



# PUC

ISSN 0103-9741

Monografias em Ciência da Computação  
nº 07/09

## **Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet**

**Jacques Douglas Varaschim**

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO  
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900  
RIO DE JANEIRO - BRASIL**

## **Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet \***

Jacques Douglas Varaschim

jacques@corp.globo.com

**Abstract.** Internet companies need to implement products rapidly and efficiently, respecting time-to-market. In addition, as the popularity of agile development, companies are discovering that simply breaking down projects into small iterations is not enough. Agile methods require changes in management, architecture, testing as well as project management. Given the large adjustments required, enterprises are looking for some guidance.

**Keywords:** Project Management, Scrum, Agile Development.

**Resumo.** As empresas de internet necessitam disponibilizar seus produtos aos usuários de forma rápida e eficiente. Além disto, com a popularização do desenvolvimento ágil elas estão descobrindo que a simples divisão de um projeto em pequenas iterações não é suficiente. Os métodos ágeis requerem mudanças em gestão, arquitetura, testes e garantia da qualidade, tanto quanto em gerenciamento de projetos. Estes são os desafios destas empresas, implantar o desenvolvimento ágil em meio a necessidade de inovação de produtos.

**Palavras-chave:** Gerenciamento de Projetos, Scrum, Desenvolvimento Ágil.

\* Trabalho Final da disciplina Processos e Ambientes, ministrada pelo prof. Arndt Von Staa, em 2008.

# Sumário

1	Introdução	1
2	O Desenvolvimento de Software para a Internet	1
3	Scrum	2
3.1	Histórico	2
3.2	Seleção do Scrum	2
3.3	Ciclo	3
3.4	Papéis e Responsabilidades	4
3.4.1	Scrum Master	4
3.4.2	Scrum Team	4
3.4.3	Product Owner	5
3.5	Etapas do <i>Sprint</i>	5
3.5.1	Sprint Planning	6
3.5.2	Daily Scrum	7
3.5.3	Sprint Review	8
3.5.4	Sprint Retrospective	8
3.6	Artefatos	9
3.6.1	Product Backlog	9
3.6.1.1	Planning Poker	9
3.6.2	Sprint Backlog	10
3.6.3	Impediment Backlog	11
3.6.4	Burndown Chart	11
4	Scrum na Globo.com	12
4.1	Implantação	13
4.1.1	Scrum Master	14
4.1.2	Product Backlog	15
4.1.3	Scrum Team	15
4.1.4	Daily Scrum	16
4.1.5	Sprint Planning	16
4.1.6	Sprint	16
4.2	Pós Scrum	17
5	Conclusão	17
	Referências	18

## 1 Introdução

No Brasil as metodologias de desenvolvimento ágil têm gerado grande entusiasmo entre seus usuários e na comunidade acadêmica. O princípio comum do desenvolvimento ágil é o desenvolvimento cooperativo, baseado mais nas pessoas e suas iterações, em vez de focar em grandes esforços de planejamento de requisitos e processos rígidos. É importante ressaltar que o desenvolvimento ágil prega o simples, entretanto, ser simples implica em grande carga de disciplina e organização.

Implementar uma metodologia ágil requer uma mudança de mentalidade nas pessoas que participam do processo de desenvolvimento. Em certos casos, a mudança de papéis e as novas responsabilidades podem gerar desconforto nos participantes durante a implantação do modelo.

A partir de 2001 temos uma maior clareza relacionada aos modelos ágeis de desenvolvimento. Naquele ano foi publicado o Manifesto para o Desenvolvimento Ágil (Agile Manifesto, 2001). O Manifesto é uma compilação das mais variadas técnicas, como Extreme Programming, Scrum, Crystal, Lean Development e outros, e tem como objetivo identificar padrões existentes entre estas metodologias. Deste encontro surgiram alguns valores que devem ter maior peso no processo o desenvolvimento ágil:

- **Indivíduos e Iterações sobre Processos e Ferramentas**
- **Software Pronto sobre Documentação Detalhada**
- **Colaboração Cliente/Time sobre Contratos**
- **Capacidade de Mudança sobre Seguir Planos**

Nos casos acima vale salientar que os itens a esquerda tem maior valor dentro do modelo ágil, não devendo os da direita ser esquecidos.

O objetivo deste trabalho é explorar quais são as características necessárias aos participantes do desenvolvimento ágil e de seu ambiente de trabalho, assim como descrever o processo de implantação em uma empresa de internet. Para este propósito é usada a implantação do Scrum dentro da Globo.com, desde a seleção do processo até sua implementação em times multidisciplinares que trabalham em diferentes projetos e as lições aprendidas.

## 2 O Desenvolvimento de Software para a Internet

O ambiente de desenvolvimento de software para a internet possui características similares ao de um modelo de gerenciamento de projetos ágil. Esta similaridade faz com que sua adoção de métodos ágeis vá ao encontro das necessidades das empresas que estão neste mercado.

Neste contexto podemos destacar:

- ciclo de desenvolvimento propício a mudanças de requisitos, baseado em iterações curtas e com entrega de software funcionando. Isto permite que o resultado seja facilmente avaliado pelo cliente e ajustes de estratégia do produto sejam realizados.

- equipes motivadas contribuem para o melhor gerenciamento do projeto, assim como com sugestões de inovação.
- a disponibilização do produto em curtas iterações para o usuário final permite avaliações constantes da satisfação do cliente, tanto interno quanto externo.

## 3 Scrum

### 3.1 Histórico

Baseado em casos de estudo da indústria automobilística, de computadores e de impressoras, Hirotaka Takeuchi e Ikujiro Nonaka descreveram em seu artigo "The New New Product Development Game" de 1986, um novo modelo que aumentava a velocidade e a flexibilidade do desenvolvimento de novos produtos comerciais. Em seu estudo eles comparavam o novo método em que as fases possuíam forte intersecção e todo o processo era desenvolvido por times multifuncionais com o "rugby", onde todo o time trabalha em conjunto para avançar. Em 1991, DeGrace e Stahl, em "Wicked Problems, Righteous Solutions: A Catalog of Modern Engineering Paradigms" foram os primeiros a mencionar o nome Scrum (Wikipedia, 2009).

No início dos anos 90, Ken Schwaber iniciou o uso do modelo em desenvolvimento de software dentro da sua empresa a Advanced Development Methods. Na mesma época, Jeff Sutherland utilizou um modelo semelhante na Easel Corporation sendo o primeiro a utilizar o nome Scrum para o desenvolvimento de software.

### 3.2 Seleção do Scrum

A seleção do Scrum foi baseada particularmente pelos seus princípios básicos de objetividade, papéis bem definidos e facilidade de aprendizado. Além disto, pesquisa de 2007 (VersionOne, 2008) 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, 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. O detalhamento do funcionamento das equipes pode ser visto no item 3.4.2 Scrum Team.
- 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. Ver item 3.5 Etapas do *Sprint*.
- Requisitos de produtos organizados em uma lista de itens chamada "*Product Backlog*". Ver item 3.6.1 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.

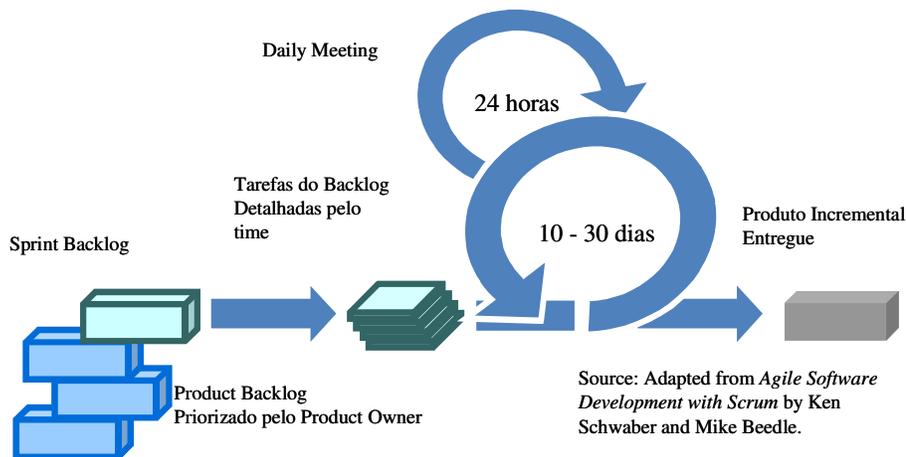
### 3.3 Ciclo

O Scrum tem o progresso de desenvolvimento baseado em iterações com duração entre duas e seis semanas, chamadas de *Sprints*. O primeira etapa dentro do Sprint é a reunião de planejamento (*Sprint Planning*), onde o time (*Scrum Team*), em conjunto com o cliente (*Product Owner*) define o que será implementado na iteração, sendo responsabilidade do cliente realizar a priorização do trabalho a ser feito.

A próxima etapa é a de execução, onde o time detalha as tarefas necessárias para implementar o que foi solicitado pelo cliente e posteriormente inicia a execução das mesmas. Durante o *Sprint* o time realiza reuniões diárias (*Daily Meeting*) com não mais de 15 minutos para averiguar o acompanhamento do progresso do desenvolvimento, usando para isto o *BurnDown Chart*, que é um gráfico usado para o acompanhamento das tarefas a realizar, em andamento e já realizadas.

Ao final do *Sprint* é realizada uma reunião para a validação da entrega (**Sprint Review**), onde o cliente e quem mais tiver interesse no produto pode verificar se o objetivo do Sprint foi atingido. Logo após, é realizada apenas pelo time uma reunião (**Sprint Retrospective**) onde o Sprint é avaliado sob a perspectiva de processo, time ou produto, quais foram os acertos e os erros com o objetivo de melhorar o processo de trabalho.

Abaixo podemos ver uma figura representativa do ciclo de desenvolvimento utilizando o Scrum:



**Ciclo de Desenvolvimento Scrum**

Com isto em mente, verifica-se que o Scrum é altamente recomendável para empresas que tenham no seu ciclo de projetos produtos dinâmicos e que possuam alta taxa de mudança de requisitos. Vamos agora entender melhor o Scrum, conhecendo seus papéis, etapas e artefatos.

## 3.4 Papéis e Responsabilidades

### 3.4.1 Scrum Master

O Scrum Master é um novo papel de gerente introduzido pelo Scrum. A principal função do Scrum Master é ser o facilitador do processo e ajudar as pessoas a resolver os problemas, enquanto dissemina e aplica o processo dentro da organização.

O papel do Scrum Master é fundamental, sendo durante a fase inicial de implantação uma função extremamente desafiadora. Seu papel de remover impedimentos, sejam dentro do time ou da própria empresa, faz com que questionem até mesmo o modelo de funcionamento da companhia.

Como tarefas adicionais temos:

- manter o time funcional e produtivo
- proteger o time de interferência externas

Para o papel de Scrum Master não necessariamente precisamos ter pessoas com treinamento técnico, pois seu principal papel é o de entender as dificuldades do time e manter uma boa comunicação. No início da implantação do modelo será mais difícil o papel do Scrum Master em razão da empresa estar em fase de adaptação ao modelo.

### 3.4.2 Scrum Team

O time Scrum deve ser multidisciplinar e auto-organizado, devendo ser pequeno com no máximo 10 a 15 participantes. O conjunto do conhecimento dos seus participantes deve englobar todas as características necessárias para a implementação do projeto a que se propõe. Embora o Scrum Team seja responsável por atingir os objetivos do Sprint, ele deve ser também autônomo e ter controle sobre o seu processo de desenvolvimento, sendo de sua responsabilidade ao final de cada Sprint mostrar os resultados do trabalho para o cliente. Durante um Sprint os recursos devem permanecer dedicados mas seus membros podem ser alterados entre Sprints caso um conhecimento específico seja necessário em determinada etapa.

A partir da implantação de métodos ágeis nota-se uma mudança no perfil dos participantes dos times, que iniciam como especialistas e aos poucos tornam-se colaborativos em outras áreas. A execução de tarefas de outras disciplinas leva ao aprendizado, sendo este um do ponto importante na adoção de um modelo ágil. Durante esta mudança as pessoas começam a ter incertezas sobre o modelo de carreira a ser seguido e como será seu crescimento profissional, sendo este item um ponto de atenção a ser tratado pela empresa.

A integração de equipes que trabalham em mais de um projeto simultaneamente também deve ser revista. Por exemplo, em algumas empresas as áreas de garantia da qualidade são separadas do desenvolvimento, sendo necessária uma adequação ao modelo quando se agrega garantia da qualidade aos Sprints.

Segundo Ken Schwaber (Schwaber, 2004), grandes projetos que requerem mais de um time são um problema que pode ser tratado a partir da formação de um **Team of Team Member's**. Neste caso um representante de cada time participa de uma equipe central que tem o objetivo de sincronizar as ações de todos os times envolvidos.

É importante ressaltar que a responsabilidade pela qualidade do produto é do time, ele tem o dever de verificar se suas práticas de desenvolvimento são compatíveis com as necessidades de qualidade e disponibilidade da aplicação. A partir do momento que o time verifica necessidade de refatorar o seu código ou construir novos testes automatizados ele deve criar histórias e demonstrar esta necessidade ao Product Owner para que as mesmas sejam priorizadas e desenvolvidas pelo time. A história é o elemento que descreve uma funcionalidade, uma melhoria ou uma correção a ser implementada dentro do sistema. Ela indica o que será desenvolvido e não como será feito.

### **3.4.3 Product Owner**

O Product Owner é o cliente de um time Scrum, ele é responsável pelo retorno do investimento de um produto. Entre suas tarefas estão:

- Definir todas as funcionalidades que o produto deve possuir. É importante salientar que sugestões de histórias podem ser feitas por qualquer pessoa mas a inclusão destas no Product Backlog é de gerência do Product Owner
- Priorizar as funcionalidades que devem ser incluídas a cada Sprint de acordo com o valor de negócio e inclui no Sprint
- Aceitar ou Rejeitar as entregas do time

Além disto, faz parte da tarefa do Product Owner avaliar a entrega de seu produto para os demais clientes e repriorizar e rever funcionalidades. O Product Owner tem o poder para interromper um Sprint em caso de urgência e até mesmo de suspender um projeto caso detecte que seu retorno não será o esperado pela empresa.

O perfil de um Product Owner está relacionado à área de produto, sendo necessário que o mesmo consiga compartilhar os objetivos do produto que está sendo construído pelo time. Ele deve ser visto como um líder pelo time, deve ser o principal motivador e ter bom relacionamento com a equipe. Isto é importante, pois o time precisa assumir que os desafios do Product Owner são os seus desafios.

### **3.5 Etapas do Sprint**

De acordo com Ken Schwaber (Schwaber e Beedle, 2002), o Sprint é definido como um ciclo de desenvolvimento curto em que o time foca em atingir o objetivo proposto pelo Product Owner. Durante este período o time deve ter total autoridade para seu gerenciamento, não devendo sofrer interferências externas do Product Owner ou mesmo de outras pessoas.

Em alguns casos os participantes de times vêem os métodos ágeis como uma tentativa de microgerenciamento já que metodologias como o Scrum aceleram o processo de desenvolvimento e os desenvolvedores integram mais com seus gerentes em menores espaços de tempo (Cohn e Ford, 2003).

O Sprint conceitualmente deve ter um tempo de execução de duas a quatro semanas. Este tempo reduzido tem como objetivo fazer com que as funcionalidades sejam entregues e validadas pelo Product Owner. Um time, na medida do possível, deve manter o tamanho de seus Sprints constante, permitindo que as métricas coletadas e as estimativas se tornem cada vez mais eficientes.

### 3.5.1 Sprint Planning

A primeira atividade dentro de um Sprint é uma reunião denominada Sprint Planning. Durante o Sprint Planning é realizada a seleção das histórias que serão implementadas durante aquele ciclo. É importante ressaltar que a tarefa do Sprint Planning já supõe que exista um Product Backlog, com suas histórias, valores de negócio e complexidade. Veremos mais adiante em "3.6.1 Product Backlog" o que é necessário para a montagem do mesmo.

O Sprint Planning geralmente é feito em um dia, mas de acordo com o tamanho do Sprint este tempo pode variar. No caso de Sprint curtos, por exemplo uma semana, o tempo do Sprint Planning pode ser menor, visto que o número de histórias a ser detalhado é menor. Um dos grandes influenciadores no atraso dos Sprints é a inexistência de um Product Backlog definido.

O Sprint Planning tem dois momentos, o primeiro, onde o time juntamente com o Project Owner e o Scrum Master planejam as funcionalidades que serão contruídas durante o Sprint que segue.

São de grande importância para esta etapa:

- Deve existir um Product Owner
- possuir um backlog estimado anteriormente com relação à complexidade de cada pendência;
- priorizar os itens que devem ser desenvolvidos, tarefa realizada pelo Product Owner
- reavaliar a complexidade dada aos itens priorizados, verificando sua corretude e se o item pode ser incluído no Sprint, tarefa realizada pelo time
- a correta avaliação da velocidade do time, item que só é passível de melhor avaliação após alguns Sprints, já que o time vai aprendendo em seu novo fluxo de trabalho

A partir destas informações a primeira etapa do Sprint Planning é:

1. Product Owner prioriza os itens em ordem de importância
2. time seleciona os itens que considera possíveis de desenvolvimento dentro do Sprint

Durante esta primeira etapa de reunião a tarefa do time e o Product Owner é avaliar as histórias existentes no Product Backlog e definir segundo o Product Owner aqueles que possuem maior valor para o produto. Na maioria das vezes é avaliada a relação entre o valor de negócio da história e a complexidade do desenvolvimento ditada pelo time.

Neste momento o Product Owner tem a autoridade para definir as histórias que serão encaixadas no próximo Sprint. É necessário salientar que a responsabilidade pela qualidade do produto é do time, e o mesmo tem a possibilidade de incluir histórias no Product Backlog relacionadas com a qualidade, arquitetura e outros. É responsabilidade do time também, mostrar ao Product Owner que determinadas histórias criadas por ele impactam diretamente o negócio e têm grande valor, devendo portanto ser desenvolvidas.

Ao final desta etapa as histórias devem ter sido selecionadas e devem se encaixar na velocidade do time para um Sprint chegando então o momento da segunda etapa que é o detalhamento das tarefas.

Na segunda parte do Sprint Planning o time tem como objetivo desmembrar as histórias em tarefas que serão cumpridas durante o Sprint. O objetivo desta segunda etapa é bem claro, o time deve realizar um detalhamento de todas as necessidades para desenvolver uma história e verificar se a estimativa inicial dada está de acordo com todas as atividades existentes.

Para isto é necessário que eles façam o seguinte:

1. para cada item o time detalha as atividades e estima o tempo necessário para o desenvolvimento de cada uma delas. As tarefas devem requerer no máximo um dia para que possam ser acompanhadas no Daily Meeting. Este modelo pode ser ajustado pela empresa e existem modelos em que as tarefas são definidas em número de horas e seu acompanhamento é realizado desta forma. Como citado anteriormente o Scrum é um modelo e deve ser adaptado às características e necessidades da empresa.
2. ao finalizar as estimativas é revisto se o tempo do Sprint é suficiente para a realização de todas as tarefas. Caso o time detecte que após o detalhamento existe mais trabalho do que o Sprint prevê, é necessário dialogar com o Product Owner para rever as histórias que compõem o Sprint. Do mesmo modo se sobrar tempo deve-se prever a possibilidade de adicionar mais histórias do Product Backlog.

Não são alocados recursos para as tarefas, atividade esta que é diária e realizada nos Daily Meetings.

### **3.5.2 Daily Scrum**

O Daily Scrum é uma reunião rápida que deve ter o conceito de tempo fechado, realizada todo dia, em pé durante a qual os membros do time explicam entre si o que foi feito desde a última reunião, o que será feito até a próxima e quais são os impedimentos que surgiram.

Quais são as principais características do Daily Scrum

- As reuniões devem começar no horário marcado. Frequentemente o time define penalidades por atrasos.
- Todos podem assistir as reuniões mas apenas o time tem o direito de falar durante a reunião. Caso seja necessário eles podem solicitar informações para o Scrum Master ou Product Owner.
- Deve durar até 15 minutos. Em times maiores que o recomendado esta reunião pode se estender um pouco mais. Caso o time esteja muito grande avalie a possibilidade da criação de dois times.
- Os participantes devem permanecer em pé
- Deve ser realizada sempre no mesmo local na mesma hora do dia.

As perguntas que são respondidas durante a reunião são:

- O que você fez desde ontem?
- O que você planeja fazer até amanhã?

- Existiu algum impedimento?

O objetivo desta reunião, além de acompanhar o desenvolvimento do projeto e distribuir as tarefas e criar a pressão entre os membros do time para a entrega do objetivo. A cobrança é do time para o time, não existem chefes e subordinados, apenas a equipe em busca do objetivo da entrega.

É importante verificar que a disciplina é extremamente importante no Scrum, o time é auto-gerenciado e deve agir como tal, possíveis atrasos são vistos no dia-a-dia e devem ser avaliados pelo time, bem como o que deve ser feito para superar os problemas. Todas as informações relacionadas ao andamento das tarefas devem ser organizadas e repassadas para a ferramenta de controle do andamento do Sprint que é o BurnDown Chart (Ver em 3.6.4 Burndown Chart).

### **3.5.3 Sprint Review**

O Sprint Review acontece no final de cada Sprint. Durante esta reunião o time mostra o resultado do seu trabalho para o Product Owner e para outros clientes opcionalmente.

Os itens avaliados durante o Sprint Review foram definidos como requisitos durante o Sprint Planning através de histórias de aceitação. Estes serão os itens levados em consideração para a verificar a completude do trabalho realizado.

Como boa prática o Sprint deve ser acompanhado pelo Product Owner que verifica o andamento do desenvolvimento das histórias. Durante o review o Product Owner já possui noção de praticamente todo o andamento do Sprint e quais são as histórias que foram completadas.

O que deve ser realizado durante esta etapa:

A equipe irá demonstrar cada uma das histórias desenvolvidas, sendo cada uma delas avaliada de acordo com o os casos de aceitação existentes

O Product Owner realizará os testes de cada uma das histórias para verificar se as condições estabelecidas foram satisfeitas

No caso de histórias não finalizadas a equipe deve discutir com o Product Owner os motivos do não atingimento do objetivo. Um balanço dos problemas encontrados, sejam eles corporativos, de erro de estimativa, mudanças de última hora, é necessário para que o Product Owner esteja ciente do ocorrido.

O ideal é que a validação seja realizada em ambiente de produção dado que o Scrum prega software funcionando como final da etapa, mas isto, também de acordo com a empresa pode variar, sendo possível em alguns casos validar em um ambiente de pré-produção.

### **3.5.4 Sprint Retrospective**

A reunião chamada de Sprint Retrospective tem como principal foco o time. Ela é realizada sempre ao final de cada Sprint e deve ser utilizada para a correção dos problemas detectados pelo time no processo de desenvolvimento, bem como para a implementação de melhorias a partir de sugestões.

As melhorias no processo de desenvolvimento são contínuas e podem demorar até um ano. Por ser um processo iterativo todas as mudanças devem ser gradativas, fazendo com que cada nível novo seja implementado, avaliado, testado e só posteriormente nova complexidade seja adicionada.

Outro item importante é manter o foco nos novos métodos pois existe a tendência dos times a retornarem ao uso de velhas práticas. Sem a melhoria contínua existe a possibilidade dos times perderem em performance já que não existem novos desafios.

Participam na retrospectiva o time e o Scrum Master, para os demais é necessário convite por parte do time. Este é o momento em que o Scrum Master tem o trabalho de ajudar o time a encontrar seus verdadeiros problemas. Um exemplo disto é quando a entrega realizada não contemplou todas as histórias previstas, faz parte do trabalho do Scrum Master ajudar o time a verificar qual foi o verdadeiro problema.

## **3.6 Artefatos**

### **3.6.1 Product Backlog**

O Product Backlog é a lista que contem as funcionalidades de negócio, os requisitos técnicos e os erros encontrados no sistema que precisam ser desenvolvida. É importante salientar que esta lista está em constante evolução, seja porque novos requisitos de negócio surgem, seja porque tecnicamente faz-se necessária manutenção no software ou um refatoramento.

É função do Product Owner manter a lista com suas prioridades e mostrar para o time quais são as perspectivas para o projeto. Um dos maiores desafios do Scrum é manter um Product Backlog organizado, seja com relação as prioridades ou com relação a complexidade necessária para o cumprimento das histórias.

Cada história existente no Product Backlog deve ter um valor de negócio e uma complexidade associada. O primeiro é de responsabilidade do Product Owner e o segundo do time, que deve ser sempre solicitado a reavaliar as estimativas constantes no Product Backlog.

É importante que sempre antes de um Sprint Planning o Product Backlog esteja pronto, esta tarefa não faz parte do Sprint Planning e a utilização do mesmo para realizar a tarefa de priorização completa e estimativa pode fazer com que o Sprint Planning dure dias ao invés de horas.

A definição de valor de negócio é extremamente complexa. Uma das técnicas utilizadas para a criação de um Product Backlog inicial é dar um valor de pontos fechado para a Product Owner dividir entre suas histórias. Isto pode ser descrito da seguinte maneira, imaginemos que o Product Owner precise construir um carro e para isto lhe são dados 2000 (dois mil) pontos de complexidade. Assim ele necessariamente terá um volume de pontos limitado e realmente dará os pontos de valor para aqueles que realmente importam (Carroceria, Motor, Câmbio) deixando menos pontos para aqueles que neste caso por exemplo seriam menos importantes (retrovisor, bancos de couro, etc..). O modelo anterior facilita pois desta forma o Product Owner tem um limite para gastar, da mesma forma que a empresa tem um orçamento limitado.

A questão da priorização é resolvida pelas necessidades do negócio e a complexidade geralmente utiliza uma ferramenta chamada Planning Poker.

#### **3.6.1.1 Planning Poker**

O Planning Poker é um exercício de estimativa que pode ser realizada em grupo e usa a sequência de Fibonacci (1,2,3,5,8,13,21,...) como base para a avaliação de complexidade. A lógica associada a utilização desta sequência está na diferença

existente entre cada uma o que permite claramente ver quão complexa é cada funcionalidade. As diferenças no caso mapeiam melhor as incertezas associadas a avaliação.

Tudo começa selecionando-se o item que todos acreditem ser o mais fácil de implementar, sendo a ele associado o menor valor da sequência. Posteriormente os demais itens são avaliados levando-se em consideração sua complexidade relativa a de menor valor. Cada um dos participantes dá sua nota de complexidade para cada história e caso haja consenso entre eles a complexidade é validada e atribuída a história. Em caso de divergência é aberto espaço para debate entre os membros que tiveram maior diferença e cada um pode expor seu ponto de vista. Após isto é realizada nova rodada e assim segue até o consenso. É responsabilidade do Scrum Master intervir caso o time não consiga chegar a um consenso na complexidade de uma história.

É importante que a equipe sempre mantenha o Product Backlog com estimativas coerentes com a sua velocidade de desenvolvimento. Por este motivo a cada Sprint o time deve reservar um parcela de seu tempo para rever o Backlog do produto e se necessário fazer novas estimativas ou corrigir estimativas já feitas baseados na experiência do projeto corrente.

### **3.6.2 Sprint Backlog**

O Sprint Backlog é gerado a partir das histórias retiradas do Product Backlog e que serão implementadas durante o Sprint.

Durante o Sprint Planning as histórias aceitas pela equipe são desmembradas em tarefas menores, de no máximo um dia, para que possam ser acompanhadas pela equipe durante o Sprint, além disto as histórias de aceitação devem ser redigidas e registradas para o aceite do Sprint, gerando novas tarefas. Este trabalho gera o Sprint Backlog, dele fazem parte todas as tarefas necessárias para alcançar o objetivo do Sprint. É importante salientar que durante o Sprint novas tarefas podem surgir, ou porque foram esquecidas ou porque determinada tarefa terá que ser dividida, estas novas tarefas fazem parte do Sprint e do Sprint Backlog.

Durante o Sprint caso todas as tarefas do Sprint Backlog já tenham sido realizadas é possível que o time, com o consentimento do Product Owner, adicione mais histórias do Product Backlog ao Sprint Backlog. Neste caso o time assume o compromisso de finalizar aquelas histórias até o final do Sprint.

O Sprint Backlog em conjunto com o Burndown Chart devem estar em constante acompanhamento pela equipe. É este acompanhamento que permite que a equipe veja com antecedência a possibilidade de não conseguir entregar histórias do Sprint.

Outro item importante sobre as histórias do Backlog é que elas em geral não devem ser desenvolvidas em paralelo. O paralelismo de histórias só deve acontecer se já não existem mais atividades na história anterior que não possam ser feitas. A tentativa de evitar o paralelismo tem como objetivo evitar que a equipe chegue no final do Sprint com todas as histórias quase prontas, mas nenhuma completamente terminada. Neste caso perde-se totalmente o conceito de velocidade da equipe visto que não se pode estimar o real tempo necessário para terminar as tarefas finais em paralelo.

### 3.6.3 Impediment Backlog

O Impediment Backlog possui a lista de impedimentos que podem gerar problemas na entrega, seja de um Sprint ou de um Projeto como um todo. Geralmente consiste de tarefas que não puderam ser realizadas pela equipe por pendências externas. Este backlog deve ser tratado pelo Scrum Master que deve agir como facilitador na resolução destes impedimentos.

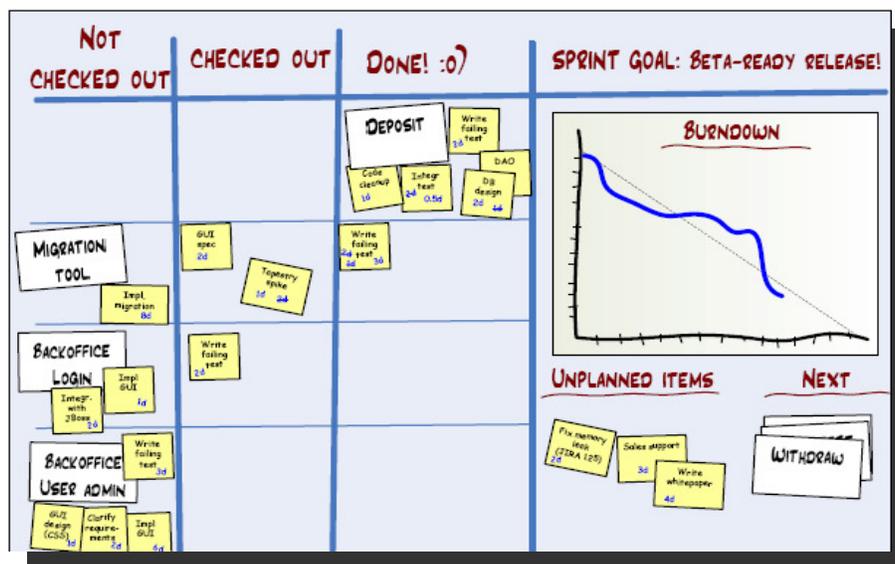
Do Impediment Backlog podem fazer parte tarefas como instalar um servidor ou simplesmente apressar o help desk para fazer a instalação de um software para um dos desenvolvedores e que será utilizado na entrega do Sprint.

Em grandes empresas costuma-se formar uma equipe de Gerentes e Diretores que tem como objetivo ajudar na resolução de problemas da companhia. A formação desta equipe geralmente é difícil mas em alguns casos é a que traz mais resultados no longo prazo. Para muitos dentro da empresa problemas recorrentes da estrutura não são vistos e são tratados como decorrentes do modelo da empresa. Uma equipe deste grau de maturidade tem como objetivo quebrar paradigmas e transformar a empresa em uma empresa ágil.

### 3.6.4 Burndown Chart

Como descrito anteriormente o acompanhamento das tarefas realizadas pelo time é controlado pelo próprio time. Durante o Sprint o time não deve ser influenciado por fatores externos e o acompanhamento das tarefas é diário através do Daily Meeting.

Nos Daily Meetings participam time e Scrum Master, podendo haver outros participantes, desde que como ouvintes. A partir das respostas das perguntas do Daily Meeting, tarefas que estão distribuídas são finalizadas e novas são distribuídas. A distribuição das tarefas é feita pelo time e cada membro da equipe seleciona as tarefas que pode realizar seguindo a ordem de execução definida pelo time. A partir da informação das tarefas realizadas no dia anterior é traçado um gráfico que indica a completude de tarefas do Sprint e seu andamento em comparação com o número de tarefas planejado.



## Exemplo de Burndown Chart

O Burndown é um gráfico simples que se baseia em atividades que não ultrapassem um dia de trabalho. O eixo X indica o número de tarefas existentes no Sprint e o eixo Y os dias que representam o tamanho do Sprint. A partir da quantidade de tarefas realizadas diariamente pode-se verificar se o Sprint está dentro do esperado. Caso as estimativas estejam erradas e o número de tarefas diários não é cumprido a linha azul tende a ir para a direita e distanciar-se da reta de atividades diária. Este é o indicativo de que provavelmente as tarefas foram mal avaliadas ou existe algum outro problema com a equipe.

A situação inversa é quando a linha azul fica abaixo da reta indicando que as tarefas foram superestimadas e o Sprint deve acabar antes.

Nas situações acima a equipe e o Product Owner devem tomar a decisão em conjunto do que deve ser feito.

Além do Burndown pode-se notar que o acompanhamento de tarefas ainda é realizada através de um quadro onde são divididas as tarefas à realizar, em andamento e já realizadas.

O acompanhamento diário e com atenção desta ferramenta faz a diferença na entrega das histórias no prazo. Como tarefas podem ser adicionadas diariamente uma análise incorreta do que está acontecendo com o desenvolvimento do Sprint pode gerar um grande atraso e penalizar o Sprint.

## 4 Scrum na Globo.com

As metodologias ágeis sempre foram assunto dentro da Globo.com principalmente por se tratar de um empresa de internet sujeita a grandes modificações de requisitos e ao ambiente inovador a que está inserida.

Várias pessoas já conheciam o conceito das metodologias ágeis mas na prática utilizávamos uma adaptação do Rational Unified Process como modelo de desenvolvimento de projetos.

A partir da necessidade da implantação de um projeto que todos achavam ser inviável no modelo existente, decidiu-se por adotar o Scrum em uma linha de desenvolvimento com o objetivo de entregar o aplicativo em questão.

O cenário existente era de uma empresa departamental, em que todas as áreas de desenvolvimento eram separadas (Design, Arquitetura da Informação, Desenvolvimento, Teste) e a interface entre as áreas tinha certos requisitos incompatíveis com a velocidade desejada. Os projetos eram modelados no seu limite e quando chegavam para a implementação em certos casos já existiam funcionalidades que necessitavam de ajuste.

A implementação começou com uma equipe de quinze pessoas, desenvolvendo um projeto de ampla visibilidade com um grande potencial de acesso por parte do usuário final. Como não existia nenhuma outra equipe utilizando o Scrum o processo gerava uma certa desconfiança por parte dos participantes da equipe, mesmo que a maioria deles estivesse extremamente comprometida com o objetivo do projeto.

Existia uma grande preocupação com o modelo do Scrum, onde não existe paralelismo. Uma história tem que ser trabalhada até que se chegue ao seu final e novas histórias não devem ser desenvolvidas até que a anterior seja finalizada. O

detalhamento do projeto não era comparável ao de outras vezes o que gerava certa insegurança sobre qual era o tamanho das funcionalidades e qual era sua complexidade.

Os primeiros resultados:

- no segundo Daily Scrum a primeira história foi concluída
- percepção de maior velocidade
- faltou planejamento no primeiro Sprint
- apenas metade do planejado foi desenvolvido mesmo com a percepção de maior velocidade
- sobraram recursos, algumas pessoas não tinham tarefas para fazer
- equipe resolveu melhorar planejamento e diminuir a equipe para nove pessoas

A partir dos primeiros Sprints notou-se maior motivação da equipe. Poder discutir o que seria implementado e ajudar na definição das funcionalidades fez com que a equipe se sentisse dona do produto. A incorporação de técnicas ágeis de desenvolvimento diminuiu o número de bugs e o tempo de testes pôde ser usado de forma mais objetiva para funcionalidades complexas.

Após quatro Sprints um produto funcional já existia, permitindo que os Clientes utilizassem o sistema e novas funcionalidades fossem sendo testadas e incorporadas.

#### **4.1 Implantação**

Após o sucesso do uso do Scrum no projeto do Aplicativo tomou-se a decisão de implantar o modelo em toda a empresa. Para isto definiram-se algumas premissas:

- treinar os Scrum Master e Product Owners no processo
- implantar em times piloto com o objetivo de avaliar o impacto nas pessoas, estruturar treinamento e avaliar os benefícios da metodologia
- implantar práticas ágeis de engenharia tais como Integração Contínua, Desenvolvimento baseado em Teste com o objetivo de melhorar a qualidade de software e facilitar o refatoramento das aplicações
- treinar os times. Esta premissa foi deixada de lado assumindo-se que o Scrum Master e o Product Owner conseguiriam assumir esta responsabilidade sendo o maior erro cometido na implantação. Inicialmente o time não conhece suas novas responsabilidades gerando grande desgaste interno.

Durante a implantação alguns desafios foram maiores, principalmente relacionados a manutenção da motivação da equipe e a integração das diversas áreas. Notou-se também a necessidade de um ambiente de colaboração entre Product Owner e Scrum Master com a necessidade de definição de objetivos comuns entre eles e com a própria equipe.

Os gerentes, antes departamentais, tiveram que assumir novas funções relacionadas a manutenção de padrões entre times e práticas de desenvolvimento, fazendo de forma mais próxima o acompanhamento do crescimento profissional dos seus recursos.

O Scrum tem como objetivo mostrar os problemas da empresa, estes problemas são levantados como impedimentos e no princípio a inexperiência no tratamento destes

impedimentos acabaram gerando desgaste desnecessário até 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. Mesmo com o treinamento é importante deixar claro antes de começar as atividades
- paralelismo de histórias. Os times retornam aos velhos métodos e paralelizam as histórias, no final do Sprint temos várias histórias inacabadas
- falta de Product Backlog Priorizado e Estimado
- não gerenciar as reuniões no formato de tempo fechado (*timebox*). Sprint Plannings demorando três dias.

Além dos itens acima existem também alguns cuidados que devem ser tomados com relação aos papéis, etapas e artefatos do Scrum que serão descritos abaixo.

#### **4.1.1 Scrum Master**

O Scrum Master é crucial durante a fase de implantação e o mais difícil neste caso é conseguir dividir seu tempo entre as tarefas de disseminar o processo e se preocupar com as pessoas.

Muitas vezes o time vê no Scrum Master alguém que está tentando apenas implantar o processo. Quando o Scrum Master pede que o time se auto-organize o time entende aquilo como se o Scrum Master estivesse forçando os membros do time para este tipo de organização, confundindo o Scrum Master novamente como o antigo gerente.

Durante os Daily Scrums iniciais membros do time detalham suas atividades mais para o Scrum Master do que para o próprio time parecendo mais uma reunião de estado atual de projeto, isto deve ser evitado.

O Scrum Master deve ser visto como o facilitador para a resolução dos problemas e não como quem resolve, o que é uma visão difícil para o time no início da implantação do processo.

Ao dar ênfase na implantação do Scrum deve-se ter em mente de que em alguns momentos o processo precisa ser adaptado e não somente as pessoas precisam se adaptar ao processo. Esta visão vem do Scrum Master que deve detectar dentro dos times o que deve ser mudado no processo.

Embora o dito acima seja verdadeiro, o Scrum Master deve ter a preocupação em manter a disciplina do processo, as estruturas e objetivos das reuniões e o time ciente das regras básicas como o comprometimento com as entregas do Sprint. Sem isto a tendência dos times é relaxar com relação ao processo e voltar a adotar velhas práticas.

### 4.1.2 Product Backlog

Neste caso a grande preocupação relacionada está no fato de que em alguns casos é difícil para o Product Owner tomar algumas decisões de priorização sobre o Product Backlog e mostrar para a equipe uma visão de produto consistente e que possa criar um time motivado.

Ter prioridades claras e a direção a ser tomada para um produto é de grande dificuldade. Aqui a idéia é usar a técnica do Scrum de entregar pequenas parcelas de um produto funcional para que seu cliente final ajude o Product Owner a decidir o que tem maior prioridade.

Após os passos anteriores e com uma visão clara do produto é sempre importante manter o Product Backlog priorizado e estimado. O Product Owner deve sempre solicitar ao time que reveja itens do Product Backlog e refaça as estimativas. Sem estimativas coerentes com a velocidade do time um Product Backlog é inútil na hora do planejamento do Sprint.

A questão da manutenção e a correção de problemas deve ser levada em consideração no Product Backlog. Depois de alguns Sprints quando a velocidade do time estiver ajustada é necessário avaliar quanto tempo está sendo gasto neste tipo de atividade. Além disto, incluir práticas de engenharia que diminuam o número de erros de produção deve ser uma constante na melhoria do processo, principalmente quando se utilizam técnicas ágeis.

O Product Owner deve estar atento para histórias geradas pelo time relacionadas a débitos técnicos que devem ser corrigidos. Este tipo de história se indefinidamente postergado tende a diminuir a produtividade do time.

### 4.1.3 Scrum Team

Com relação aos times uma grande preocupação é adequar os especialistas ao novo modelo de trabalho com iterações curtas. Em alguns casos é importante preservar o especialista fazendo com que ele participe apenas das atividades nas quais é necessário e em alguns casos nem mesmo fazendo parte do time tendo apenas o compromisso de suas entregas.

Ter somente especialistas no time pode diminuir a colaboração, deve-se então encontrar atividades que sejam possíveis de divisão. O importante nestes casos é ajustar a equipe com parcimônia e mostrar que os objetivos continuam sendo do time e não apenas daqueles que podem realizar determinada tarefa.

É importante que todos dentro do time estejam dispostos a colaborar, o trabalho individual não deve impactar na entrega do time.

Para times mais antigos que entram no novo modelo é importante inculir o conceito de pressão entre pares. Muitas vezes acostumados com o trabalho em equipe e o acompanhamento direto de um coordenador ou gerente, determinados membros do time tendem a ter uma entrega menor no novo modelo. Se o time não cobra o membro e opta por uma posição de ignorar, retirando deste indivíduo atividades importantes o time perde em produtividade e terá em pouco tempo um membro sem participação e entrega. Neste caso o Scrum Master deve estar atento para este tipo de comportamento que deve ser rapidamente corrigido.

É importante que durante a reunião de retrospectiva todos os aspectos, inclusive os pessoais, sejam discutidos. A colaboração dentro do time é fundamental e quando uma

das peças começa a não entregar, o time deve ficar preocupado e ser o primeiro a solicitar a substituição.

#### **4.1.4 Daily Scrum**

É importante manter a restrição de tempo e diminuir as interferências externas a equipe. Participação contínua e que influencie na entrega do time deve ser rapidamente gerenciada pelo Scrum Master.

#### **4.1.5 Sprint Planning**

Dois problemas são mais visíveis no Sprint Planning:

- time assumir muita responsabilidade
- time entrar na zona de conforto e não assumir desafios

Com relação a primeira, existe uma tendência natural inicial do time em assumir mais itens do que o normal, visto que as histórias inicialmente tem pouca descrição e o projeto não está tão detalhado. Isto tende a diminuir com o tempo, assim que a equipe consegue um melhor parâmetro de estimativas e sua velocidade se torna mais previsível.

A segunda é um problema maior. A partir dos erros cometidos ao pegar muitas histórias, a relação entre Product Owner e Scrum Team pode se deteriorar um pouco. Isto leva o time a assumir menos tarefas do que as que pode realmente entregar, ficando desta forma na zona de conforto e entregando de forma correta todos os seus Sprints. Neste caso Scrum Master e Product Owner devem acompanhar de perto a equipe e avaliar a entrega de cada um dos indivíduos para verificar se existe margem para solicitar maior complexidade nos próximos Sprints.

#### **4.1.6 Sprint**

Embora forneça ferramenta para o acompanhamento diário das atividades e controle sobre as tarefas executadas no Sprint o Burndown Chart deve ter cuidado especial tanto da equipe quanto do Scrum Master e Product Owner.

Em muitos casos a equipe não faz uma correta divisão das tarefas a executar, gerando durante o Sprint a divisão de várias delas. Quando isto acontece e as tarefas geradas somadas tem tempos adicionais a que a gerou, o Sprint fica comprometido. Por este motivo é importante que as equipes vejam a necessidade de um bom planejamento, inclusive com a criação correta das tarefas relacionadas a cada história, pois somente isto faz com que seja possível ter idéia de que o acompanhamento está correto.

Além disto é importante lembrar que a atualização do Burndown e das tarefas é de responsabilidade do time, mantê-lo com as informações corretas, para que possa ser acompanhado por todos na empresa é tarefa diária.

A utilização de ferramentas para manutenção do backlog e do acompanhamento deve ser adiada. Inicialmente os times tem que participar do processo e entender inclusive o que pode ser melhorado para que algo que tenha realmente valor seja implantado.

## 4.2 Pós Scrum

Pode-se notar após a implantação do Scrum que as equipes ficaram mais motivadas. O processo de estimativas e o auto-gerenciamento fez com que se criasse uma relação mais próxima com a área de produto que entendeu melhor o processo de desenvolvimento. 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 em Sprints permitiu que os projetos fossem melhor priorizados e seu tamanho discutido, eliminando funcionalidades que aparentemente não possuem valor de negócio relevante.

Mudanças culturais são iniciadas pela implantação do processo e durante este período é importante manter claro o objetivo para todos, desde o time até a presidência, visto que novas necessidades de adequação do processo surgem diariamente.

Após doze meses do início da implantação, hoje temos dezesseis times Scrum trabalhando, todos em estágios diferentes de maturidade, cada produto tem sua estrutura de trabalho e diversos passos ainda para percorrer, mas o resultado é positivo e visível.

## 5 Conclusão

O processo de desenvolvimento ágil para empresas que trabalham com internet é uma realidade. O modelo de conceituação de produto e a necessidade cada vez maior de velocidade só se faz possível com um processo que permita pequenas entregas.

Este modelo dá ao consumidor final a oportunidade de testar rapidamente o produto e, ao atingir uma gama extremamente diferente de usuários o Product Owner pode reavaliar os requisitos e redirecionar o desenvolvimento para atender as necessidades corretas.

O Scrum possui um modelo de implementação simples, dele fazem parte poucas especificações e artefatos, o grande desafio é integrá-lo as necessidade da empresa, do cliente e do time, fazendo as adaptações necessárias.

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

SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. New Jersey: Prentice Hall, 2002.

SCHWABER, K. **Agile Project Management with Scrum**. Redmond: Microsoft Press, 2004.

SCHWABER, K. **The Enterprise and Scrum**. Redmond: Microsoft Press, 2007.

VERSIONONE. **The State of Agile Development**. Disponível em:

<http://www.versionone.com/pdf/StateofAgileDevelopmentSurvey.pdf> Acesso em: 30 jun.2008

COHN, M.; FORD, D. Introducing an Agile Process to an Organization. **Computer**, v. 36,p. 74, 2003.

AGILEMANIFESTO. **Manifesto for Agile Development**. Disponível em:

<http://www.agilemanifesto.org> Acesso em: 30 jun.2008

MOUNTAIN GOAT SOFTWARE, **The Scrum Development Process**. Disponível em:

<http://www.mountangoatsoftware.com/Scrum> Acesso em: 30 jun.2008

WIKIPEDIA, **Scrum (development)**. Disponível em:

[http://en.wikipedia.org/wiki/Scrum\\_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development)) Acesso em: 01 jan.2009