

Scrum em Ação

**Gerenciamento e Desenvolvimento Ágil
de Projetos de Software**

**Andrew Pham
Phuong-Van Pham**

Original English language edition Copyright © 2012 Course Technology, a part of Cengage Learning. Portuguese-language edition for Brazil copyright © 2012 by Novatec Editora. All rights reserved.

Edição original em Inglês Copyright © 2012 Course Technology, parte da Cengage Learning. Edição em Português para o Brasil copyright © 2012 pela Novatec Editora. Todos os direitos reservados.

© Novatec Editora Ltda. 2012.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Tradução: Edgard B. Damiani

Revisão gramatical: Patrícia Zagni

Editoração eletrônica: Carolina Kuwabata

ISBN: 978-85-7522-285-0

Histórico de impressões:

Dezembro/2011 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Fax: +55 11 2950-8869

E-mail: novatec@novatec.com.br

Site: www.novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Pham, Andrew
Scrum em ação : gerenciamento e desenvolvimento
Ágil de projetos de software / Andrew Pham e
Phuong-Van Pham ; [tradução Edgard B. Damiani]. --
São Paulo : Novatec Editora : Cengage Learning,
2011.

Título original: Scrum in action : Agile
software project management and development.
Bibliografia.
ISBN 978-85-7522-285-0

1. Administração de projetos 2. Desenvolvimento
Ágil de software 3. Scrum (Desenvolvimento de
software) 4. Software - Desenvolvimento I. Pham,
Phuong-Van. II. Título.

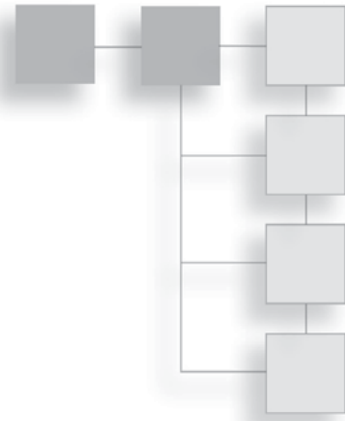
11-11327

CDD-005.1

Índices para catálogo sistemático:

1. Scrum : Gerenciamento e desenvolvimento Ágil
de projetos de software : Ciência da
computação 005.1
VC20111208

Preparando o palco: Ágil e Scrum



Semelhantemente a muitas pessoas, estamos animados com a adoção do gerenciamento e desenvolvimento de projetos Ágeis (adaptáveis) em geral e com a adoção do Scrum em particular.

Diferentemente de várias pessoas que são muito dogmáticas em relação à metodologia Ágil ou ao Scrum, no entanto não acreditamos que o mundo corporativo atual tenha se reorganizado completamente para o Scrum. Talvez algumas das empresas comerciais tenham feito isso, mas a maioria das corporações não despediu seus especialistas ou não lhes deu um novo título generalista, como os dogmatistas do Scrum acreditam que deveria acontecer. A maioria dessas empresas ainda mantém sua organização atual do jeito que está, com várias funções separadas em seus departamentos de TI. CapitalOne, uma das empresas que adotou a metodologia Ágil e costuma ser citada por vários instrutores de Scrum, ainda está contratando analistas de negócios, analistas de sistemas de negócios e gerentes de projeto. Isso também se aplica a muitas das grandes empresas conhecidas por terem feito a transição para o Scrum, como Sabre, Verizon, NBC Universal, General Dynamics, Texas Instruments e American Airlines. com, apenas para citar algumas delas.

Vamos falar mais sobre a origem e os fundamentos do Scrum neste capítulo, mas, por ora, digamos que o Scrum, tal qual no esporte rugby, é uma forma de reiniciar o jogo, seja após um incidente, seja depois que a bola sai de jogo. A ideia é manter o jogo (desenvolvimento de software) rolando.

Mesmo que o Scrum possa ser usado fora do desenvolvimento de software, neste livro, vamos focar o Scrum exclusivamente como uma estrutura de processo Ágil para gerenciar e desenvolver projetos de software.

Tendo gerenciado projetos no estilo tradicional de controle e comando por um longo tempo, ambos testemunhamos, de tempos em tempos, que a metodologia Ágil e o

Scrum parecem auxiliar nossas equipes a produzirem resultados de software com mais eficácia do que suas contrapartes de comando e controle.

Isso não significa que o estilo tradicional orientado a planos não funciona mais em circunstância nenhuma; existe uma dose de rigor na abordagem tradicional da qual até mesmo a metodologia Ágil e o Scrum podem se beneficiar. O que estamos simplesmente dizendo é que podemos fazer melhor se soubermos como retroceder alguns passos para refletirmos sobre como fazemos e aprendemos, para melhorarmos nossas técnicas e processos com novos conceitos e ideias.

Como as pessoas e as empresas tendem a abraçar as mudanças lentamente, só o fato de o movimento Ágil e, especialmente, o Scrum terem decolado e gerado tanta excitação é um bom sinal. Com este livro, juntos esperamos contribuir para difundir a adoção da metodologia Ágil e do Scrum, que deriva de vários anos de experiência em gerenciamento e desenvolvimento de projetos de software na vida real.

Como muitos de nós, se você conseguir superar os obstáculos, descobrirá que o método Ágil e o Scrum são incrivelmente revigorantes e também aprenderá a implementá-los com sucesso em um projeto. Se isso não acontecer, sua trajetória será repleta de dificuldades, que, em última instância, podem levar seu projeto ao fracasso.

Existem muitas razões por que as equipes às vezes falham com o Scrum. Uma das razões mais comuns é que muitos profissionais ainda estão desacostumados a essa metodologia, mesmo após participar de uma aula de Scrum ou ler alguns artigos sobre ele. A segunda razão pode ser o fato de seu projeto ser tão complexo a ponto de precisar de técnicas mais avançadas que reduzam a complexidade dele e consigam controlá-lo. Uma terceira razão poderia ser que as empresas ainda não estão preparadas para o Scrum ou suas equipes não sabem como o usar segundo as restrições atuais da empresa. Isso poderia ocorrer seja pela falta de experiência, seja porque a equipe foi mal aconselhada por ScrumMasters ou instrutores dogmáticos, algo que vamos tratar neste guia prático.

Independentemente da razão, o método Ágil e o Scrum trazem mudanças, e estas são sempre difíceis, a menos que sejam gerenciadas apropriadamente. Mas com um bom gerenciamento, tais dificuldades podem ser transformadas em oportunidades.

Um de nossos objetivos neste livro é preparar você e sua equipe para as mudanças que passarão com o método Ágil e o Scrum. Queremos ajudá-los a tornar o processo o mais suave possível, enquanto damos uma chance à sua equipe de se adaptar a seu projeto Scrum.

Quais são os fundamentos do gerenciamento de projeto e desenvolvimento Ágil de software?

Os fundamentos do gerenciamento de projeto e desenvolvimento Ágil de software são, sem dúvida, o Manifesto Ágil e a Declaração de Interdependência.

Em 2001, um grupo de especialistas em software se reuniu no resort Snowbird, em Utah, para rascunhar o que ficou conhecido como o Manifesto Ágil (www.agilemanifesto.org):

“Estamos descobrindo melhores maneiras de desenvolver software, fazendo-o e ajudando outros a fazê-lo. Ao longo desse trabalho, começamos a valorizar:

- indivíduos e interações em vez de processos e ferramentas;
- software funcional em vez de documentação extensiva;
- colaboração com o cliente em vez de negociação de contrato;
- respostas à mudança em vez de seguimento de um plano.

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

[© 2001 Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas.]

Além desses quatro valores, o Manifesto Ágil tem doze princípios:

1. Nossa prioridade máxima é satisfazer o cliente por meio da entrega antecipada e contínua de software com valor.
2. Abraçar as mudanças de requisitos, mesmo quando o desenvolvimento já está avançado. Os processos Ágeis usam as mudanças para trazer vantagens competitivas ao cliente.
3. Entregar software funcional frequentemente, indo de algumas semanas a alguns meses, com preferência para a escala de tempo mais curta.
4. Pessoas ligadas ao negócio e ao desenvolvimento devem trabalhar juntas diariamente, ao longo do projeto.
5. Construir projetos em torno de indivíduos motivados. Dê a eles o ambiente e suporte necessários, e confie que eles realizarão o trabalho.
6. O método mais eficiente e efetivo de transmitir informações para e dentro de uma equipe de desenvolvimento é conversando cara a cara.
7. Software funcional é a primeira medida de progresso.

8. Processos Ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. Atenção contínua à excelência técnica e um bom projeto melhoram a agilidade.
10. Simplicidade – a arte de maximizar a quantidade de trabalho que não precisa ser realizado – é essencial.
11. As melhores arquiteturas, os melhores requisitos e os melhores projetos emergem de equipes auto-organizáveis.
12. A intervalos regulares, a equipe reflete sobre como tornar-se mais efetiva, então se alinha e ajusta seu comportamento de acordo com essa resolução.

Apesar de o Manifesto Ágil ter sido rascunhado em 2001, alguns anos depois de o Scrum ter sido anunciado na Object Oriented Programming Systems, Languages and Applications (OOPSLA) em 1996, é um fato bem conhecido entre os especialistas que ele teve grande influência sobre o Scrum. Essa influência ficou óbvia no segundo livro de Ken Schwaber, *Agile Project Management with Scrum*, no qual ele escreveu que o Scrum é um dos processos Ágeis com valores e princípios semelhantes aos descritos no Manifesto Ágil.

Enquanto o Manifesto Ágil lida com desenvolvimento de software, a “Declaração de Interdependência” (Declaration of Interdependence, ou DOI) da Gestão de Projeto Ágil, que outro grupo de especialistas elaborou em 2005, focaliza mais o gerenciamento de projetos (<http://pmdoi.org>):

“Somos uma comunidade de líderes de projeto que tem sido altamente bem-sucedida em entregar resultados. Para alcançar tais resultados:

- **aumentamos o retorno do investimento**, tornando o fluxo contínuo de valor o nosso foco;
- **entregamos resultados confiáveis**, engajando os clientes em interações frequentes e propriedade compartilhada;
- **esperamos incertezas** e gerenciamos levando-as em conta, por meio de iterações, antecipação e adaptação;
- **promovemos criatividade e inovação** reconhecendo que os indivíduos são a fonte última de valor e criamos um ambiente em que eles fazem a diferença;
- **impulsionamos o desempenho** por meio do compromisso do grupo em obter resultados e da responsabilidade compartilhada pela eficácia do grupo;

- **melhoramos a eficácia e a confiabilidade** por meio de estratégias situacionais específicas, processos e práticas.”

[© 2005 David Anderson, Sanjiv Augustine, Chistopher Avery, Alistair Cockburn, Mike Cohn, Doug DeCarlo, Donna Fitzgerald, Jim Highsmith, Ole Jepsen, Lowell Lindstrom, Todd Little, Kent McDonald, Pollyanna Pixton, Preston Smith e Robert Wysocki.]

Saber se o Manifesto Ágil ou a DOI ocorreu aos especialistas antes ou depois de terem sido influenciados, de alguma maneira, pelo Scrum ou por quaisquer outros processos Ágeis existentes na época, é uma questão irrelevante.

O importante é que se você realmente entender o significado do manifesto e da DOI, terá vantagem ao adaptar o Scrum, caso seja necessário, sem trair suas fundações Ágeis.

Origem do Scrum

Historicamente, o termo Scrum surgiu em um artigo publicado por Hirotaka Takeuchi e Ikujiro Nonaka na *Harvard Business Review* de 1986. Nesse artigo, intitulado “The new new product development game” (“O novo novo jogo do desenvolvimento de produtos”), Takeuchi e Nonaka descreveram uma abordagem holística, na qual equipes de projeto são compostas de pequenas equipes multifuncionais, trabalhando com sucesso rumo a um objetivo comum, que os autores compararam à formação Scrum do rugby.

Enquanto trabalhava na construção de uma ferramenta de Análise e Projeto Orientados a Objetos (Object-Oriented Analysis and Design, ou OOAD), Jeff Sutherland, então vice-presidente de Engenharia da Easel, Inc., percebeu que sua equipe de software precisaria de uma versão aprimorada do Desenvolvimento Rápido de Aplicações (Rapid Application Development, ou RAD). O que ele queria era um processo semelhante ao Scrum, em que, ao final de iterações curtas, o CEO da Easel pudesse ver a demonstração de código funcional em vez de gráficos de Gantt em papel.

Durante mais ou menos esse mesmo período, Ken Schwaber (veja as leituras recomendadas na bibliografia) estava pesquisando ativamente sobre como poderia ajudar sua empresa, Advanced Development Methods (ADM), a melhorar seu processo de software com o objetivo de aumentar a produtividade das equipes.

Após uma profunda análise de como outras vendedoras de software independentes (Independent Software Vendors, ou ISV) bem-sucedidas construam softwares, Ken percebeu que o processo de desenvolvimento de todas elas era semelhante, no sentido de que todas usavam processos empíricos que exigiam inspeção e adaptação constantes.

A pedido da Object Management Group (OMG) em 1995, Jeff e Ken trabalharam em conjunto para resumir o que haviam aprendido ao longo do anos; eles criaram uma

nova metodologia, que foi chamada de Scrum, descrita em um artigo de Schwaber, “Scrum and the perfect storm” (“Scrum e a tempestade perfeita”), disponível em www.controlchaos.com/my-articles.

Como o Scrum funciona

Perceba na figura 1.1 que a equipe de Scrum, que deveria ser uma equipe multifuncional, é composta de um ScrumMaster, de um Product Owner (Dono do Produto) e da equipe de desenvolvimento (ou, simplesmente, “equipe”), com todas as habilidades necessárias (como coleta de requisitos, projeto, codificação e teste) para construir o produto de software.

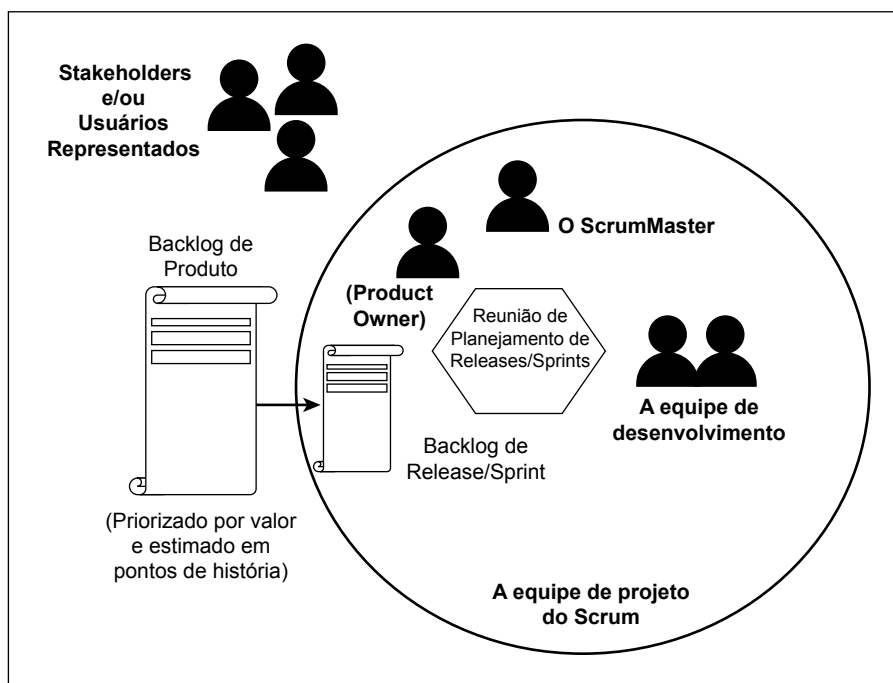


Figura 1.1 – Backlog de Produto, Backlog de Release e Backlog de Sprint.

Mesmo nas melhores situações, quando a composição da equipe de Scrum é completamente multifuncional, isso não significa que todos os membros do projeto Scrum, em termos organizacional e hierárquico, façam parte da estrutura permanente da equipe de Scrum. A menos que sua empresa seja uma firme proponente da implantação do Scrum em âmbito organizacional e já tenha se reorganizado completamente, seguindo as recomendações de estrutura organizacional do Scrum, as equipes de Scrum ainda são emprestadas dos diferentes departamentos aos quais pertencem, como o Setor de Qualidade, Arquitetura de Empresa, Pré-produção ou DBA.

Mesmo algumas das empresas, comumente citadas como tendo adotado o Scrum, ainda possuem o Gerenciamento de Projeto, Setor de Qualidade, Pré-produção, Grupos de Análise de Sistemas de Negócios e a Arquitetura de Empresa trabalhando como funções separadas no setor de Tecnologia de Informática (TI), e as equipes de projeto têm de negociar com esses grupos para manter seus projetos Scrum andando. Muito raramente uma empresa demite todos os gerentes de projeto (o Scrum não tem, propriamente dito, um gerente de projeto), ou altera o título deles para ScrumMaster, ou reorganiza todos os seus grupos de TI separados em torno de equipes de projeto de Scrum multidisciplinares. Isso seria um sonho, mas ainda não é a realidade atual.

Na figura 1.1, você pode verificar como um projeto Scrum é iniciado. Como pode ver no diagrama, tudo começa com o Product Owner, que é responsável por obter informações dos stakeholders, ou usuários que os representem, para elaborar uma lista de requisitos e criar um Backlog¹ de Produto.

O Backlog de Produto é uma lista de requisitos priorizada, que pode incluir de tudo: de aspectos do negócio a tecnologias, questões técnicas e correções de bugs.

Alguns praticantes e autores, como Henrik Kniberg, em seu livro, *Scrum and XP from the Trenches*, preferem manter o Backlog de Produto no âmbito empresarial para incluir apenas requisitos de negócio, semelhantemente ao que fazemos.

Como você vai aprender no capítulo 4, “Coleta de requisitos visuais para o Backlog de Produto”, os requisitos de usuário para o Backlog de Produto do Scrum costumam ser coletados como histórias de usuário curtas, durante um workshop de um ou dois dias, anteriormente à reunião de Planejamento de Releases (Planejamento de Lançamentos) e à reunião de Planejamento de Sprints.

Enquanto o Planejamento de Releases era, de certa forma, opcional nos primórdios do Scrum, ele provou ter ajudado muitas equipes de Scrum a serem ainda mais eficazes ao longo dos anos. Portanto, recomendamos enfaticamente que o Product Owner passe pelo Planejamento de Releases com sua equipe, mesmo que isso seja um exercício difícil, já que exige que você aprenda a respeito do produto antes da reunião de planejamento.

Quanto mais você, como Product Owner, conhecer o produto, mais poderá ajudar a equipe. O objetivo-chave do Planejamento de Releases é permitir que a equipe de Scrum identifique todos os lançamentos que o produto de software deve ter, com uma agenda de entregas prováveis. Normalmente, o Planejamento de Releases deve durar quatro horas no caso de uma equipe de Scrum com Sprints de quatro semanas.

1 N. de T.: Backlog refere-se a um log (resumo histórico) de acumulação de trabalho num determinado período de tempo (Fonte: Wikipédia).

Além do Planejamento de Releases, a equipe de Scrum também deve passar por alguns Planejamentos de Sprints, seja como parte do processo de Planejamento de Releases, seja independentemente, após a realização do Planejamento de Releases.

Normalmente, a reunião de Planejamento de Sprints deve durar em torno de oito horas, no caso de Sprints de quatro semanas, e deve ser ajustada para quatro horas, no caso de Sprints de duas semanas.

Como prática comum, a reunião de Planejamento de Sprints deve ser dividida em duas reuniões de quatro horas cada uma.

Durante a primeira parte da reunião, o Product Owner percorrerá os requisitos, na forma de histórias de usuário, para decidir, com o feedback da equipe, quais deveriam fazer parte de quais Sprints e quais são seus objetivos. A primeira reunião serve, principalmente, para responder à questão “O quê”.

Durante a segunda parte da reunião de Planejamento de Sprints, que focaliza o “Como”, a equipe de desenvolvimento tentará identificar tarefas (tasks) a partir das histórias previamente escolhidas e deduzir quanto tempo (em horas) a equipe levará para transformar essas tarefas em incrementos de produto potencialmente entregáveis. A menos que a equipe utilize algum tipo de software de planejamento, todas as tarefas de desenvolvimento que fazem parte do Sprint normalmente serão registradas em um Quadro de Tarefas (Task Board), algum tipo de quadro-branco em uma parede, para uma fácil alocação e rastreamento por parte da equipe.

Tão logo as reuniões de Planejamento de Releases e Planejamento de Sprints tenham terminado, inicia-se o trabalho propriamente dito dos Sprints (Figura 1.2), com os 15 minutos do Daily Scrum (Scrum Diário) ou Daily Standup (algo como Reunião Diária em Pé).

Originalmente, a Reunião Diária poderia durar até cerca de 30 minutos, mas, como parte da evolução ou ajuste do Scrum, sobre a qual falaremos mais no capítulo 12, “Como adaptar o Scrum (sem destruir suas fundações Ágeis ou fazer Scrum-buts negativos)”, sua duração foi cada vez mais reduzida na prática, até chegar aos 15 minutos atuais.

Normalmente, a duração de um Sprint varia de uma a quatro semanas. Exceto em circunstâncias muito especiais, não se pode adicionar ou excluir itens do Backlog de Sprint enquanto o Sprint está ocorrendo, a menos que a equipe e o Product Owner concordem com esse procedimento, mas isso é mais uma exceção do que a regra.

Diferentemente dos processos tradicionais, em que o gerente do projeto é responsável por organizar reuniões de status semanais para acompanhar o estado do projeto, no Scrum a equipe se reunirá todo dia, não para inspecionar o estado do projeto, e sim

para inspecionar o progresso da equipe rumo ao objetivo do Sprint. Essa é a razão de você ouvir com certa frequência as pessoas dizerem que o Daily Scrum não é uma reunião de status.

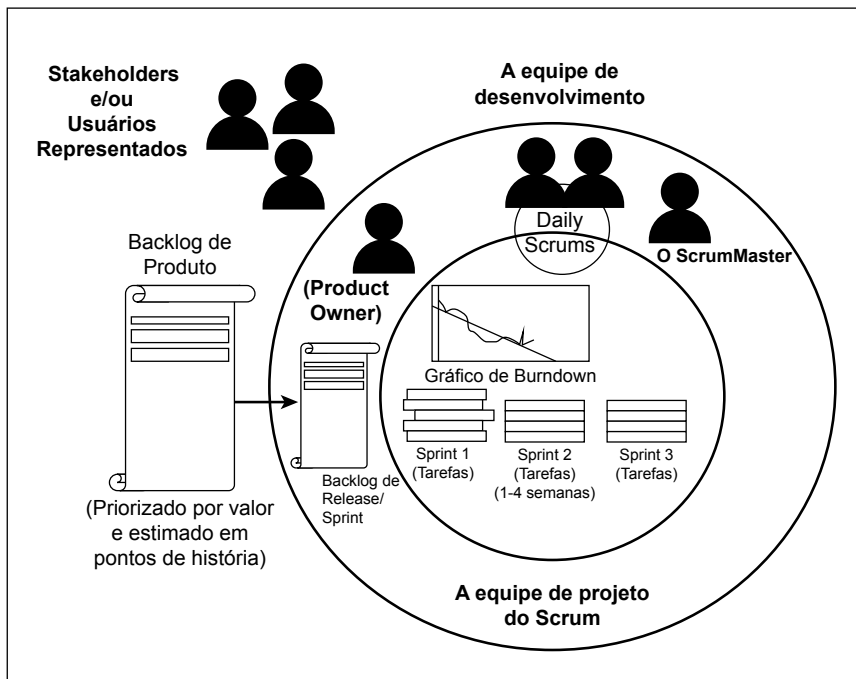


Figura 1.2 – Sprints, um Gráfico de Burndown e Daily Scrums.

Para manter um registro do progresso da equipe rumo ao objetivo do Sprint, um Gráfico de Burndown (ou Gráfico de Consumo) será criado para mostrar quanto trabalho ainda resta até que a equipe tenha terminado o Sprint. Mesmo que a criação desse Gráfico de Burndown seja responsabilidade da equipe, pode ser atualizado pelo ScrumMaster sempre que a equipe não tiver tempo para isso.

Antes do final de cada Sprint, a equipe se reunirá com o Product Owner, como parte do Mecanismo de Inspeção e Adaptação do Scrum, para percorrer o que é conhecido como uma Revisão de Sprint organizada pelo ScrumMaster (Figura 1.3). Essa é outra reunião cronometrada, que geralmente dura quatro horas, no caso de um Sprint de quatro semanas, ou duas horas, no caso de um Sprint de duas semanas.

O objetivo dessa reunião é multifacetado: em primeiro lugar, serve para que a equipe de Scrum e o Product Owner discutam o que foi e o que não foi feito. Em segundo lugar, a reunião serve para demonstrar para o Product Owner aquilo que foi feito e obter seu feedback. Por fim, o terceiro objetivo é obter atualizações do Product Owner relativas a quaisquer mudanças na direção do mercado ou do produto.

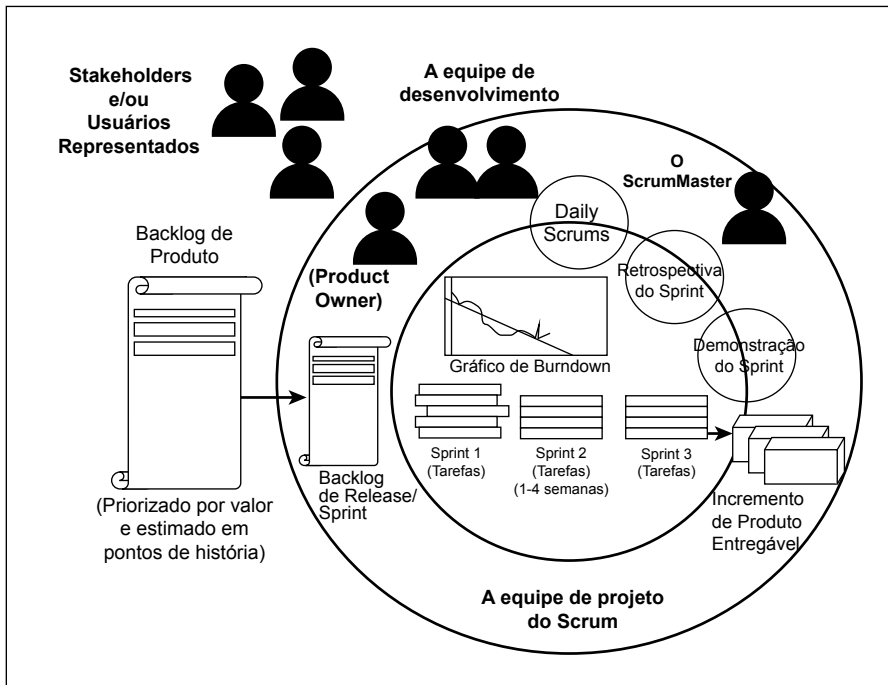


Figura 1.3 – Revisão do Sprint e Retrospectiva.

Logo após a Revisão do Sprint, mas antes do Sprint seguinte, a equipe de Scrum também se reunirá para percorrer uma Retrospectiva do Sprint, com o objetivo de identificar o que funcionou e o que não funcionou durante o Sprint atual. A intenção é ver como os participantes poderiam tornar sua colaboração ainda mais eficaz ao seguir para o próximo Sprint.

Como prática comum, a Reunião de Retrospectiva deveria durar três horas para um Sprint de um mês, mas sua duração deve ser ajustada, como é o caso de todas as outras reuniões cronometradas, em proporção à duração do Sprint, como duas horas, no caso de um Sprint de duas semanas.

A figura 1.4 fornece um gráfico geral das responsabilidades colaborativas dos membros da equipe de Scrum.

O Scrum não soa muito complicado como uma estrutura de projeto geral, certo? Sim, mas apesar de o Scrum soar simples na teoria, pode ser muito difícil de implementar, especialmente se sua empresa ainda for nova nas práticas do Scrum ou se você deseja turbinar seu projeto Scrum com algumas das práticas atuais.

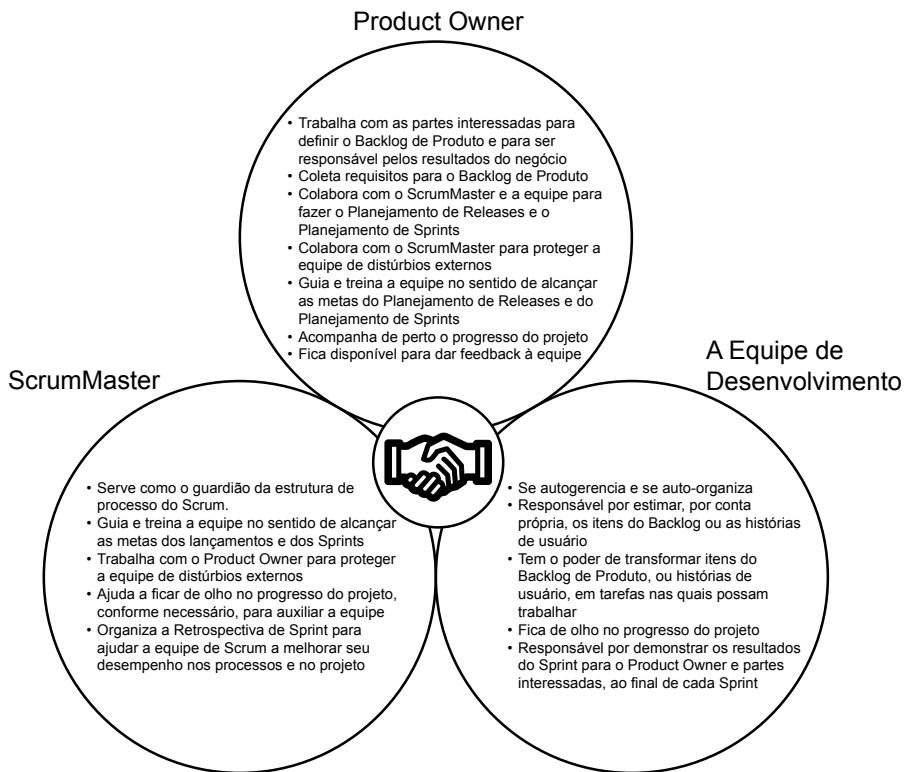


Figura 14 – Tudo se resume à colaboração entre a equipe, o ScrumMaster e o Product Owner.

Por que a metodologia Ágil e o Scrum são eficazes na gestão de projetos de software?

Mesmo que a metodologia Ágil e o Scrum em particular sejam difíceis de implementar, têm provado serem extremamente eficazes quando implantados de maneira apropriada.

À medida que for lendo este livro, você verá o por que de a metodologia Ágil e o Scrum normalmente serem mais eficazes no gerenciamento e desenvolvimento de projetos. No entanto, existem quatro vantagens que gostaríamos de enfatizar *a priori*:

- Um mecanismo de redução sistemática de risco: todos nós, que somos responsáveis pelo planejamento e execução de projetos, sabemos a importância de reduzir o nível de risco ou incerteza a zero, ou para o menor nível possível.

Apesar de existirem quatro maneiras distintas de lidar com riscos (evitar, transferir, aceitar e mitigar), no final das contas, gerentes de projeto geralmente acabam tendo que mitigá-los. É aqui que o Scrum brilha, com seu ciclo frequente de Inspeção e Adaptação.

- Um ciclo de vida de desenvolvimento de software mais enxuto:

Nas figuras 1.5 e 1.6, vemos a grande diferença na linha do tempo, com uma das equipes usando um ciclo de vida mais longo (Figura 1.5), enquanto a outra equipe está utilizando um ciclo de vida mais enxuto (Figura 1.6), fazendo um uso mais eficiente do tempo produtivo.

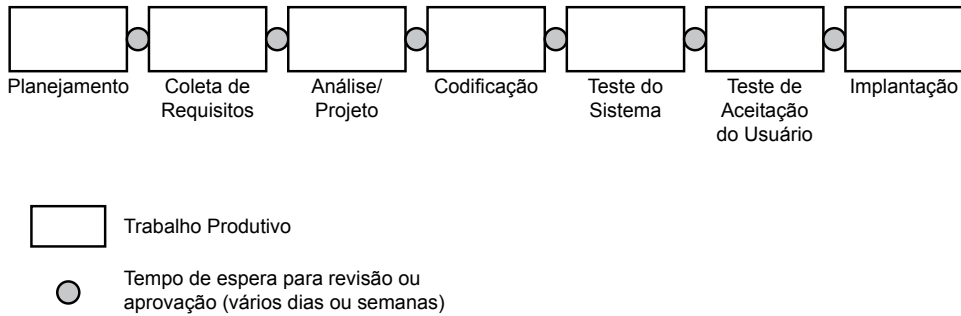


Figura 1.5 – Fluxo de valor tradicional.

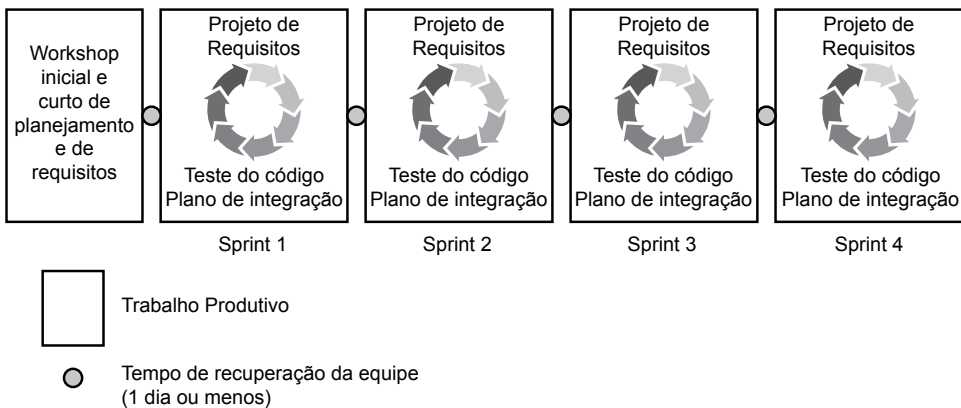


Figura 1.6 – Fluxo de valor do Scrum.

- Um processo de gestão de projeto mais adaptativo. Diferentemente do processo sequencial usado dentro do ambiente em cascata, que considera a estabilidade do projeto como sua fundação (Figura 1.7), o Scrum se parecerá mais com a figura 1.8, em que a mudança é considerada a única constante.
- Uma estrutura de processo de gestão e desenvolvimento de projeto baseado na motivação e no orgulho das pessoas. Mais do que tudo, talvez este seja um dos princípios mais importantes da metodologia Ágil e do Scrum. O novo foco não recai no gerente tendo que ditar aos membros da equipe o que deveriam estar fazendo, e sim em deixar a equipe decidir por conta própria como vai caminhar para realizar seu trabalho.

O Scrum propõe uma nova estrutura de gestão de software, que está baseada em auto-organização, motivação, propriedade e orgulho de uma equipe em realizar suas aquisições.

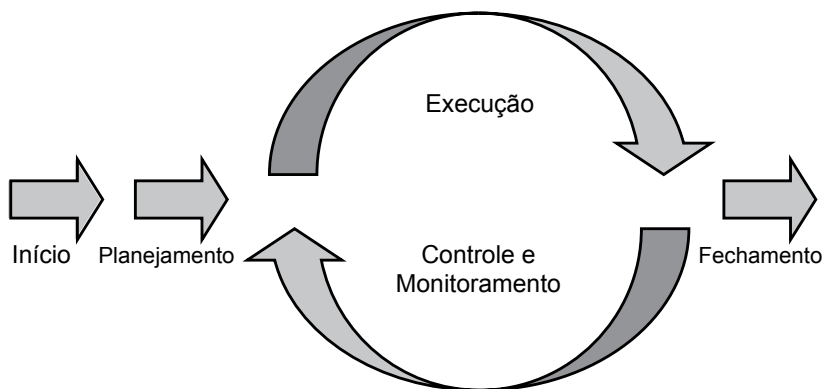


Figura 1.7 – O processo de gestão de projeto tradicional.

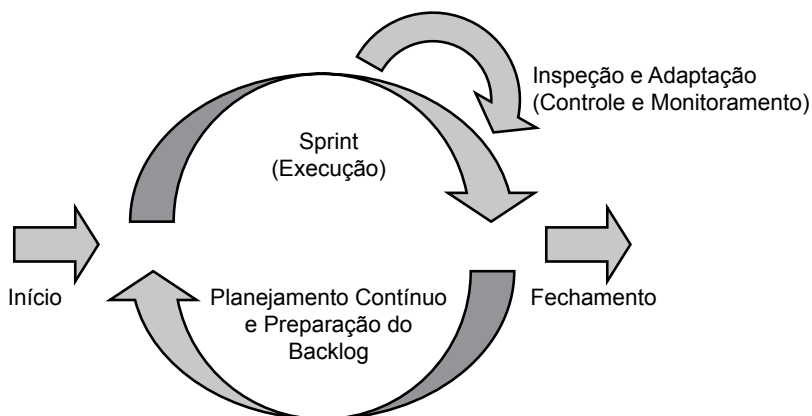


Figura 1.8 – Estrutura de gerenciamento de projeto adaptativo usando o Scrum.

Apesar de curta, essa introdução à metodologia Ágil, e especialmente ao Scrum, fornecerá a você a compreensão necessária para ser capaz de seguir em frente com o restante deste livro.

Mas antes disso, também devemos dizer que já que nosso foco será o Scrum, vamos nos referir apenas a ele no restante do livro.

Resumo

A metodologia Ágil nasceu em 2001, quando um grupo de especialistas se reuniu em Utah para rascunhar aquilo que agora é conhecido como Manifesto Ágil, focalizando no lado do desenvolvimento de software.

Como complemento ao Manifesto Ágil, outro grupo de especialistas se reuniu em 2005 para rascunhar um outro documento chamado de Declaração de Interdependência (DOI) da Gestão de Projetos, que deu ênfase à gestão de projetos.

Juntos, o Manifesto Ágil e a DOI formam a fundação do movimento Ágil, e de todo o desenvolvimento Ágil e de seus processos de gestão de projeto, incluindo o Scrum.

Já que o Scrum não pode ser utilizado em vários contextos sem algum tipo de adaptação, um bom conhecimento das fundações do movimento Ágil e de seus processos lhe permitirá saber como adaptar o Scrum para seu ambiente.

Apesar de que você verá todas as vantagens do Scrum ao longo do livro, quatro delas merecem ser apontadas de imediato: um mecanismo de redução de riscos, um processo de software mais enxuto, um processo de gestão de projetos de software mais adaptativo e uma estrutura baseada na auto-organização, motivação, propriedade e orgulho de uma equipe.

Como o foco deste livro é o Scrum, vamos nos referir apenas a ele ao longo dos capítulos.