

Nesta seção você encontra artigos voltados para as práticas e métodos ágeis.

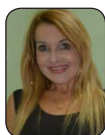


Implantando práticas do Scrum e do XP em um projeto de software

Veja como aplicar duas das mais importantes metodologias ágeis

Fique por dentro:

Este artigo é útil a todas as pessoas que escolhem o desenvolvimento de software com o uso de métodos ágeis. Através dele é possível observar a simulação prática de um projeto de software com o uso dos frameworks ágeis Scrum e Extreme Programming (XP). O artigo inicia com conceitos e práticas do Scrum e do XP e depois realiza a simulação de um projeto com base em algumas práticas.



Celina Ferreira Ribeiro

ribeira.f.celina@gmail.com
celinafr Ribeiro.wordpress.com

Sou graduada no curso de Tecnologia em Sistemas para a Internet do Instituto Federal do Tocantins. Atuei como líder de projetos de tela e analista de requisitos no projeto SIGA-EDU da Rede Federal de Ensino (Renapi). Trabalhei como analista de sistemas e chefe de tecnologia da informação no Instituto de Gestão Previdenciária do Tocantins. Atualmente, trabalho com projetos e publicações de artigos e tenho artigos publicados nas revistas .Net Magazine, Engenharia de Software, entre outras revistas e eventos nacionais.

Por muito tempo os padrões para o desenvolvimento de software, tanto na academia quanto na indústria, foram norteados pelos métodos tradicionais. Entretanto, ao observar o resultado da indústria de software, que apresentava um acentuado número de projetos fracassados, alguns profissionais da área começaram a adotar processos de trabalho diferentes dos preconizados pelas metodologias tradicionais. Devido à baixa formalidade existente nas novas metodologias adotadas (ágeis) quando comparadas com os métodos tradicionais de desenvolvimento, elas foram inicialmente chamadas de leves. Ganham destaque em ambientes acadêmicos e empresariais, mas geraram divergências entre os profissionais principalmente em relação à confiabilidade dos processos e à qualidade de software.

As consideráveis mudanças que os métodos ágeis trouxeram em sua essência tornaram-se polêmicas e não inspiraram confiança aos profissionais mais conservadores. Por outro lado, uma parcela crescente de profissionais vem utilizando métodos ágeis em projetos de software, ganhando principalmente o apoio inicial do cliente já que prazos mais curtos para entregas efetivas representam um dos principais objetivos da metodologia.

Assim, é cada vez mais comum o desenvolvimento de software com a utilização de métodos ágeis. Esses objetivam reduzir o ciclo de vida do software e, consequentemente, acelerar

o seu desenvolvimento. Além disso, fornecem outros diversos benefícios, como melhor definição do objetivo, organização diária para alcançar a meta definida, mais independência e produtividade para a equipe, respostas rápidas às mudanças, aumento da comunicação da equipe e melhor atendimento ao cliente.

São muitas as propostas ágeis, porém neste artigo vamos trabalhar com o Scrum e com a Extreme Programming. O Scrum não é um processo ou uma técnica para desenvolver produtos, é um framework pelo qual se podem empregar várias técnicas ou processos. Ele pode ser utilizado para desenvolver softwares comerciais, sites, aplicações para redes de computadores, smartphones, etc. O framework possui diversas características, entre elas destacam-se:

- A transparência no planejamento e no desenvolvimento de tarefas;
- A participação dos clientes;
- As entregas frequentes com as funcionalidades concluídas;
- Os planos de mitigação de riscos;
- O fato dos problemas não serem ignorados;
- As reuniões diárias para discutir a evolução do projeto; e
- A possível aplicação em projetos de software de pequeno e grande porte.

Já a Extreme Programming (XP) busca favorecer o cumprimento das estimativas e satisfazer as necessidades do cliente, permitindo construir um projeto de software com tempo menor. O XP é um processo leve, focado em pessoas e permite a qualidade no desenvolvimento do software. Ele descarta o excesso de comunicação, a burocracia no projeto e o excesso de documentação, sendo a documentação do software o próprio código fonte.

A metodologia possui alguns princípios, como feedback rápido, simplicidade, mudanças incrementais, disposição a mudanças e trabalho de qualidade. O XP é fundamentado em práticas, as quais estão representadas na **Figura 1**.

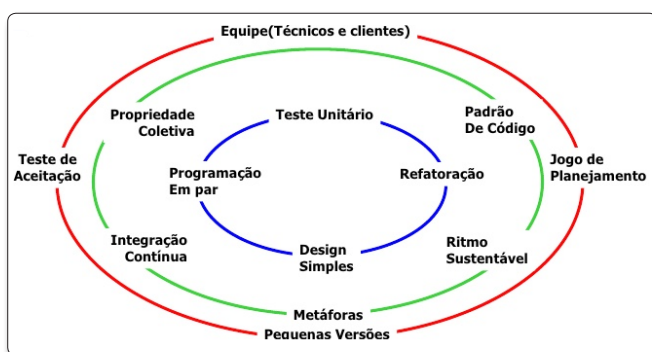


Figura 1. Práticas do XP.

Ele vem atraindo cada vez mais desenvolvedores já que sua metodologia aproxima o cliente da equipe de desenvolvimento de forma a proporcionar a integração total no desenvolvimento do sistema. Assim como o Scrum, o XP também é um método ágil de desenvolvimento de software que permite

a utilização de algumas práticas no seu projeto, não sendo obrigatório o uso de todas elas. Nada impede, por exemplo, utilizar algumas práticas do Scrum juntamente com algumas do XP.

Implantando práticas do Scrum e do XP

A empresa fictícia Hotel Belo Mar necessita de um portal para reservas online e contratou uma equipe para desenvolvê-lo. A equipe foi criada conforme a metodologia Scrum, e seus profissionais possuem habilidades para realizar qualquer tarefa no desenvolvimento do software. O Scrum recomenda que todos os integrantes da equipe tenham habilidades para desenvolver qualquer papel, isso inclui a análise, desenvolvimento, integração e testes da funcionalidade.

Com o time montado, designou-se o Product Owner e o Scrum Master para trabalhar no projeto de reservas online e, depois, agendou-se uma reunião para apresentar o projeto para o time de desenvolvimento. Falarei mais sobre o Product Owner e o Scrum Master ao longo do artigo, mas caso deseje ler mais sobre eles acesse o guia oficial do Scrum disponível na seção **Links**.

A apresentação do projeto para o time de desenvolvimento é denominada no Scrum de “visão do produto”. Nela, o Scrum Master utiliza o material de apoio (por exemplo: vídeos, slides e filmes) para facilitar o entendimento da equipe. Finalizada a apresentação do projeto, a próxima etapa a realizar é o Product Backlog, que pode ser entendido como uma lista de requisitos e histórias. Porém, antes de falar sobre ele precisamos primeiramente entender como funciona a especificação de requisitos em metodologias ágeis e o conceito de histórias.

A etapa de requisitos em metodologias ágeis é chamada de Requisitos Ágeis (RA). Diferentemente das metodologias tradicionais, que são mais formalmente documentadas e controladas, em RA existe uma minimização do foco na documentação com um direcionamento para entregas de software cada vez mais rápidas. Enquanto que nos métodos tradicionais a especificação dos requisitos é feita geralmente através de casos de uso, nos métodos ágeis essa especificação muitas vezes é feita através de histórias de usuário.

Histórias de usuário são artefatos de desenvolvimento utilizados em sistemas geridos segundo metodologias ágeis. Elas fracionam os requisitos para que seja possível estimar o esforço para realizar determinado objetivo de forma mais fácil e objetiva. São descrições simples de uma funcionalidade, e é recomendável que sejam escritas segundo o ponto de vista do usuário.

Uma história de usuário deve ser escrita considerando três variáveis: (1) quem, que representa as pessoas envolvidas, (2) o que, que representa as ações e necessidades do cliente e (3) por que, que representa o efeito no negócio. Para que esse direcionamento seja possível, foi definido como boa prática o seguinte padrão de descrição para a definição de histórias do usuário: Como (usuário/papel), quero (meta) para que eu possa (motivo). Dois exemplos que consideram esse formato são:

- Como candidato a emprego, quero procurar um trabalho para que eu possa avançar na minha carreira;

- Como recrutador, quero postar uma vaga de trabalho para que eu possa encontrar um novo membro da equipe.

Com esses conceitos bem definidos, voltemos ao Product Backlog (a **Figura 2** apresenta um Product Backlog). Essa lista de requisitos e histórias funciona da seguinte forma:

1. O Product Owner se reúne com as partes interessadas no projeto e define os itens do Product Backlog.
2. Os itens do Backlog são inseridos na sequência correta, sendo os de maior prioridade no início da lista e os de menor prioridade no final.

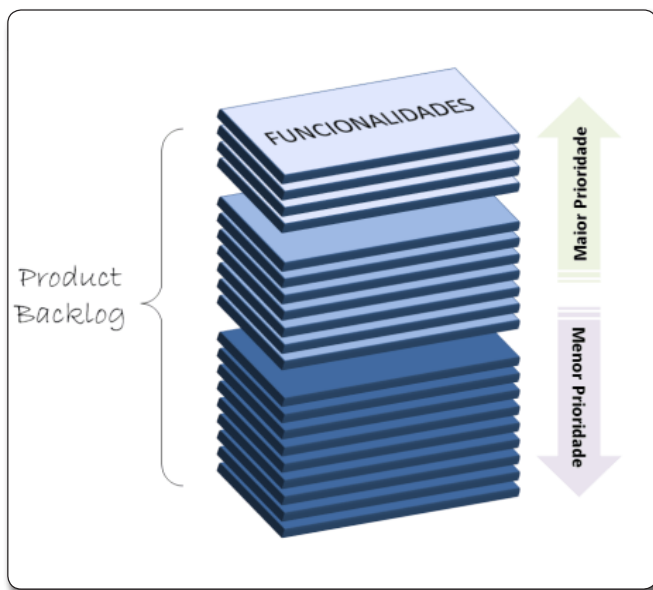


Figura 2. Product Backlog

O Product Backlog possui os seguintes campos:

- **Nome:** define-se um nome para cada história, por exemplo “promoção”. É importante que o nome seja claro e pequeno, de forma que a equipe entenda e consiga separar uma história das demais;
- **Como demonstrar:** breve descrição de como a história será exibida na exposição da Sprint (veja o **BOX 1**), por exemplo: “o portal deve permitir a inserção de promoções”.

BOX 1. Sprint

No Scrum, as tarefas são divididas em iterações, ou ciclos, de curta duração, que recebem o nome de Sprint.

A **Tabela 1** apresenta o Product Backlog para a empresa fictícia Hotel Belo Mar antes da reunião de planejamento de Sprint.

No exemplo da **Tabela 1**, o primeiro item do Product Backlog é “Reserva”, dessa forma, a reserva é o item de maior prioridade. O Product Backlog pode ser criado em planilhas, desenvolvidas, por exemplo, no software Excel com compartilhamento habilitado para facilitar a visualização de todos os envolvidos no projeto.

Nome	Como demonstrar
Reserva	O site deve permitir reservas online.
Visualização de quartos	O portal deve disponibilizar a visualização de quartos.
Promoção	O portal deve permitir a inserção de promoções.

Tabela 1. Product Backlog

Na confecção do Product Backlog foi possível a adoção de valores da Extreme Programming, como a comunicação e o feedback, valores esses muito importantes para a definição dos itens do Backlog

Planejamento da Sprint

Concluído o Product Backlog, é necessário planejar a Sprint. A reunião de planejamento da Sprint fornece ao time informações suficientes para trabalharem por algumas semanas. O resultado dessa reunião deve ter:

- Um objetivo da Sprint (o que se pretende entregar);
- Um Backlog para a Sprint (lista de histórias da Sprint originada do Product Backlog);
- A data escolhida para a apresentação da Sprint;
- A data e local previsto para as reuniões diárias.

Na reunião de planejamento, o Product Owner resume o objetivo da Sprint e as histórias mais importantes, e, em seguida, o time analisa e estima o tempo de cada história. A estimativa de tempo é a estimativa inicial do time em relação ao tempo necessário para implementar cada história. A unidade de tempo usada é “pontos por história”, que corresponde a medida ideal “homem/dias” trabalhando.

Para fazer o planejamento, a equipe de negócios, acompanhada pela equipe de desenvolvimento, descreve as funcionalidades que necessita nos cartões de história. Após a definição das tecnologias e ferramentas que serão utilizadas, os desenvolvedores estimam o volume de trabalho de cada funcionalidade e anotam no próprio cartão de história. Todo o processo de escrita de cartões, priorização e estimativas faz parte do “Jogo do Planejamento”. Através desses elementos a equipe fará o planejamento das versões do produto, agrupando os cartões de acordo com funcionalidades de maior relevância para os usuários.

Um exemplo para estimativa de tempo é: “José, se você pegar essa história, em quantos dias você termina?”. A resposta de José é: “consigo fazer em 3 (três) dias”. Dessa forma, temos 3 (três) como o total de pontos pois José trabalhará sozinho na história.

O Scrum fornece uma boa técnica para estimar tempo, de nome Planning Poker. Na Planning Poker os integrantes do time recebem baralhos contendo treze cartas:

- Cartas numeradas: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40 e 100;
- Cartas com símbolos: ? (interrogação) e uma carta com o desenho de uma xícara de café.

As cartas especiais significam:

- 0: história concluída ou quase concluída;

- **100:** história grande, precisando ser dividida em tarefas menores;
- **?:** história indefinida;
- **Xícara de café:** pausa para o café, água ou descanso.

A **Figura 3** esboça as cartas do Planning Poker:



Figura 3. Cartas do Planning Poker

Antes de iniciar o Planning Poker é necessário definir um valor de referência, ou seja, é preciso utilizar uma história que demanda um tempo menor para servir de referência para pontuar as demais. Definida a primeira história, é preciso ler as outras e pontuá-las conforme a numeração das cartas e o grau de dificuldade.

O Planning Poker funciona da seguinte maneira:

1. Com as treze cartas em mãos, o Product Owner ou o Scrum Master lê a descrição da história;
2. Cada membro do time escolhe uma carta para representar a sua estimativa de tempo (em pontos por história) e a deixa virada sobre a mesa;
3. Quando todos estimarem, as cartas são reveladas simultaneamente;
4. Ocorrendo diferença exorbitante entre duas estimativas, o time discute até chegar em um consenso.

O Planning Poker permite resultados satisfatórios devido à contribuição de todos os membros do time, assim como da responsabilidade deles com a estimativa de tempo. A **Figura 4** ilustra a realização de um Planning Poker.



Figura 4. Planning Poker

Concluído o Planning Poker, o Product Owner adiciona a estimativa no Product Backlog e em seguida cria o nível de prioridade dos itens do Backlog. Para determinar a prioridade, podem-se responder as seguintes questões:

1. Quais os itens do Product Backlog retornam mais valor ao negócio?
2. Quais itens devem ser entregues primeiro?

O Scrum recomenda que a priorização dos itens do Product Backlog seja por categoria, já que a maioria das histórias possui dependência. Uma técnica que ajuda a priorização é a Theme Screening.

Para trabalhar conforme a Theme Screening é preciso seguir alguns passos:

1. Verificar os fatores importantes na priorização;
2. Escolher um Baseline, ou seja, decidir o tema que será base para os demais;
3. Determinar a tabela de pesos:
 - +: melhor que a referência estabelecida;
 - 0: igual a referência estabelecida;
 - : pior que a referência estabelecida.
4. Classificar conforme o score. Quanto maior o score, maior a prioridade do item do Product Backlog.

A **Tabela 2** esboça a priorização dos itens para a empresa fictícia Hotel Belo Mar com base na Theme Screening.

A **Tabela 3** apresenta a estimativa definida no Planning Poker e o nível de prioridade dos itens no Product Backlog.

Com base no Planning Poker da empresa fictícia Hotel Belo Mar (**Tabela 3**) foram estimados: reserva (40 pontos), visualização de quartos (20 pontos) e promoção (20 pontos). As ordens de prioridade são: reserva (2), visualização de quartos (1) e promoção (0).

	Temas		
Critérios para Análise	Promoção	Reserva	Visualização de Quartos
Baseline	0	0	0
Reserva para carnaval	0	+	+
Reserva para férias	0	+	0
Score	0	2	1

Tabela 2. Theme Screening

Nome	Como demonstrar	Estimativa	Prioridade
Reserva	O site deve permitir reservas online.	40	2
Visualização de quartos	O portal deve disponibilizar a visualização de quartos.	20	1
Promoção	O portal deve permitir a inserção de promoções.	20	0

Tabela 3. Product Backlog com estimativa e prioridade

Concluído o Product Backlog, determina-se a hora e local da reunião diária e, em seguida, finaliza-se o planejamento da Sprint.

Decompondo histórias em tarefas

Existe diferença entre histórias e tarefas. Tarefas são atividades estimadas para serem desenvolvidas e entregues, enquanto que as histórias são conjuntos de tarefas. Decompor história em tarefas é bem simples, por exemplo: a história “reserva” pode ser decomposta em tarefas como colher requisitos, documentar requisitos, implementar tela e realizar testes. A **Tabela 4** exibe a história “reserva” da empresa fictícia Hotel Belo Mar decomposta em tarefas.

História	Tarefas
Reserva	<ul style="list-style-type: none"> - Colher requisitos. - Documentar requisitos. - Implementar tela. - Realizar testes.

Tabela 4. Decompondo história em tarefas

Sprint Backlog

Para dar continuidade ao projeto, é preciso criar um Sprint Backlog. O Sprint Backlog é uma lista de tarefas extraída do Product Backlog que o time se compromete a fazer em uma Sprint. Ele pode ser feito no software Excel, Jira ou em um quadro de tarefas fixado na parede.

O Sprint Backlog desenvolvido em quadro de papel fixado em parede geralmente possui os seguintes elementos: número da Sprint, objetivo da Sprint e as atividades não iniciadas, em andamento e concluídas. Um exemplo de Sprint Backlog é apresentado na **Tabela 5**.

Sprint n°: 01	Objetivo: Conclusão do módulo reserva	
Não iniciado	Em andamento	Concluído
<div>Reserva</div> <div>1. Colher Requisitos</div> <div>2. Documentar</div> <div>3. Implementar</div> <div>4. Fazer Testes</div>		

Tabela 5. Sprint Backlog

As atividades da Sprint número 01 para o projeto da empresa fictícia Hotel Belo Mar são: colher requisitos, documentar, implementar e fazer testes. Essas atividades apresentam status inicial de “não iniciada”. Ao serem iniciadas, passarão para o status “em andamento” e, posteriormente, para “concluído”.

Reuniões

No Scrum, as reuniões são constantes. A Daily Scrum acontece diariamente no mesmo local e horário. Nela, os integrantes ficam em pé para facilitar a circulação no ambiente, bem como para reduzir os riscos de ultrapassar o limite de quinze minutos.

Na Daily Scrum a equipe responde a três perguntas:

1. O que fiz ontem?
2. O que farei hoje?
3. Existe impedimento?

Um exemplo de Daily Scrum é apresentado na **Figura 5**, onde se tem os integrantes dispostos em pé em frente ao Sprint Backlog do projeto.



Figura 5. Daily Scrum

Na Daily Scrum, o Sprint Backlog é atualizado conforme as respostas das três perguntas (“o que fiz ontem?”, “o que farei hoje?” e “existem impedimentos?”). Em algumas equipes, as tarefas do Sprint Backlog são atualizadas na Daily Scrum pelos próprios integrantes da equipe, já em outras o Scrum Master é que é o responsável pela atualização das tarefas.

No término da primeira Daily Scrum do projeto da empresa fictícia Hotel Belo Mar foi definido que toda a equipe ficasse responsável pela atualização do Sprint Backlog de forma que todos tenham conhecimento do status da Sprint.

Finalizada a primeira Daily Scrum do projeto, o Sprint Backlog que estava com a atividade **1. Colher requisito** no status “não iniciado” foi atualizado conforme a **Tabela 6**, passando para “em andamento”.

A Sprint Review (revisão de Sprint) acontece ao final de cada Sprint. Nessa reunião são realizadas alterações, quando necessárias, no Product Backlog. Na Sprint Review, o Product Owner elucida os itens concluídos e não concluídos do Product Backlog, o time discute os fatores positivos e negativos ocorridos na Sprint e o Product Owner cogita as datas de conclusão baseadas em progressos do projeto. A Sprint Review possui *time-box* de no máximo quatro horas de duração para uma Sprint de trinta dias. O artefato gerado no término dessa reunião é a revisão do Product Backlog.



Sprint nº: 01	Objetivo: Conclusão do módulo reserva	
Não iniciado	Em andamento	Concluído
		

Tabela 6. Sprint Backlog atualizado

Outra reunião do Scrum é a Sprint Retrospective. É nela em que se avalia o que deu certo e o que deu errado durante a Sprint. Depois, os ajustes para a próxima Sprint são feitos. Assim como as outras reuniões do Scrum, essa também possui *time-box* definido, sendo esse de no máximo três horas de duração para uma Sprint de um mês. Geralmente na Sprint Retrospective os participantes preenchem um quadro contendo três colunas:

1. Bom: o que faríamos do mesmo jeito se fossemos refazer a Sprint;
2. Ruim: o que faríamos de outra maneira se fossemos refazer a Sprint;
3. Possíveis melhorias: possíveis ideias concretas de melhorias futuras.

Um exemplo de quadro da Sprint Retrospective para a empresa Hotel Belo Mar segue na Tabela 7.

Bom	Ruim	Possíveis Melhorias
Tecnologia definida	Dificuldade em coletar os requisitos de usuário.	Atualizar conhecimento de ferramentas tecnológicas.

Tabela 7. Quadro de Retrospectivas de Sprint

Inserindo práticas do XP

Para o projeto da empresa Hotel Belo Mar, a equipe adotou algumas práticas da Extreme Programming, como a programação em par, o design simples, a propriedade coletiva do código e a padronização do código.

A programação em par do XP é aquela em que dois desenvolvedores se sentam no mesmo computador para escrever um código fonte, sendo um deles responsável por codificar, pensar nos algoritmos e na lógica de programação e o outro, por observar o código fonte produzido, pensar em como melhorá-lo, torná-lo mais simples, apontar os erros e possíveis falhas.

Na programação em par do XP, as duplas e os papéis são trocados constantemente para que todos os integrantes da equipe obtenham conhecimento sobre todas as partes do projeto. Essas trocas são muito úteis uma vez que permitem que o conhecimento sobre as regras de negócio do projeto seja compartilhado, fortalecem a prática coletiva do código fonte e nivelam o conhecimento técnico dos desenvolvedores da equipe.

O design simples é outra prática do XP muito importante, pois permite que o código se torne simples e claro de forma a facilitar o entendimento e possível continuidade de desenvolvimento por qualquer outro integrante da equipe do projeto. Algumas características do design simples são: os testes são executados com sucesso, o projeto não possui lógica duplicada e possui o menor número de classes e métodos possível.

A propriedade coletiva do código faz com que toda a equipe trabalhe unida em busca de qualidade no código fazendo melhorias e refatoramentos a qualquer momento. Como o XP prega a propriedade coletiva do código, tornam-se necessários padrões de codificação a fim de facilitar o entendimento do código e futuras alterações.

Os métodos ágeis surgiram com o objetivo de reduzir a quantidade de problemas enfrentados por projetos de software desenvolvidos utilizando metodologias tradicionais. Embora venham alcançando alguns resultados positivos, existem algumas características que, caso não sejam tratadas, podem levar ao fracasso projetos que utilizam métodos ágeis.



Veja na seção **Links** as 10 causas mais comuns de problemas no desenvolvimento de projetos que utilizam XP e/ou Scrum segundo Henrik Kniberg.

Para lidar com problemas que surgem, nada impede o uso do Scrum e do XP em um só projeto. Para um bom projeto ágil, uma dica é utilizar as práticas que melhor se adequem a ele. Combinar o Scrum e XP é fácil, pois o Scrum é focado nas práticas de gerenciamento e organização, enquanto o XP, nas tarefas de programação.

Este artigo tratou sobre a teoria e a prática dos frameworks de princípios ágeis Scrum e Extreme Programming. Na teoria, citaram-se os principais conceitos e, na prática, simulou-se um projeto para um hotel combinando os dois frameworks.

O Scrum e o XP são bastante utilizados em projetos ágeis, e conhecer as suas práticas e combinar os seus recursos favorece a integração da equipe, o entendimento dos membros em todo o projeto e a certeza de que o projeto será desenvolvido conforme os requisitos do cliente.

As práticas gerenciais são diretamente adotadas pelo Scrum, já algumas práticas mais técnicas são adotadas pelo XP. Além de combinar práticas das duas metodologias, podemos também utilizar outras práticas de outros frameworks, como as definidas no Project Management Body of Knowledge (PMBOK).

Links:

Scrum: A metodologia ágil explicada de forma definitiva

<http://www.mindmaster.com.br/scrum>

Um guia definitivo para o Scrum: As regras do jogo

<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>

Guia oficial do Scrum (em inglês)

<http://www.scrumguides.org/>

11 Passos essenciais para implantar o Scrum no seu projeto

<http://www.mindmaster.com.br/scrum-11-passos/>

Como criar, estimar, priorizar e manter o Product Backlog

<http://pt.slideshare.net/Ridlo/workshop-como-criar-estimar-priorizar-e-manter-o-product-backlog>

10 ways to screw up with scrum and XP (vídeo)

<https://www.infoq.com/presentations/Fail-Scrum-Henrik-Kniberg>

10 ways to screw up with scrum and XP (resumo)

<http://gc.blog.br/2008/08/07/agile-2008-conference-henrik-kniberg-10-ways-to-screw-up-with-scrum-and-xp/>

