review. Ele será responsável por validar a apresentação e concluir se o Sprint Goal foi atingido. Após esta, reuniões de debrief são executadas, chamadas Retrospective Meetings. Reuniões de retrospectiva levam a reflexão acerca do que passou e quais atitudes devem ser tomadas para o correto atingimento dos objetivos. São levantados aspectos positivos (WWW – what went well) e negativos (WCBI – what can be improved) tanto organizacionais quanto relativos a Release. Todos os envolvidos devem estar presentes e são responsáveis por minimi-

zar todo e qualquer risco ao projeto e também por manter os aspectos positivos levantados.

E o mesmo se repete por todos os Sprints planejados, com a diferença de que o último Sprint possui uma reunião de Release Review, que tem como guia o Release Goal definido. Nunca é demais lembrar que o planejamento é feito durante toda a Release e portanto, alterações podem ser feitas. Se constitui em uma boa prática poder refazer o planejamento durante cada Sprint Planning, mas durante o Sprint, o Scrum Master deve procurar “blindar” o Scrum Team de grandes modificações.

As estimativas de tamanho de cada funcionalidade/requisito são executadas obrigatoriamente pelo Scrum Team. Dessa forma, todos sabem para onde ir (goal) e é desta forma que se obtém o comprometimento da equipe para com o projeto. A viabilidade de continuidade dos planos é constantemente verificada.

A comunicação de um projeto que utiliza o Scrum é mais efetiva e direta, além de informações estarem sempre à tona com a utilização de quadros brancos (Agile Radiator). A Figura 4 exemplifica um Agile Radiator monitorando um projeto real. Eles garantem visibilidade do projeto a todos os envolvidos. Não há como mascarar o real andamento. O goal fica afixado e os requisitos – através de Post-its - (Selected backlogs) e seus desdobramentos (Sprint backlogs) são posicionados na situação onde se encontram (se ainda não iniciados - planejados, se sendo executados no momento - em andamento e se terminados – 100% concluídos). Eles devem ser posicionados de acordo com a prioridade dos mesmos – Business Value declarado pelo Product Owner. Post-its localizados no topo nos dizem ser de maior prioridade que os posicionados no rodapé do quadro branco.

Impedimentos (Impediment Backlog) que ocorrem durante o Sprint também são evidenciados. O Scrum tem um gráfico, de nome BurnDown Chart, que exhibe o real andamento do mesmo. O Scrum Team diariamente, através de reuniões chamadas Daily Scrum, atualiza o Agile Radiator com a situação atual, movendo post-its e indicando o trabalho executado durante o período.

Vale destacar que a granularidade de cada Sprint Backlog deve ser pequena. Sua estimativa não deve ultrapassar 8 horas de trabalho e isso se constitui em uma boa prática no gerenciamento do andamento das atividades. Através desta reunião, o time consegue avaliar se é necessário mudar o plano para acertar o rumo, se deve priorizar alguma outra atividade, se muitos impedimentos estão ocorrendo e como minimizá-los, se há algum recurso que está necessitando de ajuda, etc. É real! Isso tudo sempre à luz do goal definido. Além de todas estas vantagens, ainda proporciona visibilidade a todos os envolvidos.

Durante cada reunião de Daily Scrum, post-its são “movimentados” pelo Scrum Team contemplando cada uma das perguntas associadas a esta reunião. Atividades executadas (o que foi feito hoje) são movidas para a seção “Concluídos”. Atividades planejadas (o que será feito amanhã) são colocadas na seção “Em Andamento” e impedimentos são registrados (que impedimentos surgiram? Repare a seção em vermelho no rodapé a direita).

Além destes artefatos, o Burndown Chart deve ser atualizado ao final da reunião contabilizando o tamanho total entregue pelo Scrum Team no dia a

ser representado. O Burndown possui duas retas: uma que exibe o planejado e outra que reflete o realizado. Ele está registrado no topo à esquerda. Ele contém o total de trabalho restante e só de “batermos o olho” no gráfico conseguimos verificar a evolução do trabalho. Todo o Agile Radiator tem esta característica. Se estivermos vendo muitos impedimentos, temos que verificar o que está sendo feito para prevenir novos e se estes estão sendo corretamente removidos. Se atividades de maior prioridade estão sendo deixadas para trás, a própria equipe pode verificar o que está havendo. O que foi bom (WWW) e o que pode ser melhorado (WCBI) também ficam afixados no quadro branco.

A cada novo Sprint, o Agile Radiator deve ser modificado para retratar a situação atual novamente. E este ciclo se inicia: planejamento, execução, controle e avaliação do que foi feito para cada nova Release necessária.

Considerações Finais

Uma confusão ou uma análise superficial que não podemos deixar de citar é o entendimento de que metodologias ágeis são pouco formais ou que indo por este caminho, você estará li-

vre de documentações ou burocracias. É necessária muita disciplina para seguir esta abordagem. Seus princípios e valores são baseados em dedicação e bom senso de todos os envolvidos. Suas práticas pregam inspeções constantes - para o feedback rápido - e aceitação das mudanças e adaptações que o processo deva passar. O comprometimento entre todos os envolvidos também se constitui como um grande diferencial do framework. O processo de gestão está permeado entre os diversos papéis existentes como falado anteriormente.

Estas são somente práticas de um framework de processo que nos guia em como fazer. Ele está baseado no Manifesto da Agilidade, que em 2001 reuniu vários gurus para a discussão sobre processos de desenvolvimento de software. Não é uma anarquia! É um convite a pensar diferente. Cabe a cada um de nós descobrir o melhor processo dentro da organização que trabalhamos. Nossa experiência nos leva a combinar algumas abordagens. E elas têm se demonstrado bastante efetivas. Mas, conforme dissemos, isso se dará de forma diferente para cada um. O importante é encontrar o seu caminho. ●



Figura 4. Agile Radiator

Bibliografia

Agile Alliance, 2002. Agile Manifesto, <http://www.agilealliance.org>
 Artigo: “Get Ready for Agile Methods, with Care”,
 Computer, 35, 1 (January 2002) 64-69 escrito por
 Boehm, B.
 Agile Software Development with Scrum, Prentice
 Hall, (October 2001). Schwaber, K; Beedle, M,
 Agile Software Development, Cockburn Highsmith
 Series Editors, Alistair Cockburn 2000

Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:
www.devmedia.com.br/esmag/feedback

