SCRUM

A ARTE DE FAZER O DOBRO DO TRABALHO NA METADE DO TEMPO

JEFF SUTHERLAND
COCRIADOR DO SCRUM

LeYa

DADOS DE COPYRIGHT

Sobre a obra:

A presente obra é disponibilizada pela equipe ***Le Livros*** e seus diversos parceiros, com o objetivo de oferecer conteúdo para uso parcial em pesquisas e estudos acadêmicos, bem como o simples teste da qualidade da obra, com o fim exclusivo de compra futura.

É expressamente proibida e totalmente repudiável a venda, aluguel, ou quaisquer uso comercial do presente conteúdo

Sobre nós:

O ***Le Livros*** e seus parceiros disponibilizam conteúdo de dominio publico e propriedade intelectual de forma totalmente gratuita, por acreditar que o conhecimento e a educação devem ser acessíveis e livres a toda e qualquer pessoa. Você pode encontrar mais obras em nosso site: ***LeLivros.site*** ou em qualquer um dos sites parceiros apresentados neste link.

"Quando o mundo estiver unido na busca do conhecimento, e não mais lutando por dinheiro e poder, então nossa sociedade poderá enfim evoluir a um novo nível."

Ficha Técnica

Copyright © 2014 by Jeff Sutherland and Scrum, Inc.
Todos os direitos reservados.
Tradução para a língua portuguesa © 2014 Texto Editores Ltda.
Título original: Scrum: The art of doing twice the work in half the time

Preparação de texto: Meggie Monauar
Revisão: Paula Jacobini
Diagramação: Cristiane Viana
Capa: Ideias com Peso

Dados Internacionais de Catalogação na Publicação (CIP)
Angélica Ilacqua CRB-8/
7057
Sutherland, Jeff
Scrum : a arte de fazer o dobro do trabalho na metade
do tempo / Jeff Sutherland; tradução de Natalie Gerhardt. -
São Paulo : LeYa, 2014.
Bibliografia
ISBN 9788544100882
Título original: Scrum : The art of doing twice the work
in half the time
1. Scrum (Desenvolvimento de software) 2. Negócios 3.
Administração I. Título II. Gerhardt, Natalie
14-0701 CDD 658
Índices para catálogo sistemático:
1. Administração - negócios

2014
TEXTO EDITORES LTDA.
Uma editora do Grupo LeYa
Rua Desembargador Paulo Passaláqua, 86
01248-010 — Pacaembu — São Paulo, SP
www.leya.com.br

Jeff Sutherland

SCRUM

A arte de fazer o dobro do trabalho na metade do tempo

Prefácio

Por que Scrum?

Eu criei o Scrum, junto com Ken Schwaber, há vinte anos, para ser uma forma mais rápida, eficaz e confiável de criar softwares para o setor de tecnologia. Até aquele ponto — e até 2005 —, a maior parte do desenvolvimento de software era feita usando o método em cascata, no qual um projeto era concluído em todos os estágios distintos e seguia, passo a passo, em direção ao lançamento para os consumidores, ou usuários. O processo era lento, imprevisível e, em geral, nunca resultava em um produto que as pessoas queriam ou estavam dispostas a pagar para obter. Atrasos de meses ou até mesmo de anos eram endêmicos ao processo. Os planos iniciais de passo a passo, expostos em detalhes reconfortantes em diagramas de Gantt, asseguravam aos gestores que tínhamos total controle do processo de desenvolvimento — no entanto, quase sempre, nós rapidamente ficávamos atrasados em relação ao cronograma, e desastrosamente acima do orçamento.

Para superar essas falhas, em 1993, inventei uma nova forma de fazer as coisas: o Scrum. Trata-se de uma mudança radical das metodologias prescritivas e de cima para baixo usadas na gerência de projetos no passado; já o Scrum é semelhante aos sistemas autocorretivos, evolucionários e adaptativos. Desde o começo, a estrutura do Scrum se tornou *a* forma de o setor tecnológico criar novas aplicações de software e produtos. Contudo, embora ele tenha se tornado muito bem-sucedido no gerenciamento de projetos de software e hardware no Vale do Silício, ainda permanece pouco conhecido em outros setores de negócios. E foi por isso que escrevi este livro: para revelar e explicar o sistema de gerenciamento do Scrum para setores de negócios fora do mundo da tecnologia. Aqui eu falo sobre a sua origem no Sistema Toyota de Produção e no ciclo OODA da aviação de combate. Discuto como organizamos projetos em torno de equipes pequenas — e por que essa é a forma mais eficaz de se trabalhar. Explico como priorizamos as diversas tarefas nos projetos; como definimos Sprints de uma semana a um mês para criarmos uma força e tornarmos todos na equipe responsáveis; como fazemos reuniões breves diárias

para manter o controle de tudo o que foi feito e dos desafios que inevitavelmente aparecem; e como o Scrum incorporou os conceitos de aprimoramento contínuo e produtos minimamente viáveis para obter feedback imediato dos consumidores, em vez de esperar até que o projeto tenha sido concluído. Como você verá, usamos o Scrum para construir qualquer coisa, desde carros viáveis que fazem 42 quilômetros por litro de combustível até trazer os sistemas de banco de dados do FBI para o século 21.

Leia este livro. Acho que você verá como o Scrum pode ajudar a transformar o modo como a sua empresa trabalha, cria, planeja e pensa. Eu realmente acredito que esse sistema pode ajudá-lo a revolucionar a forma como as empresas funcionam em praticamente todos os setores, assim como ele revolucionou a inovação e a velocidade para o mercado em uma gama estarrecedora de novas empresas e uma variedade enorme de novos produtos surgindo fora do Vale do Silício e do mundo da tecnologia.

— Jeff Sutherland, PhD

CAPÍTULO 1

A maneira como o mundo funciona está quebrada

Jeff Johnson tinha quase certeza de que aquele não seria um bom dia. Em 3 de março de 2010, o Federal Bureau of Investigation (FBI) cancelou o seu projeto mais ambicioso de modernização — aquele que poderia ter evitado o 11 de setembro, mas que se transformou em um dos maiores fiascos da indústria de software de todos os tempos. Por mais de uma década, o FBI tentou atualizar seu sistema de computação, e tudo indicava que iria falhar. *De novo*. E agora o projeto era dele.

Ele havia chegado ao FBI sete meses antes, atraído pelo novo Diretor-Executivo de Informação (*Chief Information Officer*, CIO), Chad Fulgham, com quem trabalhara na Lehman Brothers. Jeff era o diretor-assistente da nova Divisão de Engenharia de Tecnologia da Informação (TI), e tinha um escritório no último andar do edifício J. Edgar Hoover, no centro de Washington, D.C. Era uma sala grande com vista para o Monumento de Washington. Mal sabia Jeff que, pelos dois anos seguintes, acabaria em um escritório sem janelas e do tamanho de uma caixa de fósforos no porão do prédio, tentando consertar algo que todos diziam ser impossível.

"Não foi uma decisão fácil", conta Jeff. Seu chefe e ele decidiram declarar derrota e cancelar um programa no qual haviam trabalhado por quase dez anos e custara centenas de milhões de dólares. Àquela altura, fazia mais sentido trazer o projeto para "dentro de casa" e tentar fazê-lo por conta própria. "Mas aquilo precisava realmente ser feito, e muito bem feito".

Tratava-se do aguardado projeto para um novo sistema de computação que efetivamente trouxesse o FBI para a era moderna. Em 2010 — a era do Facebook, Twitter, Amazon e Google —, a agência ainda preenchia a maioria de seus relatórios no papel, e utilizava um sistema chamado Automated Case Support. Ele rodava em computadores gigantescos que, em algum período da

década de 1980, eram o que existia de mais moderno. Mas, naquele momento, muitos agentes especiais nem o usavam mais; era ainda inconveniente e muito lento para uma época de ataques terroristas e criminosos espertos.

Quando um agente do FBI queria fazer algo — *qualquer coisa*, na verdade —, desde pagar a um informante para perseguir um terrorista até fazer um relatório sobre um ladrão de bancos, o processo não era muito diferente do que era feito trinta anos antes. Johnson o descreve da seguinte maneira:

> Era necessário escrever o relatório usando um processador de texto e depois imprimi-lo em três vias. Uma seria enviada para aprovação, outra arquivada localmente para o caso de a primeira se perder; e, na terceira via, você teria de pegar uma caneta vermelha — não, eu não estou brincando, uma caneta vermelha mesmo — e circular as palavras-chave que deveriam ser inseridas no banco de dados. Você tinha de indexar o próprio relatório.

Quando um pedido era aprovado, a via em papel recebia determinado número — sim, um número escrito em um pedaço de papel é o modo como o FBI mantém todos os seus arquivos de casos. Esse método era tão antiquado e furado que recebeu parte da culpa quando a agência não conseguiu "unir os pontos" que mostravam vários ativistas da Al-Qaeda entrando no país pouco tempo antes dos atentados terroristas de 11 de setembro. Um escritório tinha um suspeito; outro se perguntava por que tantos estrangeiros suspeitos estavam assistindo a aulas de pilotagem de avião. E outro ainda tinha um suspeito em sua lista de vigilância, mas não repassou a informação. Assim, ninguém no FBI foi capaz de reunir todas aquelas peças.

Depois dos ataques, a Comissão do 11 de Setembro investigou a fundo para tentar descobrir o principal motivo de aquilo ter acontecido, e chegou à conclusão de que os analistas não conseguiram ter acesso às informações necessárias. Diz o relatório: "A ineficiência dos sistemas de informação do FBI significava que tal acesso dependia em grande parte das relações interpessoais do analista com pessoas em outras unidades ou equipes que detinham tais informações".

Antes da tragédia, o FBI nunca havia concluído uma avaliação da ameaça global do terrorismo dentro dos Estados Unidos. Houve uma série de razões para isso, desde foco no avanço da carreira dos funcionários até uma total falta de compartilhamento de informações. No entanto, o relatório apontou a ausência de sofisticação tecnológica como talvez o principal motivo por que o FBI falhara de forma tão drástica nos dias que antecederam os ataques. "Os

sistemas de informação do FBI eram completamente inadequados", concluiu o documento. "O FBI não tinha capacidade para saber o que sabia: não havia qualquer mecanismo adequado para acessar ou compartilhar o conhecimento institucional".

Quando os senadores começaram a fazer perguntas desconfortáveis para a agência, o FBI praticamente disse "não se preocupem, temos um plano de modernização já em andamento". O sistema planejado se chamava Virtual Case File (VCF) e deveria mudar tudo. Sem deixar que a crise passasse em branco, os funcionários relataram que precisavam apenas de mais US$ 70 milhões, além dos outros US$ 100 milhões já orçados, para concluírem o trabalho. Se você ler os relatórios sobre o VCF da época, perceberá que as palavras *revolucionário* e *transformação* são usadas de forma generosa.

Três anos depois, o programa foi cancelado. Não funcionava. Nem um pouquinho. O FBI tinha gastado US$ 170 milhões dos contribuintes para comprar um sistema de computador que nunca seria usado — nem uma linha de código ou aplicação ou clique do mouse. Tudo aquilo era o mais absoluto desastre. E não era comparável a um erro da IBM ou da Microsoft. A *vida* das pessoas estava, literalmente, em risco. O Senador Patrick Leahy, de Vermont, democrata e então Presidente do Comitê Judiciário do Senado, declarou, na época, ao *Washington Post*:

> Nós tínhamos informações que poderiam ter impedido os ataques terroristas de 11 de setembro. Estavam bem ali, diante de nós, e ninguém fez nada... Eu não estou vendo os problemas serem corrigidos... Talvez cheguemos ao século XXII antes que consigamos ter a tecnologia do século XXI.[1]

É bastante esclarecedor dizer que muitas das pessoas que trabalhavam no FBI quando o desastre do Virtual Case File aconteceu não estão mais trabalhando lá.

Em 2005, a agência anunciou um novo programa, o Sentinel. Daquela vez, ia funcionar. Daquela vez, tomariam todas as precauções necessárias, fariam os procedimentos orçamentários corretos e usariam as ferramentas certas de controle. Já tinham aprendido a lição. O preço? Meros US$ 451 milhões. E estaria em pleno funcionamento em 2009.

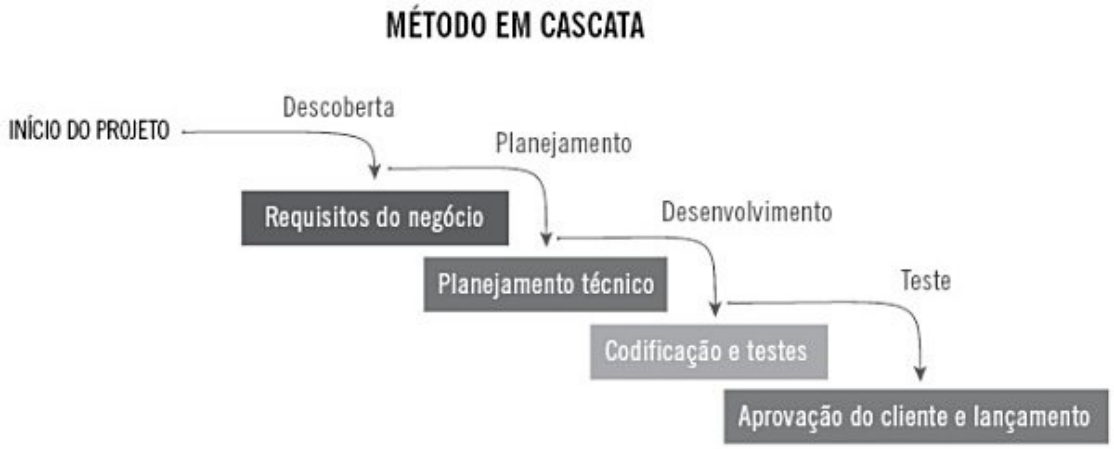
O que poderia dar errado? Em março de 2010, a resposta caiu na mesa de Jeff Johnson. A Lockheed Martin, empresa contratada para desenvolver o sistema Sentinel, já tinha gastado US$ 405 milhões do orçamento. Tinham desenvolvido apenas metade do projeto e já estavam um ano atrasados. Uma

análise independente estimou que levariam outros seis a oito anos para concluir o projeto, e que ainda teriam de investir mais US$ 350 milhões do dinheiro dos contribuintes.

Encontrar uma solução para aquilo era problema de Johnson.

O que tinha dado errado e como a situação foi resolvida são o motivo por que estou escrevendo este livro. Não era uma questão de inteligência. Não era que a agência não tivesse as pessoas certas nos lugares certos e também não era uma questão de tecnologia errada. Também não tinha nada a ver com ética no trabalho ou com o estímulo adequado de competitividade. Era por causa da *maneira* como as pessoas estavam trabalhando. A *maneira* como a maioria das pessoas trabalha. A maneira como nós achamos que o trabalho *precisa* ser feito, porque foi assim que aprendemos a fazê-lo.

Quando você ouvir o que aconteceu, vai achar que, em um primeiro momento, *parece* fazer sentido: as pessoas na Lockheed se reuniram antes de entrar na concorrência para o contrato, analisaram os requisitos e começaram a planejar como desenvolver um sistema que atenderia a todas as necessidades do cliente. Eles tinham muitas pessoas inteligentes trabalhando por meses a fio, planejando tudo o que precisava ser feito. Então, dedicaram mais alguns meses planejando *como* fazê-lo. Desenharam lindos diagramas com tudo isso, e o tempo que levaria para atingir os objetivos. Então, com uma seleção cuidadosa de cores, apresentaram um fluxo que mostrava cada uma das fases do projeto como uma cascata.



Esses fluxos se chamam “diagramas de Gantt”, em homenagem a Henry Gantt, que os desenvolveu. Com o advento dos computadores pessoais na década de 1980, tornou-se bem mais fácil desenvolver diagramas complicados

— e torná-los realmente *complexos* —, e eles se tornaram verdadeiras obras de arte. Cada etapa do projeto está detalhadamente definida; cada evento importante, cada data de entrega. Esses diagramas são realmente algo impressionante de se ver. O único problema com eles é que estão sempre *errados*. Sempre.

Henry Gantt inventou esses famosos diagramas por volta de 1910. Eles começaram a ser usados na Primeira Guerra Mundial pelo General William Crozier, quer era o Oficial Chefe do Armamento do exército dos Estados Unidos. Todos que já estudaram essa guerra sabem que sua capacidade organizacional eficiente não foi exatamente um ponto notável. O porquê de um artefato da Primeira Guerra Mundial ter se tornado uma ferramenta *prática* utilizada no gerenciamento de projetos no século 21 nunca ficou muito claro para mim. Nós já desistimos de guerras de trincheiras, mas, de alguma forma, as ideias que as norteiam ainda são populares.

É muito tentador ter todo o trabalho que precisa ser feito exposto para todos verem. Já visitei diversas empresas com funcionários cujo *único trabalho* é atualizar diariamente os diagramas de Gantt. O problema é que quando aquele plano elegante se depara com a realidade, ele cai por terra. Só que, em vez de descartar o plano ou o modo como pensam nele, os gerentes contratam pessoas para fazer com que pareça que o plano está funcionando. Basicamente, o que eles fazem é contratar pessoas para mentir por eles.

Esse padrão desastroso é um eco daqueles relatórios que os politburos[2] soviéticos recebiam na década de 1980 um pouco antes do colapso da URSS. Uma miragem completa. Assim como naquela época, os relatórios se tornaram mais importantes do que a realidade que deveriam descrever, e se houvesse discrepâncias, o problema estava na realidade, não nos diagramas.

Quando eu era um cadete em West Point, dormia no quarto antigo de Dwight Eisenhower. À noite, as luzes da rua refletiam um brilho dourado no consolo da lareira e, às vezes, isso me acordava. Havia uma placa que dizia DWIGHT D. EISENHOWER DORMIU AQUI. E eu me lembraria de que Eisenhower, certa vez, disse que planejar o combate é importante, mas assim que o primeiro tiro fosse disparado, o plano viraria fumaça. Pelo menos ele teve o bom senso de não usar um diagrama de Gantt.

Então, a Lockheed apresentou ao FBI todos aqueles lindos diagramas, e a agência assinou o contrato. Supostamente, a tarefa agora estava tão bem planejada que nada poderia dar errado. “Olhem, está no plano codificado por cor, com definição de hora e gráficos de barras.”

Ainda assim, quando Jeff e seu chefe, o CIO Chad Fulgham, olharam o plano na primavera de 2010, sabiam o que estavam vendo, o que *todos* aqueles diagramas eram na realidade: uma farsa completa. Quando os dois começaram a analisar o desenvolvimento real e o que já tinha realmente sido entregue, perceberam que o problema estava muito além de poder ser resolvido. Novos defeitos estavam sendo descobertos no software a uma velocidade muito maior do que conseguiam corrigir os antigos.

Chad informou ao Inspetor Geral do Departamento de Justiça que eles conseguiriam concluir o projeto Sentinel se o desenvolvessem internamente, cortando o número de desenvolvedores, e, assim, conseguiram entregar a parte mais desafiadora do projeto em menos de um quinto do tempo com menos de um décimo da quantia orçada. O ceticismo nos relatórios geralmente secos do Inspetor Geral (IG) para o Congresso foi palpável. No relatório de outubro de 2010, depois de expor nove pontos de preocupação sobre a proposta, os assistentes do IG concluíram: “em suma, temos preocupações e questões significativas em relação à capacidade de essa nova abordagem finalizar o projeto Sentinel dentro do orçamento e do prazo estipulados e com funcionalidade semelhante...”.[3]

Uma nova forma de pensar

Esta nova abordagem se chama Scrum. Eu a criei vinte anos atrás. Agora essa é *única* maneira comprovada de ajudar projetos desse tipo. Existem duas formas de fazer as coisas: o método antigo da “cascata” que gasta centenas de milhões de dólares e não entrega nenhum resultado, ou a nova forma, que, com menos gente e em menos tempo, consegue mais resultados com mais qualidade e menos custos. Sei que soa bom demais para ser verdade, mas a prova está nos resultados. Funciona mesmo.

Há vinte anos, eu estava desesperado. Precisava de uma nova forma de pensar sobre o trabalho. E, por meio de muita pesquisa e experiências e análise de dados passados, percebi que todos nós precisávamos de uma nova forma de organizar os empreendimentos humanos. Nada disso é ciência espacial, já se falou disso antes. Existem estudos que datam da Segunda Guerra Mundial mostrando algumas das formas como as pessoas trabalham melhor. Contudo, por algum motivo, elas nunca juntaram as peças. Nas últimas duas décadas, tentei fazer exatamente isso, E, agora, essa metodologia se tornou onipresente

no primeiro campo ao qual eu a apliquei: o desenvolvimento de softwares. Em gigantes como Google, Amazon e Salesforce.com e em pequenas start-ups sobre as quais você ainda nem ouviu falar, essa estrutura mudou radicalmente a forma como as pessoas fazem as coisas.

O motivo por que ela funciona é simples. Eu olhei a forma como as pessoas *realmente* trabalham, em vez de como elas *dizem* que trabalham. Analisei uma pesquisa realizada por décadas e as melhores práticas de empresas em todo o mundo, analisei mais a fundo as melhores equipes nessas empresas. O que as tornava superiores? O que as tornava diferentes? Por que algumas equipes atingiam resultados excepcionais e outras apenas resultados medíocres?

Por motivos que explicarei melhor em capítulos mais adiante, eu chamei de "Scrum" essa estrutura de desempenho de equipe. O termo vem do jogo de rúgbi e se refere à maneira como um time trabalha junto para avançar com a bola no campo. Alinhamento cuidadoso, unidade de propósito, clareza de objetivo, tudo se unindo. Trata-se de uma metáfora perfeita para o que uma equipe deseja fazer.

Tradicionalmente, a gerência quer duas coisas em qualquer projeto: controle e previsibilidade. Isso resulta em um vasto número de documentos, gráficos e diagramas, exatamente como os da Lockheed. Meses de esforço para o planejamento de todos os detalhes, para que não haja nenhum erro, para que o orçamento não estoure, e para que tudo seja entregue no prazo.

O problema é que o cenário cor-de-rosa nunca vira realidade. Todo o esforço investido no planejamento, tentando restringir mudanças e adivinhar o imponderável, não serve para absolutamente nada. Todo projeto envolve a descoberta de problemas e surtos de inspiração. Qualquer tentativa de restringir o empreendimento humano de qualquer natureza a diagramas coloridos é bobagem e está fadada ao fracasso. Não é dessa maneira que as pessoas trabalham e não é dessa forma que os projetos avançam. Não é como as ideias florescem ou como as coisas excepcionais sãos feitas.

Em vez disso, esse tipo de diagrama gera frustação porque as pessoas não conseguem o que realmente queriam. Os projetos atrasam e estouram o orçamento e, em muitos casos, acabam não dando certo. Isso é verdadeiro principalmente nos casos que envolvem equipes criativas trabalhando em algo novo. Na maioria das vezes, a gerência não toma conhecimento do caminho em direção ao fracasso, até que milhões de dólares e milhares de horas tenham sido investidos em vão.

O Scrum pergunta por que leva tanto tempo e tanto esforço para as coisas

serem feitas, e por que somos tão deficientes para perceber quanto tempo e esforço algo vai exigir. A catedral de Chartres levou 57 anos para ser construída. É bem seguro apostar que, no início do projeto, os pedreiros tenham olhado para o bispo e dito “Vinte anos no máximo. Talvez seja possível terminar em quinze”.

O Scrum acolhe a incerteza e a criatividade. Coloca uma estrutura em volta do processo de aprendizagem, permitindo que as equipes avaliem o que já criaram e, o mais importante, *de que forma* o criaram. A estrutura do Scrum busca aproveitar a maneira como as equipes realmente trabalham, dando a elas as ferramentas para se auto-organizar e, o mais importante, aprimorar rapidamente a velocidade e a qualidade de seu trabalho.

Em essência, o Scrum tem como base uma ideia simples: ao começar um projeto, por que não fazer paradas regulares para verificar se o que está sendo feito está seguindo na direção certa, e se, na verdade, os resultados são os que as pessoas desejam? E verificar se existem maneiras de aprimorar a forma como se está trabalhando para obter resultados melhores e executados mais rapidamente, e quais seriam os obstáculos que impedem as pessoas de obtê-los.

Isso é chamado de ciclo de “Inspeção e Adaptação”. De tempos em tempos, pare de fazer o que está fazendo, revise o que já fez e verifique se ainda deveria estar fazendo aquilo e como você pode fazê-lo melhor. É uma ideia simples, mas executá-la exige reflexão, introspecção, honestidade e disciplina. Estou escrevendo este livro para mostrar como fazer isso, e não apenas para empresas de desenvolvimento de software. Eu já vi o Scrum ser usado com sucesso para fabricar carros, gerenciar uma lavanderia, dar aulas, construir foguetes espaciais, planejar um casamento — e até mesmo, como a minha esposa o usa, para se certificar, nos fins de semana, de que lista de afazeres domésticos que ela me passou foi cumprida.

Os resultados finais do Scrum — ou o objetivo do projeto, se preferir — são equipes que melhoram drasticamente a produtividade. Nos últimos vinte anos, montei essas equipes repetidas vezes. Já fui Diretor-Executivo (CEO, na sigla em inglês), Diretor de Tecnologia da Informação (CTO, na sigla em inglês) e Chefe do Departamento de Engenharia de várias empresas, desde pequenas start-ups, com poucos funcionários em uma sala, até grandes empresas com escritórios espalhados pelo mundo. Já prestei serviços de consultoria e *coaching* para centenas de outras.

Os resultados podem ser tão drásticos que grandes empresas de pesquisa e análise, como a Gartner and Standish, agora afirmam que o antigo estilo de

trabalho se tornou obsoleto. As empresas que ainda insistem nas ideias já tentadas e malogradas de comando e controle e que tentam impor um nível rígido de previsibilidade estão simplesmente fadadas ao fracasso se seus concorrentes usarem o Scrum. A diferença é grande demais. Empresas de capital de risco, como a OpenView Venture Partners em Boston, da qual sou conselheiro, dizem que o Scrum oferece uma vantagem competitiva grande demais para não ser usado. Os profissionais dessas empresas não são calorosos nem confusos, são homens de negócio de visão aguçada, e simplesmente afirmam: “Os resultados são inquestionáveis. As empresas têm duas opções: mudar ou morrer”.

Consertando o FBI

No FBI, o primeiro problema que a esquipe do Sentinel enfrentou foram os contratos. Cada uma das mudanças acabava se tornando uma negociação contratual com a Lockheed Martin. Assim, Jeff Johnson e Chad Fulgham passaram meses deslindando os contratos, trazendo o desenvolvimento para ser feito internamente e cortando a equipe de centenas de pessoas para menos de cinquenta. A equipe principal era ainda menor.

Na primeira semana, eles fizeram o que várias pessoas fazem quando se encontram na mesma situação: imprimiram toda a documentação de requisitos. Se você nunca viu o que isso significa em um projeto de grande porte, posso dizer que o material pode chegar a centenas e centenas de páginas. Vi pilhas que tinham vários centímetros de altura. E já vi isso em vários projetos: pessoas cortando, colando e desenvolvendo um documento clichê, sendo que ninguém, na verdade, leria até o final aquelas milhares de páginas. Não dá mais para fazer isso, e esse não é o objetivo. Eles construíram um sistema que os força a endossar uma fantasia.

“Havia 1.100 requisitos. Os documentos formavam pilhas com muitos centímetros de altura”, conta Johnson. Só de pensar neles faz com que eu sinta pena das pessoas que dedicaram semanas de suas vidas produzindo documentos que *não serviam para nada*. O FBI e a Lockheed Martin não estão sozinhos nessa empreitada — já vi esse tipo de situação em quase todas as empresas nas quais trabalhei. Aquela pilha alta de futilidade é um dos motivos por que o Scrum pode ser uma mudança tão poderosa para as pessoas. Ninguém deveria passar a vida fazendo um trabalho sem significado algum.

Isso não apenas é ruim para os negócios, como também destrói a alma das pessoas.

Então, depois de ler a pilha de documentos, eles analisaram e priorizaram cada um dos requisitos do sistema, o que é de total importância e mais difícil do que parece. Em geral, as pessoas dizem que tudo é importante. Mas a pergunta que precisa ser respondida, e foi o que as equipes do Sentinela fizeram, é: o que agregará mais *valor* para o projeto? Faça essas coisas primeiro. No desenvolvimento de um software existe uma regra, criada a partir de décadas de pesquisa, que afirma que 80% do valor de qualquer parte dele está em 20% de suas funcionalidades. Pense nisto: qual foi a última vez que você usou a função do Editor de Visual Basic no Microsoft Word? Provavelmente, você nem sabe o que é Visual Basic, quanto mais por que você precisaria usar tal ferramenta. Mas ela está lá, e alguém dedicou seu tempo implementando essa funcionalidade, mas eu garanto que ela não representa um aumento significativo no valor agregado do Word.

Fazer as pessoas priorizarem de acordo com o valor as obriga a produzir aqueles 20% primeiro. Em geral, depois que eles foram concluídos, elas se dão conta de que não precisam dos outros 80%, ou que o que parecia importante no início do projeto, na verdade, não era.

Para a equipe do Sentinel, a pergunta se tornou: “Tudo bem, nós vamos desenvolver este projeto enorme que é de vital importância e já gastamos centenas de milhões de dólares nele. Quando ele vai ser concluído?”. Depois de pensar nisso, eles prometeram entregar o software no outono de 2011. O relatório do Inspetor Geral do outono de 2010 é um estudo da descrença:

> O FBI afirmou que vai utilizar uma “metodologia ágil” para concluir o desenvolvimento do Sentinel, usando menos funcionários do FBI, da Lockheed Martin, além das empresas que forneceram os principais componentes-padrão do Sentinel. Além disso, o FBI planeja reduzir o número de funcionários contratados trabalhando no Sentinel de cerca de 220 para quarenta. O FBI também afirmou que o número de funcionários do FBI designados diretamente para o projeto também será reduzido de trinta para 12 [...] O FBI nos informou que acredita que conseguirá concluir o Sentinel com os aproximadamente US$ 20 milhões restantes do orçamento e no prazo de 12 meses a partir da implementação dessa nova abordagem.[4]

O uso da expressão “metodologia ágil” mostra como o IG sabia pouco sobre o Scrum. O termo “ágil” data de uma reunião de 2001, na qual eu e 16 outros líderes de desenvolvimento de software escrevemos o que se tornou conhecido

como “Manifesto Ágil”. Nele declaramos os seguintes valores: pessoas em vez de processos; produtos que realmente funcionem em vez de documentação dizendo como o produto deveria funcionar; trabalhar com os clientes em vez de negociar com eles; e responder às mudanças em vez de seguir um plano. Scrum é a estrutura que eu construí para colocar esses valores em prática. Não existe uma metodologia.

É claro que a promessa de 12 meses feita por Johnson causou uma impressão um pouco errada. Porque, na verdade, eles não sabiam; não tinham como saber. O FBI desconhecia a velocidade com que suas equipes poderiam trabalhar. Isto é algo que eu sempre digo para os executivos: “Eu só poderei dizer a data à medida que eu vir o aprimoramento da equipe. Ou seja, o quanto vão ficar mais rápidos. O quanto eles vão conseguir *acelerar*”.

É claro que também é essencial que os membros da equipe descubram o que poderia *impedi-los* de acelerar. Nas palavras de Jeff Johnson: “Eu lidei com a remoção do obstáculo”. Um “obstáculo” é uma ideia que vem da empresa que concebeu várias das ideias nas quais o Scrum se baseia: a Toyota. E, para ser mais específico, do Sistema Toyota de Produção desenvolvido por Taiichi Ohno.

Não entrarei em detalhes aqui, mas um dos conceitos-chave que Ohno apresentou foi a ideia de “fluxo”. Ou seja, a produção deveria fluir de forma calma e rápida por todo o processo, e ele dizia que uma das principais tarefas da gerência era identificar e remover os obstáculos para tal fluxo. Tudo que fica no caminho constitui um desperdício. Ohno nos dá um conceito de desperdício, assim como um valor de negócios, em seu livro clássico, *O sistema Toyota de produção*:

> Não é exagero dizer que, em um período de crescimento lento, tal desperdício seja mais um crime contra a sociedade do que uma perda nos negócios. A eliminação do desperdício deve ser o principal objetivo de uma empresa.[5]

Ohno fala muito sobre diferentes tipos de desperdícios e obstáculos que podem atrapalhar a produção. Para que o Scrum realmente funcione, alguém da equipe sênior de gerenciamento precisa compreender a fundo que os obstáculos são praticamente criminosos. Vou explicar *como* eliminar o desperdício posteriormente neste livro. Por ora, basta dizer que o efeito da eliminação é drástico, e as pessoas não costumam fazer isso porque requer honestidade consigo mesmas e com os outros.

Jeff Johnson sabia que aquele era o seu trabalho.

A equipe do Sentinel levou cerca de três meses para descobrir quanto tempo *realmente* levaria para concluir o projeto. O motivo? A resposta nos leva de volta àquele ciclo de "Inspeção e Adaptação" sobre o qual falei antes. O Scrum funciona com a definição de objetivos sequenciais que devem ser concluídos em um período definido. No caso do FBI, eles decidiram por ciclos de duas semanas, compreendendo que, ao final de cada ciclo, deveria haver uma parte *concluída* do produto. Isso significava que eles precisavam ter alguma coisa funcionando, algo que poderia ser mostrado para qualquer pessoa que quisesse ver, mas certamente para os *stakeholders* e, idealmente, para aqueles que realmente usariam o programa.

Essa metodologia permite que as equipes tenham um feedback quase que imediato do trabalho realizado. Eles estão na direção certa? O que planejam fazer em seguida é *realmente* o que deveriam fazer, considerando tudo que descobriram durante aquele ciclo?

No Scrum chamamos esses ciclos de Sprint [corrida de velocidade de curta distância]. No início de cada ciclo, acontece uma reunião para planejar o Sprint. A equipe decide a quantidade de trabalho que acredita ser capaz de realizar nas duas semanas seguintes. Eles escolhem as tarefas na lista de prioridades, as anotam em *post-its* e os colam na parede. A equipe decide quantas tarefas será capaz de executar em duas semanas.

No final do Sprint, a equipe se reúne e mostra o que conseguiu realizar naquele tempo. Eles analisam quantos dos *post-its* da parede realmente foram concluídos. Será que tinham escolhido tarefas demais e não conseguiram concluir todas? Ou será que haviam escolhido poucas? O importante era que começassem a estabelecer uma base para sentir o ritmo do trabalho — a velocidade que conseguiam atingir.

Depois de mostrarem o que conseguiram fazer — e é aqui que as ideias de Ohno entram —, a equipe discute não o que fizeram, mas *como* fizeram. Eles perguntam: "Como podemos trabalhar melhor no próximo Sprint? Quais foram os obstáculos que tivemos de remover durante esse período? Quais são os obstáculos que estão diminuindo o nosso ritmo?".

E foi por isso que Jeff Johnson precisou de alguns meses antes de poder dizer quanto tempo o projeto demoraria. Ele queria mensurar a velocidade de cada equipe com base em alguns Sprints e, depois, ver o quanto eles poderiam melhorar — o quanto mais poderiam acelerar e avançar. Uma vez que analisou quantas tarefas cada equipe concluiu em cada Sprint e verificou quantas ainda

havia até o final do projeto, ele foi capaz de projetar uma data de conclusão.

Além de descobrir a velocidade das equipes, ele também queria saber quais eram os obstáculos que as atrasava. O que ele realmente queria fazer era *acelerar* as equipes para que produzissem mais rápido. Mas não trabalhando mais tempo (explicarei posteriormente por que esse é um caminho inútil que acaba fazendo com que as coisas levem mais tempo), mas sim trabalhando *melhor* e de forma *mais inteligente*. Jeff relatou que suas equipes aumentaram a produtividade em um *fator de três*, ou seja, estavam avançando três vezes mais rápido depois que começaram a se mover, em relação ao início do projeto. O motivo? Ficaram melhores no trabalho em equipe, sim, mas o mais importante: descobriram o que os atrasava e, a cada Sprint, tentavam se livrar dos obstáculos.

Por fim, foram necessários 18 meses de codificação para implementar o sistema de banco de dados do projeto Sentinel, e mais dois meses para disponibilizá-lo para todo o FBI. “Foi uma tremenda pressão de tempo”, contou Johnson em uma entrevista. “E vocês têm de entender que o sistema é usado para *tudo*: pagamento de informantes, armazenamento de provas, arquivos dos casos, agendas. Esta *reunião* está no Sentinel”.

E, na opinião dele, qual era a parte mais eficaz do Scrum? “As demonstrações. O trabalho voltado para um produto demonstrável com frequência”. A cada duas semanas a equipe do Sentinel se reunia e demonstrava tudo o que tinha conseguido. E esse sistema de mostrar e contar não era apenas para eles. A equipe levava o que tinha feito e o mostrava para as pessoas que realmente usariam o programa. Todos que tinham interesse no projeto enviavam um representante, o que significava uma apresentação com casa cheia. Gravações. Inteligência. Agentes especiais. Um funcionário do Inspetor Geral. Representantes de outras agências governamentais. Algumas vezes, o Diretor ou Vice-Diretor do FBI estava na sala, assim como o próprio Inspetor Geral. *Não* era um público muito fácil de se lidar.

E foi isso que fez as coisas funcionarem, afirma Johnson. “Scrum não é sobre os desenvolvedores. Mas sim sobre os clientes e os *stakeholders*. Sério, era uma mudança organizacional. Mostrar o *produto real* era a parte mais eficaz.”

Na verdade, mostrar o produto era extremamente válido porque as pessoas estavam bastante descrentes, para dizer o mínimo, quanto ao progresso do trabalho. Elas não conseguiam acreditar que o Sentinel continuava provocando avanços em um ritmo cada vez mais rápido. “Eu disse para o Congresso que com 5% do orçamento e em vinte meses nós conseguiríamos o que a Lockheed

não tinha conseguido fazer com 90% do orçamento em um período de dez anos", conta Johnson.

> Todos estavam céticos. Nós tínhamos de fazer relatórios para o Assistente do Procurador Geral da Justiça. Tínhamos de ser transparentes em relação ao status do projeto, mas o nosso público continuava desconfiando de que havia algo desonesto acontecendo. Eles já tinham visto aqueles tipos de indicadores no passado, os relatórios começaram a ficar menos detalhados e outra coisa *estava* acontecendo.

O ceticismo contagiou o resto do FBI. *Os caras no porão vão ferrar com tudo de novo*, era o que pensavam. Aquele seria apenas um sistema temporário que os deixaria na mão de novo e eles teriam de voltar a usar o papel.

Jeff falou para sua equipe sobre uma citação que tivera de decorar quando ainda era um cadete naval. Era um trecho do discurso "Cidadania numa República", que Teddy Roosevelt fez em Sorbonne, em 1910. Ele costuma ser bastante citado, e você talvez já o conheça:

> Não é o crítico que conta, não o homem que aponta como o homem forte tropeça, ou onde o executor de ações poderia ter feito melhor. O crédito pertence ao homem que está realmente na arena, com o rosto desfigurado pela poeira e suor e sangue; que se esforça corajosamente; que erra, que tenta de novo e de novo, porque não existe esforço sem erros e lacunas; mas quem realmente se esforça para fazer as obras, quem conhece grandes entusiasmos, as grandes devoções; quem consome-se em uma causa digna é quem melhor conhece, no final, o triunfo das grandes realizações, e que, na pior das hipóteses, se falhar, pelo menos não será enquanto não tiver ousado muito, de tal forma que seu lugar nunca será junto às almas frias e tímidas que não conhecem a vitória nem a derrota.[6]

A equipe enfrentou alguns atrasos enquanto tentava descobrir exatamente com que rapidez conseguiria trabalhar e o nível de dificuldade de tudo que tinha pela frente. Por fim, em julho de 2012, eles ativaram o Sentinel. E tiveram de ativar todo o sistema e para todos. Não tinha como fingir uma coisa daquelas.

"Aconteceu de um dia para outro. Em um caso criminal ou de contra terrorismo, algo acontecendo em Los Angeles poderia ser relacionado a algo acontecendo em Chicago", explicou Jeff Johnson. "Não poderíamos permitir que pistas fossem perdidas. *Em cada detalhe nós tínhamos de apresentar uma solução boa, conhecida e clara.*"

E tal solução deveria ser clara e boa o suficiente para ser mantida em um

julgamento. Os dados no Sentinel seriam usados em julgamentos, e sua integridade precisava estar acima de qualquer suspeita.

Jeff estava frenético e nervoso naquele primeiro dia. Foi até o escritório e ligou o Sentinel. Ele carregou. Bom sinal. Então, tentou aprovar um documento com uma assinatura eletrônica — uma tarefa básica que dezenas de milhares de funcionários do FBI costumavam fazer o tempo todo. Ele recebeu uma mensagem de erro. Não estava funcionando. Ele entrou em pânico, e visões de desastre encheram sua cabeça. Então, leu com cuidado o código de erro que tinha recebido e se deu conta do que aquilo significava: ele não havia inserido o cartão com o qual o computador verificaria a sua identidade. Ele, então, inseriu o cartão, clicou com o mouse e o Sentinel estava pronto.

O sistema teve um efeito drástico no FBI. A capacidade de comunicação e compartilhamento de informações mudou fundamentalmente o que a agência seria capaz de fazer. Em janeiro de 2013, um agente de campo do FBI foi chamado quando uma conta de uma pequena empresa foi invadida por um hacker. O valor de US$ 1 milhão fora transferido para outro país antes que os bancos norte-americanos pudessem tomar qualquer atitude. Usando o Sentinel, o agente local coordenou uma ação conjunta com um colega na embaixada do país de destino, que alertou as autoridades locais, que, por sua vez, impediram que a transferência fosse concluída antes de entrar no sistema bancário. Isso aconteceu em questão de horas, algo que simplesmente não seria possível na época das três vias de documentos e canetas vermelhas. Foi a diferença entre pegar um bandido ou permitir que ele se safasse.

No porão do FBI, a equipe do Sentinel ainda está lá, e as divisórias foram removidas das baias para que um colega possa olhar para o outro. Na parede há um pôster grande com os princípios do “Manifesto Ágil”— princípios que eu ajudei a escrever e dediquei minha vida para disseminar pelo mundo. Por mais estranho que pareça para uma sala sem janelas, uma alfazema cultivada cresce, saudável, sob a luz fluorescente. “Alfazema” era o codinome do protótipo do Sentinel. Os membros da equipe ainda estão nos seus postos, aperfeiçoando e acrescentando novas funcionalidades ao sistema que desenvolveram.

Existe uma piada antiga na comunidade do Scrum: uma galinha e um porco estão caminhando pela estrada, e a galinha diz:

“Ei, porco, eu estava pensando que a gente devia abrir um restaurante.”

“E qual vai ser o nome do restaurante?”, pergunta o porco.

“Que tal ‘Presunto e Ovos’?”

“Não, obrigado”, responde o porco. “Eu teria que me comprometer, mas você

só teria de se envolver.”

A ideia do Scrum é que os “porcos” são aqueles que estão completamente comprometidos com o projeto e são responsáveis diretos pelos resultados. As “galinhas” são as pessoas informadas sobre os progressos realizados, ou seja, os *stakeholders* ou partes interessadas. Na parede da sala do Sentinel, temos um sino na forma de uma cabeça de porco. Quando ele soa, as pessoas que fizeram tudo aquilo que todos disseram que não poderia ser feito sabem que estão sendo chamadas. Temos outra campainha do lado de fora, mas aquela é só para as galinhas.

A cada dia o mundo se torna um lugar mais complicado, e o trabalho que realizamos está cada vez mais complexo, em um ritmo cada vez mais rápido. Pense em carros, por exemplo. Eu costumava passar muito tempo fazendo consertos simples no meu carro. Há uns trinta anos, eu conseguiria reconstruir um radiador. Agora, quando abro o capô do meu carro, posso muito bem estar olhando para a parte interna de um computador. Na verdade, é exatamente o que *estou* fazendo, já que o novo Ford tem mais linhas de código do que o Facebook e o Twitter juntos. Criar algo tão complexo assim é um grande peso para a humanidade. Sempre que as pessoas estão envolvidas em um esforço criativo e complexo, seja para enviar um foguete para o espaço, inventar um interruptor de luz melhor ou capturar um criminoso, os métodos tradicionais de gerenciamento simplesmente não funcionam.

E nós sabemos disso — tanto como indivíduos quanto como sociedade. Nós enxergamos os ecos da nossa vida real representados em distopias fictícias como as mostradas no desenho *Dilbert* ou no filme *Como enlouquecer seu chefe*. Todos nós já chegamos em casa e contamos para a nossa família ou amigos sobre a loucura que é a “organização” corporativa moderna. Já disseram para todos nós que preencher corretamente o formulário é mais importante do que fazer o trabalho ou que precisamos fazer uma reunião para nos preparar para uma reunião preparatória. É uma loucura. Ainda assim, continuamos fazendo isso. Mesmo diante do absoluto e completo fracasso.

O lançamento do Healthcare.gov, website no qual os norte-americanos deveriam poder se cadastrar para o seguro de saúde, é um ótimo exemplo disso. A interface era linda; inteligente, clara — um primor de design. Ela foi concluída em três meses usando o Scrum. A funcionalidade, porém, era um fracasso. Simplesmente não funcionava. Ela deveria conectar os bancos de dados da receita federal aos bancos de dados estaduais, das seguradoras, do departamento do serviço de saúde. Trata-se de um trabalho complexo que

envolveu mais de vinte empresas trabalhando em diferentes partes, e elas planejaram tudo usando as técnicas em cascata e só testaram o site ao final do projeto por alguns dias, em vez de realizar testes incrementais ao longo de todo o processo.

A tragédia é que todo mundo sabia o que ia acontecer. As pessoas que trabalham para aquelas empresas não são burras. Elas sabiam. O problema foi que todo mundo disse “Não é minha responsabilidade”. Cada empresa entregou a sua parte e ficou por isso mesmo. Eles nunca olharam o site do ponto de vista do usuário, mas apenas do próprio ponto de vista. E fizeram isso porque não estavam alinhados entre si — não estavam unidos em prol de um objetivo comum. O que o Scrum faz é promover a união das equipes para criar grandes projetos, e isso exige que todos não apenas vejam o objetivo final, mas que também façam entregas incrementais para atingi-lo. Não havia ninguém responsável pelo projeto do site Healthcare.gov que tenha insistido para que tudo fosse testado à medida que fosse desenvolvido e, infelizmente, em termos de fracasso, a história desse site dificilmente pode ser considerada atípica.

Quantas vezes você ouve falar sobre algum projeto enorme com custo de milhões e milhões de dólares ser cancelado não apenas porque os custos ultrapassaram o orçamento, mas também por que simplesmente *não funcionavam*? Quantos bilhões de dólares são gastos a cada ano para não produzir *nada*? Quanto tempo da sua vida é desperdiçado em um trabalho que tanto você quanto seu chefe sabem que não cria nenhum valor? É como se você estivesse cavando buracos para tapá-los em seguida, se você for considerar todo o impacto que está causando.

Não precisa ser assim. Não precisa mesmo. Só porque todo mundo sempre disse para você que é assim que o mundo funciona não significa que eles estão certos. Existe *sim* uma maneira diferente de fazer as coisas — uma maneira diferente de se trabalhar.

E se você *não o fizer*, você será substituído. Ou a sua empresa vai morrer. O mundo hipercompetitivo do século 21 não tem espaço para bobagens.

Um ponto ainda mais importante: trabalhar de uma forma produtiva ao máximo — com o modo Scrum — não precisa se restringir aos negócios. E se as pessoas usassem esse método para resolver os grandes problemas da humanidade, tais como a dependência do petróleo, problemas de educação, falta de água potável nas partes mais pobres do mundo, ou o aumento nos índices de criminalidade? E se realmente existir uma maneira melhor de viver, trabalhar e resolver os problemas de uma forma diferente? Uma maneira que

realmente possa mudar o mundo? Pois *existe*. Há pessoas usando o Scrum para lidar com cada uma dessas questões que eu mencionei, e elas estão causando um grande impacto.

Neste livro você vai aprender algumas das maneiras fundamentais como as pessoas trabalham melhor, por que somos péssimos ao fazer estimativas e por que fazer hora extra vai resultar em mais atrasos ao projeto. Vou mostrar todas as pesquisas e aplicações que pessoas comuns, cientistas e organizações fizeram diligentemente durante anos, e mostrar como o Scrum une tudo isso de uma forma que você pode começar a usá-lo imediatamente.

Eu vou mostrar *como* fazer isso. Primeiro, porém, eu quero contar a história de como cheguei até aqui.

PONTOS PRINCIPAIS

Planejar é útil. Seguir cegamente os planos é burrice. É simplesmente *tentador demais ficar desenhando diagramas sem fim*. Todo o trabalho que precisa ser feito em um projeto de grande porte deve ser definido para todos verem — mas, quando os planos detalhados se deparam com a realidade, eles viram ruínas. Inclua no seu método de trabalho a possibilidade de mudança, descoberta e inovação.

Inspeção e Adaptação. De tempos em tempos, pare de fazer o que está fazendo, revise o que já fez e verifique se isso ainda é o que você deveria estar fazendo e se existe uma maneira de fazer melhor.

Mudar ou morrer. Ficar preso ao modo antigo de fazer as coisas, de mandar ou controlar e manter uma previsibilidade rígida resultará apenas no fracasso. Nesse meio tempo, a concorrência que estiver disposta a mudar deixará você comendo poeira.

Fracasse rápido para que possa corrigir o problema o quanto antes. A cultura corporativa costuma dar mais valor a formulários, procedimentos e reuniões do que à criação de valor palpável que pode ser verificado a curtos possibilita de tempo pelos usuários. O trabalho que não resulta em valor real é loucura. Trabalhar em um produto em ciclos curtos possibilita um feedback inicial do usuário, permitindo que você possa eliminar imediatamente tudo aquilo que obviamente constitui um desperdício de esforço.

1 Dan Eggen e Griff Witte, *The FBI's Upgrade That Wasn't; $170 Million Bought an Unusable Computer System, The Washington Post*, Washington, 18 de agosto de 2006, Seção A, p. 1.

2 Politburo era o comitê central do Partido Comunista da antiga União Soviética (URSS), e politburos eram seus afiliados. (N.T.)

3 U.S. Department of Justice, Office of the Inspector General, *Status of the Federal Bureau of Investigation's Implementation of the Sentinel Project*, Relatório de 11 de outubro de 2010.

4 *Ibid.*

5 Taiichi Ohno, *Toyota Production System: Beyond Large-scale Production*, Cambridge, MA, Productivity,

1988. [Em português, *O sistema Toyota de produção: além da produção em larga escala*, Porto Alegre, Bookman, 1997.]

6 Roosevelt, Theodore, *Citizenship in a Republic*. Discurso proferido na Université Paris-Sorbonne, França, 23 de Abril de 1910. [Em português, *Cidadania numa República.* Disponível em: <http://wallacecezar.wordpress.com/2010/11/24/o-homem-na-arena-cidadania-numa -repblica/>. Acesso em: 31 jul. 2014]

CAPÍTULO 2

As origens do Scrum

Para os pilotos de combate no Vietnã, o período de serviço militar era sinônimo de sobrevoar o território inimigo em cem missões. Metade dos pilotos foi derrubada, alguns foram resgatados, mas a maioria nunca conseguiu voltar. Em 1967, ainda como um jovem piloto de combate inexperiente, fui enviado da base militar Mountain Home Air Force, em Idaho, para a base Udorn Royal Thai Air Force, na região norte da Tailândia, para realizar o trabalho mais perigoso da Força Aérea dos Estados Unidos: reconhecimento.

Isso aconteceu muito antes da era das missões Predator com *drones* e imagens de satélite confiáveis. Todas as armas foram retiradas do meu RF-4C Phantom, e a aeronave equipada com câmeras e um tanque extra de combustível. A minha missão era sobrevoar o território inimigo, de forma que o meu navegador pudesse tirar fotos antes e depois das missões de bombardeio. A maioria das operações era realizada à noite, e eu voava pela escuridão tropical a apenas algumas centenas de pés do chão, quase esbarrando na copa das árvores. No momento em que cruzava a fronteira do Vietnã do Norte, o meu *Heads-Up Display*[7] se acendia como uma máquina de fliperama, e o sistema de aviso contra mísseis soava com uma fúria de bipes e apitos. O céu era cortado por balas disparadas de canhões antiaéreos, e eu sabia que, em questão de minutos, um radar antimísseis estaria com a minha aeronave na mira, a não ser que quinhentos pés fosse baixo o suficiente para me manter invisível para o radar.

Naquelas situações a adrenalina corria nas minhas veias, mas eu nunca perdi a calma. Ao contrário, o perigo quase me acalmava. Acredito que isso se deve ao treinamento que recebi na Força Aérea para aprender a controlar o risco. Lá aprendi quatro coisas: observar, avaliar, decidir e agir. Para ser mais específico, observava a região-alvo, planejava o melhor caminho para entrar na zona de ataque, avaliava as opções diante de eventos inesperados, e, por fim, agia de forma decisiva com base nos instintos e no treinamento. A hesitação resultava

na morte de pilotos, assim como a imprudência. Assim que o meu navegador tirava as fotografias, eu puxava o manche e saía da zona de guerra, com meu campo de visão se reduzindo, devido à força *g*. O navegador costumava desmaiar por causa da força e, em alguns casos, perdia o controle dos intestinos. Mas nunca reclamou. Eu sempre consegui nos levar de volta a salvo.

Naquela época eu era apenas um jovem piloto rezando para sobreviver às missões designadas a mim. Eu não sabia que a minha experiência nessa função e o treinamento que recebi sobre como pensar e agir em situações de vida ou morte definiriam o modo como eu trabalharia pelo resto da vida. Cheguei ao Vietnã em 1967 com dois esquadrões de pilotos de combate F-4 e dois da aeronave de reconhecimento RF-4C, em um total de cem aviões. O avião de reconhecimento substituiu duas esquadrilhas de RF-101. Daquelas cinquenta aeronaves RF-101, apenas quarto não foram derrubadas em um ano. E essas quatro restantes tinham tantos buracos de bala que era impossível voar com elas. Não sei como seus pilotos conseguiram pousá-las depois da última missão. O RF-4C era uma nave de combate mais resistente, mas, mesmo assim, metade dos aviões foi abatida em um ano. Melhoramos a taxa de sobrevivência, mas, ainda assim, 50% dos que chegaram comigo não voltaram para a base, embora alguns tenham sido resgatados na floresta antes que se tornassem prisioneiros.

Quando voltei da guerra, fiz mestrado em estatística em Stanford e passei o maior tempo possível no laboratório de inteligência artificial da universidade. Depois, tornei-me professor de matemática da Air Force Academy, onde entrei em um programa de PhD em biometria na University of Colorado Medical School. Lá, perguntei ao meu orientador, dr. John Bailar, um dos pesquisadores mais reconhecidos em medicina e estatística, como eu poderia escrever uma tese que fosse útil e não terminasse em uma prateleira empoeirada na biblioteca. Ele me entregou trezentos artigos médicos sobre câncer, cada um deles exibindo dados estatísticos que variavam drasticamente entre humanos e animais e tipos de tumor. Bailar disse que se eu conseguisse explicar por que todos eram diferentes, ele me daria o título de doutor. Então, foi exatamente o que fiz, e consegui meu PhD.

Para isso, passei anos tentando descobrir o que acontecia quando uma célula se tornava cancerosa. Aprendi muito sobre teoria de sistemas e como um sistema apresenta apenas determinados estados estáveis. Quando uma célula evolui, ela passa de um estado estável para outro. Dediquei quase dez anos para descobrir as regras para transpor um sistema adaptativo complexo de um estado

para outro, e como fazer com que o próximo estado fosse positivo em vez de negativo.

Anos mais tarde, ocorreu-me que organizações, equipes e pessoas constituem sistemas adaptativos complexos. Os elementos que fazem com que uma célula passe de um estado a outro são os mesmos que movem as pessoas. Para mudar uma célula, é necessário, primeiro, injetar energia no sistema. Primeiro temos o caos — parece não haver regras, tudo está no fluxo. Quando você faz isso em organizações como uma tentativa de mudança, as pessoas costumam se apavorar, não conseguem entender o que está acontecendo e não sabem como agir. No entanto, de forma bastante rápida, assim como ocorre com as células, a organização entra em um novo estado estável. A única questão é se o novo estado é melhor do que o antigo. A célula é cancerosa ou saudável? Eu me perguntava *como podemos descobrir algumas regras simples que possam guiar as equipes a entrar em um estado mais produtivo, feliz, incentivador, divertido e arrebatador?* Passei os quinze anos seguintes tentando descobrir a resposta.

Durante o governo Reagan, as verbas destinadas à pesquisa científica foram drasticamente cortadas, incluindo as destinadas ao estudo do câncer realizada pelos National Cancer Centers, onde eu trabalhava como analista dos dados coletados nos estudos clínicos e epidemiológicos do Colorado Regional Cancer Center. Enquanto eu pensava no que fazer, uma empresa chamada MidContinent Computer Services entrou em contato comigo porque ouviram dizer que eu era o principal especialista na área da mais nova tecnologia da empresa. A MidContinent trabalhava prestando serviços para 150 bancos na América do Norte e seu mais novo e desejável produto chamava-se “Caixa Automático de Autoatendimento” (ATM, na sigla em inglês), ou caixas eletrônicos. Estávamos em 1983, quando sacar dinheiro significava ficar em uma fila no banco ou passar de carro em um *drive-through* bancário, quando você precisava preencher um cheque para “sacar” a quantia que queria e entregá-lo ao caixa.

Os caixas eletrônicos seriam a solução, mas, naquela época, a empresa enfrentava problemas para fazer suas redes se comunicarem entre si. Eles precisavam de alguém que conhecesse sistemas, então me fizeram uma proposta lucrativa para ser o vice-presidente de sistemas avançados. As máquinas que formavam a rede de computadores eram iguais às que eu passara anos rodando os meus programas de doutorado, então, eu era uma boa opção.

Ou foi apenas o que pensei. Nada nunca é fácil, não é? Quando entrei na empresa, deparei-me com um departamento que usava o método em cascata

para o planejamento dos projetos. Havia centenas de programadores de computador que se sentavam em suas mesas o dia inteiro trabalhando de maneira ostensiva, mas sem conseguir entregar nada dentro do prazo ou do orçamento. Para os caixas automáticos, os custos eram 30% mais altos do que a receita. A falta de eficiência surpreendia.

No início, passei um tempo tentando entender como as coisas funcionavam. Dá para imaginar como a alta diretoria tratava a minha equipe. Houve muitos gritos, microgerenciamento, comportamento passivo-agressivo e exigências de mais trabalho com horas-extras. Mas não importava o quanto a diretoria pressionasse, os atrasos eram crônicos, os projetos estavam acima do orçamento previsto e nada era entregue quando deveria ser.

Decidi que a melhor opção seria mudarmos tudo. A operação estava defeituosa demais para ser consertada de forma gradual; assim, decidi criar uma empresa dentro de uma empresa. Pedi ao nosso CEO, Ron Harris, que me deixasse montar uma organização separada com todos que estavam envolvidos nas redes ATM. Teríamos a nossa própria equipe de vendas, nossa própria equipe de marketing e nosso próprio departamento financeiro. Ron era um CEO brilhante e criativo que confiava no meu trabalho. Talvez sob a direção de outra pessoa, isso nunca tivesse acontecido. Depois de ouvir a minha ideia, ele disse: “Sutherland, se você quer uma dor de cabeça dessa, fique à vontade”.

E foi o que fiz. Cheguei aos desenvolvedores e gerentes e disse a eles: “Nós temos que parar de fazer as coisas que estão nos matando”. A situação era como aquela antiga piada sobre ficar batendo com a cabeça contra um muro de tijolos só para se sentir melhor depois que parar. “Temos de descobrir um modo melhor de trabalhar”, afirmei, “e temos de começar já”.

Então, comecei a dirigir uma pequena empresa como uma equipe dividida em subequipes. Os bônus não tinham como base o desempenho individual, mas sim o desempenho da empresa como um todo. Descobrimos ferramentas que abriram o caminho até o Scrum dez anos mais tarde — por exemplo, os conceitos de Dono do Produto (Product Owner), Pendências do Produtor (Product Backlog) e Sprints semanais, os quais abordarei em mais detalhes mais adiante. Em seis meses, éramos a divisão mais lucrativa da empresa, e a receita era 30% mais alta do que os custos. Os nossos sistemas Nonstop Tandem constituíram os primeiros computadores de transações on-line que os bancos tiveram confiança o suficiente para usar, então nós os distribuímos por toda a América do Norte. Hoje, você encontra caixas eletrônicos praticamente em qualquer lugar do país, e cada um deles sabe exatamente quanto dinheiro

você tem. A minha equipe teve muito a ver com isso. E, sim, você pode agradecer.

Aprendendo a pensar como um robô

Depois da minha experiência na carreira militar seguida pela carreira acadêmica, eu me considerava meio que um intruso no mundo dos negócios. No entanto, aquela perspectiva constituiu um dos meus ativos mais valiosos. Desde o primeiro dia, eu não conseguia entender por que as pessoas insistiam em trabalhar de uma forma que *sabiam* que era ineficiente, destrutiva, desumanizadora e depressiva. Acho que elas imaginam que, se todo mundo trabalha assim, então essa deve ser a melhor forma de se trabalhar.

Eu realmente adorei a época em que trabalhei para a MidContinent, mas estava ávido para testar as minhas habilidades diante de novos desafios. Nas duas décadas seguintes, acabei prestando meus serviços para empresas pequenas e grandes como Vice-Presidente de Engenharia ou Diretor Executivo de Tecnologia. Em cada um deles, tentei fazer as equipes trabalharem juntas de formas mais eficientes. E, em uma dessas empresa, eu me vi em um prédio em Cambridge, Massachusetts, a apenas alguns quarteirões do MIT. Alguns doutores e professores haviam acabado de fundar uma nova empresa para construir robôs, e eles a dirigiam em uma sala no laboratório do MIT. Ao final, acabaram sublocando espaço da minha empresa.

Algumas semanas depois que se mudaram, o inesperado aconteceu: um robô de seis pernas e do tamanho de um gato entrou no meu escritório e começou a me perseguir em volta da minha mesa de trabalho. O inventor entrou nervoso e se desculpou pela máquina, mas, a cada dois ou três dias, o episódio se repetia. Um dos robôs fugia do laboratório e começava a correr pelo prédio. Eu conseguia ouvir o som das pernas mecânicas passando pelos corredores.

Nas tardes de sexta-feira, eu sempre servia vinho e cerveja no escritório, para que os funcionários relaxassem e socializassem depois de uma semana pesada de trabalho. Eu convidava o cientista robótico que trabalhava no final do corredor para participar desses eventos e, certa vez, Rodney Brooks apareceu; ele era professor de inteligência artificial no MIT e um dos fundadores da empresa de robótica. Eu perguntei como os robôs errantes funcionavam.

“Há décadas tentamos construir uma máquina que realmente tenha a capacidade de pensamento inteligente”, disse ele. “Gastamos bilhões de

dólares, muitos anos de trabalho para construir os maiores computadores que podemos, com os maiores bancos de dados, mas tudo que conseguimos foi um computador que consegue vencer seres humanos em um jogo de xadrez".

Ele me explicou que tinha usado uma abordagem completamente diferente com seus robôs. Em vez de tentar construir algo com um cérebro central, sua equipe fez um robô cujas pernas tinham, cada uma, um cérebro próprio. Além disso, um processador na espinha trabalhava com algumas regras simples: siga em frente, volte, não bata nas outras pernas. O chip da rede neural na cabeça do robô conhecia tais regras e agia como um árbitro para todas as partes. Ele dizia a elas o que via através da câmera, quando ele atingia um obstáculo, por exemplo.

O que é interessante, explicou Brooks, é que cada vez que você liga o robô, ele aprende a andar pela primeira vez. Não há banco de dados de como as coisas estão dispostas na sala; em vez disso, o mundo é o banco de dados. O robô descobre tudo pela primeira vez a cada vez que é ligado. Ele bate nas coisas com base no lugar em que se encontra, o que significa que pode se adaptar a qualquer ambiente.

"Deixe eu mostrar para você", sugeriu enquanto me levava ao laboratório. Ele enfiou um chip neural em branco em um dos robôs insectoides, e observei enquanto aquela máquina ganhava vida. Hesitante no começo, ele tropeçava pela sala como um filhote de gamo[8] se erguendo sobre as pernas pela primeira vez. No entanto, a cada passo, ia se tornando mais seguro. As pernas aprendiam rapidamente a colaborar entre si e trabalhar juntas. Em questão de poucos minutos, o robô corria pela sala. Nenhum dado era armazenado e não havia uma programação sobre como andar; em vez disso, algumas regras simples faziam com que aqueles componentes trabalhassem juntos. As pernas não pensavam, apenas agiam. Fiquei maravilhado como o sistema era engenhoso e simples ao mesmo tempo. Diante de mim havia algo fazendo exatamente o que eu fora treinado para fazer enquanto voava no Vietnã: observar, avaliar, decidir e agir. Ele absorvia o ambiente em que se encontrava e se comportava de forma decisiva com base nos dados daquele lugar.

Perguntei a Brooks: "o que aconteceria se a gente conseguisse descobrir um conjunto simples de instruções para equipes de pessoas trabalharem juntas exatamente como aquelas pernas? Elas se auto-organizariam e se auto-otimizariam exatamente como aquele robô".

"Eu não sei", respondeu ele. "Por que você não tenta e depois me diz se funcionou?"

Não saia em busca de cascatas

Cada vez mais eu percebia que, se conseguisse criar um sistema que, como aquele robô, pudesse coordenar pensadores independentes com feedback constante sobre o seu ambiente, obteria níveis de desempenho muito mais elevados. O aperfeiçoamento do fluxo de informações entre as “pernas” de um grupo resultaria em níveis de eficiência nunca antes alcançados.

A minha conversa com Brooks aconteceu há mais de duas décadas. Durante muitos anos, ele foi o chefe do departamento de robótica e inteligência artificial do MIT, e aquele robô, semelhante a uma aranha, que conheci, nomeado Genghis Khan, agora se encontra no Smithsonian como item de colecionador. Você já deve conhecer uma das empresas de Brook, a iRobot, que fabrica o aspirador de pó Roomba e usa a mesma inteligência adaptativa para limpar o chão que Genghis Khan usava para me perseguir no meu escritório. A sua mais nova inovação na Rethink Robotics, o robô Baxter, consegue trabalhar de forma colaborativa com seres humanos no mesmo espaço de trabalho.

O trabalho de Brook me inspirou. E, em 1993, levei aquelas ideias comigo para uma empresa chamada Easel, que me contratara como vice-presidente de tecnologia. Os executivos queriam que a minha equipe desenvolvesse uma nova linha de produtos em seis meses, a qual seria oferecida a alguns dos seus maiores clientes — como a Ford Motor Company, que usava o software deles para desenvolver e construir aplicações internas. Reuni-me com a minha equipe de desenvolvimento e disse que sabia que eles não conseguiriam fazer aquilo usando o mesmo modelo antigo de desenvolvimento de software.

O sistema antigo era o método em cascata que descrevi no capítulo anterior: tudo relacionado a um projeto cuidadosamente disposto naqueles diagramas gigantescos de Gantt, com cada tarefa medida em horas precisas e destacadas em cores atraentes. Aqueles diagramas eram lindos em sua precisão. E também eram uma farsa completa.

Na Easel eu sabia que a metodologia em cascata atrasaria nosso prazo em meses, talvez até em anos. Precisávamos descobrir uma maneira completamente diferente de fazer as coisas. Eu fui até o CEO e informei que íamos rasgar o diagrama de Gantt. Ele ficou chocado e exigiu saber o motivo.

“Quantos diagramas de Gantt você já viu na sua carreira?”, perguntei.

“Centenas”, respondeu ele.

“Quantos estavam certos?”

Ele fez uma pausa.

“Nenhum.”

Então, eu disse a ele que entregaria um software em perfeito funcionamento no final do mês, em vez de um digrama de Gantt não cumprido. Ele mesmo poderia testá-lo para verificar se estávamos no caminho certo. E *teríamos* de estar se quiséssemos cumprir nosso prazo.

Minha equipe e eu passamos algumas semanas lendo centenas de documentos, livros e artigos sobre organização de equipes e desenvolvimento de produtos. Até que um dos desenvolvedores trouxe um artigo publicado na Harvard Business Review, em 1986, escrito por dois professores de administração japoneses, Hirotaka Takeuchi e Ikujiro Nonaka. O título era “The New New Product Development Game” [O novo jogo para o desenvolvimento de novos produtos]. Takeuchi e Nonaka analisaram equipes de algumas das empresas mais produtivas e inovadoras: Honda, Fuji Xerox, 3M, Hewlett-Packard, entre outras. Eles argumentavam que o método antigo usado para o desenvolvimento de novos produtos, simbolizado pelo Sistema de Planejamento e Fases da NASA — um sistema em cascata — era fundamentalmente falho. Em vez de tais sistemas, as melhores empresas usavam um processo de desenvolvimento de sobreposição, que era mais rápido e mais flexível. As equipes eram multifuncionais. Tinham autonomia e um objetivo transcendente: estavam em busca de algo maior do que elas mesmas. A direção não impunha ordens — ao contrário, os executivos eram líderes facilitadores focados em retirar os obstáculos do caminho, não em determinar o que deviam fazer e como deveriam desenvolver o produto. Os professores japoneses compararam o trabalho de equipe a um time de rúgbi e diziam que as melhores equipes agiam como se houvesse um *scrum*: “[...] a bola é passada pelo time conforme ele avança, como uma unidade, pelo campo”.[9]

O artigo de Takeuchi e Nonaka chamou atenção quando foi publicado pela primeira vez, mas aquilo tinha sido sete anos antes da nossa leitura na Easel. Todos admiraram o conceito, mas ninguém fez nada com aquilo. O gerente americano médio não foi capaz de compreender, mesmo que a Toyota estivesse aumentando rapidamente a sua fatia de mercado usando aquela abordagem. Na Easel, não tínhamos nada a perder, por isso decidimos tentar, embora o artigo se concentrasse na fabricação de produtos e não no desenvolvimento de softwares. Pensei que a ideia deles abordava algo fundamental, um processo descritivo de como os seres humanos trabalhavam melhor juntos em qualquer empreendimento. Ela fluía por todas as outras experiências que eu já conduzira

desde o meu primeiro trabalho no setor privado na MidContinent.

Esse foi o nascimento formal do Scrum. Entregamos o produto na Easel dentro do prazo de seis meses, abaixo do orçamento, e com menos *bugs* do que qualquer outro produto anterior.

Fiquei tão animado com as possibilidades dessa nova forma de gerenciamento de projetos que todo o meu trabalho futuro se concentrou no refinamento do Scrum para as empresas. Em 1995, apresentei um artigo com Ken Schwaber, intitulado “SCRUM Development Process” [Processo de desenvolvimento SCRUM], o qual sistematizava tais práticas em uma conferência de pesquisa da Association for Computing Machinery. Desde então, desistimos de usar o nome em maiúsculas e adaptamos um pouco mais a ideia, mas os princípios fundamentais continuam os mesmos — e as empresas que adotam o processo costumam ver benefícios imediatos.[10]

Inspeção e adaptação

As equipes Scrum que trabalham bem conseguem obter o que chamamos de “hiperprodutividade”; é difícil acreditar, mas costumamos ver uma melhora entre 300% e 400% na produtividade entre as que executam bem o Scrum. As que melhor se desenvolvem atingem um aumento de até 800% na produtividade, e repetem o mesmo sucesso várias vezes. Além disso, elas também conseguem mais que dobrar a *qualidade* do trabalho.

Então, como construir autonomia, transcendência e fecundação cruzada em uma equipe Scrum e, com tal combinação, conseguir obter hiperprodutividade? Bem, este livro é exatamente sobre isso, mas vou delinear a estrutura básica agora.

Uma vez que o Scrum se baseia em técnicas utilizadas na indústria japonesa, vale a pena conhecer um pouco sobre onde os japoneses a aprenderam. Ironicamente, muitas delas foram ensinadas por um americano: W. Edwards Deming, que trabalhava para o General Douglas MacArthur durante a ocupação americana no Japão depois da Segunda Guerra Mundial. A abordagem de MacArthur para reconstruir a economia consistia em demitir a maioria dos altos gerentes nas empresas japonesas, promover os gerentes de produção e importar dos Estados Unidos especialistas em operações de negócios, como Deming.

A influência de Deming na indústria japonesa foi incrível. Ele treinou

centenas de engenheiros no que chamamos de “controle de processo estatístico” baseado em uma ideia simples de medir exatamente o que estava sendo feito, sua qualidade e lutar por um “aprimoramento contínuo”. Não basta melhorar uma vez, é preciso fazê-lo constantemente. Sempre procure algo que possa ser aprimorado. Nunca, jamais, conforme-se com o lugar onde está. A maneira como você atinge seus objetivos é sempre experimentar, até que consiga perceber se atingiu um ponto mais alto. Se eu tentar esse método, o resultado será melhor? E quanto a esse outro? E se eu mudar esse pequeno detalhe?

Deming fez um discurso famoso para líderes empresariais japoneses em 1950. Na audiência, havia pessoas como Akio Morita, o fundador da Sony. No discurso, Deming declarou:

> [...] não importa o quanto seus técnicos sejam excelentes, vocês, que são líderes, devem buscar sempre avançar no aprimoramento da qualidade e na uniformidade do produto, para que os seus técnicos consigam fazer melhorias. Portanto, o primeiro passo pertence à gerência. Primeiro, os técnicos da sua empresa e suas fábricas precisam saber que vocês se dedicam com fervor ao avanço da uniformidade e qualidade dos produtos e com um senso de responsabilidade em relação à qualidade do produto. Nada acontecerá com isso se você só falar sobre o assunto. As ações são importantes.[11]

E o método é agir e aplicar o conceito que tornou Deming tão famoso, ou seja, o ciclo PDCA (**P**lan [planeje], **D**o [faça], **C**heck [verifique], **A**ct [aja]). É possível aplicá-lo à produção de qualquer produto, seja um carro, um *video game* ou, até, um avião de papel.

Quando treino as pessoas para usar o Scrum, é este o exemplo que uso: aviões de papel. Divido-as em equipes e digo que o objetivo é construir o maior número de aviões de papel que conseguirem e que sejam capazes de voar até o outro lado da sala. Haverá três papéis a serem desempenhados em cada equipe: uma pessoa verificará quantos dos aviões feitos conseguem voar de verdade; outra trabalhará na produção, mas também estará atenta ao processo em si e buscará maneiras de fazer com que a equipe construa aviões melhores e acelere o trabalho. Os demais participantes deverão se concentrar em fazer o maior número de aviões capazes de atravessar a sala voando no tempo definido.

Então, eu conto que faremos três ciclos de seis minutos para o processo de construção. As equipes têm um minuto de cada ciclo para planejar (P) como vão construir o avião; três minutos para fazer (D) — construir e testar quantos aviões realmente voam; e, por fim, terão dois minutos para verificar (C). Nessa

fase a equipe busca meios para melhorar o processo de construção. O que deu certo? O que deu errado? O design deve ser alterado? Como o processo de construção pode ser melhorado? Então, eles vão agir (A). No universo de Deming, “agir” significa mudar o modo de trabalhar, com base em resultados e informações ambientais reais. É a mesma estratégia usada pelo robô de Brooks.

Realize esse ciclo três vezes, esteja você construindo aviões de papel ou espaçonaves de verdade, você vai melhorar de forma significativa (em uma razão de duas a três vezes mais rápido e com pelo menos o dobro da qualidade). Esse ciclo PDCA, uma ideia radical quando Deming a apresentou para os japoneses, foi o que fez a Toyota se tornar a empresa automobilística número 1 do mundo. E é dessa forma que qualquer tipo de produção “enxuta” (o termo americano para o uso dos conceitos do Sistema Toyota de Produção), ou desenvolvimento Scrum de produção, é feito.

Mude ou morra

O motivo por que um novo modo de fazer as coisas era imperativo e por que uma parte tão grande de empresas o adotou pode ser explicado, em parte, pelo estado deplorável em que o desenvolvimento de software se encontrava. Os projetos quase sempre estavam atrasados, acima do orçamento e, em geral, simplesmente não funcionavam. E isso não acontecia porque as pessoas eram burras ou gananciosas, mas sim pelo modo como pensavam no trabalho. Elas insistiam no método em cascata, insistiam que tudo podia ser planejado com antecedência e, até mesmo, que as coisas *não mudariam* durante um projeto de multicamadas. Isso é a mais completa loucura.

Aprendi isso em primeira mão na BellSouth, quando visitei a empresa como consultor há alguns anos. Eles possuíam engenheiros de primeira linha, muitos da famosa Bell Labs., que executavam projetos em cascatas de forma perfeita e entravam em concorrência para projetos enormes que chegavam a valores entre US$ 10 milhões e US$ 20 milhões. Esses engenheiros reuniam todos os requisitos do cliente, então sumiam por 18 meses e entregavam no prazo e dentro do orçamento exatamente o que o cliente pedira. Essa era uma das pouquíssimas empresas em todo o mundo que conseguia fazer isso. O problema era que, quando entregavam o projeto, o cliente não queria mais o que dissera antes, ou seja, as circunstâncias tinham mudado. Isso significa que os ciclos de negócios estavam cada vez menores, e os clientes exigiam serviços

mais ágeis.

Fui chamado com o propósito de ajudar a BellSouth a descobrir o que estavam fazendo de errado. Logo percebi que era exatamente o modo como trabalhavam. Isso pode ser difícil de ouvir quando você tem a impressão de que estava fazendo tudo certo. Então, um dia eu me coloquei diante de uma sala com 150 engenheiros da BellSouth e disse a eles que, a menos que mudassem para um modelo diferente, mais responsivo ao cliente, eles não durariam muito. A plateia era difícil, formada por homens e mulheres muito inteligentes, mas acreditavam que as minhas ideias não passavam de mais um modismo gerencial. Eu não consegui convencê-los, então, simplesmente dei de ombros e fui embora deixando um aviso final de “Mudem ou morram”. Como você já deve ter notado, a BellSouth não está mais entre nós.

Shu Ha Ri

O Scrum tem suas raízes no pensamento e na prática japoneses. Quando viajei recentemente para encontrar o professor Ikujiro Nonaka, ele deixou claro para mim que, no Japão, o Scrum não é visto como o último modismo de trabalho. Eles o consideram um modo de fazer, de ser, um modo de vida. Quando ministro aulas, costumo falar sobre a minha experiência pessoal no estudo da arte marcial japonesa *aikido* durante anos.

O Scrum, assim como o *aikido*, ou até mesmo como o tango, é algo que você só consegue aprender fazendo, e no qual seu corpo, sua mente e seu espírito ficam alinhados por meio da prática e do aprimoramento constantes.

Nas artes marciais aprende-se um conceito chamado *Shu Ha Ri*, que se refere a diferentes níveis de habilidade. No estado *Shu*, você conhece todas as regras e formas, as quais deve repetir, como os passos de uma dança, de modo que seu corpo os absorva. Resumindo, você nunca se distrai.

No estado *Ha*, quando as formas já estiverem dominadas, você poderá fazer inovações, como incluir um gingado novo no passo da dança, por exemplo.

No estado *Ri* você será capaz de descartar formas, já terá dominado totalmente a prática e poderá ser livremente criativo, porque o significado do *aikido*, ou do tango, já está tão profundamente arraigado em você, que qualquer passo que fizer expressará a sua essência.

O Scrum é bem parecido com isso. Exige prática e atenção, mas também um esforço contínuo para chegar a um novo estado no qual as coisas apenas fluem

para que todo o resto aconteça. Se você já assistiu a um grande dançarino ou ginasta, sabe que seus movimentos parecem ser realizados quase sem esforço, como se não estivessem fazendo nada, apenas existindo. Melhor, a impressão que se tem é que eles não poderiam ser nada além daquilo que são naquele momento. Eu passei por uma experiência assim quando um pequeno mestre de *aikido* me lançou facilmente no ar e de uma forma que fez com que eu caísse gentilmente no tatame, como seu eu fosse um bebê sendo colocado, com carinho, em um berço.

É nesse ponto que você precisa chegar com o Scrum. Esse é o estado que eu quero que todos cheguem em suas vidas. O trabalho não precisa ser um saco. Pode fluir, pode ser uma expressão de alegria, um alinhamento em direção a um objetivo superior. Pode ser melhor. Podemos ser maravilhosos! Só precisamos praticar.

No decorrer do livro, cada capítulo se concentrará em um aspecto particular do Scrum. Esse mergulho profundo é necessário para explicar o raciocínio por trás dos conceitos e o motivo por que o Scrum é estruturado dessa forma. Os fundamentos básicos do Scrum (uma descrição definitiva do método) são descritos no Apêndice, mas ele só explica o que você deve fazer. Se você me acompanhar pelos capítulos a seguir, eu vou explicar o *porquê*.

PONTOS PRINCIPAIS

Hesitação é a morte. Observe, avalie, decida, aja. Saiba onde está, avalie suas opções, tome uma decisão e aja!

Procure respostas. Sistemas adaptativos complexos seguem algumas regras simples, que aprendem a partir do ambiente em que se encontram.

Grandes equipes. São multifuncionais, autônomas, capazes de tomar decisões e motivadas por um objetivo transcendente.

Não adivinhe. Planeje (*Plan — P*), Faça (Do — D), Verifique (*Check — C*), Aja (*Act — A*): planeje o que vai fazer. Faça. Verifique se o resultado é o que desejava. Aja de acordo com as informações e faça as coisas de outra maneira. Repita em ciclos regulares e, ao fazer isso, obtenha um aprimoramento contínuo.

Shu Ha Ri. Primeiro, aprenda as regras e as formas e, uma vez que as dominar, faça inovações. Por fim, em um estado elevado de domínio, descarte as formas e apenas seja — com todo o aprendizado internalizado as decisões são tomadas de forma quase inconsciente.

7 Painel transparente que fica entre o piloto e a frente da aeronave e cuja função é permitir que o piloto se concentre mais no seu trabalho, sem precisar baixar a cabeça para olhar para o painel de instrumentos. (N.T.)

8 Gamo é um animal semelhante ao cervo, que vive em regiões no norte da África, sul da Europa e na Ásia.

(N.T.)

9 H. Takeuchi e I. Nonaka, The New New Product Development Game, *Harvard Business Review,* Jan./Fev., 1986, pp. 285-305.

10 K. Schwaber, “Scrum Development Process” in *OOPSLA Business Object Design and Implementation Workshop*, J. Sutherland et al., (eds.), Londres, Springer, 1997.

11 W. Deming, *To Management*. Discurso feito no centro de conferências Mt. Hakone, Japão, 1950.

CAPÍTULO 3

Equipes

As equipes são responsáveis por fazer as coisas no mundo do trabalho. Existem equipes que fazem carros, atendem telefone, realizam cirurgias, programam computadores, dão notícias e invadem apartamentos ocupados por terroristas. É claro que existem artesãos que trabalham sozinhos, mas as equipes são responsáveis por fazer o mundo girar. E é nelas que o Scrum se baseia.

Todo mundo sabe disso, mas, no mundo dos negócios, é comum que a gente acabe se concentrando unicamente em indivíduos, mesmo que a produção seja um esforço coletivo. Pense em bônus por desempenho ou promoções ou em contratações: o foco está sempre em atores individuais, em vez de na equipe. E isso acaba sendo um grande erro.

Os gerentes tendem a focar no indivíduo porque, intuitivamente, isso faz sentido. Você quer as pessoas mais competentes, mas elas são diferentes; então, se você se concentrar em conseguir os melhores profissionais, terá os melhores resultados, certo? Bem, não é tão simples assim.

Consideremos, por exemplo, o processo utilizado para avaliar os alunos em uma disciplina. Em um curso de programação de computadores ministrado pelo professor Stanley Eisenstat, em Yale, há uma matéria famosa por sua dificuldade: a CS 323. Certa vez, os alunos começaram a reclamar sobre o tempo que levavam para fazer cada um dos trabalhos, e então o professor, em vez de tornar os trabalhos mais simples, começou a rastrear quanto tempo cada aluno precisava para concluí-los. Joel Spolsky, que fizera o curso de Eisenstat na década de 1980 e agora possuía a própria empresa de software, comparou os dados com as notas que os alunos tiravam. Ele queria saber se havia qualquer correlação entre o tempo dedicado ao projeto e a nota que o aluno recebia. O interessante é que não há. Algumas pessoas são rápidas e conseguem um 10, enquanto outras trabalham de forma meticulosa e tiram a mesma nota. A única diferença é o tempo que cada um dedicou à tarefa. Portanto, que aplicação esse

dado pode ter no mundo dos negócios?

Bem, se você for um gestor, acredito que não vai querer contratar apenas os alunos que alcançaram a nota máxima, mas aqueles que o conseguiram em menor tempo. No estudo de Yale, os alunos mais rápidos ultrapassaram seus colegas mais lentos em uma razão de 10:1. Eles eram dez vezes mais rápidos e conseguiram a mesma excelente nota. Ser dez vezes mais rápido é bem significativo, certo? Então, a impressão que se tem é que as empresas deveriam se concentrar em contratar as pessoas mais rápidas e livrar-se das que trabalham mais devagar. Mas essa somente *parece* ser a melhor abordagem para aumentar a produtividade, porque outros fatores podem ser ainda mais cruciais.

Se você olhar para equipes em vez de indivíduos, notará algo interessante. Existem estudos que avaliaram cerca de 3.800 projetos diferentes, variando desde trabalhos realizados em firmas de contabilidade, passando por empresas de desenvolvimento de software para navios de batalha, até projetos tecnológicos na IBM. Os analistas não consideraram os dados de desempenho individual, mas sim os de desempenho da equipe. E, quando você examina como essas equipes se saíram, se depara com algo surpreendente. Se a melhor equipe conseguia realizar uma tarefa em uma semana, quanto tempo você acha que a pior equipe levou? Talvez você chute a mesma razão observada em Yale — 10:1 (ou seja, a equipe mais lenta levou mais de dois meses para conseguir o que a equipe mais rápida realizou em uma semana). A resposta verdadeira, porém, é que há uma diferença muito maior no desempenho coletivo do que no desempenho individual. Na verdade, a pior equipe não levou dez semanas para fazer o que a melhor fez em uma semana: ela levou *duas mil* semanas para fazer o mesmo trabalho. Essa é a grande diferença entre o melhor e o pior. Então, onde você deve concentrar a sua atenção? No nível do indivíduo, onde talvez consiga obter uma melhora de dez vezes, se conseguir transformar, em um passe de mágica, todos os seus funcionários em gênios? Ou no nível da equipe, aumentando a produtividade a uma enorme magnitude, mesmo que você apenas torne as suas piores equipes em... equipes medíocres? É claro que se você tiver como meta a mediocridade, certamente vai conseguir alcançá-la. Mas e se você conseguir tornar todas as suas equipes excelentes?

Em determinadas épocas, locais e com determinados grupos pequenos de pessoas, tudo se torna possível. Mesmo que nunca tenha feito parte de uma equipe assim, você já deve ter visto alguma em ação. Você ouve história sobre elas; são contadas lendas sobre o que conseguem fazer. Eu cresci em Boston e

moro lá agora, então, alguns dos grandes times que me vêm à mente são os Celtics da década de 1980 ou o New England Patriots da era de Tom Brady. Quando esses times entravam em campo era como se estivessem jogando um jogo diferente dos outros. Lances que antes pareciam impossíveis de repente se tornavam parte da estratégia da partida. Era como se um estado de graça tivesse descido sobre os jogadores e, por um instante, não fosse possível errar. Larry Bird atravessando o campo e passado a bola sem olhar para uma direção onde parecia não haver ninguém. Mas, quando se imagina que a bola vai sair do campo, Kevin McHale simplesmente *aparece, e* exatamente no lugar onde deveria estar. Então, ele faz um lance lateral — novamente parecendo nem ver o que está fazendo — e Robert Parish está na posição perfeita para arrematar a jogada. Aquele alinhamento absoluto de objetivo e confiança é o que cria a grandiosidade.

Nós já vimos esses times. Alguns de nós já até tiveram a sorte de estar em uma equipe assim — ou em *mais* de uma — na vida. Quando eu estava desenvolvendo o Scrum, olhei para o que as equipes de alto desempenho possuíam e as outras não. Eu me perguntava por que algumas delas mudavam o mundo, e outras apenas se atolavam na mediocridade. Quais são os elementos comuns que as equipes realmente notáveis possuem? E, o mais importante, será que é possível reproduzi-los?

Acontece que a resposta é que sim, isso é possível.

No artigo original intitulado “The New New Product Development Game” [“O novo jogo para o desenvolvimento de novos produtos”], que descrevia o que se tornou o Scrum, os professores Takeuchi e Nonaka descreveram as características das equipes que viram nas melhores empresas do mundo:

1. **Transcendência:** elas têm um senso de propósito além do comum. O objetivo percebido por todos permite que eles transformem o ordinário em extraordinário. De uma forma bastante verdadeira, a decisão de não estar na média, mas ser grandioso, muda o modo como eles se enxergam e o que são capazes de fazer.
2. **Autonomia:** as equipes se auto-organizam e se autogerenciam; têm o poder de tomar as próprias decisões sobre como fazer o próprio trabalho e têm o poder de fazer com que tais decisões sejam acatadas.
3. **Interfuncionalidade:** as equipes possuem todas as habilidades necessárias para concluir o projeto: planejamento, projeção, produção, vendas e distribuição. E tais habilidades alimentam e reforçam umas as outras. Nas palavras de um membro da equipe que projetou uma nova câmera revolucionária da Canon: “Quando todos os membros da equipe se encontram em uma sala grande, a informação de alguém se torna sua, mesmo sem você perceber. Então, você começa a pensar no que é melhor fazer

primeiro e o que é melhor fazer depois pelo grupo como um todo, e não apenas de acordo com o que é melhor para você”.[12]

Então, como criar uma equipe que determine um objetivo superior, seja capaz de se organizar e consiga, constantemente, alimentar as habilidades de cada um dos seus membros? Dediquei muito tempo refletindo sobre isso. Afinal, você não pode simplesmente gritar com as pessoas dizendo que elas têm de se organizar melhor e ser transcendentes; isso é algo que precisa vir de dentro. A imposição vai matar o que você está tentando fazer. Será que existe algum conjunto de regras mais simples que estimule essa mágica?

A longa fileira cinza

Pensei em quando fiz parte de uma dessas equipes mágicas. Foi no início da década de 1960, quando eu era um cadete na Academia Militar dos Estados Unidos, mais conhecida como West Point. No meu último ano lá, fui escolhido como oficial de treinamento do meu regimento de cadetes, a L2, ou “Loose Deuce”.

Em 1963, havia 24 regimentos em West Point. A1 até M1 e A2 até M2. Três vezes por semana, esses regimentos tomavam o campo de paradas com rifles, espadas e todo o tipo de armas, faixas brancas e equipamento cuidadosamente acomodado. Tais formações em parada constituíam uma competição na academia havia mais de duzentos anos. Em 1963, a Loose Deuce era a última colocada daquela classificação havia mais de um século.

O oficial de treinamento não possui qualquer poder direto. Não faz parte da estrutura de comando do regimento. Ninguém é subordinado a ele. Ninguém tem de obedecê-lo. Mas, depois de cada parada, esses oficiais se reuniam e avaliavam cada regimento de acordo com diversos critérios. Como oficial de treinamento da Loose Deuce, decidi que o que eu poderia fazer era tornar as coisas mais transparentes. Fiz diagramas coloridos mostrando o que saíra certo e o que saíra errado e os colocava no quartel, onde todos do meu regimento poderiam vê-los todos os dias.

A princípio, as críticas eram simples. *Charlie enfiou a espada na lama. Jim não se virou em sincronia com o grupo. A saudação de Dave foi desleixada.* Não havia qualquer punição ou culpa; eram apenas fatos que foram expostos pelos outros oficiais de treinamento e levantados durante as avaliações. Ainda

assim, aqueles eram os motivos por que a L2 ficava sempre em último lugar.

Em algumas semanas, os cadetes aguçaram o jogo, e as avaliações ruins agora apontavam para o comandante do regimento. As suas ordens não eram claras o suficiente; assim como a cronometragem não era precisa. Não foi surpresa nenhuma eu receber uma advertência por ter criticado o comandante, mas respondi simplesmente: “As avaliações foram feitas. Eu só estou mostrando quais foram elas. As outras tropas conseguiram se organizar. *Você* agora é o problema. Você quer corrigir isso? Ou quer ser o pior para sempre?”. Algumas semanas depois, a L2 era o regimento número um em West Point.

O cadete com mais honrarias na história de West Point foi o General Douglas MacArthur. Ele recebera a mais alta classificação do que qualquer outro cadete graduado ali, e fora comandante-chefe nas duas guerras mundiais. Como um general cinco estrelas e ganhador de uma Medalha de Honra, ele tinha uma ligação especial com o Corpo de Cadetes.

Um ano antes de eu começar a treinar meu regimento, em maio de 1962, ele fizera seu último discurso em West Point. É necessário imaginar a cena de maneira correta para compreender o impacto total. Havia três mil homens vestindo o uniforme cinza dos cadetes, sentados no imenso salão de pedra com grandes colunas e lustres gigantescos pendurados no teto alto. A cerca de nove metros de altura, em uma das paredes, havia uma plataforma com vista para todo o salão. O General MacArthur, já frágil, estava naquela plataforma e fez o discurso que hoje é conhecido como “Long Gray Line” [Longa Fileira Cinza]:

> Vocês são a levedura que une todo o nosso sistema nacional de defesa. Das suas fileiras, surgem os grandes capitães que seguram o destino da nação nas mãos, no momento em que soa o alarme da guerra.
>
> A longa fileira cinza nunca fracassou. Se isso acontecesse, um milhão de espíritos em verde-oliva, cáqui, azul e cinza se levantariam de suas cruzes brancas e bradariam estas palavras mágicas: Dever, Honra, Pátria.[13]

Ao ouvir aquilo, lembro-me de que parecia que as legiões de tais espíritos tinham se erguido atrás de MacArthur enquanto ele deixava a corporação com sua última obrigação cumprida. E três mil homens, treinados para a guerra, e cujas lágrimas não desciam fácil, começaram a chorar.

> Nos meus sonhos, ouço o som de armas, o som estridente da fuzilaria, o murmúrio triste e estranho do campo de batalha. Mas, na noite da minha lembrança, eu volto a West Point, onde ouço os ecos

repetidos: Dever, Honra, Pátria.

Hoje marco a minha última chamada para vocês. Mas eu quero que saibam que quando eu cruzar o rio, os meus últimos pensamentos conscientes serão sobre o Exército e o Exército e o Exército.

Até hoje, todo cadete na academia é obrigado a decorar tudo isso, linha por linha, palavra por palavra, para poder se formar. Esse discurso se tornou o guia espiritual da corporação de cadetes e do exército norte-americano como um todo: Dever, Honra, Pátria.

Quase um ano depois, o General MacArthur faleceu. Um regimento foi escolhido para marchar no seu funeral. Ao som lento e ritmado dos tambores, a Loose Deuce, o mesmo regimento que fora o pior por mais de cem anos, marchou atrás do caixão que carregava um dos maiores generais dos Estados Unidos.

Alguns meses após do funeral, marchei pela última vez com a Loose Deuce na minha formatura. Todos os 24 regimentos participaram, mas o L2, por causa da ordem alfabética, foi o penúltimo a marchar. Depois da cerimônia, meu futuro sogro me perguntou: “Aquele regimento, o penúltimo a entrar. Eles foram diferentes de todos. Os outros marchavam, eles pareciam flutuar. Quem são eles?”.

“Aquele é o meu regimento”, respondi. “Aqueles homens participaram do enterro do General MacArthur”.

Meu regimento atingiu o nível de transcendência.

Scrum em período de rebelião

Em geral, quando as pessoas falam sobre grandes equipes ou times, eles *só* falam sobre aquele senso transcendente de objetivo. No entanto, embora esse seja um elemento crítico, é apenas uma das pernas de um banco de três pernas. Tão crítica quanto, mas talvez menos celebrada, é a liberdade de fazer o seu trabalho do modo que você acredita ser o melhor, ou seja, ter autonomia. Em todas as grandes equipes, são os seus membros que decidem de que forma vão atingir os objetivos definidos pelos líderes da organização.

A praça Tahrir se tornou sinônimo da revolução egípcia e as lutas contínuas no país, mas, antes de janeiro de 2011, ela não passava de outra rotatória suja e engarrafada no centro do Cairo. Ao norte, o Museu Egípcio se ergue e, ao sul, os altos muros da Universidade Americana no Cairo e o icônico Muqawama, o

palácio do governo. A sede do Partido Democrático Nacional do ditador egípcio Hosni Mubarak ficava a oeste, assim como a sede da Liga Árabe. Por mais estranho que pareça, na extremidade leste da praça, entre todas as coisas, havia um restaurante da franquia KFC, que logo se tornou o plano de fundo de protestantes que atiram pedras e enfrentam a polícia.

No final de janeiro de 2011, um grupo pequeno de protestantes decidiu usar aquela rotatória do trânsito para protestar contra o assassinato brutal de um jovem chamado Khaled Said, cometido pela polícia egípcia. O que teria sido apenas mais um pequeno protesto contra um regime repressivo teve enorme sucesso, acendendo a imaginação dos egípcios e atraindo, por fim, milhões de pessoas para a praça. No decorrer do mês seguinte, o impensável aconteceu. Apenas pelo fato de as pessoas terem se reunido para dizer não, um dos regimes ditatoriais mais antigos e mais poderosos do Oriente Médio caiu. As pessoas se reuniam dia após dia, noite após noite, enchendo a praça e criando um país alternativo, no qual o ditador Hosni Mubarak não governava, e a população podia expressar sua opinião. Eles mudaram o mundo deles.

Para os jornalistas, foi uma grande história com significado histórico. Entre os que foram para o Cairo estava a equipe da National Public Radio (NPR), uma das primeiras instalações de jornalismo dos Estados Unidos. Pegos de calça-curta no princípio, os produtores e repórteres da NPR perderam prazos, não cobriram algumas histórias e lutavam para atender as exigências dos executivos em Washington.

J. J. Sutherland, meu filho, foi chamado para resolver as coisas. Correspondente de guerra e produtor de longa data, foi enviado ao Cairo para fazer a cobertura funcionar. A história era importante demais para não entrar no ar todos os dias, em todos os programas, a cada hora. J. J. foi deixado em um país no qual os aeroportos foram fechados, os estrangeiros tentavam desesperadamente fugir, e as redes de celular e internet foram desligadas. Ele era o produtor sênior no local, mas, assim como um oficial de treinamento em West Point, um produtor da NPR funciona como um facilitador e um organizador — uma pessoa para ajudar e estimular, em vez de um líder ou gerente típico. O trabalho de J. J. era ajudar a equipe a fazer o melhor trabalho que poderiam. Não era *mandar* as pessoas fazerem as coisas, mas dar aquilo de que precisavam. As ordens da gerência eram contar o fato e entrar no ar várias vezes ao dia, e a equipe no local descobriu como fazer para superar aquele desafio, decidindo quais histórias contar e como conta-lás, usando o rádio como mídia.

Por mais estranho que pareça, o fato de ser tão difícil se comunicar com os executivos em Washington foi precisamente o que fez a equipe alcançar tanto sucesso. Da forma mais verdadeira possível, eles estavam sozinhos. Com a impossibilidade de receberem uma supervisão direta e constante de Washington, e os eventos acontecerem tão rápido, a equipe teve de se organizar. Um dos conceitos-chave do Scrum é que os membros da equipe decidem, *eles mesmos*, como irão executar o trabalho. A responsabilidade da direção da empresa é estabelecer os objetivos estratégicos, e o trabalho da equipe é decidir como atingi-los. No Cairo, não havia como alguém que *não* estivesse lá, no território, acompanhar o que estava acontecendo. Quase todos os dias, a equipe da NPR relatava uma série de histórias para o dia seguinte que se tornariam instantaneamente obsoletas devido à velocidade do desdobramento dos eventos. Aconteceria um confronto grande na praça, um discurso ou uma batalha, e todo o trabalho da equipe teria sido em vão. De repente, eles estavam lutando para conseguir colocar no ar alguma notícia importante em tempo hábil.

E conseguiram isso usando o Scrum. Os prazos definidos determinavam que eles deviam se apresentar a cada 12 horas, nos programas *Morning Edition* e *All Things Considered.* A cada ciclo, J. J. conversava com a equipe e fazia três perguntas simples: “o que você fez desde a última vez que conversamos?”; “O que você vai fazer antes de voltarmos a conversar?”; “O que está atrapalhando o seu trabalho?”. O fato de ele fazer essas três perguntas, que é um dos rituais regulares do Scrum, forçava os correspondentes a falarem e a dividirem informações uns com os outros. E o principal trabalho de J. J., sendo um verdadeiro Mestre Scrum, era certificar-se de que tudo que estivesse atrapalhando a equipe em uma reunião estivesse resolvido na seguinte. O obstáculo poderia ser qualquer coisa — desde lidar com a burocracia egípcia, até conseguir um quarto de hotel seguro ou encontrar motoristas e tradutores para tirá-los da custódia da temida polícia secreta do Egito, a *Mukhabarat*.

Como tudo funcionou? Bem, aquilo que começara como um caos, com disputas pessoais e fracasso de dar as notícias de forma rápida, tornou-se uma máquina bem azeitada que a direção nem precisava gerenciar. Ao contrário, os membros da equipe se autogerenciavam. Nas semanas que se seguiram, a pequena equipe da NPR no Cairo fez a maior cobertura que qualquer um achava possível. E tudo com qualidade superior à da concorrência, resultando, no final, no recebimento de vários prêmios. Aquilo foi um feito que não teria sido possível se a equipe não estivesse imbuída com um senso de propósito

(contar uma das maiores histórias de suas carreiras) e não contassem com autonomia (a capacidade de decidirem por si mesmos como produzir o maior número de matérias para cobrir a história).

Agora, o Scrum é usado em todos os departamentos na NPR, desde o web design, passando pelo jornalismo e chegando à criação de novos programas de rádio. As equipes do Chicago Tribune, do New York Times, do Washington Post e do ProPublica também estão usando o Scrum. Quando os prazos são apertados, eles precisam de velocidade.

Uma equipe para concluir o trabalho

A terceira perna do banco para grandes equipes é que todas elas tenham as habilidades necessárias para concluir o trabalho. Em uma estrutura de organização clássica, você talvez encontre a equipe de planejamentos, seguida pela equipe de construção, equipe de testes, de produção e de entrega. Cada equipe ao longo do caminho precisa concluir sua parte da ação antes de o projeto poder avançar para o próximo passo. Nenhuma equipe individual pode realmente finalizar o produto.

O exemplo clássico disso é o processo *phase-gate* [fase/pontos de decisão] da NASA. Foi essa abordagem que conduziu o programa do ônibus espacial e outros projetos desenvolvidos nas décadas de 1960, 1970 e 1980. É tudo bem diferente agora, então vou explicar como o antigo processo funcionava. Primeiro, temos a “fase” de descoberta, na qual as pessoas decidem o que vão conseguir — talvez seja a construção de um foguete que chegue à Lua. Um bando de estrategistas se reúne em uma sala e fica pensando em como aquilo funcionaria. Temos aí um “ponto de decisão”, quando um gestor ou um grupo de gestores precisa avaliar se vale a pena seguir com o projeto. Então, chega a fase de definição do escopo, na qual a “equipe de requisitos” decide o que o produto vai fazer. Assim, é hora de decidir mais uma vez; acontecem outras reuniões e, em seguida, toda a imensa quantidade de documentos gerados segue para a nova fase: construir um plano de negócios e de projeto. Depois, todos os planos são levados para mais algumas reuniões de aprovação e, então, seguem para *outra* fase, a de desenvolvimento, na qual o produto será realmente construído. Mais algumas reuniões e documentos, e o produto é entregue para outro grupo na fase seguinte, a fase de testes. Aquelas pessoas nunca tinham visto o produto antes, mas vão testá-lo, aprová-lo e levá-lo a outro ponto de

decisão, ou a mais um conjunto de reuniões intermináveis, com mais um monte de documentos que ninguém leu. E, então, somente então, o produto passa para o *sexto* grupo de pessoas que irá realmente lançá-lo. Escrever sobre isso já é exaustivo. E essa era a maneira como a NASA construía as coisas.

No início da década de 1980, executivos de Fujitsu foram para os Estados Unidos estudar como a famosa agência espacial funcionava. Ao implantarem os mesmos procedimentos quando voltaram para o Japão, viram imediatamente a qualidade cair, a taxa de fracasso subir, e sua capacidade de entrega descer ralo abaixo. Eles foram rápidos em abandonar o processo, afirmando que era bastante provável que ele produzisse um erro catastrófico. A Comissão Rogers que examinou o desastre da *Challenger* em 1986 concordou. Nas famosas palavras escritas pelo físico Richard Feynman no apêndice F do relatório da Comissão: "Parece que, seja qual for o objetivo, seja para consumo interno ou externo, o gerenciamento da NASA exagera a confiabilidade de seus produtos chegando ao ponto da fantasia".[14]

O fato é que, quando você olha para as melhores equipes — como as que existiam na Toyota ou na 3M quando Takeuchi e Nonaka escreveram seu artigo, ou as do Google, da Salesforce.com ou as da Amazon hoje —, não existe essa separação de papéis. Cada equipe funciona com todas as pessoas participando de todos os processos, do início ao fim.

Nicola Dourambeis é responsável pelas práticas "ágeis" na Salesforce.com. Ela é responsável por cerca de duzentas equipes Scrum na empresa que figura rotineiramente nas listas das "100 melhores empresas para se trabalhar" da *Fortune* e na lista das "Empresas mais inovadoras do mundo" da *Forbes*. Nicola diz que considera o Scrum o "ingrediente secreto" deles. "Quando ainda éramos uma start-up", diz ela, "fazíamos um lançamento importante três ou quatro vezes por ano. Conforme crescemos e aumentamos, gerenciando projetos usando o típico modelo em cascata, o número caiu para uma vez por ano em 2005-2006. Aquilo tinha que mudar. Então, começamos a usar o Scrum. Desde então, fazemos três lançamentos por ano. Não há tantas grandes empresas que consigam fazer isso".

O que Dourambeis busca nas equipes é a diversidade — de conjunto de habilidade, pensamento e experiência. Ela quer equipes que sejam altruístas e autônomas, mas também precisa que sejam interfuncionais. Equipes que consigam fazer com que o projeto inteiro seja concluído.

Um dos testes para ela saber se a equipe está no caminho certo é quando pergunta, digamos, para um engenheiro de rede: "Em que equipe você está?".

Se a pessoa responder mencionando o produto em que estão trabalhando (automação ou integração, por exemplo) em vez de sua especialidade (ou seja, engenharia de rede), ela assente em aprovação. Quando um especialista se identifica mais com a sua especialidade do que com o produto que está desenvolvendo, Nicola sabe que ainda tem trabalho a fazer.

O Scrum na guerra

Um dos exemplos mais drásticos de equipe interfuncional vem das forças armadas. A divisão chamada American Special Operations Forces (SOF) [Forças Americanas de Operações Especiais] agia exatamente como descrevo a seguir. Uma típica "equipe classe A" das forças armadas especiais é formada por 12 pessoas. Um líder, que é um comandante oficial, um primeiro e um segundo sargentos (que conduz a equipe nas operações de rotina), um sargento de inteligência, e dois terceiros sargentos de cada especialidade: armamentos das forças especiais, extermínio, medicina e comunicação. Cada equipe tem todas as habilidades para realizar suas missões do início ao fim. E elas realizam treinamento cruzado entre cada conjunto de habilidades. Eles querem se certificar, por exemplo, que se ambos os médicos forem mortos, o especialista em comunicação será capaz de fazer uma bandagem no especialista em armamentos. Outra diferença entre a forma como as Forças Especiais operam e a maioria das outras unidades militares "regulares" é que eles não separam a inteligência do planejamento das operações. Não há entrega de uma equipe para outra, que é quando os erros podem ocorrer. As Forças Especiais não querem desastres como o *Challenger*. Assim, existe um fluxo de comunicação constante entre as pessoas que coletam a inteligência, as que planejam o que fazer com tais informações e aqueles que sairão pela porta.

Durante a Guerra do Iraque, as Forças Especiais mostraram que eram excelentes nas missões para matar pessoas. Eram capazes de localizar um alvo rebelde e liquidá-lo naquela mesma noite. Entre 2003 e 2007, realizaram milhares de missões bem-sucedidas com o objetivo de acabar com qualquer insurgência iraquiana, principalmente a Al-Qaeda no Iraque. Tanto em termos táticos quanto operacionais, as Forças Especiais quase sempre eram bem-sucedidas em suas missões. As equipes interfuncionais altamente treinadas estavam entre as forças mais letais que o mundo já tinha visto. Ainda assim, apesar de sua habilidade e talento, tiveram um impacto estratégico quase nulo.

Durante aqueles primeiros quatro anos de guerra, os ataques às forças americanas e aos civis iraquianos crescia quase diariamente. Durante os piores momentos da guerra, ocorriam mais de cem ataques por dia às forças americanas, e até mesmo a letalidade das Forças Especiais não conseguia lutar contra a maré. No final de 2006 e início de 2007, quase todos os comentaristas consideravam o Iraque uma causa perdida. Assim, cada morte de um soldado americano passou a ser considerada um sacrifício fútil.

Então, em 2007, o general David Petraeus liderou a operação que ficou conhecida como "Surge" [Tempestade], que consistia em colocar dezenas de milhares de tropas adicionais no país e deixá-los viver entre a população local. Essa nova estratégia teve um resultado extraordinário. Um dos motivos foi que o povo iraquiano passou a acreditar que os americanos estavam realmente do lado deles, lutando contra os rebeldes que estavam explodindo bombas nas vizinhanças e fazendo uma limpeza étnica. Outro motivo foi o fato de os militares americanos conseguirem convencer inúmeros antigos rebeldes a passar para o lado dos Estados Unidos, em um programa chamado "Sons of Iraq" [Filhos do Iraque]. Mas havia um terceiro componente estratégico, que usava algo que o jornalista Bob Woodward considerou tão revolucionário quanto a invenção do tanque de guerra ou do avião.

Essa arma não era um novo dispositivo ou *drone* não tripulado. Era o que o general Stanley McChrystal, comandante das operações conjuntas especiais na época, chamou de "guerra colaborativa", porque usava equipes interfuncionais de todo o governo norte-americano com o objetivo de atingir e acabar com as redes da Al-Qaeda. Conforme o Washington Post descreveu no dia 6 de setembro de 2008:

> A CIA fornece analistas de inteligência e material de espionagem com sensores e câmeras que podem acompanhar os alvos, veículos ou equipamentos por até 14 horas. Os especialistas forenses do FBI dissecam os dados, desde informações de telefones celulares até itens encontrados nos bolsos dos extremistas. Funcionários do Ministério da Fazenda rastreiam fundos transferidos entre extremistas e governos. Funcionários da Agência Nacional de Segurança interceptam conversas e dados de computador, e os membros da Agência de Informação Geoespacial usam equipamentos de alta tecnologia para apontar o local exato onde extremistas suspeitos estão usando telefones ou computadores.[15]

O que eles fizeram foi criar uma equipe interfuncional que possuía todas as habilidades necessárias para concluir o trabalho. Em vez de ter todos esses

especialistas em equipes separadas que raramente compartilhavam informações, todos trabalharam juntos, na mesma sala, compartilhando informações de inteligência e planejando rastrear e matar os agentes da Al-Qaeda.

Antes disso, uma organização de inteligência definiria o alvo e transferiria a operação para a equipe das Forças Especiais. Então, aquela equipe passaria qualquer informação de inteligência que obtivera para que outra equipe a analisasse. Aqueles que estavam usando esse modelo de transferência descobriram o que Fujitsu descobrira décadas antes, quando tentaram implementar o sistema de fase/ponto de decisão usado pela NASA, e um dos motivos do desenvolvimento do Scrum: sempre que há transferências entre equipes, existe a possibilidade de desastre. Conforme um artigo intitulado "Employing ISR: SOF Best Practices" ["Aplicação da pesquisa de sistemas de informação: melhores práticas das forças especiais de operação"] publicado no Joint Force Quarterly explicou:

> As equipes com membros de várias agências possibilitaram a eliminação de fendas organizacionais entre os diferentes atores da coalisão no Iraque, colocando um "olho que não pisca" sobre alvos de alto valor [...] A transferência de responsabilidades entre unidades e organizações representava uma "piscada organizacional", perdendo força e talvez possibilitando a fuga de um alvo.[16]

Compartilhar informações e recursos assim não é fácil em qualquer cenário. Já vi gestores quase terem um ataque quando seus recursos são alocados em uma equipe fora de seu controle direto. Abrir mão do microgerenciamento e do controle diário é difícil, mas fazer isso no mundo secreto da inteligência e operações especiais é ainda mais difícil — tão difícil que, apesar da eficácia, as equipes no Iraque foram rapidamente desfeitas depois que a operação "Surge" foi considerada um sucesso. Christopher Lamb e Evan Munsing escreveram um artigo fascinante intitulado "Secret Weapon: High-value Target Teams as an Organizational Innovation" ["Arma secreta: equipes para alvos de alto valor como uma inovação organizacional"], descrevendo o seguinte:

> […] assim que o quase fracasso no Iraque foi revertido, o apoio burocrático para as equipes formadas por membros de várias agências começou a cair. Em 2008, outros departamentos e agências, particularmente uma agência de inteligência não identificada, começou a remover as pessoas e parou de cooperar, por acreditar que o compartilhamento de informações e a colaboração já tinham ido longe demais.[17]

A arma mais poderosa no arsenal dos Estados Unidos, aquilo que Bob Woodward considerara tão importante quanto a invenção do tanque de guerra e do avião, foi deixada de lado em virtude de interesses departamentais provincianos e preocupações dos gerentes médios com as próprias carreiras. Já vi isso acontecer repetidamente em uma grande instituição financeira em Boston. Eles me ligam apavorados quando têm um projeto crítico nas mãos que está enfrentando problemas. Pedem para eu ensinar o Scrum para dezenas de pessoas e formar equipes capazes de resolver a emergência. Eles direcionam pessoas da organização para formarem equipes interfuncionais para lidar com a questão. E, então, fica tudo resolvido. Assim que a crise faz parte do passado, eles desmontam a equipe, enviando cada membro para seu respectivo feudo gerencial. A transparência e o compartilhamento de informações de uma equipe realmente fantástica ameaça estruturas arraigadas em segredos e cegueira. Os gestores não costumam querer que outros gestores, suas próprias equipes ou outras pessoas na estrutura organizacional de poder saibam exatamente o que estão fazendo ou o que conseguem fazer e em quanto tempo. Eles acham que manter tal conhecimento em segredo é essencial para seu poder. Em vez de se alinharem com o interesse do bem maior, eles o fazem de acordo com as próprias motivações, que, de forma geral, se resumem a ganância e ambição. Foi exatamente esse tipo de pensamento que levou ao enorme fracasso gerencial que resultou no mais recente colapso econômico. Em muitas empresas, as ações se baseiam apenas no que havia ali para o indivíduo, a curto prazo. Não se pensa no que poderia beneficiar a todos ou limitar os danos à economia global.

O tamanho importa, sim, mas não do modo como você imagina

Não é porque a interfuncionalidade pode obter grandes resultados, que você deve bancar Noé e colocar um par de cada em uma equipe. A dinâmica da equipe só funciona bem em grupos *pequenos*. A fórmula clássica são sete pessoas, podendo haver variação de duas a mais ou a menos, embora eu já tenha visto equipes de três atingirem alto nível de funcionamento. O fascinante é que os dados mostram que se você tiver mais do que nove pessoas em uma equipe, a velocidade costuma cair. Isso mesmo. Mais recursos tornam a equipe mais lenta.

No desenvolvimento de software, existe um conceito chamado “Lei de Brooks”, que Fred Brooks cunhou em 1975, no seu excelente livro *O mítico homem-mês*. Para simplificar, ele diz que “acrescentar mais gente em um projeto de software atrasado resulta em um atraso ainda maior”.[18] Isso já foi demonstrando em vários estudos. Lawrence Putnam é uma figura lendária na indústria de desenvolvimento de software e tornou sua missão de vida estudar quanto tempo as coisas demoram para ser feitas e por quê. Seu trabalho sempre demonstrou que projetos com vinte ou mais pessoas precisavam de mais esforço do que aqueles que contavam com cinco ou menos. Não era só um pouco mais, mas muito mais. Uma equipe grande levaria cinco vezes mais horas do que uma equipe pequena para realizar algo. Ele viu isso repetidas vezes e, em meados da década de 1990, decidiu tentar realizar um estudo amplo para determinar o tamanho certo de uma equipe. Ele analisou 491 projetos de médio porte em centenas de empresas diferentes; todos exigiam a criação de produtos novos ou de novas funcionalidades, e não novas versões de antigos produtos. Lawrence dividiu os projetos de acordo com o tamanho das equipes e logo notou algo interessante: assim que uma equipe ficava com mais do que oito membros, ela passava a levar muito mais tempo para realizar seu trabalho. Grupos de três a sete pessoas precisavam de 25% menos esforço do que os de nove a 12 pessoas para realizar o mesmo trabalho. Esse resultado se repetiu em centenas e centenas de projetos. Grupos muito grandes fazem *menos* parecer ser uma regra de ferro da natureza humana.

Mas por quê? Para responder a essa pergunta, temos de considerar as limitações do cérebro humano. Você já deve ter ouvido falar do estudo clássico de George Miller publicado em 1956, que mostrou que o número máximo de itens que uma pessoa normal consegue reter na memória de curto prazo é sete. Supostamente, esse é o motivo pelo qual os números de telefone são formados por sete dígitos[19]. O problema do trabalho de Miller é que a sua pesquisa foi refutada mais tarde.

Em 2001, Nelson Cowan, da Missouri University, se perguntou se aquela regra mágica de sete era realmente verdadeira e realizou um amplo estudo com todas as novas pesquisas sobre o assunto. Acontece que o número de itens que uma pessoa consegue reter na memória de curto prazo não é sete, mas quatro.[20] As pessoas costumam achar que conseguem decorar mais do que isso, usando técnicas mnemônicas ou apenas concentrando-se mais. No entanto, a pesquisa é bastante clara ao afirmar que nós só conseguimos nos lembrar de quatro “pedaços” de dados. O exemplo clássico é fornecer a alguém a seguinte série

de 12 letras *fbicbsibmirs*. As pessoas costumam lembrar de quatro letras — *a não ser* que percebam que elas podem ser "partidas" em quatro acrônimos conhecidos FBI, CBS, IBM, IRS. Se você conseguir ligar as coisas na sua memória de curto prazo fazendo associações com a sua memória de longo prazo, então conseguirá guardar mais informações. Contudo, a parte da mente que se concentra — a parte consciente — só é capaz de reter cerca de quatro itens diferentes de cada vez.

Portanto, existe um limite físico do que seu cérebro consegue reter a qualquer momento, e isso nos leva de volta a Brooks. Quando ele tentou descobrir por que aumentar o número de pessoas em um projeto fazia com que levasse mais tempo para ser concluído, descobriu dois motivos. O primeiro é o tempo que as pessoas levam para se tornar rápidas. Como era de se esperar, trazer uma nova pessoa e fazê-la se tornar rápida faz com que todos os outros fiquem mais lentos. O segundo motivo tem a ver não apenas com *como* nós pensamos, mas, quase literalmente, com *o quê* o nosso cérebro é capaz de pensar. O número de canais de comunicação aumenta drasticamente de acordo com o número de pessoas, e nosso cérebro simplesmente não consegue lidar com isso.

Se você quiser calcular o impacto do tamanho do grupo, pegue o número de pessoas em uma equipe e multiplique-o pelo "mesmo número menos um" e divida por dois. Canais de comunicação = n (n-1)/2. Então, por exemplo, se você tem uma equipe com cinco pessoas, você tem dez canais. Seis pessoas, 15 canais. Sete pessoas, 21. Oito, 28. Nove, 36. Dez, 45. O nosso cérebro simplesmente não é capaz de acompanhar esse número de pessoas de uma vez. Nós não sabemos o que cada um está fazendo. E ficamos mais lentos quando tentando descobrir.

Assim como em uma equipe das Forças Especiais, todos em uma equipe Scrum têm de saber o que os outros estão fazendo. Todo o trabalho que está em andamento, os desafios enfrentados, os progressos feitos, tudo tem de ser transparente para todos. E se as equipes ficarem grandes demais, a capacidade de todos se comunicarem de forma clara, ao mesmo tempo, fica confusa. Há informações cruzadas demais. Em geral, a equipe se divide social e funcionalmente em subequipes que começam a trabalhar com objetivos cruzados. A interfuncionalidade se perde. As reuniões que levavam minutos agora duram horas.

Não faça isso. Mantenha suas equipes pequenas.

O Mestre Scrum

Com a primeira equipe Scrum, eu exibia, regularmente, um vídeo do All Blacks se preparando para jogar. Os All Blacks, lendário time de rúgbi da Nova Zelândia, são uma equipe transcendente. Antes de cada jogo, eles realizam uma cerimônia dos guerreiros Maori de *haka*, uma dança de guerra que energiza pessoas prestes a entrar no campo de batalha. Ao assistir àquilo, você quase consegue ver a energia emanando de cada jogador e misturando-se até formar uma unidade. Com pisadas sincronizadas, palmas e cantos — movimentos ritualizados que simulam a ação de cortar a garganta dos inimigos — você assiste a homens comuns se transformarem em algo maior, algo grandioso. O time invoca o espírito guerreiro que não aceita a derrota nem o desânimo.

A equipe precisou assistir algumas vezes, mas depois aqueles programadores de computador meio fora de forma acabaram começando a falar sobre como conseguiriam agir da mesma maneira que o All Blacks. Foi então que determinaram quatro aspectos dignos de emulação: o primeiro foi o foco intenso no objetivo, construído e energizado com os cantos Maori. O segundo foi a colaboração radical — braços e corpos unidos, esforçando-se em busca do mesmo objetivo. O terceiro, o anseio de aniquilar — qualquer coisa que entrasse no caminho deveria ser eliminada. E o último consistia da excitação universal quando qualquer membro do time conseguia passar com a bola. Não importava quem. Só o fato de acontecer era motivo de comemoração.

Então, definimos essa estrutura de Sprints, reuniões diárias, revisões e retrospectivas, e percebi que precisávamos de alguém cujo trabalho seria certificar-se de que o processo em si era eficaz. Não era um gerente, e sim um líder da equipe, algo entre um capitão e um treinador. Enquanto assistíamos ao All Blacks todos os dias, perguntei à equipe como deveríamos chamar essa pessoa; eles decidiram chamá-la Mestre Scrum. O Mestre deveria ser o facilitador de todas as reuniões, avaliar se havia transparência e, o mais importante, ajudar a equipe a descobrir quais eram os obstáculos no caminho. O importante desse trabalho era perceber que, em geral, os obstáculos não são tão simples quanto uma máquina que não funciona ou que Jim é uma droga em contabilidade — o processo em si. O trabalho do Mestre Scrum é guiar a equipe em direção ao *aprimoramento contínuo* — sempre perguntar “Como podemos fazer melhor o que fazemos?”.

Idealmente, ao fim de cada iteração, cada Sprint, a equipe deve olhar para si

mesma com muita atenção — para suas interações, práticas e processos — e fazer a si mesma duas perguntas: “O que podemos mudar no modo como trabalhamos?” e “Qual é o nosso maior problema?”. Se essas duas perguntas forem respondidas de maneira direta e aberta, uma equipe pode realizar o trabalho mais rápido do que qualquer um jamais tenha imaginado.

Não odeie o jogador, odeie o jogo

Costuma acontecer que moral, coesão e produtividade baixas se baseiam em uma falta de compreensão fundamental sobre o modo como os seres humanos trabalham. Quantas vezes você se encontrou com um colega e começou a falar mal de um terceiro que “não está fazendo a sua parte” ou que “sempre nos atrasa” ou que “toma decisões idiotas”? Ou você estava em um grupo tentando resolver um problema e a primeira coisa que todos fazem é tentar encontrar um culpado?

Estou disposto a apostar que cada um de vocês já participou de uma reunião desse tipo. Também aposto que cada um de vocês, em uma situação ou outra, foi considerado o culpado pelo problema. Mas eu também estaria aberto a apostar que quando você está colocando a culpa em alguém, você encontra falhas a partir de um ponto de vista pessoal, ao passo que quando *você* está levando a culpa, está muito mais ciente dos fatores situacionais que levaram ao problema e dos motivos que o levaram a agir daquele modo. E você sabe o que mais? Quando você fala de si mesmo, está absolutamente certo. Quando fala dos outros, porém, está cometendo um dos erros mais comuns — e destrutivos — dos seres humanos ao julgar as ações de outras pessoas. Isso até tem um nome: “Erro Fundamental de Atribuição”.

Alguns estudos fascinantes relacionados a isso estão descritos no livro *Induction: Processes of Inference, Learning, and Discovery* [Indução: processos de inferência, aprendizagem e descoberta], escrito por John H. Holland, et al. Um estudo citado no livro foi publicado no início da década de 1970, então não é novidade nenhuma. São conceitos velhos que foram reproduzidos diversas vezes. Tudo sobre o que faz os seres humanos agirem como agem. De qualquer forma, esse grupo de pesquisadores reuniu vários alunos do sexo masculino e fez duas perguntas simples: “O que levou você a escolher esse curso para se formar?” e “Por que você está namorando a sua namorada?”. Então, pediram para os sujeitos da pesquisa responderem às

mesmas perguntas em relação aos melhores amigos. Apareceram diferenças importantes. Quando falavam sobre si mesmos, não o faziam de forma pessoal, mas sobre a pergunta em si. Respondiam coisas como “Química é um campo com altos salários” sobre a escolha do curso, e “Ela é uma pessoa muito amorosa” sobre a namorada. Mas quando falavam sobre os amigos, falavam sobre as capacidades e as necessidades desses amigos. Coisas como “Ele sempre foi muito bom em matemática” ou “Ele é meio dependente e precisa de uma mulher que tome conta de tudo”.[21]

Essa forma de perceber o mundo é engraçada quando você a vê nos outros. É tão óbvio que eles estão fazendo um julgamento injusto. Antes de rir, porém, você precisa admitir que *você* faz isso o tempo todo também. Todo mundo faz. Todos nós achamos que estamos respondendo a uma situação, ao passo que os outros são motivados pelo próprio caráter. Um efeito colateral engraçado é que quando nos pedem para descrever as características da nossa personalidade e as dos nossos amigos, nós sempre nos representamos como alguém bem mais chato. Nós dizemos que temos bem menos personalidade do que eles.

Os autores de *Induction* traçaram um paralelo interessante entre a maneira como pensamos erroneamente sobre as motivações sociais e como pessoas que não são cientistas ou pessoas com um pouco de compreensão intuitiva de física veem o mundo físico.

Um físico intuitivo pode explicar por que uma pedra cai afirmando que ela possui a qualidade intrínseca da gravidade, em vez de dizer que a gravidade faz parte de um sistema de forças agindo sobre a pedra. Do mesmo modo, quando falamos sobre os outros, abordamos suas propriedades inatas, em vez perceber as qualidades relacionadas com o ambiente externo. Na verdade, são tais interações, como nosso ambiente, que estimulam nosso comportamento. O *sistema que nos cerca*, em vez de qualquer qualidade intrínseca, é o responsável por grande parte do nosso comportamento. O Scrum foi criado para mudar tal sistema. Em vez de buscar culpados e falhas, ele recompensa o comportamento positivo porque se concentra nas pessoas trabalhando em conjunto e conseguindo cumprir as tarefas.

Talvez a mais famosa demonstração dessa reação humana aos sistemas seja o experimento de Milgram sobre obediência a figuras de autoridade, o qual foi realizado no início da década de 1960, em Yale. O experimento foi simples e, para o mundo moderno, um pouco cruel. Também era devastador e poderoso, e ensinado no primeiro ano do curso de psicologia. O dr. Stanley Milgram, um professor da universidade, fazia uma pergunta bastante pertinente para a época.

Três meses antes do início dos primeiros experimentos, Adolf Eichmann, um dos criadores do Holocausto, fora a julgamento.

Um dos questionamentos mais persistentes cercando o Holocausto era como tantos milhões de pessoas poderiam ter se tornando cúmplices condescendentes em tal horror. Será que os alemães eram fundamental e moralmente repreensíveis? Será que havia algo intrinsicamente mal na formação daquela cultura? Ou será que eles estavam apenas seguindo ordens? É muito fácil olhar para os crimes cometidos contra a humanidade e colocar a culpa de seus atos em indivíduos. É a coisa certa a se fazer, não é? A pergunta de Milgram, porém, era: será que os americanos comuns são tão diferentes dos alemães? Será que teriam reagido de forma diferente na mesma situação? E a resposta desagradável é que não, os americanos *não teriam* se comportado de forma diferente. Na verdade, considerando o número de países que reproduziu o experimento, *ninguém* teria agido diferente. Diante da situação adequada, todos nós somos capazes de nos transformar em nazistas.

O experimento funcionava da seguinte forma: o sujeito, uma pessoa comum, recebia instruções de alguém usando um jaleco branco de laboratório (o que lhe conferia uma imagem de autoridade científica) para aplicar choques elétricos cada vez mais fortes em uma terceira pessoa, um ator, que estava em uma outra sala. O sujeito poderia ouvir, mas não ver, o ator. À medida que os choques aumentavam, o ator começava a gritar e implorar que parassem. Por fim, o ator (que, em algumas versões do experimento, dizia ao sujeito que tinha problemas cardíacos) começaria a bater na parede, gritando para que cessassem o experimento. Por fim, ele ficava em silêncio.

Algumas pessoas pararam em 135 volts, enquanto o ator gritava, e perguntavam sobre o objetivo do experimento. Quase todos continuaram depois de saber que não seriam considerados responsáveis. Alguns sujeitos exprimiram uma risada nervosa enquanto ouviam os gritos de agonia no quarto ao lado. Quando alguém quisesse parar, o "cientista" apenas dizia: "*Continue*, por favor". E se o sujeito não recomeçasse, insistia: "O experimento precisa que você *continue*". Se ainda assim nada acontecesse, ele acrescentava: "É absolutamente essencial que você *continue*". A maioria das pessoas pareceu ficar sob grande estresse e suava muito. Exibiam pulso acelerado e temperaturas elevadas quando o instinto de "lutar ou fugir" tomava conta deles. Então, se eles ainda não apertassem o botão, o cientista tentava uma última vez: "Você não tem outra escolha. Você *tem* que continuar".

Quase todo mundo continuou, dando o choque final em alguém que tinha

gritado e, depois, ficado em silêncio. Milgram resumiu as implicações do experimento em seu artigo “The Perils of Obedience” [“Os perigos das obediência”], de 1974, da seguinte forma:

> Pessoas comuns, apenas fazendo seu trabalho, e sem qualquer hostilidade particular de sua parte, podem se tornar agentes em um processo destrutivo terrível. Além disso, mesmo quando os efeitos destrutivos de seu trabalho se tornam evidentemente claros, e eles recebem instruções para continuar fazendo ações incompatíveis com padrões fundamentais de moralidade, relativamente poucas pessoas têm os recursos necessários para resistir à autoridade.[22]

Em geral, quando tal experimento é discutido em sala de aula, o professor chama a atenção dos alunos para o fato de que o culpado é o *sistema* no qual aquelas pessoas comuns estão inseridas, e não elas. Mas é uma lição difícil de internalizar, porque se você aceitar a verdade, o que isso diz sobre você?

A resposta é que *todos* somos criaturas do sistema onde estamos inseridos, e o que o Scrum faz é aceitar tal realidade e, em vez de procurar um culpado, tenta analisar o sistema que produziu o erro e corrigi-lo.

Outro experimento semelhante foi realizado em um seminário teológico no início da década de 1970. Você poderia imaginar que seminaristas devem ser as pessoas mais misericordiosas do planeta, certo? Os sujeitos desse experimento receberam a instrução de darem um sermão do outro lado do campus. Alguns receberam a informação de que tinham de se apressar, porque as pessoas já estavam aguardando, e eles estavam atrasados. Outros não receberam essa informação. Enquanto atravessavam o campus da faculdade, cada um dos seminaristas passou por alguém gemendo e pedindo ajuda em uma porta. Quantas das pessoas que receberam instruções de se apressarem pararam para ajudar? Dez por cento. De *seminaristas*.

Ainda assim, as pessoas querem culpar indivíduos, não sistemas. É melhor assim. O Erro Fundamental de Atribuição agrada o nosso senso de justiça. Se pudermos culpar outra pessoa, nos protegemos do fato de que faríamos a mesma coisa — que provavelmente também apertaríamos aquele botão, diante de um contexto que nos impulsionasse a isso.

Como esse erro de culpar indivíduos em vez de sistemas se manifesta nos negócios? Eu tenho dois bons exemplos, o primeiro ocorrido na fábrica de carros New United Motor Manufacturing, Inc. (NUMMI), em Fremont, Califórnia. Tratava-se de um empreendimento conjunto entre a General Motors e a Toyota. A fábrica tinha sido fechada pela GM em 1982; a administração da

empresa considerava a mão de obra a pior dos Estados Unidos. As pessoas bebiam no trabalho, faltavam e sabotavam, sutilmente, os carros (colocando, por exemplo, uma garrafa de Coca-Cola dentro de uma porta, onde ela ficaria batendo e irritando os clientes). A Toyota reabriu a fábrica em 1984. A GM contou a eles como os trabalhadores eram horríveis, mas que os gestores eram ótimos e que eles poderiam recontratá-los. Em vez disso, a Toyota decidiu não recontratar os gestores, mas sim a maioria dos trabalhadores — chegando a enviar alguns deles ao Japão para aprender o Sistema Toyota de Produção. Quase imediatamente, a fábrica NUMMI passou a produzir carros com a mesma precisão e com um percentual de defeitos tão baixo quanto no Japão. As mesmas pessoas, um sistema diferente. A GM nunca atingiu aqueles níveis de qualidade em nenhuma de suas outras fábricas americanas. Ela saiu do contrato de operação conjunta no mesmo ano em que abriu falência.

O *segundo* exemplo que vem à mente é um pouco diferente. Ele faz com que eu me pergunte qual nível de uma "posição de negligência" é necessário para que as pessoas comecem a procurar alguém a quem culpar, em vez de buscar uma solução. Isso tem a ver com a forma como os capitalistas de risco com quem trabalho agem quando decidem investir em uma empresa. Fiquei surpreso logo que entrei para a OpenView Venture Partners e percebi que, diferente de muitas empresas de capital de risco, eles realmente não ligam a mínima para como a corporação gastou o dinheiro *antes* de seu investimento. O passado não importa. A OpenView decide se quer investir com base no estado atual do negócio — todo o resto é insignificante. Eles querem saber como o dinheiro *deles* será gasto; como a empresa gastou o dos outros não é importante. Só o futuro — apenas as *soluções* — realmente importam.

Em busca da "grandiosidade"

Parece mágica quando uma equipe começa a se alinhar e entrar em total sincronia. Você sente quando entra em uma sala com eles: parece que entraram em campo. Tem-se a impressão de que estão flutuando —eles se tornaram maiores do que eles mesmos.

Estive recentemente na casa de um amigo em Copenhague. Como você pode imaginar, já que ele é europeu, é um grande fã de futebol. Não sei ao certo qual era o campeonato em que o time dele estava jogando, mas era intenso observá-lo pular, sentar e gritar com a televisão. Eu estava diante de um fã de esportes

enfurecido. E, então, chegou o momento em que o jogo estava empatado, os segundos avançavam, e o time dele tinha a posse de bola. Então, talvez um pouco atrás do meio de campo, *sem olhar para onde os outros jogadores de seu time estavam*, um centroavante chutou a bola para uma massa de jogadores em frente ao gol. O problema era que não havia ninguém para receber a bola. Por um instante fiquei desanimado; de repente, um jogador apareceu — no lugar certo e na hora certa, e fez o gol. Ele correra a toda velocidade do meio de campo e entrou no meio da massa de adversários em frente ao gol, onde viu a oportunidade de cabecear a bola. Foi uma surpresa completa. O centroavante que chutara a bola tivera fé, porém, que o seu colega estaria onde deveria estar. E aquele colega acreditara que a bola chegaria onde ele poderia fazer alguma coisa com ela. Foi inspirador assistir àquele tipo de sincronia.

E é nesse lugar que eu quero ajudar as pessoas a chegarem com o Scrum. Não é impossível. Não são apenas as elites, os atletas e as pessoas especiais que conseguem fazer isso. O Scrum é sobre estabelecer o sistema certo com os devidos incentivos e dar às pessoas liberdade, respeito e autoridade para fazerem as coisas por si mesmas. A grandiosidade não pode ser imposta, precisa vir de dentro. E, o melhor, ela existe em todos nós.

PONTOS PRINCIPAIS

A alavanca certa. Mude o desempenho da equipe. Isso tem um impacto *muito* maior — em várias ordens de grandeza enorme — do que o desempenho individual.

Transcendência. Grandes equipes têm um objetivo muito maior do que o individual — por exemplo, participar do enterro do General MacArthur, vencer o campeonato de basquete da NBA.

Autonomia. Dê às equipes a liberdade de tomarem as próprias decisões e o modo como vão agir — para ser respeitado como mestre de sua arte. A capacidade de improvisar fará toda a diferença, esteja a equipe cobrindo uma revolução do Oriente Médio ou vendendo algum produto.

Interfuncionalidade. A equipe deve ter todas as capacidades necessárias para concluir um projeto, seja a missão entregar um software para a Salesforce.com, seja ou capturar terroristas no Iraque.

Pequenas vitórias. Equipes pequenas trabalham mais rápido do que equipes grandes. A regra prática são sete membros por equipe — podendo variar dois a mais ou a menos.

Procurar culpados é burrice. Não procure pessoas ruins; procure sistemas ruins — aqueles que incentivam mau comportamento e premiam desempenhos medíocres.

12 H. Takeuchi e I. Nonaka. The New New Product Development Game *Harvard Business Review,* Jan./Fev., 1986, pp. 285-305.

13 ouglas MacArthur, *The Long Gray Line*. Discurso em West Point, Nova York, 1962.

14 Richard Feynman, *Presidential Commission on the Space Shuttle Challenger Accident, Appendix F —*

"Personal Observations on Reliability of Shuttle", Relatório de 1986.

15 Jobs Warrick e Robin Wright. U.S. Teams Weaken Insurgency In Iraq, *The Washington Post*, 6 de setembro de 2006.

16 Michael Flynn et al., Employing ISR: SOF Best Practices *Joint Force Quarterly, Issue* 50, 3º trimestre, 2008 p. 60.

17 C. Lamb e E. Munsing. *Secret Weapon: High Value Target Teams as an Organizational Innovation*, Institute for National Strategic Studies, Strategic Perspectives, n. 4, 2011.

18 Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Reading, MA, Addison-Wesley, 1975. [Em português, *O mítico homem-mês.* Rio de Janeiro, Elsevier, 2009.]

19 Nos Estados Unidos, os números telefônicos possuem dez dígitos no total. No entanto, os três primeiros correspondem ao código da área. (N.T.)

20 N. Cowan, The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity, *Behavioral and Brain Sciences, Vol.* 24, 2001, pp. 87-185.

21 Richard Nisbett et al., Behavior as Seen by the Actor and as Seen by the Observer, *Journal of Personality and Social Psychology, Vol.* 27(2), 1973, pp. 154-64.

22 Stanley Milgram, The Perils of Obedience, *Harper's Magazine*, 1974.

CAPÍTULO 4

Tempo

O tempo é o limite insuperável do empreendimento humano, afetando tudo, desde o modo como trabalhamos, passando pelo tempo que as coisas levam, até definir se somos bem-sucedidos ou não. O fluxo impiedoso do tempo define fundamentalmente o modo como enxergamos o mundo e a nós mesmos. Conforme o famoso verso do poeta inglês do século 17, Andrew Marvell, "Se nós tivéssemos mais tempo e mundo", qualquer coisa poderia ser feita. É claro, porém, que um senso de mortalidade paira sobre todos os nossos esforços. Sabemos que nosso tempo é limitado. Dessa forma, não é um verdadeiro crime desperdiçá-lo? Nas palavras de Marvell, novamente:

> *Verdade, o sol não pode ser parado*
> *correndo é que ele fica ao nosso lado.*[23]

Mas como podemos fazer isso? É fácil gritar "*Carpe diem*!" de um palco para inspirar a plateia, mas como você pode conseguir isso de verdade? Um monte de trabalho diz para as pessoas se sentarem, se prepararem e trabalharem horas a fio. "Não pense no mundo lá fora" é o que os nossos chefes nos dizem de forma implícita. "Não se preocupe com os seus filhos nem com o jantar — só trabalhe, e, então, trabalhe mais, e será recompensado. Você será promovido. Conseguirá fechar aquela venda. Você terminará aquele projeto".

Embora eu não tenha nada contra promoções, vendas ou projetos, é fato que os seres humanos são terríveis quando trabalham dessa forma. Não somos bons em manter o foco, passamos muito mais horas do que o necessário no escritório, e somos péssimos em fazer estimativas de quanto tempo algo levará para ser feito. Estou falando de *todas* as pessoas — é assim que os seres humanos *são*.

Quando eu me sentei para desenvolver o Scrum, não tinha a menor intenção de criar um novo "processo". Meu projeto era apenas juntar toda a pesquisa

que tinha sido feita durante décadas e selecionar o que havia de mais interessante e útil naquilo. Queria incorporar as melhores práticas e “roubar” qualquer uma das boas ideias com as quais eu me deparasse. Um pouco antes do primeiro Scrum de verdade, na Easel, em 1993, eu estava trabalhando em uma empresa a alguns quarteirões do laboratório de mídia do MIT, e “roubei” uma ideia do laboratório que se tornou o centro do Scrum: o Sprint.

O Sprint

No início da década de 1990, o laboratório de Mídia do MIT inventava todo tipo de maravilhas. Na época em que a internet ainda estava engatinhando, eles faziam desde robôs até a tinta eletrônica que possibilitou o desenvolvimento dos dispositivos eletrônicos de leitura, e novas maneiras de codificar o som. Foi um período inacreditável, e eu tendia a contratar alunos que saíam do laboratório, porque eles estavam com a cabeça cheia de ideias e tinham uma capacidade incrível de pensar em coisas legais e construí-las rapidamente.

Essa velocidade vinha de uma política do laboratório de mídia que se aplicava a todos os seus projetos. De três em três semanas, cada equipe tinha de demonstrar para os colegas o projeto no qual estavam trabalhando. Tratava-se de uma exposição aberta; qualquer um poderia assistir. E se não funcionasse ou não fosse legal, os diretores do laboratório acabavam com o projeto. Isso forçava os alunos a construírem coisas maravilhosas e funcionais de forma rápida, e, o mais importante, eles recebiam um feedback imediato sobre o trabalho.

Pense nos diversos projetos que você faz. Aposto que raramente você recebe um feedback até que eles estejam concluídos — e isso pode levar meses, até anos. Você pode estar seguindo na direção *completamente* errada durante muito tempo, sem suspeitar de nada, e isso vai acabar subtraindo uma grande parte da sua vida. Nos negócios, isso pode significar a diferença entre o sucesso e o fracasso. Eu já vi esse tipo de situação acontecer o tempo todo: uma empresa passa anos em um projeto que parecia uma ótima ideia quando os funcionários começaram, mas, quando cruzam a linha de chegada, o mercado já sofreu mudanças fundamentais. Quanto mais cedo você entregar o produto para os clientes, mais rápido eles serão capazes de dizer se você está fazendo algo de que precisam.

Então, quando comecei o primeiro Scrum na Easel e disse para o CEO que

não ia mostrar para ele um enorme e detalhado diagrama de Gantt que nós dois sabíamos que estava errado, ele respondeu: “tudo bem. O que você *vai* me mostrar?”. E eu respondi que uma vez por mês mostraria a ele uma parte funcional do software. Não algo que funciona por trás do programa. Não alguma parte da arquitetura do projeto. Mas uma parte do software que um cliente pudesse realmente usar. Uma característica completamente implementada.

“Tudo bem, então”, concordou ele. “Pode fazer assim.”

Então, a minha equipe embarcou no que chamamos de “Sprints”. Escolhemos esse nome por evocar uma qualidade de intensidade. Todos vamos trabalhar por períodos curtos e, então, paramos para ver aonde chegamos.

A Team WIKISPEED é um grupo fundado por Joe Justice, que fabrica carros. Carros que fazem mais de 42 quilômetros por litro de combustível; são aprovados para trafegar nas ruas; têm classificação cinco estrelas; atingem 220 km/h; e custam menos do que um Toyota Camry. A WIKISPEED está constantemente aprimorando o veículo, mas se você quiser comprar um, precisa fazer um depósito de US$ 25 mil na wikispeed.com, e eles entregarão o carro em um prazo de três meses. E eles usam o Scrum para fazer isso. Assim como muitas das melhores equipes hoje, trabalham com Sprints de uma semana. Toda quinta-feira, eles se sentam e olham para uma lista grande das pendências, que inclui tudo: desde o protótipo de um novo painel até testar as setas dos carros. Eles organizam a lista por prioridade e dizem: “tudo bem, considerando essa lista, quantas coisas conseguiremos fazer nesta semana?”; e quando dizem “fazer”, o que eles querem dizer é “concluir” — de forma completa. Essas novas características funcionam. O carro pode ser dirigido. Toda semana. Em cada Sprint.

Ao entrar na oficina da Team WIKISPEED, no norte de Seattle, em uma quinta-feira normal, você verá o caos glorioso e organizado que é uma oficina de construção. Há caixas de ferramentas, serrotes, equipamento eletrônico, prendedores e chaves de fenda. Uma fresadora CNC Router está em um canto ao lado da estrutura inacabada de um carro na baia 3. Uma furadeira e uma curvadora de metal estão largadas ali como filhotes ávidos para brincar. No dia em que os visitamos, havia uma foto do comprador do carro sobre a estrutura inacabada — Tim Myer. Ele gosta de escalar montanhas, comer batata frita e tomar cidra. Ele não gosta de não saber o que está acontecendo, muito menos de não ter opções. Você pode encontrá-lo nas montanhas nos fins de semana ou dançando no Tractor nas noites de segunda-feira.

Na frente, na baia 1, está o primeiro carro que a Team WIKISPEED fez — o carro que participou de um concurso XPrize, que pagava um prêmio de US$ 10 milhões para carros que atingem uma eficiência de combustível de 42 quilômetros por livro de combustível. A empresa ficou em décimo lugar, vencendo mais de cem competidores, desde grandes empresas automobilísticas até universidades. Como resultado, foram convidados, em 2011, para participar do Detroit Auto Show e posicionados bem na frente, entre a Chevy e a Ford. Esse carro agora é o melhor laboratório de teste para novas ideias.

Ao lado do carro, há um quadro branco cobrindo uma parede de três metros e meio de altura, que se estende por toda a largura da loja. Nele, você encontra dezenas e mais dezenas dos artefatos mais comuns encontrados no Scrum: *post-its*. Em cada um desses quadradinhos de papel, há algo que precisa ser feito: “furar o tubo para o encaixe da engrenagem de direção”, “preparar o molde interno”, “revestimentos internos do para-lama para evitar borrifos dos pneus” etc.

O quadro está dividido em algumas colunas: “Pendências”, “Fazendo”, “Feito”. A cada Sprint, os membros da equipe da WIKISPEED colocam na coluna “Pendências” quantos *post-its* eles acham que conseguem concluir no decorrer da semana. Conforme o tempo passa, um membro da equipe pegará uma daquelas tarefas e a colocará na coluna “Fazendo”. Quando tal tarefa é finalizada, o *post-it* passará para a coluna “Feito”. Todos na equipe conseguem visualizar o que cada um está fazendo o tempo todo.

Um ponto importante: nenhuma tarefa vai para a coluna “Feito”, a não ser que possa ser cumprida pelo cliente. Em outras palavras: você pode *dirigir* o carro. E se alguém dirige o carro e diz “ei, a seta está agarrando”, então, esse problema será resolvido no próximo Sprint.

Os Sprints às vezes são chamados de “caixas de tempo”, porque são definidos para ter certa duração. Você não pode fazer um Sprint de uma semana e, depois, um de três semanas. Você precisa ser consistente: quer estabelecer o ritmo de trabalho no qual as pessoas saibam o que pode ser feito em determinado período. Em geral, a quantidade os surpreende.

Um elemento crucial de um Sprint individual, porém, é que uma vez que a equipe se compromete com o objetivo, as tarefas são bloqueadas. Nada mais pode ser acrescentado por ninguém fora da equipe. Explicarei os motivos disso mais adiante, mas, por ora, saiba apenas que interferir e distrair a equipe reduz drasticamente sua velocidade.

Conforme mencionei, no primeiro Scrum, definimos Sprints de quatro

semanas. Quase no final do primeiro Sprint, percebemos que não estávamos sendo rápidos o suficiente, poderíamos fazer mais. Assistimos ao vídeo do All Blacks fazendo a *haka* e ultrapassando a linha dos oponentes. *Por que não somos assim?*, perguntamos. *Por que não temos esse tipo de espírito?* O nosso objetivo não era apenas ser uma boa equipe, mas ser a melhor. Como conseguiríamos isso? Uma vez mais, a resposta foi algo simples que "roubamos" de outra pessoa — uma reunião diária.

Reunião diária

Fora dos limites de uma cidade que não posso citar, em uma empresa que não posso mencionar, um grupo de pessoas se reúne todos os dias para ponderar como levar outras pessoas ao espaço. Já que foguetes são, na verdade, mísseis balísticos intercontinentais com uma carga humana, existe certa quantidade de segurança e sigilo em torno do esforço privado para viagens espaciais. E *trata-se* de um negócio, não apenas de uma ideia bilionária e um sonho impossível. Enquanto escrevo, um outro foguete privado acabou de ser atracado na Estação Espacial Internacional pela segunda vez. Nem mesmo o governo dos Estados Unidos tem essa capacidade atualmente.

Contudo, nesse prédio e dia específicos, essas pessoas estão lutando para determinar o tamanho da caixa que conterá toda a aviônica do foguete. A aviônica é responsável por informar à espaçonave onde esta se encontra, para onde vai e como chegará lá. Pense nisso como a mente do foguete.

Existem duas equipes: uma de hardware e outra de software, cada qual composta por sete pessoas. Todos os dias as equipes se reúnem diante de um quadro branco que cobre a parede de cima a baixo e de um lado a outro. Assim como na WIKISPEED, há algumas colunas desenhadas no quadro: "Pendências", "Fazendo", "Feito". Listadas nas colunas estão apenas as coisas que a equipe precisa fazer durante aquele Sprint. As tarefas variam desde trabalhar com um dos diversos fornecedores de circuitos eletrônicos especiais até mapear como o acelerômetro se comunicará com o restante da nave. O Mestre Scrum, a pessoa responsável por dirigir o processo, faz três perguntas para cada um dos membros da equipe:

1. O que você fez ontem para ajudar a equipe a concluir o Sprint?
2. O que você vai fazer hoje para ajudar a equipe a concluir o Sprint?
3. Que obstáculos a equipe está enfrentando?

Só isso. Essa é a reunião. Se ela levar mais que 15 minutos, você está fazendo algo errado. O objetivo dessa reunião é ajudar todos na equipe a saberem exatamente em que ponto do Sprint eles estão. Todas as tarefas serão concluídas no prazo? Existem oportunidades para ajudar outros membros da equipe a superar obstáculos? Não há tarefas definidas por alguém superior — a equipe é autônoma; *eles* fazem isso. Não há relatórios detalhados para a diretoria. Qualquer um da diretoria ou de outra equipe pode passar e olhar para o quadro Scrum de aviônica para saber exatamente em que ponto o projeto está.

Então, quando a primeira equipe Scrum quis descobrir como poderiam se tornar os All Blacks, eles consultaram a literatura para descobrir como as melhores equipes conseguiam isso. Algo muito legal no desenvolvimento de software era que a situação inicial estava tão ruim, e tanto dinheiro estava sendo desperdiçado — bilhões e bilhões de dólares por ano —, que as pessoas dedicavam muito tempo analisando o motivo, e havia dados sobre tudo.

Uma das pessoas que passou anos estudando como as coisas eram feitas na indústria de software foi Jim Coplien, do lendário Bell Labs da AT&T. Coplien, que é chamado por todos e por si mesmo de “The Cope”, passou anos analisando centenas de projetos de software, tentando descobrir por que uma pequena minoria se saía tão bem, enquanto a grande maioria era desastrosa. No início da década de 1990, ele foi convidado para analisar um projeto que a Borland Software Corporation estava desenvolvendo: um novo produto de planilhas chamado Quattro Pro para o Windows. Para esse projeto, já tinham sido criadas mais de um milhão de linhas de código de software. Eles levaram 31 meses para produzir isso e o trabalho foi realizado por oito pessoas. Isso significava que cada membro da equipe produzira mil linhas de código por semana. Aquela era a equipe mais rápida da história, e Jim queria saber como eles conseguiram aquilo.

Então, ele mapeou todo o fluxo de comunicação da equipe — quem falava com quem, onde a informação fluía bem e onde isso não acontecia. Esse tipo de mapeamento é uma ferramenta clássica que pode ser usada para detectar gargalos ou pessoas que não passam informações adiante. Quanto maior a saturação de comunicação — quanto mais todos sabem de tudo —, mais rápida é a equipe. Basicamente, a métrica conseguida com esse tipo de análise mede o quão bem todos sabem do que precisam para realizar o próprio trabalho. A Borland recebeu a maior classificação já vista: 90%. A maioria das empresas

fica em torno de 20%.

Mas como conseguiríamos alcançar aquele nível na nossa equipe? O que enfraquece tal saturação é a especialização — o número de papéis e cargos em um grupo. Se as pessoas têm um cargo específico, elas tendem a fazer apenas as coisas que parecem combinar com esse cargo. E, ao proteger o poder desse papel, provavelmente elas irão reter conhecimentos determinados.

Então, nós nos livramos de todos os nomes de cargos. Chamei todo mundo e disse para rasgarem seus cartões de visita. Se alguém quisesse colocar um cargo no seu currículo, só o fariam para uso externo. Ali, onde o trabalho era feito, só havia membros de uma equipe.

O outro ingrediente do "molho secreto" da equipe da Borland era uma reunião diária para discutir o que estava sendo feito. Unir o pessoal em uma sala é a chave, porque isso dá à esquipe a oportunidade de se organizar em relação aos desafios. Se alguém estava empacado em um problema — se o acelerômetro não se comunica com o altímetro —, todos percebiam que aquele obstáculo poderia bloquear o Sprint por inteiro, então todos precisavam se juntar para se certificarem de que iriam resolver a situação.

Na Borland, a reunião durava pelo menos uma hora. Aquilo me pareceu um tempo muito longo, então, analisei as informações centrais que precisavam ser comunicadas e cheguei àquelas três perguntas.

E foi assim que as reuniões diárias passaram a funcionar. Nós tínhamos certas regras. Os encontros aconteciam no mesmo horário todos os dias, não importando em qual, e *todos* deviam estar lá. Se a equipe inteira não estivesse presente, a comunicação simplesmente não acontecia. A questão era dar à equipe um ritmo regular.

A segunda regra era que a reunião não podia durar mais do que 15 minutos. Queríamos que ela fosse rápida, clara e direto ao ponto. Se algo precisasse ser discutido mais a fundo, anotávamos e nos reuníamos depois. A ideia era obter as informações mais valiosas e práticas no menor tempo possível.

A terceira regra era que todo mundo tinha de participar de forma efetiva. Para isso, eu disse que todos deviam ficar de pé. Desse modo, haveria conversas ativas, onde todos falavam e ouviam, e também manteria as reuniões curtas.

Esse é o motivo por que tais reuniões costumam ser chamadas de "*Stand-up* diário" ou "Scrum diário". Mas não importa muito *como* você a chama. O importante é que ela deve ocorrer no mesmo horário todos os dias, com as mesmas três perguntas, que todos estejam de pé e que ela não dure mais do que 15 minutos.

O problema que costumo ver é que as pessoas têm a tendência de tratar as reuniões diárias simplesmente como um relatório individual. “Eu fiz isso... Eu vou fazer aquilo” — e depois passam para a outra pessoa. A abordagem mais eficaz fica mais próxima de quando os jogadores de futebol americano se juntam para preparar uma jogada. Um receptor pode dizer “estou com problemas com a linha de defesa”, ao que um bloqueador da barreira pode responder “pode deixar comigo. Vou abrir a linha pra você”. Ou um zagueiro pode dizer “nosso jogo está se deparando com uma muralha, vamos surpreendê-los com um passe para a esquerda”. A ideia é que a equipe converse de forma rápida de como poderão seguir em direção à vitória — ou seja, concluir um Sprint. A passividade não é apenas algo indolente, mas também algo que afeta o desempenho do restante da equipe. Assim que detectada, deve ser eliminada, imediatamente.

Eu quero equipes agressivas — com membros que saiam dessas reuniões sabendo a coisa mais importante que precisam atingir naquele dia. Uma pessoa ouve outra dizer que vai levar um dia para concluir determinada tarefa, mas outra talvez saiba como concluí-la em uma hora se trabalharem juntas. Eu quero equipes que saiam da reunião dizendo coisas como “Vamos nessa. Vamos conseguir”. A equipe precisa *querer* ser grandiosa.

O meu discurso padrão para equipes grandes e pequenas é: “Vocês querem ser horríveis para o resto da vida? É essa a sua motivação? Porque existe uma escolha, sabe? — você não *precisa* ser assim”. Uma equipe precisa exigir de si mesma a grandiosidade.

Na Easel, com a primeira equipe Scrum, implementamos as reuniões diárias durante o terceiro Sprint. Tínhamos planejadas quatro semanas de trabalho para ele — que consistia mais ou menos na mesma quantidade de trabalho do mês anterior. Terminamos tudo em uma semana. Tivemos uma melhora de 400%. Naquela primeira sexta-feira, todos da equipe olharam um para os outros e exclamaram “uau!”. Naquele momento eu soube que talvez eu tivesse descoberto algo.

Repetições

Aquele tipo de aprimoramento foi acrescentado ao Scrum a partir daquele terceiro Sprint. Trata-se do objetivo do Scrum. Em certos casos, já vi equipes altamente disciplinadas aumentarem oito vezes a produtividade. É isso que

torna o Scrum revolucionário. Você consegue que mais coisas sejam feitas de forma mais rápida e mais barata — o dobro de trabalho na metade do tempo. E lembre-se: não é só nos negócios que o tempo é importante. O tempo constitui sua vida, então, desperdiçá-lo, na verdade, é uma forma lenta de suicídio.

O que o Scrum faz é alterar o modo como você pensa sobre o tempo. Depois de fazer parte de alguns Sprints e reuniões diárias, você para de ver o tempo como uma flecha linear atirada para o futuro, e começa a vê-lo como algo fundamentalmente cíclico. Cada Sprint é uma oportunidade de fazer algo totalmente novo; a cada dia, uma chance de se aprimorar. O Scrum encoraja uma visão holística do mundo. A pessoa que se compromete com isso passará a valorizar cada momento como um ciclo recorrente de fôlego e de vida.

Sempre me espanta o tempo que leva para uma casa ser reformada. Minha esposa e eu costumávamos sempre nos lembrar, um ao outro, que isso sempre levará o dobro do tempo e do custo que estimamos. E isso se tivermos sorte. Estou certo de que você já ouviu as mesmas histórias horríveis que eu: a reforma da cozinha que levaria duas semanas acabou levando seis, fazendo com que a família tivesse de comer fora por mais de um mês; o reparo na rede elétrica que demorou o triplo do tempo estimado; um trabalho pequeno que pareceu demorar uma eternidade para ser concluído. Bem, há uns dois anos, o meu amigo e colega na corrente de pensamento ágil, Eelco Rustenburg, contou-me em um jantar que tinha decidido reformar a casa — completamente, do chão ao teto. Ele disse que mexeria em todos os aposentos, mudaria a fiação, instalaria novos aparelhos, pintaria tudo. E o trabalho levaria apenas seis semanas.

Todos rimos e começamos a presentear Eelco com as *nossas* histórias dolorosas de reformas. “Seis semanas para a casa inteira?”, perguntei, rindo. “Não vai dar. Só para reformar a nossa *cozinha* levou esse tempo, embora eles tenham prometido concluir tudo em duas semanas. Você vai ter que morar em um hotel até o final do ano”.

“Não”, respondeu ele. “A obra vai terminar no tempo e vai ficar dentro do orçamento. Eu vou fazer tudo usando o Scrum.”

Aquilo me deixou animado — a ideia de usar o Scrum em uma área completamente diferente da indústria de software. Encontrei Eelco por acaso uns seis meses depois e perguntei como foram as coisas. “Tudo ótimo”, respondeu ele. “Seis semanas cravadas. Mas, com o meu *vizinho*, a história foi outra”.

Eis o que aconteceu. Eelco decidiu fazer os operários trabalharem como uma

equipe Scrum. Ele tinha projetos semanais para colocar na coluna “Feito”, e, no trailer do empreiteiro estacionado na frente do jardim, havia um quadro Scrum cheio de *post-its* listando as tarefas. Todas as manhãs, ele se reunia com carpinteiros, eletricistas, encanadores e quem quer que fosse necessário para o Sprint daquela semana, e repassava o que tinha sido feito no dia anterior, o que fariam naquele dia, e quais eram os obstáculos no caminho.

Eelco me contou que isso fez com que eles pensassem e se comunicassem sobre o projeto de um modo diferente do que pensavam antes. Os encanadores e os carpinteiros conversaram sobre como um poderia ajudar o outro a trabalhar mais rápido. A falta de material foi detectada antes que sua ausência fizesse a obra parar. Mas ele me contou que o que as reuniões diárias conseguiram mesmo foi acabar com as dependências. Em qualquer trabalho de construção, passa-se um tempo enorme aguardando que uma parte do trabalho seja concluída antes que outra possa começar, e, em geral, essas fases envolvem diferentes conjuntos de habilidades — instalação elétrica e colocação de gesso, por exemplo. A reunião diária colocou todas essas pessoas em uma sala onde elas rapidamente descobriram como poderiam trabalhar juntas como uma equipe. Eles não eram mais indivíduos com habilidades separadas, mas uma equipe tentando levar uma casa para a coluna de “Feito”.

E funcionou. Seis semanas depois o projeto estava pronto. Eelco e a família voltaram para casa. A vida era boa. Quando ele me contou isso, fiquei surpreso, mas o parabenizei por ter conseguido ótimos empreiteiros. “Mas olha só”, disse ele, então, “a história não acaba aí”. No final do quarteirão da casa dele, um vizinho queria fazer uma reforma praticamente igual. Os dois moraram em uma parte antiga da Holanda, e a casa deles tinha sido construída exatamente na mesma época, com uma planta exatamente igual. O vizinho viu o excelente trabalho que os empreiteiros tinham feito na casa de Eelco e imaginou que conseguisse repetir a mágica.

Os mesmos trabalhadores foram contratados, mas, daquela vez, eles levaram *três meses* para concluir o trabalho. Os mesmos trabalhadores. O mesmo tipo de casa. O mesmo trabalho. O dobro do tempo e, é claro, o dobro do custo. A única diferença foi que o vizinho não usou o Scrum. Os problemas que o Scrum força a aparecerem logo no início não foram descobertos até que já fosse tarde demais. As pessoas não se coordenaram entre si do mesmo modo e foram obrigadas a esperar que alguém terminasse um trabalho para poderem começar o seu. O vizinho acabou gastando o dobro do que Eelco, com a maior parte do custo pago para as pessoas esperarem outras concluírem a *sua* parte.

Pense no seu trabalho. Quanto tempo é desperdiçado enquanto você espera alguém concluir uma tarefa, ou receber alguma informação, ou porque está tentando fazer um monte de coisas ao mesmo tempo? Talvez *você* prefira trabalhar o dia todo — quanto a mim, *eu* prefiro surfar.

PONTOS PRINCIPAIS

O tempo é finito. Trate-o dessa forma. Divida seu trabalho em unidades que possam ser realizadas em um período definido, curto e regular — o ideal são quatro semanas. E, se você pegar a febre Scrum, pode chamar essas unidades de Sprint.

Demonstre ou morra. No final de cada Sprint, você precisa ter algo *pronto* — algo que possa ser usado (para voar, dirigir ou qualquer outra coisa).

Jogue seus cartões de visita fora. Cargos são marcadores especializados de status. Seja conhecido pelo que *faz* e não pelo modo como as pessoas se referem a você.

Todo mundo sabe de tudo. A saturação da comunicação acelera o trabalho.

Uma reunião por dia. Quando se trata de verificar o trabalho da equipe, uma vez por dia é o suficiente. Reúna-se por 15 minutos na reunião diária, veja o que pode ser feito para aumentar a velocidade e faça isso.

23 Andrew Marvell, *To His Coy Mistress*, Poets.org. [Em português, À sua senhora pudica], trad. Guilherme Gontijo Flores, Disponível em: <http://escamandro.wordpress.com/2014/04/14/andrew-marvell-em-traducao/>. Acesso em: 17 jul. 2014.]

CAPÍTULO 5

O desperdício é um crime

O coração do Scrum é o ritmo. O ritmo é profundamente importante para os seres humanos. Sua batida é ouvida na passagem do sangue pelas veias e arraigada nos recessos mais profundos do nosso cérebro. Estamos sempre em busca de padrões, levados a procurar o ritmo em todos os aspectos da nossa vida.

No entanto, os padrões que buscamos não são necessariamente recompensadores ou otimizados para nos trazer felicidade. Por exemplo, existem os ritmos negativos seguidos pelos viciados e pelos deprimidos. Você pode andar pelos corredores de praticamente qualquer prédio de escritórios e ver esses padrões se manifestarem. É provável que você os encontre em algum lugar onde alguém se sinta frustrado por estar sendo minado, ou desesperado por dentro quando se dá conta de que está preso em um sistema indiferente às pessoas, ou raiva por ser visto como uma engrenagem em uma máquina.

Isso faz parte da experiência humana. Você pode voltar mil anos no tempo e ler os textos de outras pessoas como nós, cujas vidas ficaram presas em um sistema diante do qual elas se sentem impotentes. Parece, porém, que aprendemos a controlar essa sensação de estar preso em algum momento do século 20. Principalmente no ambiente corporativo, criamos uma despersonalização aguda que parece ser imposta pelo destino.

O que o Scrum faz é criar um tipo diferente de padrão. Ele aceita o fato de que somos criaturas movidas pelo hábito, buscamos ritmo e somos, até certo ponto, previsíveis, mas também somos seres mágicos e capazes de atingir a grandiosidade. Quando criei o Scrum, pensei: *e se eu conseguir pegar os padrões humanos e torná-los positivos, em vez de negativos? E se eu conseguir desenvolver um ciclo virtuoso que sempre se reforce e que promova o que temos de melhor e diminua o que temos de pior?* Ao dar ao Scrum um ritmo

diário e semanal, acho que o que eu estava tentando era oferecer às pessoas a chance de *gostarem* da pessoa que veem no espelho.

Mas existem armadilhas. O que parecem padrões virtuosos pode acabar sendo nada além de tolices — nada além de desperdício. É isto que vou abordar neste capítulo: o desperdício que infecta o nosso trabalho, o câncer que acaba com a nossa produtividade, a nossa organização, a nossa vida e a nossa sociedade.

Outro dia, eu estava entrevistando um candidato a uma vaga na Scrum Inc., e perguntei por que ele queria trabalhar em uma empresa Scrum. Então, ele me contou que trabalhava em uma empresa que produzia livros e produtos afins, como livros de atividades, materiais didáticos, apresentações etc. O trabalho dele era identificar os principais estudiosos em um campo específico e trabalhar junto com eles para desenvolver tais produtos. De certa forma, era excitante. Ele era formado em história, um estudioso do período colonial dos Estados Unidos, e tinha a chance de trabalhar com as principais mentes em seu campo.

“Trabalhei lá por um ano”, contou ele. “Um ano que passei desenvolvendo dezenas de produtos diferentes. Ao final daquele período, analisamos pela primeira vez tudo o que tínhamos conseguido. E metade do trabalho foi jogada fora. Não porque eu não fosse bom, mas porque não havia mercado para aqueles produtos, ou a direção tinha mudado. Seis meses da minha vida foram totalmente desperdiçados.”

Naquele ponto, um certo tom de indignação e raiva transpareceu na sua voz. E, então, determinação. “Eu espero que o Scrum não deixe isso acontecer, que o meu trabalho tenha um objetivo, e que eu realmente importe”.

Talvez você ache que esse é um exemplo extremo. Cinquenta por cento de desperdício. Na verdade, porém, esse resultado até que estava bom. Quando entro em uma empresa, em geral, me deparo com cerca de 85% de esforço desperdiçado. Apenas um sexto do trabalho feito realmente produz algo de valor. Bem no fundo, enquanto repetimos o ritmo dos nossos dias, sabemos que isso é verdade. É por isso que rimos, às vezes meio nervosamente, quando ouvimos piadas sobre a insanidade e o desperdício de vida que encontramos em uma corporação moderna.

Estou aqui para dizer para vocês que isso não deveria ser motivo para riso. Isso deveria ser motivo para vergonha. Deveríamos chorar pela vida e pelo potencial que estamos desperdiçando. Eu fiz uma breve apresentação de Taiichi Ohno da Toyota no primeiro capítulo deste livro, e o que ele disse:

“desperdício é mais um crime contra a sociedade do que uma perda nos negócios”. Sua opinião sobre o desperdício teve profunda influência em mim, e eu quero investir algum tempo falando sobre isso.

Ohno fala sobre três tipos diferentes de desperdício. Ele usou palavras japonesas: *Muri*, desperdício causado pela irracionalidade; *Mura*, desperdício causado pela inconsistência; e *Muda*, desperdício causado pelos resultados. Essas ideias estão bastante alinhadas com o ciclo PDCA de Deming, sobre o qual escrevi um pouco antes: **P**lan [planejar], **D**o [fazer], **C**heck [verificar], **A**ct [agir]. **Planejar** significa evitar o *Muri*; **Fazer**, evitar o *Mura*; e **Verificar**, evitar o *Muda*. **Agir** é sinônimo de vontade, motivação e determinação para fazer tudo isso. Vou abordar todos esses passos, um de cada vez, e apontar o que se deve evitar — desde o desperdício no estoque, até o desperdício de se trabalhar demais e o desperdício emocional de expectativas irracionais.[24]

Fazer uma coisa de cada vez

Costumo ouvir pessoas se gabarem da sua capacidade de serem multitarefa. Tenho certeza de que você também. Se você mesmo não faz isso, você conhece alguém que faz — o cara que gerencia três projetos de uma vez, que dirige e fala ao celular ao mesmo tempo, que promove sua competência ao reclamar em alto e bom som sobre o malabarismo que tem de fazer para conseguir cumprir todas as suas tarefas todos os dias. Esse tipo de “ostentação de ocupação” está se tornando parte da nossa cultura de trabalho. Nas descrições de tarefas de um cargo, agora se veem requisitos como “deve ser capaz de gerenciar cinco projetos simultâneos de forma equilibrada”.

A capacidade de fazer malabarismo parece tão *atraente* — principalmente em uma era na qual as informações fluem por milhares de canais diferentes e a frase “preciso disso *pra* agora” está se proliferando. Queremos ser o *cara* — o *supermalabarista*. Dizemos para nós mesmos que vamos conseguir. Infelizmente, *não vamos conseguir*. E quanto mais acharmos que vamos conseguir, pior ficamos.

Um exemplo bem esclarecedor é a prática de dirigir e falar ao celular ao mesmo tempo. E as pesquisas são muito claras sobre isso: as pessoas que dirigem enquanto falam ao telefone envolvem-se em mais acidentes do que as que não o fazem. O problema é ainda mais alarmante quando você considera que, de acordo com a National Highway Transportation Safety Administration,

uma agência norte-americana de segurança nos transportes, a qualquer dado momento, 8% das pessoas nas estradas estão falando ao telefone celular.

Esse é o legado do conceito de multitarefa.

Eis uma citação do meu artigo favorito sobre o assunto:

> […] mesmo quando os participantes direcionam o olhar para objetos enquanto estão dirigindo, eles não conseguem "enxergá-los" quando estão falando ao celular porque sua atenção está afastada do meio externo e concentrada no meio interno, um contexto cognitivo associado à conversa telefônica.[25]

Isso mesmo, as pessoas realmente olham para um objeto, o carro cuja traseira eles estão prestes a bater, ou a árvore na qual eles vão enfiar o carro, mas *elas não veem*. Ainda assim, nós insistimos em dirigir e falar ao telefone ao mesmo tempo.

Eu consigo ler a sua mente agora mesmo. Você está pensando "bem, outras pessoas não conseguem fazer isso. Mas eu vou conseguir, porque sou um executivo poderoso" ou "Eu sou uma mulher inteligente. Eu consigo; eles não". Os estudos, porém, são bastante claros sobre o assunto: se você acha que é bom nisso, você é, na verdade, pior do que todo mundo. A Utah University, que realizou uma série de pesquisas interessantes nessa área, perguntou às pessoas se elas achavam que eram boas em multitarefas — tais como usar o telefone celular ao dirigir, e depois as testou para conferir se estavam certas. Eis as conclusões dos pesquisadores:

> Descobrimos que a percepção das pessoas acerca da própria capacidade de serem multitarefa é muito exagerada; na verdade, a maioria dos participantes se considerava acima da média. Tais estimativas tinham muito pouca fundamentação na realidade. Assim, parece que as pessoas mais propensas a realizar multitarefas e usar o celular enquanto dirigem são aquelas que apresentam uma percepção exagerada de suas capacidades.[26]

O autor principal do estudo, David Sanbonmatsu, disse ao blog da NPR, *Shots*, em janeiro de 2013: "as pessoas não são multitarefas porque são boas nisso. Elas fazem porque são mais distraídas. Elas não conseguem inibir o impulso de fazer outra atividade". Em outras palavras, as pessoas que fazem diversas coisas ao mesmo tempo *não conseguem* se concentrar, elas não conseguem evitar isso.

Eu provavelmente não deveria estar usando "elas", mas "nós". Todos nós fazemos isso. É difícil evitar. E o principal a ser lembrado aqui é que fazer isso

é burrice. Quero que você faça um pequeno exercício, que eu sempre aplico nos meus cursos de treinamento. É simples, mas mostra o profundo impacto do foco e do fluxo. Mostra como multitarefas são árduas para o nosso cérebro, e como isso reduz o seu ritmo, mesmo que você acredite que está aumentando. Demonstra como essas atividades *desperdiçam* tempo e energia.

A tarefa consiste em escrever os números de 1 a 10 em algarismos arábicos e romanos (I, II, III, IV etc.), e as letras do alfabeto, do A ao L. Prepare-se para fazer isso. Você quer realizar a tarefa o mais rápido possível. Mas eu quero que você faça isso da primeira vez da seguinte forma: escreva os numerais cardinais, os algarismos romanos e as letras do alfabeto, de forma que o papel fique assim:

1	I	A
2	II	B
3	III	C

Você deve fazer uma linha de cada vez. Cronometre seu tempo. Eu vou fazer junto com você. Eu levei 39 segundos. Agora, em vez de fazer por linha, faça por colunas. Então, primeiro você fará os numerais cardinais, em seguida, os algarismos romanos e, por último, as letras do alfabeto. Vou fazer também. Levei 19 segundos. Simplesmente por fazer uma de cada vez, em vez de ficar trocando um contexto pelo outro, eu cortei pela metade o tempo que levei para realizar a tarefa.

“Tudo bem, Sutherland”, ouço você dizer, “isso pode ser verdade para falar ao telefone e dirigir e fazer listas idiotas, mas eu dirijo uma empresa. Eu tenho de fazer um monte de coisas ao mesmo tempo — eu tenho equipes trabalhando em cinco projetos simultâneos. Eu tenho que me manter competitivo. Eu não tenho condições de fazer diferente”.

Aqui é onde a incrível quantidade de pesquisa realizada em projetos de desenvolvimento de software se torna muito útil novamente. Lembre-se: as pessoas fizeram essa pesquisa porque continuavam desperdiçando centenas de milhões de dólares todos os anos, e seus produtos só pioravam. Então, sendo os engenheiros que eram, começaram a procurar dados para mensurar tudo. Há um diagrama excelente que aparece em dos trabalhos clássicos sobre como desenvolver software para computadores, “Software com qualidade”, por

Gerald Weinberg:[27]

Número de projetos simultâneos	Porcentagem do tempo disponível por projeto	Perda com a troca de contexto
1	100%	0%
2	40%	20%
3	20%	40%
4	10%	60%
5	5%	75%

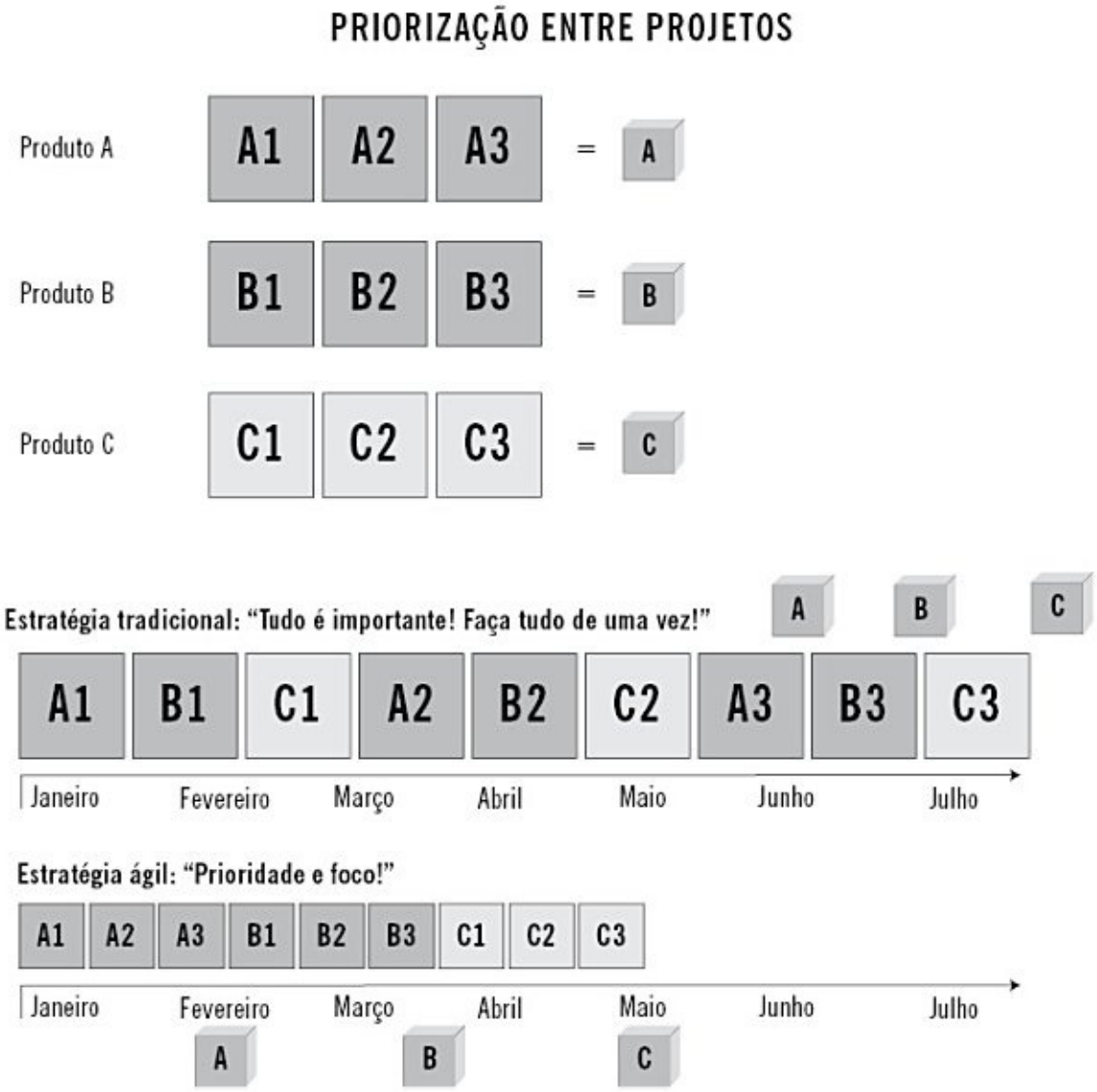
A coluna “Perda com a troca de contexto” é puro desperdício. Isso mesmo: se você tem cinco projetos, cerca de 75% do seu trabalho não vai a lugar algum — três quartos do seu dia são jogados na lata de lixo. É por isso que você não consegue escrever as linhas e as colunas com a mesma velocidade: isso se deve às limitações físicas do cérebro.

Um cientista chamado Harold Pashler demonstrou isso no início da década de 1990, e o chamou de “Interferência dual de tarefas”. Conduzindo alguns experimentos, ele pedia a um grupo para realizar uma tarefa bastante simples, como, digamos, pressionar um botão se uma luz se acendesse. Então, pedia a outro grupo para fazer o mesmo, e mais outra tarefa simples, como pressionar um botão diferente dependendo da *cor* da luz. Assim que outra tarefa foi acrescentada, não importando o quão simples, o tempo envolvido dobrava. Pashler teorizou que havia algum tipo de gargalo de processamento — que as pessoas só conseguem pensar sobre uma coisa de cada vez. Ele avaliou que certo esforço está envolvido em “empacotar” um processo, chegando à sua memória e extraindo outro e, então, voltando ao trabalho. E cada vez que você troca as tarefas, tal processo leva determinado tempo.[28]

Como resultado, você não faz isso, porque se concentra totalmente em uma coisa de cada vez. Você fala ao celular, e mesmo que só esteja falando sobre comprar leite, você literalmente não consegue enxergar o carro à sua frente. O seu cérebro não consegue processar duas coisas ao mesmo tempo. Há alguns estudos recentes que usaram imagens por ressonância magnética para mapear o cérebro que está pensando. Os dados mostram que é possível pensar em duas coisas ao mesmo tempo com apenas um processo ocorrendo em cada lobo cerebral. Mesmo assim, porém, as imagens indicam que o pensamento não ocorre de forma simultânea; em vez disso, o cérebro alterna de uma tarefa para

outra de forma seriada. Basicamente, existe uma função de controle, sendo impossível discutir consigo mesmo de forma vigorosa demais.[29]

Então, voltemos ao trabalho. O que isso significa quando você tenta fazer com que as tarefas sejam realizadas? Bem, vamos analisar uma equipe típica. Naquele ano, eles decidiram fazer três projetos — vamos chamá-los A, B e C. E eles planejaram seu ano, dizendo que vão fazer um pouco para este projeto, depois, um pouco para o outro e, em seguida, o outro, então o calendário deles era mais ou menos assim:



Ao tentarem fazer tudo de uma vez — a estratégia clássica —, concluirão os três projetos até o final de julho. Mas, se utilizarem a abordagem Scrum, ao levar cada projeto para a coluna "Feito", um de cada vez, eles minimizarão os

custos da troca de contexto e conseguirão finalizar tudo até o início de maio.

Eles não mudam o tamanho do projeto, nem os elementos envolvidos na sua criação, mas apenas fazendo uma coisa exclusivamente antes de avançar faz com que a tarefa leve um pouco mais do que a metade do tempo. *Metade*.

E quanto à outra metade? Puro desperdício. Nada mais é produzido. Nem um dólar economizado, nenhuma inovação implementada. Trata-se apenas de um desperdício de vida humana. É o trabalho sem propósito.

Então, esse é o custo da multitarefa. Todos nós vivemos em um mundo no qual existem múltiplas demandas do nosso tempo. As pessoas querem coisas diferentes de nós: o telefone toca e trata-se de uma ligação muito importante, as crianças chegam da escola, o chefe entra no nosso escritório. O que eu quero, porém, é que você se torne *consciente* do custo da troca de contexto. Ela é real e você deve tentar minimizá-la.

Se você está trabalhando em algo complexo — por exemplo, elaborar um relatório, criar uma apresentação, desenvolver um software ou editar um livro —, você tem um objeto incrivelmente complexo na sua mente. Você precisa levar em consideração dezenas de fatores, lembrar-se do que você já fez, qual é seu objetivo e quais obstáculos podem surgir, e isso é bastante complicado. Então, o que acontece se você é interrompido ou se tem de trocar rapidamente para outro projeto, mesmo que apenas por um instante? Você já deve ter adivinhado: aquela arquitetura mental cuidadosamente construída desmorona. Pode levar *horas* de trabalho apenas para voltar ao mesmo estado de consciência e atenção. Esse é o custo. Então, minimize esse desperdício tentando fazer *de uma vez só* todas as tarefas que exigem um tipo específico de concentração. Divida essas tarefas em blocos de tempo, nos quais será possível desligar o telefone e colocar na porta uma placa “Não perturbe”.

Já foram realizadas algumas pesquisas que demonstram que ser multitarefa não apenas desperdiça o seu tempo, mas também torna você burro. Um estudo feito na Universidade de Londres, em 2005 (reconhecidamente um estudo bem pequeno e não revisado por especialistas, mas ainda assim importante),[30] mensurou o quanto uma pessoa emburrece ao realizar multitarefas. O psiquiatra Glenn Wilson testou o QI de quatro homens e quatro mulheres em dois momentos: um calmo e outro cheio de distrações (telefone tocando, e-mails chegando). Durante os testes, ele mensurou a condutância térmica da pele, a frequência cardíaca e a pressão arterial. E, curiosamente, o QI médio caiu mais de dez pontos quando eles se encontravam em ambientes com distrações. Ainda mais curioso foi o fato de essa queda ser mais acentuada em homens do

que em mulheres (talvez, por algum motivo, as mulheres estejam mais habituadas a distrações).

Fazer pela metade não é o mesmo que fazer

Conforme mencionei, o Scrum pega muito dos seus conceitos no modelo de manufatura dos japoneses que foram apresentados no livro clássico *O sistema Toyota de produção: além da produção em larga escala*, escrito por Taiichi Ohno. Nos Estados Unidos, esse modelo foi considerado uma produção "enxuta". Basicamente, a ideia é eliminar ao máximo possível o desperdício no chão de fábrica. Agora, a maioria de nós não procura melhorar o fluxo em uma fábrica de automóveis, mas algumas das ideias são aplicáveis a qualquer tipo de trabalho.

Um conceito que quero abordar é o "Trabalhar em processo" ou "Inventário". A ideia é que é um desperdício ter um monte de coisas à sua volta se elas não forem usadas para construir algo. Essas coisas, sejam elas portas de carro ou peças, custam dinheiro e, se estão no chão de fábrica, significa que uma grande quantia de dinheiro está empatada em um estoque que, na verdade, não é necessário naquele momento. Isso muda a maneira como você olha as coisas que estão *em processo*. Se, por exemplo, tudo o que uma fábrica de automóveis tem é um monte de carros pela metade, ela gastou muito dinheiro e esforço, mas não criou nada de valor real. Na produção enxuta, a ideia é minimizar pela metade a quantidade de coisas construídas espalhadas pela fábrica.

O poder dessa ideia se aplica a todo tipo de trabalho. Vamos pegar algo bem simples que quase todo adulto casado no planeta tem: uma lista de tarefas domésticas. Em qualquer semana, a minha lista conta com dez a vinte tarefas que preciso cumprir. Elas variam desde pintar novamente o banheiro até comprar mais ração para o cachorro, ou pagar a hipoteca e limpar o jardim. Essas são tarefas do cotidiano, o ônus de ser um membro totalmente integrado à sociedade. Agora, existem várias maneiras de cumprir as tarefas da tal lista. Mas o maior erro que você pode cometer é tentar fazer cinco coisas de uma vez. Isso é ser multitarefa, e você provavelmente não conseguirá fazer tudo, o que vai deixá-lo com um "trabalho em processo".

Imagine (ou, se já aconteceu com você, lembre-se) que você tem cinco

tarefas *parcialmente* feitas. Você pintou uma das paredes do banheiro, a ração do cachorro ainda está no porta-malas do carro, e o cheque da hipoteca já está preenchido, mas ainda não foi enviado, e você já juntou as folhas do jardim, mas falta ensacá-las. Você se esforçou, mas não criou nada de valor. O valor só é obtido quando as latas de tinta e o pano que protegeu o ladrilho já não estão mais no banheiro, o cachorro já foi alimentado, o banco recebeu o dinheiro e o jardim está limpo. Fazer as coisas pela metade é o mesmo que não fazer nada.

Conforme eu disse, no Scrum, existe um ritmo para o trabalho. A cada iteração ou Sprint, a equipe tenta concluir determinado número de tarefas. Mas "concluir" significa um produto completo que pode ser entregue e usado por um cliente. Se algo está feito pela metade no final do Sprint, você está pior do que se nem tivesse começado. Você gastou recursos, esforço e tempo e não conseguiu nada que se encontre em um estado que pode ser entregue. Você tem um carro pela metade. Talvez tivesse sido melhor criar algo menor — algo que realmente funcione.

Outra maneira de olhar para o "trabalho em processo" ou "estoque" é simplesmente considerá-lo um estoque físico. Vamos pegar carros como exemplo. Ter toneladas de carros parados sem serem vendidos é um problema para um fabricante. Mas não ter carros disponíveis para serem vendidos também é um problema. Então, cada fabricante e agência de vendas entram em um acordo cuidadosamente equilibrado. Eles querem produzir veículos o suficiente para manter o estoque disponível, mas não tantos que sejam obrigados a investir quantias enormes de dinheiro em coisas que não estão vendendo.

Vamos colocar números reais na questão. Em dezembro de 2012, a General Motors começou um corte de pessoal em suas fábricas nos Estados Unidos. Por quê? A empresa tinha fabricado carros demais. No final de novembro daquele ano, eles tinham 245.853 picapes completas paradas em revendedoras espalhadas por todo o país. Aquilo representava 139 dias de caminhonetes. Considerando o preço médio, aqueles veículos não vendidos representavam US$ 7,5 bilhões. Bilhão com *b*. Todo esse dinheiro, nesse caso, em forma de caminhonetes, mas dinheiro mesmo assim, paradas lá, sem terem sido vendidas. Então, eles começaram a fechar fábricas e a despedir pessoas antes do Natal.

Quantos dias de estoque uma fábrica de carros *deve* ter? O padrão da indústria é cerca de sessenta dias — menos da metade do que a GM tinha. Pense nisso. Quando você compra ração para o cachorro no supermercado, não

quer comprar um suprimento para seis meses. Isso ocupa lugar na garagem e pode custar muito caro, de forma que o cheque da hipoteca não poderá ser emitido naquele mês.

Agora, você pode pensar “ei, eles *construíram* carros; pelo menos essa parte fizeram certo, não é?”. Não eram carros *pela metade* — qual é o problema? O problema é que estoque muito alto é a mesma coisa que “trabalho em processo”. Se você investe muito valor em coisas que não estão entregando valor, não terá recursos para fazer outras coisas — tais como anunciar mais, estimular mais as vendas ou explorar novas ideias. Você tem que ter *algum* estoque; a chave é minimizá-lo.

O trabalho que não é feito e produtos que não estão sendo usados são dois aspectos da mesma coisa: esforço investido sem qualquer resultado positivo. Não faça isso.

Faça certo da primeira vez

O dr. James Womack, fundador do Lean Enterprise Institute no MIT e autor de diversos livros sobre produção enxuta, conta uma história maravilhosa sobre os riscos do “retrabalho” no seu clássico *A máquina que mudou o mundo*. James e sua equipe passaram anos viajando pelo mundo analisando o maior esforço de produção já feito pelos seres humanos: fabricar carros. Ele queria saber por que algumas empresas produziam carros mais rápido e com menos defeitos do que outras. Agora, qualquer fabricante racional usa o que James decidiu chamar de “produção enxuta”, mas, naquela época, as coisas eram diferentes.

Uma das maiores diferenças entre os fabricantes se encontrava no mercado de carros de luxo. No Japão, empresas desse tipo, como a Toyota, a Honda e a Nissan, passavam em média 16,8 horas na fabricação de carros de luxo. As peças chegavam a uma parte da fábrica, e, cerca de 17 horas depois, surgia um Lexus. E eles tinham uma taxa de 34 defeitos a cada cem veículos. Nada mal.

Na Europa, porém, a história era diferente. Empresas como a Mercedes-Benz, a Audi, e a BMW levavam 57 horas para fazer um carro e tinham uma taxa de 78,7 defeitos a cada cem veículos.

Por que os europeus levavam tanto tempo? E por que a taxa de defeitos era tão alta? A BMW não é conhecida por fazer carros ruins. Eis o motivo: na fábrica da Toyota, quando um problema aparece na linha de produção, cada

trabalhador tem a capacidade de parar toda a linha. Quando isso acontece, todos se reúnem na linha que parou — não para brigar com o cara que fez isso, mas para corrigir o problema detectado. Eles não querem que qualquer carro saia na outra ponta com defeitos a serem concertados, querem é corrigir o problema de uma vez, porque assim estará resolvido para sempre. Se não fizerem isso, aquele mesmo defeito poderá aparecer em centenas de veículos.

Os fabricantes europeus de carros de luxo na Europa tinham um modo diferente de fazer as coisas. No final da linha de produção, havia dezenas de pessoas com jalecos brancos para consertar todos os problemas. Eles se certificavam de que a porta do carro tinha aquele *som* da BMW ao se fechar, que o motor ronronava no tom certo, que todas as partes estavam encaixadas da forma correta. Eles não se viam como fabricantes, mas como artesãos, artistas criando uma obra de arte. Isso é ótimo quando você produz poucos carros, mas quando fabrica milhões deles, os custos sobem muito. Conforme Womack relatou em seu livro:

> [...] a fábrica alemã estava dedicando mais esforço na correção de problemas no carro que tinha acabado de fabricar do que a fábrica japonesa precisava para fabricar um carro quase perfeito logo na primeira vez.[31]

Você leu corretamente. Os alemães gastavam mais tempo consertando um carro que tinham acabado de fabricar do que os japoneses para fabricar um carro. Existe um motivo por que a Toyota se tornou a fabricante número 1 de carros do planeta. Eles fizeram certo da primeira vez.

Mas nós nem sempre fazemos as coisas de forma perfeita na primeira vez. Nós somos humanos, cometemos erros. A maneira como você lida com esses erros pode ter um impacto extraordinário na velocidade e no nível de qualidade com que você faz as coisas. Na Toyota, conforme eu disse, cada um dos trabalhadores pode parar a linha. A ideia é que o processo seja aprimorado de forma contínua, e que o momento certo de corrigir um problema seja quando ele é detectado, e não depois.

Alguns anos atrás, eu estava na Califórnia conversando com a equipe de desenvolvimento da Palm. Eles criaram alguns dos primeiros dispositivos chamados “Assistente Digital Pessoal”, que agora nós chamamos de telefone. Eles rastreavam tudo que faziam, automaticamente. Uma das muitas coisas que mediam era quanto tempo demoravam para corrigir um *bug* — ou seja, quanto tempo um desenvolvedor de software levava para corrigir um problema que ele

introduzira no sistema. O computador rastreava isso automaticamente, cada uma das vezes em que ocorria.

Então, digamos que um dia, quando os testadores tentaram integrar o código de Matt no restante do sistema, eles detectassem um *bug*. Matt, como a maioria dos desenvolvedores de software, não ia querer voltar e corrigir aquele código imediatamente. Em vez disso, promete voltar ao problema mais tarde. Primeiro, ele tinha de escrever um *novo* código.

Na maioria das empresas, esse tipo de teste nem acontece no mesmo dia. Poderiam ser semanas ou meses antes que todo o código fosse testado e, só então, o problema seria detectado. Mas a Palm realizava testes diariamente, testes automatizados de todo o código deles, então, sabiam, logo de cara, quando havia um problema.

Eles observaram todos os "Matts" na empresa inteira — centenas de desenvolvedores — e decidiram analisar quanto tempo levava para corrigir um *bug* se o fizessem na hora em que fosse detectado, em comparação com se tentassem *corrigir* semanas mais tarde. Agora, lembre-se, software pode ser algo bem complexo e confuso, então, qual você acha que foi a diferença?

Levou 24 vezes mais tempo. Se um *bug* fosse corrigido no dia em que foi criado, levava uma hora para corrigir; três semanas depois, levava *24* horas. Nem importava se o *bug* era grande ou pequeno, complexo ou simples — sempre levava 24 horas a mais três semanas depois. Como você pode imaginar, todos os desenvolvedores de software passaram a ter de testar e corrigir o próprio código no mesmo dia.

Já escrevi muito sobre os limites da mente humana. Nós só conseguimos lembrar de algumas coisas; só podemos nos concentrar de verdade em uma coisa de cada vez. Essa tendência para o processo de corrigir coisas fica mais difícil quando se passa mais tempo — representa uma limitação semelhante. Quando você está trabalhando em um projeto, cria um espaço em volta dele na sua mente. Você conhece todos os motivos diferentes que explicam *por que* algo está sendo feito, e mantém uma construção bastante complexa na sua cabeça. Recriar isso uma semana depois é *difícil*. É preciso se lembrar de todos os fatores considerados quando se fez determinada escolha, e recriar o processo de pensamento que o levou a tomar determinada decisão. Você tem que se tornar a pessoa que era no passado de novo, colocar-se de volta dentro de uma mente que não existe mais. Fazer isso demora, e demora *muito*. Vinte e quatro vezes mais tempo do que você levaria se corrigisse o problema assim que o descobrisse.

Tenho certeza de que você já passou por essa experiência no seu trabalho, e a lição que aprendeu foi exatamente a mesma que ensinaram quando você ainda era criança: *faça as coisas certas da primeira vez*. O único elemento novo que os dados acrescentam é que, se você cometer um erro — e todos nós cometemos —, corrija-o assim que o notar. Se você não o fizer, vai pagar caro por isso.

Trabalhar demais resulta em mais trabalho

Quando Scott Maxwell, o fundador da firma de capital de risco OpenView Venture Partners, estava trabalhando como consultor da McKinsey & Company no início da década de 1990, recebeu o que considerou ser um discurso de apoio bem estranho. Jon Katzenbach, então diretor da empresa e agora autor de diversos livros e chefe do Katzenbach Center na Booz Allen Hamilton, deu alguns conselhos a Scott dos quais ele jamais se esqueceu. Jon lhe disse que nos anos 1970, quando estava começando, todos trabalhavam sete dias por semana na McKinsey. Aquela era a cultura; aquilo era esperado de todos. Se você não trabalhasse tanto tempo assim, as pessoas achariam que você não estava fazendo a sua parte e que não estava contribuindo com a equipe.

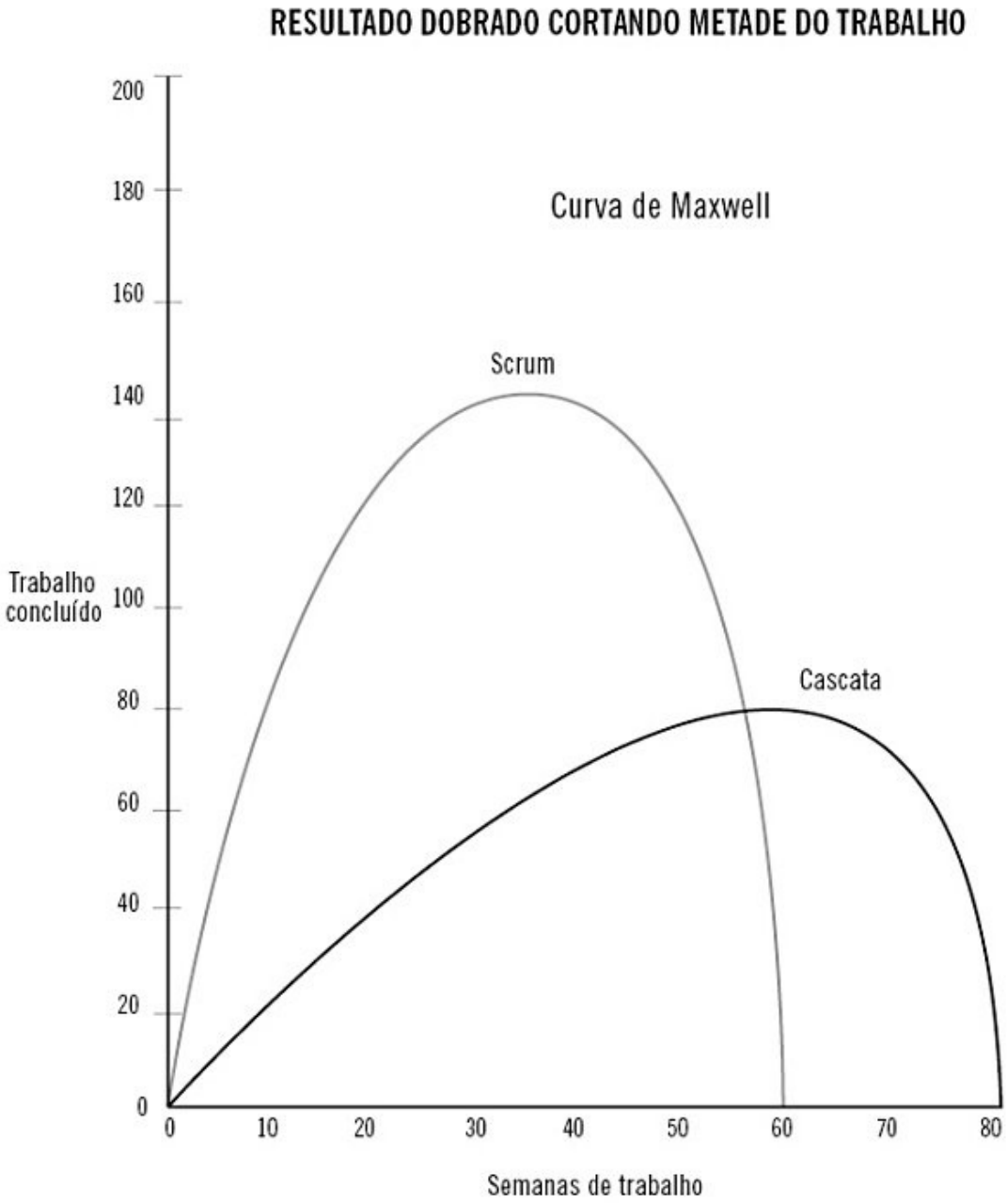
Por questões religiosas, Jon trabalhava apenas seis dias por semana. E notou algo. Mesmo trabalhando menos horas, ele conseguia produzir *mais* do que os outros "caras" — e todos eram caras naquela época — que trabalhavam todos os dias. Então, decidiu tentar trabalhar apenas cinco dias por semana. E descobriu que conseguia produzir ainda mais. "Trabalhe mais", disse ele, "e você produzirá menos".

Scott e os outros jovens consultores zombaram da ideia na época. *Trabalhar menos horas? Isso não é folga demais?* Mas Scott continuou com essa ideia por muitos anos enquanto seguia com a sua carreira, e, como CEO e fundador da OpenView Venture Partners, começou a investir em empresas de tecnologia, algumas das quais praticavam o Scrum. Ele ouviu dizer que eu tinha inventado o Scrum e morava na mesma cidade, então convidou-me para tomar um café da manhã com ele. No meio do café e *croissants*, Scott me contou a história de uma das empresas na qual investira e as equipes de desenvolvimento usavam o Scrum, e de como elas tinham melhorado a produtividade de 25% para 35%. Ele realmente estava impressionado. A minha resposta imediata foi: "25% para 35%? Eles devem estar fazendo alguma coisa errada!!!".

Scott decidiu levar o Scrum para a OpenView e implementá-lo em toda a empresa. Os caras de investimento, o pessoal de pesquisa, a alta gerência, os funcionários administrativos, todos entraram para uma equipe Scrum. E, por fim, algo aconteceu, algo que é uma das grandes vantagens do Scrum: a OpenView descobriu como as pessoas *realmente* trabalhavam em vez de como elas *diziam* trabalhar.

Na época, a OpenView era bem parecida com vários escritórios de alto nível. Arraigada na cultura corporativa estava a expectativa que as pessoas trabalhariam até mais tarde ou nos fins de semana. Eram pessoas agressivas e ambiciosas. Mas elas estavam ficando cansadas, deprimidas e desmoralizadas. Era um ambiente tão duro que algumas pessoas não conseguiam ficar ali e pediam demissão.

No entanto, à medida que a equipe começou a usar o Scrum, Scott notou uma mudança na produtividade. Trabalhar mais horas parou de produzir mais resultados. Ele me chamou ao seu escritório um dia e desenhou uma curva em um quadro branco:



O eixo y é a produtividade, e o eixo x, as horas de trabalho. O pico de produtividade começa a cair um pouco antes de quarenta horas por semana. Munido com esses dados, Scott começou a mandar as pessoas para casa mais cedo.

“Levou um tempo para eles perceberem que eu estava falando sério”, conta Scott. “Mas, por fim, eles entenderam o meu modo de pensar”.

Ele começou explicando que trabalhar até mais tarde não era um sinal de comprometimento, era um sinal de fracasso. “Não é porque eu quero que vocês

tenham uma vida equilibrada", dissera ele ao pessoal, "é porque vocês vão conseguir produzir mais".

Assim, não havia mais trabalho à noite ou nos fins de semana. Quando as pessoas entravam de férias, esperava-se que realmente entrassem de férias, nada de checar e-mails ou ligar para o escritório. Se você não pode realmente tirar um tempo sem precisar se certificar de que tudo está indo bem no escritório, parece que você não está gerenciando bem a sua equipe.

"Várias empresas não praticam isso [limite de horas de trabalho]", diz Scott. "Mas existe uma correlação direta. Você consegue uma produtividade maior. Você é mais feliz. E você tem mais qualidade". É óbvio. Trabalhar menos aumenta a produtividade e a qualidade.

Scott explica que a curva é diferente para pessoas diferentes, até para uma mesma pessoa em épocas diferentes de sua vida. "Conforme fui ficando mais velho e assumi diferentes papéis, notei que o pico de produtividade para mim é um número menor de horas do que era vinte anos atrás", explica ele. Ele acredita que boa forma física, dieta, questões pessoais e outros fatores representam um papel cada. Mas também acredita que a sua produtividade chega ao pico mais rápido conforme ele cresceu e passou a pensar profundamente no modo como trabalha. "Eu me tornei capaz de aproveitar cada vez mais oportunidade importantes".

Por que *será* que se você trabalhar menos horas, consegue uma produção maior? Não parece realmente fazer sentido. Scott explica que as pessoas que trabalham horas demais começam a cometer erros, o que, como já vimos, pode exigir mais esforço, porque consertar leva mais tempo do que para criar. Funcionários que trabalham além da conta tendem a se distrair mais e a começar a distrair os outros. Logo eles começam a tomar decisões ruins.

Os instintos de Jon Katzenbach estavam certos. As provas inquietantes revelam que temos uma capacidade bastante limitada de tomar decisões, especialmente quanto mais cansados e menos tempo de descanso tivermos.

Em abril de 2011, um grupo de pesquisadores israelenses publicou alguns estudos notáveis sobre a tomada de decisões, no periódico *Proceedings of the National Academy of Sciences of the United States of America*. O artigo deles, intitulado "Extraneous Factors in Judicial Decisions" ["Fatores externos em decisões judiciais"], analisou mais de mil decisões judiciais determinadas por juízes israelenses que presidiram dois tipos diferentes de comissão de liberdade condicional. As decisões abrangiam criminosos judeus-israelenses e árabes-israelenses — tanto do sexo feminino quanto do masculino. Os crimes

variavam desde fraude e agressão até assassinato e estupro. A grande maioria das decisões que os juízes analisavam era de pedidos para liberdade condicional.[32]

Parece bastante claro, certo? Esses estimados juízes usando seus anos de experiência e sabedoria para tomar decisões críticas que afetam não apenas a vida dos prisioneiros e de suas vítimas, mas também a comunidade como um todo. A cada dia, eles ouviam entre 14 e 35 casos.

Então, se você fosse um prisioneiro, qual seria o maior fator a ser considerado para determinar a sua sentença? Arrependimento verdadeiro, talvez? A sua transformação e seu comportamento na prisão? A gravidade do crime cometido? Nada disso, na verdade. Acontece que o que *realmente* importou foi quanto tempo fazia desde que o juiz tinha comido um sanduíche.

Os pesquisadores observaram o horário em que os juízes tomaram suas decisões, se tinham demonstrando clemência e quanto tempo fazia desde que tinham lanchado. Se eles tivessem acabado de chegar ao trabalho, ou se tivessem voltado de um intervalo para comer alguma coisa, ou se tinham voltado do almoço, eles tomavam decisões favoráveis em mais de 60% dos casos. Essa taxa caía para perto de zero conforme se aproximava o horário do intervalo seguinte.

Basicamente, um pouco depois de um intervalo curto, os juízes apresentavam uma atitude mais positiva e tomavam decisões mais lenientes. Eles demonstravam mais imaginação e capacidade de ver que o mundo e as pessoas podiam mudar, podiam ser diferentes. Mas, à medida que queimavam suas reservas de energia, começavam a tomar cada vez mais decisões que mantinham o *status quo*.

Tenho certeza de que se você perguntasse a esses juízes se eles tinham certeza de que cada uma das decisões era igualmente boa, eles se sentiriam afrontados. Mas os números e os sanduíches não mentem. Quando você não tem mais nenhuma reserva de energia, tende a começar a tomar decisões infundadas.

Esse fenômeno foi chamado de “esgotamento do ego”. A ideia é que fazer qualquer escolha envolve um gasto de energia. É um tipo estranho de exaustão — você não se sente fisicamente cansado, mas a sua capacidade de tomar boas decisões diminui. O que realmente muda é seu autocontrole — sua capacidade de ser disciplinado, cuidadoso e previdente.

Um experimento fascinante foi preparado para demonstrar exatamente isso. Um grupo de pesquisadores queria saber de que forma a tomada de decisões

afeta o autocontrole. Então, eles pegaram os soldados rasos da pesquisa psicológica — alunos universitários — e pediram que um grupo deles tomasse uma série de decisões. Foram apresentados a esses alunos diferentes produtos, e pediu-se que indicassem quais eles preferiam. Eles foram orientados a pensar com cuidado porque receberiam um presente ao final do experimento, e sua preferência afetaria o que receberiam. O outro grupo de alunos não precisou tomar qualquer decisão.[33]

O grupo de teste teve de responder a perguntas como “que aroma de vela perfumada você prefere, baunilha ou amêndoas?”; “que marca de xampu você prefere?”; “você prefere esta bala ou aquela?”. Então, eles tinham de fazer o teste clássico de autocontrole: “quanto tempo você consegue manter sua mão na água gelada?”.

Seja qual for o recurso gasto na tomada de decisão, ele também é usado no autocontrole. Os alunos que tomaram todas as decisões sobre os produtos simplesmente não conseguiram manter a mão na água gelada por tanto tempo quanto os que não tiveram de tomar nenhuma decisão.

Então, existe um número limitado de decisões sensatas que você pode tomar em qualquer dia, e, à medida que vai tomando decisões, diminui a capacidade de controlar seu próprio comportamento. Você começa a cometer erros — e pode acabar cometendo erros sérios. Conforme a curva de Maxwell mostra, essas decisões ruins têm um impacto na produtividade. Então, volte para casa às cinco horas da tarde. Desligue o celular no fim de semana. Assista a um filme. Talvez mais importante: coma um sanduíche. Ao não trabalhar demais, você vai conseguir produzir mais e com mais qualidade.

O Scrum pede às pessoas que adotam esse método para quebrar o paradigma de medir o trabalho apenas em horas. A hora por si só representa um custo. Em vez disso, a medida deve ser feita em *resultados.* Quem se importa com o número de horas que alguém trabalhou em algo? Tudo que importa é a rapidez e a qualidade com que ele é entregue.

Seja razoável

Existem três tipos de desperdício identificados por Taiichi Ohno que levam as pessoas a se dedicarem mais e por mais tempo ao trabalho do que realmente o necessário. Eu já expliquei por que essa é uma ideia incrivelmente horrível, mas reconhecer esses tipos de desperdício, que Ohno chamou de *Muri* ou

“irracionalidade”, talvez constitua as engrenagens mais poderosas de mudança que conseguiremos alcançar.

A primeira é a “Falta de lógica”. Você quer dar à sua equipe objetivos desafiadores — fazer com que queiram mais. Mas você *não* vai querer que eles lutem por objetivos absurdos ou impossíveis.

A segunda é “Expectativas exageradas”. Quantas vezes você já ouviu alguém se gabando de que seus esforços heroicos salvaram um projeto? Em geral, isso é recebido com tapinhas nas costas, alegria e congratulações. Eu enxergo isso como uma falha fundamental no processo. Uma equipe que depende de ações heroicas constantes para cumprir os prazos não está trabalhando da maneira que deveria. Entrar e sair constantemente de uma crise para outra provoca um esgotamento e não deixa espaço para aprimoramentos contínuos e racionais. É a diferença entre um *cowboy* saindo em disparada para salvar uma mocinha de um bandido e um fuzileiro naval disciplinado liberando uma zona tomada por soldados inimigos.

Ohno chamou o último tipo de desperdício de “Sobrecarga”. É o tipo de comportamento que Scott Adams costuma satirizar nas suas tirinhas de *Dilbert*. Inclui diretrizes corporativas opressoras que atrapalham, relatórios desnecessários que os funcionários têm de fazer e formulários que precisam ser preenchidos somente por preencher, além de reuniões sem sentindo que desperdiçam tempo e não produzem qualquer valor.

Embora Ohno não mencione um quarto tipo de desperdício, ainda há mais um que me vem à mente: o “desperdício emocional”. Esse tipo é gerado quando uma empresa tem um escroto no seu meio — alguém que gosta de sacudir as pessoas e deixá-las apreensivas. Pessoas escrotas costumam justificar o próprio comportamento afirmando que isso faz os colegas trabalharem melhor. Mas eles estão apenas cedendo aos aspectos negativos da própria personalidade, e nada é mais corrosivo para a capacidade de uma equipe para se sobressair do que isso.

Não seja escroto — e não permita, estimule ou aceite esse tipo de comportamento nos outros.

Fluxo

Em um mundo teoricamente perfeito, não haveria processos, reuniões, formulários ou relatórios. No lugar disso, haveria a criação *exata* daquilo que o

cliente deseja, mesmo que o cliente ainda não o saiba. Qualquer “processo” que as pessoas usam é desperdício, e isso inclui o Scrum.

Mas nós não vivemos em um mundo perfeito, e processos ruins estão tão enraizados em nossa forma de pensar que, como alternativa, precisamos de um processo o mais leve possível e com maior impacto no trabalho. O que o Scrum faz é se concentrar na tentativa de eliminar os desperdícios . Tentei criá-lo de forma que o processo em si seja uma estrutura o menos perturbadora possível, e ainda mantenha as pessoas concentradas.

O que você realmente quer no seu trabalho é um “fluxo” fácil. Nas artes marciais ou na prática de meditação, quando você alcança um senso de unidade em movimento, a atividade passa a não ser mais um esforço, mas uma energia que flui facilmente por você. Quando você assiste a grandes dançarinos ou cantores, sente que eles estão se rendendo a uma força maior do que eles mesmos, enquanto permitem que a arte se mova através deles. Chegar a esse ponto no nosso trabalho deveria ser o nosso objetivo.

No entanto, assim como o mestre de kung fu, o monge, o dançarino ou um astro da ópera dirão a você, a raiz do fluxo está na disciplina. Não pode haver desperdício de movimentos — nada irrelevante — apenas a aplicação concentrada da capacidade humana. O desperdício é qualquer coisa que o distraia disso. Se você começar a pensar no seu trabalho em termos de disciplina e fluxo, talvez consiga fazer algo incrível.

PONTOS PRINCIPAIS

Ser multitarefa emburrece. Fazer mais de uma coisa de cada vez diminui seu desempenho tanto em termos de tempo quanto de qualidade nas duas tarefas. *Não faça isso.* Se você acha que isso não se aplica a você, saiba que está errado — afeta sim.

Fazer pela metade não é fazer. Um carro pela metade só serve para prender recursos que poderiam ter sido usados para criar algo de valor ou economizar dinheiro. Qualquer coisa que esteja “em processo” custa dinheiro e energia, sem entregar nada.

Fazer certo da primeira vez. Quando você cometer um erro, corrija-o imediatamente. Pare todo o resto e resolva o problema que tem em mãos. Corrigir mais tarde pode levar vinte vezes mais tempo do que corrigir agora.

Trabalhar demais só resulta em mais trabalho. Trabalhar até tarde não resulta em *mais* produtividade; ao contrário: *diminui a produtividade*. Trabalhar demais resulta em fatiga, que leva a erros, que leva a necessidade de consertar algo que você acabou de terminar. Em vez de trabalhar até tarde ou nos fins de semana, trabalhe apenas nos dias de semana e apenas em um ritmo sustentável. E tire férias.

Não seja irracional. Os objetivos que são desafiadores estimulam a equipe; objetivos impossíveis são

apenas desanimadores.

Nada de heroísmo. Se você precisa de um herói para conseguir concluir o trabalho, você tem um problema. O esforço heroico deve ser considerado um fracasso no planejamento.

Chega de diretrizes idiotas. Qualquer diretriz que *parece* ridícula provavelmente *é ridícula mesmo.* Formulários idiotas, reuniões idiotas, aprovações idiotas, padrões idiotas são apenas isso — idiotices. Se o seu escritório parece tirado de uma tirinha de *Dilbert,* conserte isso.

Nada de escrotice. Não seja um e não tolere esse tipo de comportamento. Qualquer pessoa capaz de causar caos emocional, inspirar medo ou horror, ou que humilhe e diminua as pessoas, precisa ser detida.

Busque o fluxo. Opte pelo modo mais suave e sem problemas de concluir uma tarefa. O Scrum é sobre possibilitar o maior fluxo possível.

24 Taiichi Ohno, Toyota Production System: Beyond Large-scale Production, Cambridge, MA, Productivity, 1988. [Em português, *O sistema Toyota de produção: além da produção em larga escala.* Porto Alegre, Bookman, 1997.]

25 David Strayer et al., A Comparison of the Cell Phone Driver and the Drunk Driver, *Human Factors, Vol.* 48 (2), 2006, pp. 381-91.

26 D. M. Sanbonmatsu et al., Who Multi-Tasks and Why? Multi-Tasking Ability, Perceived Multi-Tasking Ability, Impulsivity, and Sensation Seeking, *PLoS ONE, Vol.* 8(1), 2013, e54402, doi:10.1371/journal.pone.0054402.

27 Gerald M. Weinberg, Quality Software Management, New York, NY, Dorset House, 1991. [Em português, *Software com qualidade.* São Paulo, Makron Books. Edição esgotada.]

28 Harold Pashler, Dual-task Interference in Simple Tasks: Data and Theory, *Psychological Bulletin, Vol.* 116 (2), 1994, pp. 220-44.

29 S. Charron e E. Koechlin, Divided Representation of Concurrent Goals in the Human Frontal Lobes *Science* 328.5976, 2010 pp. 360-63.

30 Wilson, Glenn. The Infomania Study. Issue brief, http://www.drglennwilson.com/Infomania_experiment_for_HP.doc.

31 James P. Womack et al., The Machine That Changed the World: The Story of Lean Production, New York, NY, HarperPerennial, 1991. [Em português, *A máquina que mudou o mundo.* Rio de Janeiro, Elsevier, 2004.]

32 Avnaim-Pesso et al., Extraneous Factors in Judicial Decisions, *Proceedings of the National Academy of Sciences of the United States of America,* 108.17, 2011.

33 K. Vohs et al., *Decision Fatigue Exhausts Self-Regulatory Resources—But So Does Accommodating to Unchosen Alternatives.*

CAPÍTULO 6

Planeje a realidade, não a fantasia

“Ei, Jeff, nós temos um problema.”

É assim que começam muitas das minhas conversas ao telefone. As pessoas se metem em encrencas, e então resolvem me ligar. Daquela vez era Mark Landy, arquiteto chefe de software da Medco. Nos Estados Unidos, as pessoas que recebem remédios prescritos pelo correio provavelmente conhecem essa empresa. Na época dessa ligação, a Medco figurava na lista Fortune 100, apresentando uma receita de quase US$ 38 bilhões, o que a fazia a maior companhia farmacêutica dos Estados Unidos, com dezenas de milhares de empregados. E a direção tinha acabado de levá-la à beira de um abismo.

Recebi o contato em dezembro de 2006. Naquele mês de julho, o seu presidente, Kenny Klepper, anunciara para Wall Street sua ideia mais recente. Mark Landy a descreveu da seguinte forma: “Estávamos tentando convencer cada vez mais pessoas a receberem seus remédios prescritos pelo correio. E existem algumas barreiras para isso”. Barreiras como o aparecimento de um inconveniente, por exemplo; mas Mark disse que havia maneiras de contorná-lo. “Olha só, quando você vai à farmácia, a experiência é minimamente clínica. Você entrega a prescrição, assina um documento afirmando que não deseja falar com o farmacêutico, e vai embora. Você pode aprimorar essa experiência”.

Uma das coisas que eles queriam fazer era colocar um farmacêutico ao telefone com o paciente — um profissional que conhecesse não apenas tal medicamento, mas todos os medicamentos já prescritos para aquele paciente. Isso era particularmente importante se o paciente apresentasse algum tipo de doença crônica, como diabetes ou doença cardíaca, que é o que ocorre com 80% dos pacientes que tomam remédios regularmente. E a maioria dessas pessoas — principalmente se forem mais idosas — toma seis ou mais

medicamentos de forma concomitante. E seus médicos — especialistas em diferentes campos da saúde — nem sempre sabem disso.

"Os médicos [nem sempre] compartilham informações uns com os outros. Nós, assim como a farmácia, sabemos mais do que eles, e ficamos sabendo disso em tempo real, [antes mesmo] que os planos de saúde", explicou Landy.

Eis a ideia de Klepper: criar farmácias especializadas, para cada tipo de medicamento, em cinco locais diferentes do país. Farmácia para medicamentos cardíacos, para diabetes, asma etc. E os farmacêuticos serão treinados não somente para atuar em cada um desses locais, mas também para conhecer interações medicamentosas, efeitos adversos etc. Assim, como eles terão uma visão mais abrangente, poderão informar aos médicos os casos nos quais talvez haja alguma contraindicação. Por exemplo: digamos que alguém seja diabético. É mais provável que esteja acima do peso e, possivelmente, apresente outras condições. Como resultado, essa pessoa metabolizará medicamentos de forma diferente. Então, se um novo médico prescreve um remédio para controlar a pressão arterial, o farmacêutico da Medco pode ligar e recomendar uma análise do fígado do paciente e um ajuste de dosagem, se necessário.

O objetivo era atrair novos clientes para a Medco, o que, na maior parte das vezes, ajudava os planos e os seguros corporativos de saúde. Ao usar tais farmácias, ou Centros de Recursos Terapêuticos, os clientes poderiam economizar não apenas cortando os custos das suas prescrições, mas também seus custos médicos totais, que aumentam quando as pessoas não tomam remédios de forma correta, ou se os medicamentos que estão tomando não apresentam boa interação um com o outro, ou com aquela pessoa em particular. Além disso, a Medco *garantiria* a economia. Se um cliente não economizasse a quantia que a Medco projetara, cobriria tal diferença.

Wall Street, para falar de forma suave, *gostou* do conceito. Uma ideia bem legal, não é? Economizar e ainda fornecer um serviço de saúde melhor. Mais clientes, mais vendas; ganha-ganha. Só havia um problema: embora Klepper tenha verificado com seus gerentes que a ideia era tecnicamente possível, ele não obtivera detalhes sobre quanto *tempo* esse plano levaria para ser implementado. As pessoas que poderiam fazê-lo funcionar só descobriram isso depois que o presidente da empresa prometera a Wall Street que o novo sistema ficaria pronto em 7 de julho de 2007. Acontecesse o que acontecesse.

Cumprir aquele prazo era particularmente importante para a Medco, porque embora eles fossem os primeiros a começar a automatizar encomendas farmacêuticas por correio, não eram os únicos, e seus concorrentes estavam

famintos. Infelizmente, a empresa tinha vários obstáculos a superar; por exemplo, uma parte grande das aplicações de software dependia de robôs locais que estavam bastante superados. Nas cinco enormes instalações da Medco, contendo quatro mil farmacêuticos processando prescrições, alguns robôs trabalhavam pegando pílulas, enquanto outros lidavam com a embalagem e o envio, e todos aqueles sistemas tinham de falar uns com os outros com um nível de precisão de 100%, ou alguém poderia morrer.

A ideia era que o novo e corajoso plano de Klepper daria à Medco uma chance de atualizar seus sistemas antigos e ficar um passo à frente dos concorrentes. A empresa levou quase seis meses para descobrir que não conseguiria cumprir o prazo. Seus cálculos mostravam que, no melhor cenário possível, eles entregariam o sistema com, pelo menos, um ano de atraso. Provavelmente mais. E foi quando me ligaram.

Por que eles demoraram *seis meses* para descobrir que não conseguiram cumprir uma tarefa no prazo é algo que merece uma análise. Não é porque eles não fossem inteligentes ou não tivessem as equipes certas, nem mesmo a melhor tecnologia. Não é porque não estivessem trabalhando com afinco ou que não fossem competitivos. Não é por isso que você não se torna a maior empresa do seu setor.

Isso aconteceu porque eles cometeram um erro bastante básico: pensaram que poderiam planejar tudo com antecedência. Investiram meses em um esforço de fazer o tipo de plano detalhado que parecesse plausível — organizar lindos diagramas e incluir etapas cuidadosamente planejadas e precisas que, quase sempre, descrevem uma realidade fictícia.

Como eu disse antes, o simples fato de planejar é tão sedutor, tão atraente, que o planejamento se torna mais importante do que o plano efetivo. E o plano, por sua vez, se torna mais importante do que a realidade. Nunca se esqueça: o mapa *não* é o terreno.

Quando uma equipe se reúne pela primeira vez para mapear um projeto, costuma haver eletricidade na sala — um senso de possibilidade, novos mundos a descobrir e novas ideia a experimentar. Trata-se realmente de uma das melhores sensações do mundo.

Então, chega o momento em que a inspiração se transforma em cálculos, e parte daquela energia se dissipa. As pessoas começam a se perguntar *"Como é que vamos conseguir sair do ponto A para chegar ao ponto B? E quando descobrirmos a resposta, quanto tempo vai levar?*".

Infelizmente, a fase de cálculos pode ser um processo tipo "lixo entra/lixo

sai”. As pessoas envolvidas podem ser altamente inteligentes, mas normalmente não *percebem* que estão alimentando seus diagramas com um monte de fantasias que desejavam que fossem realidade.

Quando Mark me explicou a situação na Medco, respondi: “Você realmente tem um problema”. Esperei um pouco e acrescentei: “Mas eu aposto que a gente consegue resolver”.

Um pouco antes do Natal, peguei um avião para Nova Jersey para passar um dia na empresa e determinar o escopo da questão. Não era algo trivial. Havia uma infinidade de documentos definindo os requisitos e as questões de conformidade, todos os tipos de relatórios e processos fase/pontos de decisão e garantia de qualidade. Enterrado ali, em algum lugar, havia o que realmente precisava ser feito, mas que ninguém realmente planejara como *fazer*.

Depois de me reunir com o pessoal-chave por um tempo, liguei para Brent Barton, um orientador Scrum com quem trabalhei em outros projetos. “Brent, eu preciso de você e de quem mais você conseguir arrebanhar para começar em um projeto no início de janeiro”, pedi. “Já temos o trabalho planejado para nós”.

Mais tarde, Brent descreveria a Medco que vira pela primeira vez como uma empresa “encalhada”. Havia tantos interesses e pessoas em desacordo que nada estava sendo feito. Naquele primeiro dia, nós nos reunimos com cerca de sete grupos diferentes, cada qual com a posse de uma parte do projeto, e nenhum deles estava realmente interessado em tentar uma nova abordagem. Mas, ele diz agora:

> A gente podia se dar o luxo do “que merda”. Você pode usar a dor e o medo como aliados quando entra como consultor. Quando a gente se deparava com resistência, só dizia: “ei, vocês podem continuar fazendo as coisas do jeito que sempre fizeram, manter o *status quo* e entregar o produto com atraso, e tudo vai ficar bem”. E eles respondiam: “não vai ficar tudo bem”.

A primeira coisa que fizemos foi chamar todo mundo para a sala de reuniões — todo pessoal-chave, todas as pessoas que realmente estavam trabalhando. E Brent pediu para todos imprimirem os documentos que tinham que descrevessem o que precisava ser feito no projeto. Não, ninguém podia enviar por e-mail; nós queríamos cópias impressas.

Estávamos em uma sala grande, talvez com mais de 15 metros de largura — sem janelas, como essas salas costumam ser, por algum motivo obscuro. No meio, havia uma mesa sobre a qual empilhamos todos os documentos que as

pessoas trouxeram. A pilha de papel tinha, pelo menos, sessenta centímetros de altura.

“Quantos de vocês realmente *leram* tudo isto?”, perguntei.

Silêncio.

“Mas veja bem”, eu disse para um dos gestores, “você assinou este documento. Aqui está a sua assinatura. Você realmente não leu?”.

Seguiu-se mais um instante de silêncio desconfortável.

Eu não queria intimidá-lo, mas o fato era que, em um número cada vez maior de projetos, as pessoas copiavam e colavam e jogavam um texto-padrão, mas ninguém realmente lia aquelas milhares de páginas. Não dava; essa era a questão. Eles definiram um sistema que os forçava a endossar uma fantasia.

Então, Brent e eu pegamos tesouras e fita adesiva, cola e *post-its*. Acontece que você realmente precisa *de tudo que aprendeu* no jardim de infância.

“Eis o que vamos fazer”, começou Brent. “Nós vamos passar por todas essas páginas e cortar tudo que realmente seja algo que precise ser feito para concluir este projeto. Então, vamos colar o que encontrarmos na parede”.

Então, nas duas horas seguintes, foi isso que as pessoas fizeram. No final, tínhamos centenas de apontamentos colados nas três paredes. Na mesa, ainda havia 50% daquela pilha de sessenta centímetros de altura. Duplicação, texto-padrão, modelos. Total e absoluto desperdício.

Então, eu disse para as equipes: “agora, precisamos estimar quanto trabalho cada um desses *post-its* vai levar para ser feito”. Não quanto tempo, mas sim quanto *trabalho*.

Demonstrarei as melhores formas de se fazer isso um pouco mais adiante neste capítulo, já que os seres humanos são, na verdade, péssimos para fazer estimativas de trabalho. Mas eu ensinei a eles um método rápido e porco que é o melhor entre um monte de maneiras ruins de fazer isso, e eles seguiram adiante.

Levaram um tempo, mas conseguiram. Na parede, havia tudo o que precisava ser feito para concluir o projeto, e dividido em tarefas gerenciáveis. Eles estimaram quanto esforço achavam que cada uma delas exigiria; e estavam bastante animados, na verdade. Uma pilha de documentos impossível de ser lida se transformara em partes compreensíveis de trabalho. É como aquela antiga charada: “Como você come um elefante? Um pedacinho de cada vez”.

Uma das coisas-chave que fizemos com cada p*ost-it* foi escrever não apenas o que precisava ser criado, mas também como saberíamos quando aquilo tivesse sido feito. Foi assim que incorporamos todas as exigências de

conformidade do FDA (Food and Drugs Administration), garantia de qualidade e relatórios de processo que precisariam considerar. Nós simplesmente dizíamos “para esta tarefa ser feita, ela precisa atender a estes objetivos”. Acrescentamos aquilo ao projeto no nível de unidade de trabalho, em vez de esperar que tudo fosse feito, para, então, descobrir o que não estava em conformidade com alguma regulação federal ou métrica interna de qualidade. Desse modo, todos na equipe, e não apenas os responsáveis pelas questões de conformidade, tinham de trabalhar para atender primeiro àquele nível de qualidade, antes de seguir para o item seguinte. A quantidade de retrabalho que isso tira de um projeto é inacreditável. Chamo esse padrão do que precisa ser atendido de “Definição de Feito”. Todo mundo sabe quando algo está concluído ou não; existem padrões claros aos quais cada parte do trabalho deve atender.

Ao olhar para todos aqueles *post-its* colados na parede, todos foram tomados por uma sensação de realização; eles agora conseguiam realmente enxergar o que precisavam fazer.

“Tudo bem”, disse Brent, “o que precisamos fazer primeiro?”.

Cerca de cinco pessoas responderam.

“E depois?”

Outras cinco pessoas com ideias diferentes das primeiras.

“E depois?”

O que queríamos que eles fizessem era algo que, às vezes, ninguém quer realmente fazer: *priorizar o trabalho*. Em geral, as pessoas simplesmente dizem que tudo é importante. Mas o que ele estava perguntando era “O que vai trazer mais valor ao projeto? Vamos fazer essas coisas primeiro”.

No final, tínhamos seis fileiras diferentes de *post-its* nas paredes, cada qual com uma cor diferente representando uma equipe diferente. As listas cobriam três paredes da sala. Agora, eu sabia que poderíamos, ao menos, começar.

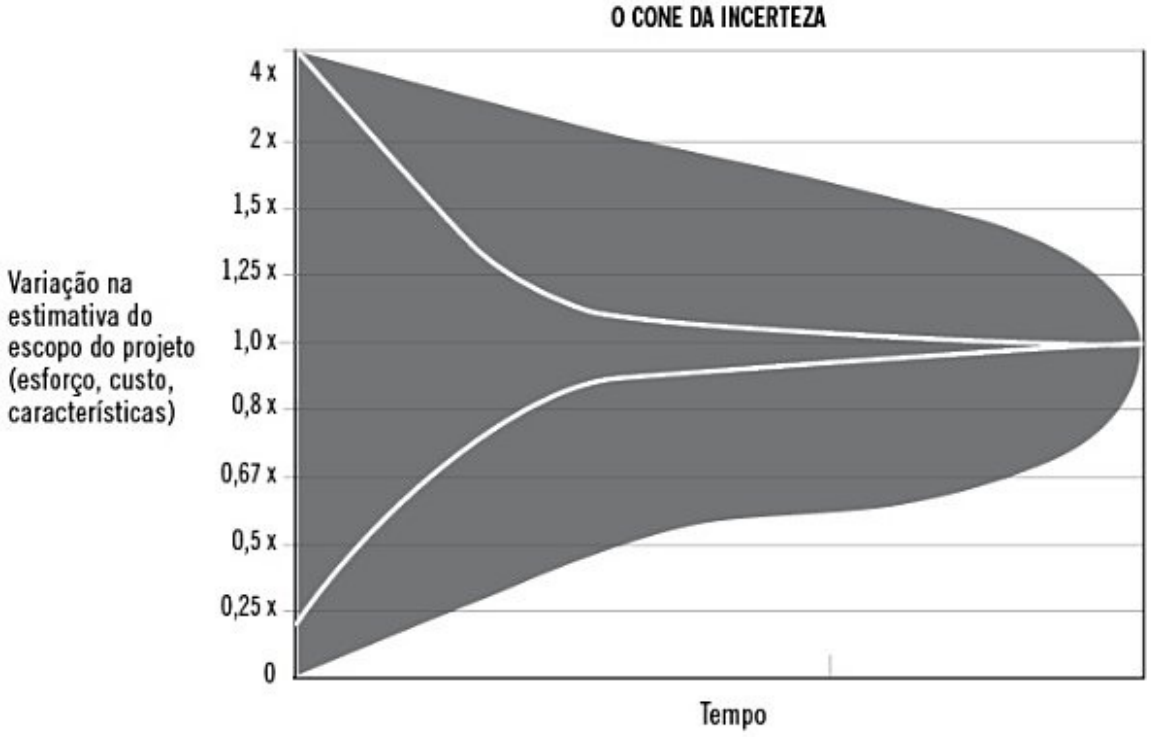
Planejando um casamento

Pode parecer simples quando descrevemos isso, mas permita que eu exemplifique as etapas no processo, usando algo menor em escala: um casamento. A cerimônia e a festa de um casamento são um projeto com uma série de coisas que precisam ser feitas até determinada data e, como você sabe, se já se casou — ou vai descobrir, se decidir se casar —, tudo vai dar errado e

precisará de um esforço quatro vezes maior do que você imaginou.

É claro que pode acontecer o contrário também: algo que você achava que levaria horas para resolver acaba levando apenas 15 minutos. A pergunta que não quer calar é: por que somos tão horríveis na hora de estimar o tempo de alguma coisa?

E, cara, nós somos ruins. Já vamos voltar ao casamento, mas, primeiro, permita que eu apresente a você um gráfico com um dos melhores títulos que já vi: “Cone da incerteza”.



Esse gráfico mostra que estimativas iniciais de trabalho podem variar de 400% do tempo que o trabalho realmente levou a 25% do tempo gasto. Trata-se de uma magnitude de erro de dezesseis vezes. Conforme um projeto avança e mais coisas são resolvidas, as estimativas vão se aproximando mais da linha da realidade, até que não haja mais estimativas, apenas a realidade.

Voltemos à Medco. Eles passaram meses planejando o esforço — como o produto seria e quanto tempo levaria. E, mesmo depois de todos aqueles meses, a pesquisa mostrava que era provável que errariam por um fator de quatro, para cima ou para baixo. Na minha opinião, é por esse motivo que o planejamento em cascata é uma maneira totalmente idiota de fazer as coisas.

“Tudo bem, Sutherland”, posso ouvir você dizer, “nós somos péssimos em estimativas, mas eu tenho que fazer *alguma coisa*, não é? Eu tenho que ter

algum tipo de plano”. E você está certo: precisa mesmo. Mas a chave é refinar o plano no decorrer do projeto, em vez de tentar planejar tudo com antecedência. Planeje apenas a quantidade de detalhes suficiente para entregar o próximo incremento de valor, e estime o restante do projeto em partes maiores. No Scrum, ao final de cada iteração, você tem algo palpável para tocar e mostrar para os clientes. Você pode perguntar a eles: “É isso que você quer? Isso resolve pelo menos parte do seu problema? Estamos no caminho certo?”. E se a resposta for “não”, mude o seu plano.

Então, como você pode fazer isso?

Vamos voltar àquele casamento. A primeira coisa a se fazer é criar uma lista com todas as coisas que tornam a cerimônia e a recepção um sucesso. Seria algo mais ou menos assim:

- noiva e noivo;
- flores;
- convites;
- igreja;
- salão de festa;
- comida;
- celebrante;
- vestido;
- alianças de casamento;
- música (DJ ou banda ao vivo).

O que se deve fazer em seguida é pegar todos esses elementos e classificá-los por prioridade. E isso será diferente para pessoas diferentes. Cada noiva e cada noivo veem o mundo de forma distinta. Mas, outro dia, perguntei ao meu amigo Alex como ele priorizou a lista *dele*, e esta foi a ordem a que ele chegou:

- noiva e noivo;
- celebrante;
- alianças de casamento;
- salão de festa;
- convites;
- comida;
- música;
- vestido;
- flores;
- igreja.

O objetivo do exercício é descobrir as coisas realmente importantes e fazê-las primeiro. Para Alex, a comida e a música são mais relevantes do que casar-se em uma igreja ou ter flores. É necessário ter esses dados, porque se você se deparar com restrições de data ou de custos, saberá onde começar a fazer os cortes: no final da lista. Entrarei em mais detalhes sobre isso mais adiante, no Capítulo 8, mas, por ora, é o suficiente.

Na Medco, a lista ocupava três paredes da grande sala de reuniões e havia centenas de itens divididos entre as seis equipes do projeto. Mas o conceito era exatamente o mesmo: organizar por valor, independente de qual seja ele. Pode ser valor de negócios, no caso da Medco, ou pode ser o valor de felicidade da noiva, no caso de um casamento.

O tamanho importa, mas apenas de forma relativa

Então, você tem uma lista de coisas que precisam ser feitas, e você já priorizou cada item. Agora, o trabalho é descobrir quanto esforço, tempo e dinheiro serão necessários ao projeto. Conforme eu já disse antes, nós, seres humanos, somos absolutamente péssimos nisso, mas, ao que tudo indica, *somos* bons em classificação *relativa* por tamanho — comparar um em relação ao outro. Considere, por exemplo, estimar a diferença entre camisetas pequenas, médias e grandes.

Meu exemplo favorito de classificação relativa é “Pontos caninos”. Muitos anos atrás, um amigo meu e um dos líderes do pensamento ágil, Mike Cohn, estava, como eu, lutando para descobrir como concluir seus projetos no prazo e dentro do orçamento, e como estimar isso. Amante de cachorros, embora sua esposa o proibisse de ter um, ele começou a perguntar a suas equipes de que tamanho de “cachorro” cada parte do projeto era. Ele fez uma lista de várias raças:

- labrador *retriever;*
- *terrier;*
- cão dinamarquês;
- *poodle;*
- bassê;
- pastor alemão;
- cão perdigueiro;
- buldogue.

Então, ele diria: “Tudo bem, este problema — é um bassê ou um cão dinamarquês? Se é um cão dinamarquês, então esse deve ser um labrador, não é?”. Assim, as equipes avaliariam todas as características que precisavam desenvolver, classificando-as de acordo com o tamanho de um cachorro. Então, Mike diria: “Vamos dar para cada uma das raças um valor numérico; assim ficará mais fácil. Digamos que um bassê seja 1 e o cão dinamarquês, 13. Isso pode tornar o labrador, 5, e o buldogue, 3”.[34]

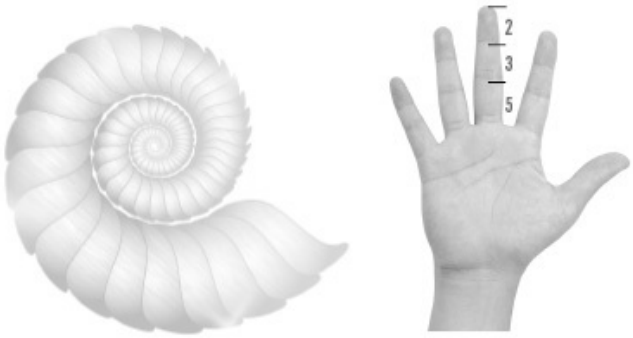
Você pode fazer o mesmo com a lista das tarefas do casamento que fizemos anteriormente. Descobrir um local para a festa, bem, isso vai exigir um pouco de pesquisa, informações sobre preço e visitar alguns lugares. É algo complexo. Vamos chamá-lo de problema do tamanho de um pastor alemão, ou seja, cinco. Noiva e noivo? Sem problemas: os dois têm apenas de aparecer. É um bassê, número 1, apenas um telefonema. Os convites, porém, são bastante complexos. Temos de fazer as nossas listas, pegar a lista da sua mãe e da minha, escolher o papel de carta, encomendar a impressão dos convites, escrever os respectivos endereços à mão. Trata-se de um grande projeto. Um cão dinamarquês. Um 13. Ou talvez dois cães dinamarqueses. E se algo for tão grande assim, você deveria quebrar em partes gerenciáveis. Que tal tornar a lista de nomes um projeto e a parte de encomendar os convites outro? Cada uma delas constitui uma tarefa do tamanho de um buldogue, certo? Ou um 3. O endereçamento será um pastor alemão, um 5. E assim por diante.

Então, isso é a classificação relativa por tamanho, comparando-se uma tarefa com outra. Agora, nem todo mundo usa cachorros, infelizmente. Mas talvez você tenha notado aí um padrão nos números que defini: 1, 3, 5, 8, 13. Cada número na série é a soma dos dois números anteriores. Chama-se sequência de Fibonacci, e há um motivo por que a usamos. Ela está em todos os lugares.

Sequência de Fibonacci: em todos os lugares

• A sequência de Fibonacci é um padrão no qual o número seguinte na sequência é igual à soma dos dois anteriores. Por exemplo: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...

• Onipresente nos sistemas naturais; por isso, os seres humanos têm milênios de experiência com essa sequência.



A sequência é como a natureza se desenvolve, seja na concha de um argonauta, nos galhos de uma árvore, nos espinhos de um abacaxi, ou nas pétalas de uma pinha. Ela aparece na couve-flor e nas curvas do cérebro humano. É a mesma, esteja você olhando para os pontinhos das folhas de uma samambaia ou a forma de uma galáxia. Trata-se de um daqueles fenômenos que, se você parar para pensar, são meio assustadores.

Existe um nome para esse fenômeno — ele se chama proporção áurea ou número de ouro. Nós a embutimos nos prédios e na arte. Do Parthenon, em Atenas, até a Grande Mesquita de Cairuão, na Tunísia. Nós a usamos para decidir o tamanho e a forma das páginas de um livro e as proporções de cartas de baralho. Os humanos são programados para achar as proporções atraentes. Para o nosso objetivo, tudo o que é importante saber é que a nossa espécie tem uma profunda compreensão das proporções da sequência de Fibonacci. Nós a conhecemos de forma intuitiva.

Os números na sequência de Fibonacci se distanciam de forma suficiente para que possamos sentir facilmente a diferença. É fácil para alguém tender a um lado ou a outro. Se uma pessoa estima algo como cinco, e outra como oito, conseguimos ver a diferença de forma intuitiva. Mas a diferença entre cinco e seis? É bastante sutil, mais sutil do que o nosso cérebro consegue registrar.

Na medicina, sabe-se muito bem que, para um paciente reportar que notou uma melhora em um sintoma, essa diferença tem de ser maior que 65%. A nossa mente não trabalha em incrementos estáveis. Somos melhores em perceber saltos de um estado para outro — e não saltos estáveis, mas sim

irregulares.

Usar a sequência de Fibonacci para calcular o tamanho de uma tarefa permite que façamos estimativas que não precisam ser 100% exatas. Nada será exatamente um cinco, um oito ou um 13, mas, ao usar esses números, temos maneiras de colher opiniões acerca do tamanho de uma tarefa na qual todos estão usando mais ou menos a mesma unidade de medida, e, dessa forma, conseguimos formar um consenso.

Fazer estimativas em grupo dessa forma é uma maneira de obter uma previsão bem mais precisa do que se tivéssemos de fazê-las sozinhos.

O oráculo de Delfos

Então, agora já sabemos que somos bons em comparar uma coisa a outra. E sabemos qual é a melhor proporção para isso. Mas como chegamos lá? Uma lista de prioridades é um bom começo, mas como vamos descobrir qual tarefa é um cinco e qual é um oito — qual é um labrador e qual é um *schnauzer*? E mesmo que alguém saiba a resposta aproximada, como podemos nos certificar de que sua estimativa está de acordo com a de todos os outros? E se a pessoa não estiver considerando alguns fatores-chave?

Como seria de se esperar, este não é um problema novo. As pessoas vêm lutando há anos para descobrir isso. Uma questão é que diferentes membros de uma equipe sabem coisas diferentes, mas outra é o que às vezes chamamos "efeito manada". Você já deve ter participado de uma reunião assim: alguém aparece com uma ideia, e todo mundo começa a falar sobre ela. E mesmo que você discorde a princípio, segue adiante porque o grupo resolveu seguir. E todos concordam com um caminho que parecia ser uma ótima ideia naquele momento, mas acabou sendo um completo fracasso. E, quando você questiona as pessoas sobre isso, quase sempre percebe que cada uma delas tinha reservas, mas não as expressou por achar que todo mundo estava animado. As pessoas presumiram que se todos resolveram seguir adiante com algo, as suas próprias reservas são tolas ou mal informadas, e elas não querem parecer burras na frente do grupo. Lembre-se de que esse pensamento em grupo não constitui um fracasso individual, mas um fracasso humano.

Na literatura, esse efeito é explicado como uma "cascata informativa". Nas palavras de Sushil Bikhchandani, David Hirshleifer e Ivo Welch, os autores do estudo "A Theory of Fads, Fashion, Custom, and Cultural Change as

Informational Cascades" ["Uma teoria sobre modismos, moda, costume e mudança cultural como cascatas informativas"]: "Uma cascata informativa ocorre quando é ótimo para um indivíduo, ao observar as ações da pessoa à sua frente, seguir o comportamento dessa pessoa sem considerar as suas próprias informações".[35]

Um grande exemplo que os autores usam é a submissão de um artigo para publicação em um periódico. Digamos que o primeiro editor do periódico o rejeite. Então, o autor submete aquele mesmo artigo para a publicação em um segundo periódico. Esse segundo editor, ao saber da primeira rejeição, fica mais propenso a rejeitá-lo também. E, se houver um terceiro periódico, aquele editor, sabendo das duas rejeições anteriores, ficará ainda mais propenso a fazê-lo. As pessoas presumem que as outras pessoas fazem julgamentos legítimos, *mesmo que tais julgamentos contradigam o seu próprio.* Isso é ruim. Ao fazer um julgamento sobre quando é provável que você vai entregar um projeto de bilhões de dólares — ou se você vai conseguir fazer tudo para o seu casamento — é essencial que você use seu *próprio* julgamento, e use outras estimativas para melhorar a sua, não para substituí-la.

Outro problema bem conhecido é o que chamamos de "efeito halo". Esse efeito ocorre quando uma característica de algo influencia o modo como as pessoas percebem outra característica não relacionada. Ele foi estudado empiricamente pela primeira vez em 1920 por Edward Thorndike. No seu estudo clássico "The Constant Error in Psychological Ratings" ["O erro constante em classificações psicológicas"], Thorndike pediu para oficiais militares classificarem seus soldados de acordo com diversas qualidades — física, intelectual, liderança, personalidade etc. Depois, ele analisou como um conjunto de qualidades afetava a outra; e descobriu que elas se correlacionavam muito proximamente. Se alguém tivesse uma classificação alta no aspecto físico, o mesmo acontecia com suas habilidades de liderança. E inteligência. E caráter. Essa pesquisa foi sustentada por estudos posteriores no decorrer dos anos, confirmando que, por exemplo, se alguém é bonito, todo mundo presume que também é inteligente e confiável.[36]

No entanto, o efeito halo estende-se muito mais do que apenas a mera beleza física; pode aparecer em qualquer lugar. Os pesquisadores apontam, por exemplo, que ONGs costumam ser tratadas como forças para o bem, mesmo que não o sejam; que companhias automobilísticas vão criar um carro "halo" para causar uma boa impressão para toda a sua linha de carros; que o iPod da Apple conferiu a todos os produtos dessa empresa uma áurea de modernidade.

Assim como o efeito manada, as pessoas que se concentram no halo não olham para os dados verdadeiros — em vez disso, gravitam em torno de algo que emite um brilho positivo. Novamente, não se trata de um fracasso do propósito; essa é a natureza das pessoas. Lutar contra isso é tolice — assim como o é lutar contra a gravidade.

Mas você pode ser inteligente em relação a isso. Na década de 1950, a Rand Corporation recebeu um pedido para responder a algumas perguntas — daquele tipo aterrorizante muito comum durante a Guerra Fria. Invocando a terminologia da Oráculo de Delfos, a sacerdotisa que podia prever o futuro, Norman Dalkey e Olaf Helmer publicaram, em 1963, um artigo com o título agradável "An Experimental Application of the Delphi Method to the Use of Experts" ["Uma aplicação experimental do método Delphi para o uso de especialistas"] com a útil referência ao "Memorandum RM-727/1-Abridged" [Memorando RM-727/1 resumido]. No artigo, eles declararam sua intenção de fazer perguntas sem que as opiniões de uma pessoa afetassem as de outra. Então, reuniram um grupo de peritos: quatro economistas, um especialista em vulnerabilidades físicas, um analista de sistemas e um engenheiro elétrico, e propuseram o seguinte:

> Aplicar a opinião de peritos, a partir do ponto de vista de um planejamento estratégico soviético, sobre a seleção de um sistema de alvo industrial dos Estados Unidos, e a estimativa do número de bombas atômicas necessárias para reduzir a produção de munições por uma quantidade prescrita.[37]

Para simplificar: a ideia era perguntar quantas armas nucleares os russos precisariam para impedir os americanos de fabricarem as próprias armas nucleares. Isso em uma época em que o conflito nuclear parecia não apenas possível, mas vencível.

A questão é que Dalkey e Helmer não queriam que seus peritos fossem influenciados uns pelos outros. E se um deles fosse o chefe de departamento de uma grande universidade e outro apenas um membro despretensioso do corpo docente de uma pequena faculdade? Como evitar que as suposições falsas de uma pessoa estragassem a opinião dos demais?

O que os dois pesquisadores fizeram foi conduzir uma série de estudos anônimos. Nenhum dos peritos sabia quem eram os demais; eles só recebiam as estimativas feitas por eles. Depois de cada questionário, a dupla de pesquisadores pegava as respostas — e os dados usados para chegar a elas — e as fornecia de volta para o grupo, retirando qualquer característica

identificadora. Limpar e repetir.

Então, no primeiro questionário, o número de bombas necessário para dar uma certeza de 50% da destruição da indústria bélica norte-americana foi estimado em uma faixa que variou entre cinquenta, na menor estimativa, e cinco mil, na maior. Quando Dalkey e Helmer analisaram as respostas, pareceu haver alguns pontos em comum no modo de pensar — vulnerabilidade de diversos alvos, a capacidade de recuperação das diversas indústrias, suprimento inicial etc. Então, eles perguntaram aos peritos se aquela análise estava correta e quais informações usaram para chegar às suas respostas.

E eles obtiveram tudo, desde o nível de força das fábricas, passando pela diferença entre vulnerabilidade física e vulnerabilidade econômica, até o tempo de fabricação de diversos componentes.

Dalkey e Helmer pegaram os dados, distribuíram entre todos os peritos e perguntaram: “tudo bem, agora, quantas bombas serão necessárias?”. A variação, então, ficou entre 89 e oitocentos. Então, eles repetiram o processo. Diversas vezes. E os resultados foram ficando cada vez mais próximos. Por fim, chegaram a uma variação entre 167 e 360 bombas nucleares.

Sair de uma variação impossivelmente ampla de 10.000% para cerca de 200% parece uma ferramenta incrivelmente poderosa para os mentores políticos. Aquilo permitiria que chegassem a um consenso geral entre peritos, sem se preocuparem com preconceitos. Essa ferramenta é tão poderosa que é usada até hoje pela Rand Corporation. Um exemplo recente foi um exercício da Delphi em 2011 analisando o conflito do Afeganistão e estimando as chances dos Estados Unidos de sucesso. O resultado, se vocês estão interessados, não foi bom.

Pôquer do planejamento

Então, a vantagem do método Delphi é que ele pega uma faixa ampla de opções, tenta remover ao máximo possível os preconceitos e, com declarações bem informadas, mas anônimas, estreita as opiniões até chegar a uma estimativa largamente aceitável. Para os nossos objetivos, a parte ruim é que isso demora muito tempo. Quando me reuni com as equipes da Medco, não dediquei nenhum tempo a estudos anônimos. Eu queria que aquelas centenas de itens fossem estimadas em questão de *horas*, não dias, e muito menos semanas.

Felizmente, *existe* uma maneira de se conseguir estimativas de forma bastante rápida e precisa: chama-se Pôquer do Planejamento.



A ideia é simples. Cada pessoa tem um baralho de cartas com aqueles números superinteressantes de Fibonacci — 1, 3, 5, 8, 13 e assim por diante. Cada item que precisa ser estimado é levado à mesa. Então, todos puxam a carta que acreditam representar o esforço necessário para concluí-lo e a colocam em cima da mesa com o número voltado para baixo. Todos viram suas cartas ao mesmo tempo. Se todo mundo está a uma carta de diferença do outro (digamos, um 5, dois 8 e um 13), a equipe apenas soma os resultados e tira a média (naquele caso, 6,6) e segue para o outro item. Lembre-se de que estávamos falando de *estimativas*, não cronogramas blindados. E estimativas em pequenas partes do projeto.

Se os números das cartas mostrarem uma diferença superior a *três*, as pessoas com as cartas mais altas e as mais baixas falam sobre o motivo por que acreditam que o seu número é o adequado. Então, fazemos uma nova rodada para a mesma tarefa. Caso contrário, eles só fazem a média das estimativas que irão se aproximar dos números aos quais aqueles estatísticos da Rand Corporation chegaram.

Eis um exemplo: digamos que você esteja pintando o interior de uma casa e precise estimar quanto tempo vai levar para terminar a sala, a cozinha e dois banheiros; e esteja fazendo isso com uma equipe com quem já trabalhou antes. Então, primeiro os dois quartos: todo mundo estima um três. Não há nenhum desacordo; vocês já fizeram isso antes e consideravam o quarto bastante

objetivo. Então, a equipe estima a sala de estar. Trata-se de um aposento bem grande. As estimativas das pessoas variam entre 5 e 13, chegando a uma média de 6. Novamente, não há necessidade de discussão. Então, chega a vez da cozinha, e temos um 3, um 8, um 13 e um 5 na mesa. A pessoa que colocou o 3 argumenta que o aposento é pequeno, envolvendo menos paredes do que os quartos. A pessoa com o 13 responde que o que vai levar muito tempo são os armários e bancadas, e que pintar todas aquelas pequenas áreas terá de ser feito com pincel e não com rolo. A equipe rapidamente faz uma nova rodada de cartas. Agora, o 3 virou um 8, e todos os outros concordam. Próximo o suficiente, eles somam, tiram a média e seguem para a próxima tarefa.

Esse método incrivelmente simples é um modo de evitar qualquer tipo de comportamento de ancoragem, tais como os efeitos manada ou halo, e permite que toda a equipe compartilhe conhecimento sobre uma tarefa específica.

Contudo, é crucial que você esteja com a equipe que realmente vai executar o trabalho para fazer as estimativas, e não alguns especialistas “ideais” para isso.

Aprendi isso pelo modo mais difícil quando eu estava trabalhando com uma empresa de comércio eletrônico na Pensilvânia, a GSI Commerce. Eles tinham sido comprados pela eBay. O que a GSI faz é projetar as lojas on-line para empresas como Levi’s, Toys “R” Us, Major League Baseball e Zales Diamonds. Não são projetos pequenos. E a GSI é muito boa nisso.

Essa empresa, porém, tinha uma ideia que parecia ser ótima na época: em vez de fazer com que cada equipe individual realizasse as estimativas, deixariam essa tarefa a cargo dos melhores estimadores da empresa — os caras mais inteligentes na sala que realmente compreendiam os projetos e a tecnologia e sabiam o que precisava ser feito. Então, eles pegaram alguns projetos e os estimaram. Esse aqui levará determinado tempo para ser concluído, aquele outro o mesmo tempo, e assim por diante. O plano era fazer estimativas para oitenta projetos de milhões de dólares, tanto para os clientes como para as equipes que realmente fariam o trabalho. Parece razoável, não é?

Bem, acontece que é um modo tão *errado* de abordar as coisas que eles desistiram do experimento na metade do caminho, depois que quarenta projetos já haviam sido concluídos. Aquilo me fez lembrar dos projetos de medicamentos que precisam ser parados porque os remédios estão matando os pacientes, em vez de curá-los. As estimativas estavam tão distantes da realidade que se tornaram inúteis. Nada foi entregue no tempo estimado, e os clientes, é claro, estavam insatisfeitos. As equipes, desmoralizadas. Foi um completo desastre. Os gerentes voltaram a usar as equipes que realmente

fariam o trabalho correto. E, vejam só, elas começaram a se alinhar novamente com a realidade.

O que eu aprendi com isso foi que apenas as pessoas que estão fazendo o trabalho sabem quanto tempo e esforço cada tarefa exige. Talvez a equipe deles seja muito boa em fazer uma coisa, mas seja péssima em outra. Talvez tenham um perito que possa ser útil em uma área específica, mas ninguém da equipe conhece uma área diferente. As equipes, conforme já disse antes, são individuais e únicas — cada uma tem seu próprio ritmo. Forçá-las a se encaixar em um processo inflexível é uma receita para o desastre.

Não existem tarefas; existem apenas histórias

Ao listar o que precisa ser feito, é tentador fazer uma lista como a que eu fiz anteriormente para o casamento de Alex: igreja, flores, celebrante, comida etc. A questão é, se você der cada um desses itens para uma equipe separada que não esteja intimamente envolvida nos resultados das decisões tomadas entre rosas brancas e margaridas, talvez você não consiga o resultado que espera.

Quantas vezes no trabalho você recebeu uma tarefa que não sabia exatamente por que deveria cumpri-la ? Alguém pede a você que determine a variação das vendas mensais na Região A, analisando lojas com mais de 55 metros quadrados. Você faz, mas não sabe *por que* isso precisa ser feito. E, por causa disso, talvez forneça o tipo errado de dados, interprete a questão erroneamente, ou talvez você apenas se ressinta de ter recebido um monte de trabalho extra. Ou, se você for o gerente, talvez fique surpreso porque a sua equipe não entendeu imediatamente que você deseja fechar as lojas pequenas e abrir lojas grandes.

O problema é que você não está conseguindo nem fornecendo informações suficientes para que o trabalho seja feito da maneira correta. As pessoas pensam em narrativas, histórias. É desse modo que compreendemos o mundo. Temos uma compreensão íntima de personagens, desejos e motivações. Nós nos encrencamos quando tentamos abstrair partes descontínuas da linha principal e tratá-las fora do contexto.

Então, a primeira coisa em que você deve pensar ao considerar uma tarefa é em um personagem ou em um papel — por exemplo, um cliente, uma noiva, um leitor, um funcionário. *Para quem* a tarefa será realizada? Através das

lentes de quem precisamos olhar quando estamos construindo isso, tomando aquela decisão, ou entregando uma peça?

Então, você precisa pensar no *quê* — no que quer que seja feito primeiro. É aqui que costumamos começar e parar. Mas é apenas o meio do processo que deveríamos estar seguindo.

Por fim, você precisa pensar em motivação. *Por que* esse personagem quer isso? Como isso vai ajudar e maravilhar esse cliente específico? E, de certo modo, essa é a parte principal. A motivação colore tudo.

Meu exemplo favorito sobre isso vem de um "meme" da Internet que vi alguns anos atrás. É apenas uma fotografia do Capitão Jean-Luc Picard (Star Trek) da USS *Enterprise*, e sob ela estava escrito: "Como capitão da nave espacial, gostaria de registrar a função de usar automaticamente a data estelar de hoje...". Faz sentido quando você pensa nisso. Você nunca se perguntou por que, em um futuro distante, o capitão de uma nave espacial tem de declarar a data estelar sempre que faz um registro? "Registro do capitão. Data estelar 4671.7. O planeta Marte está muito bonito desta órbita...". *Nós* não precisamos fazer isso hoje quando fazemos um registro em um blog. Por que ele precisa?

Mas a pergunta-chave naquela foto é *por quê*. Por que ele quer tal funcionalidade? Qual seria o seu propósito? Seria apenas para manter as entradas em ordem de data? Ou seria por algum motivo mais sério? Será que tais registros precisam se manter inalterados para atender a algum tipo de funcionalidade de auditoria feita por investigadores de cena de crime da frota estelar? São duas implementações bem diferentes. Uma é casual, a outra, robusta. A equipe precisa descobrir o que ele realmente deseja fazer, pois, em tal ponto, pode pensar em fazer as coisas de uma maneira completamente diferente, com informações mais relevantes nas quais o capitão talvez não tenha nem pensado, mas que serão de grande utilidade.

Em geral, as necessidades mudam quando temos personagens diferentes. Imagine, por exemplo, uma história com o final. "... Eu quero um carro para que eu possa ir dirigindo para o trabalho". Agora, se você começa essa frase com "Como um membro dos subúrbios..." *versus* "Como um fazendeiro dos terrenos erodidos de Dakota do Sul...", você acabará com uma interpretação bem diferente de qual seria o veículo ideal.

Então, antes de priorizar o que precisa ser feito em sua empresa, você precisa definir o personagem, o usuário, o cliente — a pessoa que vai usar o que você vai produzir. Você precisa saber do que gostam, do que não gostam, suas paixões, entusiasmos, frustrações e alegrias. Logo, precisa compreender as suas

motivações. Como esses tipos de personagens informam o que desejam? Por que eles precisam de um carro? O que eles vão fazer com aqueles registros do capitão?

Isso também influenciará o modo como você faz estimativas das coisas. Oh, eles querem uma função simples de calendário; isso é fácil. Um selo inalterável de tempo para objetivos legais — isso é um pouco mais complexo.

Escreva histórias curtas

Ao escrever as suas histórias, porém, você deve se certificar de que elas sejam curtas o suficiente para que você consiga estimá-las. Imagine a história da Amazon.com: “como cliente, eu quero a maior livraria virtual do mundo para que eu possa comprar qualquer livro no instante que eu desejar”. Agora, essa história certamente engloba a Amazon, mas, na verdade, é grande demais para que possamos fazer qualquer coisa com ela. Precisamos dividi-la. Dividi-la mesmo.

Você pode escrever histórias como estas para uma livraria virtual:

- “Como cliente, eu quero poder navegar pelos livros por gênero, para que eu possa encontrar o tipo de livros de que eu gosto.”
- “Como cliente, eu quero colocar um livro em um carrinho de compras, para que eu possa comprá-lo.”
- “Como gerente de administração de produtos, eu quero poder rastrear as compras de um cliente, para que eu possa oferecer livros específicos para ele, com base nas compras anteriores.”

Esses são os tipos de histórias que as equipes conseguem compreender. Uma discussão que possa resultar em formas de implementá-las. Elas são específicas o suficiente para serem práticas, mas não prescrevem *como* serão feitas. Lembre-se: a equipe decide de que forma o trabalho será realizado, mas *o que* será realizado é decidido pelo valor que traz aos negócios. A coleção completa de histórias que podem construir a ideia de uma livraria virtual costuma ser chamada de “Épico” — uma história grande demais para se fazer, mas que inclui um número de histórias menores que se somam até formar uma ideia única.

Tim Stoll é um desses caras cuja carreira se estende sobre o que se pode chamar de *espectro amplo de eventos*, com um foco em formar equipes capazes de realizar tarefas muito rápido. Ele era médico das Forças Especiais que serviu no Iraque e no Afeganistão, fornecedor da CIA e policial cujo trabalho era

perseguir criminosos violentos. Agora, ele é um treinador Scrum. Na verdade, *sempre* foi, ele mesmo afirma, mesmo quando liderava missões das Forças Especiais.

“Nas operações especiais, conta ele, “nós não chamamos isso de histórias, mas de Cursos de Ação. Porém, é a mesma coisa”.

Aqui está uma das poucas histórias que Tim pode contar publicamente sobre uma missão das Forças Especiais — uma missão médica em Laos: “Tínhamos dois épicos. O primeiro era um curso médico de instrução — o treinamento de forças locais em medicina de combate. O segundo épico era uma operação de retirada de minas explosivas não detonadas”.

Como médico, Tim estava à frente do primeiro épico. Ele diz que antes da missão ele se sentou e imaginou o que ele precisava conseguir e como ele deveria ordenar as histórias menores. E ele diz que começou com ideias que se encaixam muito facilmente na estrutura do Scrum.

“Como um médico das Forças Especiais, eu tenho de ensinar fisiologia básica para os meus alunos, para que eles possam compreender o corpo humano.”

Tim afirma que sabia que devia iniciar por ali quando começou a escrever suas histórias. Seus alunos precisavam compreender onde os ossos estavam para que pudessem prestar qualquer tipo de primeiros socorros. “Primeiro, eu ensino quais são os ossos maiores, depois, os menores, depois, os ossos dos punhos e tornozelos, os tendões e ligamentos”. Apenas depois que as histórias básicas tinham sido cobertas, é que ele entrou em como colocar esses ossos no lugar, liberar as vias aéreas, e parar hemorragias.

Depois de escrever essas histórias, conseguiu perceber do que precisaria para apoiar seus objetivos. Ele precisaria de um esqueleto, e de panfletos em inglês e em laosiano; então, subdividiu tudo em iterações ou Sprints. “Dois dias de voo para chegar a Laos. Uma semana de preparação. Então, duas iterações de seis semanas de instrução. Nós tínhamos de fazer com que os nossos alunos passassem do básico para o intermediário de técnicos em emergências médicas. E foi o que fizemos”.

Esteja preparado e conclua a história

Ao escrever histórias ou fazer uma lista de tarefas, é importante fazer duas perguntas: a história está pronta? E como você vai saber disso?

Vamos pegar a história de Tim como exemplo:

> Como um médico das Forças Especiais, eu tenho de ensinar fisiologia básica para os meus alunos, para que eles possam compreender o corpo humano.

Existe um mnemônico que sempre uso para dizer se uma história está pronta. Foi criado por Bill Wake, que é um pensador astuto de desenvolvimento de software. Bill diz que para uma história estar pronta ela precisa atender aos critérios INVEST:

> **I**ndependente. A história precisa ser prática e passível de ser "feita" por si só. Ela não deve ser inerentemente dependente de outra história.
> **N**egociável. Até que esteja sendo realizada, ela precisa poder ser reescrita. A permissão para alterações está embutida nela.
> **V**aliosa. Ela realmente deve acrescentar valor ao cliente, usuário ou *stakeholder*.
> **E**stimável. Você deve ser capaz de mensurá-la.
> **S**mall [pequena]. A história precisa ser pequena o suficiente para que você possa estimar e planejar facilmente. Se ela for grande demais, reescreva-a ou divida-a em histórias menores.
> **T**estável. A história precisa ter um teste no qual deve passar para ser completa. Escreva o teste antes de fazer a sua história.

Então, a história de Tim é independente: ele pode realizar sua missão sem ter de considerar, digamos, quanto combustível será necessário para o helicóptero transportar os alunos. Ela é negociável: ensinar fisiologia é a história do que ele acredita que precisa fazer, mas se ele chegar lá e descobrir que os alunos já têm esse conhecimento, ou parte dele, pode mudar a abordagem de ensino. Ela é valiosa: os alunos aprenderão conhecimentos práticos e aplicáveis sobre o corpo humano. É "small": é anatomia básica, e não como fazer uma cirurgia com a anatomia que ele está ensinando. E é testável: ele conhece as informações que deseja transmitir, e pode testar os alunos para verificar se eles aprenderam as informações.

Para cada história que se deseja fazer lá deve haver uma "definição de Pronto" (ou seja, "ela atende aos critérios INVEST?") e, por fim, uma "definição de Feito" (ou seja, "Quais condições precisam ser atendidas, que testes precisam ser feitos para considerá-la feita?"). Encontramos em projetos verdadeiros que, se as histórias estão realmente prontas, a equipe vai dobrar a velocidade da sua implementação. E, se as histórias estiverem concluídas no

final de um Sprint, as equipes dobram novamente a velocidade. Esse é um dos truques necessários para conseguir fazer o dobro do trabalho em metade do tempo.

Planejamento do Sprint

No Scrum, esse tipo de planejamento acontece em cada um dos Sprints e chama-se reunião de planejamento do Sprint. Todos se reúnem e analisam a lista de histórias que precisam ser feitas, e dizem: “tudo bem, o que podemos fazer *neste* Sprint? Essas histórias estão prontas? Elas podem ser concluídas ao final desta iteração? Poderemos demonstrá-las para o cliente e mostrar-lhe valor real?”.

A chave para responder a essas perguntas está na velocidade que a equipe está atingindo.

Conheça a sua velocidade

Podemos finalmente começar a responder às perguntas de quando as coisas serão feitas, porque agora sabemos como mensurar o que a equipe está realmente fazendo. Temos todas essas histórias, essas coisas que precisam ser feitas; e já fizemos as estimativas para cada uma delas — esta é um oito, aquela é um três etc. E, então, começamos o primeiro Sprint; digamos que ele tenha a duração de uma semana. Ao final da semana, contamos as histórias que foram concluídas, totalizamos os pontos em que foram estimadas, e esse número revela a velocidade da equipe. Uma vez que você conheça a velocidade, pode olhar para quantas histórias ainda precisa realizar e quantos pontos elas representam e, então, saberá quando o trabalho será concluído.

Além disso, uma vez que você saiba a velocidade, poderá descobrir a coisa mais importante no Scrum: o que está impedindo você de ir *mais rápido?* O que está impedindo você de *acelerar?* No capítulo anterior, falei sobre desperdício, sobre as coisas que tornam você mais lento. Essa é a maneira usada para saber se você realmente *está* se livrando do desperdício.

Voltemos à Medco, onde começamos este capítulo. Despois que estimamos todo o trabalho, eu me reuni com a alta diretoria responsável pelo projeto: vários VPs, que eram gestores gerais de unidades de negócios, e um VP Sênior.

Nós nos sentamos na mesa de conferência, e o VP Sênior tinha apenas uma

pergunta: “Você vai conseguir cumprir o prazo original?”, perguntou ele batendo com a mão na mesa.

“Eu não sei”, respondi, “mas a gente consegue bater a data revista pelo seu pessoal, ou devolvemos o seu dinheiro”.

“Isso não é uma resposta aceitável! Você vai cumprir a data original?”

“Eu não tenho como responder isso *hoje*. As equipes precisam começar a trabalhar para vermos a velocidade que conseguem atingir. Vamos fazer o seguinte: em seis semanas, eu lhe informarei a data de entrega, e não será uma de que você vai gostar. Mas”, continuei rapidamente antes que ele tivesse a chance de me interromper, “eu vou lhe dar uma lista dos obstáculos que estão impedindo as equipes de cumprirem aquela data de julho que você prometeu em Wall Street. Uma lista de obstáculos. E o seu trabalho é removê-los o mais rápido possível”.

Ele riu. “Obstáculos! Sem problemas, Jeff. Eu trabalhava na Toyota”.

Eu ri e respondi: “Parece que esse projeto já está ficando bom”.

Eu sabia que ele ia adotar a sistemática de desperdício de Taiichi Ohno e compreendia como as coisas funcionavam: livrar-se do desperdício era a chave para acelerar as equipes.

Então, depois de três Sprints mensurando a velocidade do projeto, as equipes tinham acelerado de vinte para sessenta pontos por Sprint, e eu sabia qual seria a data provável de entrega do projeto. Considerando a velocidade das equipes, e estávamos agora no início de março, levaríamos outros 19 Sprints de duas semanas — 1º de dezembro.

A diretoria não ficou satisfeita. Não era bom o suficiente. Era dia 1º de julho ou nada. Tudo dependia daquilo.

Então, entreguei um memorando com uma lista de 12 obstáculos, que variavam desde a falta de autonomia para tomar decisões até requisitos técnicos onerosos; desde pessoas que não apareciam nas reuniões, até coisas simples como não ter todo mundo da mesma equipe trabalhando na mesma sala. Havia questões de processo, personalidade e procedimento — o tipo de coisas que são endêmicas em qualquer empresa.

Esses tipos de obstáculo podem parecer instransponíveis. Quantas vezes você olhou à sua volta na empresa em que trabalha e pensou “Nós fazemos assim, nós sempre fizemos assim, e todo mundo sabe que é burrice”. Por algum motivo, porém, as pessoas veem as mudanças culturais corporativas como algo impossível. Eu costumava concordar com isso, principalmente quando se tratava de grandes empresas com cultura e diretrizes fossilizadas.

A Medco provou que eu estava errado, e eu nunca mais voltei a pensar da mesma maneira. O vice-presidente sênior que trabalhara na Toyota enviava nosso memorando para a equipe toda na segunda-feira. Cada obstáculo tinha ao lado o nome de um gestor. E cada um dos obstáculos havia desparecido na quinta-feira. Talvez as pessoas precisassem de uma arma apontada para as suas cabeças para estimular-lhes à mudança, mas aquilo serviu para mostrar o que pode ser feito se existe força de vontade (ou se você tiver um cara da Toyota no comando). Nada está escrito em pedra. Questione tudo.

Ao final do Sprint seguinte, a velocidade das equipes tinha crescido em 50%. A nova data de entrega era 1º de setembro. Ainda um atraso de três meses, mesmo que as equipes tenham acelerado de vinte para noventa pontos por Sprint, ou seja, mais de 400%!

Ainda assim, não era bom o suficiente.

Então, Brent e eu juntamos todo mundo, desde os engenheiros, passando pelo pessoal de marketing e de conformidade, até a gerência. E eles estavam com medo. Estavam com medo pelos seus empregos e carreiras se não conseguissem aquilo. Então, eu disse: “Eu vou fazer três perguntas”:

1. Existe alguma coisa que podemos fazer para acelerar mais as coisas?

“Bem”, respondeu o chefe de engenharia, “no meio do último Sprint, o pessoal de segurança de TI fechou uma porta na Internet, então, as nossas equipes na Índia e no Brasil não conseguiram fazer nada”.

“Bem, nós devemos corrigir isso, não é?”, perguntei, incrédulo. O chefe dos engenheiros olhou para o chefe de TI, que estava mais adiante na mesa. Eles achavam que aquilo talvez conseguisse adiantar mais um mês a entrega do projeto. Agora só faltavam dois meses.

2. Podemos nos livrar de alguns itens pendentes? Será que existem coisas que outras equipes possam fazer?

Ninguém teve nenhuma boa ideia.

3. Será que podemos deixar de fazer algumas coisas? Podemos reduzir o escopo do projeto de alguma forma?

Originalmente, eles me disseram que de jeito nenhum, que já tinham cortado tudo aos requisitos mínimos. “Tudo bem”, respondi, “mas vamos passar esta tarde fazendo pequenas reduções. Cada uma das tarefas terá de lutar por sua vida.

Levou algumas horas, mas conseguimos ganhar mais um mês.

Foi quando eu disse “bem, ainda estamos com um mês de atraso para a data de entrega. Se não conseguirmos pensar em mais nada, teremos de avisar à

diretoria que não conseguiremos.”

“Não”, responderam todos. “A gente vai ser despedido. Vamos analisar essas três perguntas de novo”.

Propus que nos reuníssemos com a equipe de gerenciamento. O problema não era só nosso. Era deles também, e eles poderiam ajudar.

Foi uma reunião curta. A diretoria olhou para a situação e disse: “Bem, precisamos entregar o projeto no dia 1o de julho. Quem sabe a gente possa começar usando apenas uma única fábrica primeiro? Um centro? Ou dois? Isso funcionaria?”. Houve alguns pigarros de embaraço e o reajuste de algumas coisas. Mas, por fim, eles calcularam que poderiam reduzir as características necessárias e cumprir o prazo de julho de 2007, que o presidente prometera a Wall Street.

Ao final da reunião, o Vice-presidente Sênior declarou: “vamos declarar vitória. Chame a gente se tiver qualquer problema”.

Foi incrível observar o preço das ações da Medco naquele verão. Quando iniciamos a infraestrutura, as ações começaram a subir, e quando entregamos o projeto, elas continuaram subindo. Quanto? Bem, muitos bilhões de dólares em valor, saindo de um preço de 25 e ultrapassando cinquenta em um ano. Wall Street decidiu que a empresa continuaria a crescer, atrairia novos clientes e manteria sua liderança na área. Em retrospecto, eu deveria ter pedido uma porcentagem do aumento do capital de mercado em vez de simples honorários.

Alguns anos mais tarde, a Medco usou o Scrum para construir o que chamaram de Medco 2.0. Eles reestruturaram todas as partes da empresa. Novas fábricas, novos robôs, novos processos, mais automação. Mark Landy, que àquela altura já era o CTO da empresa, disse que sem a experiência do Therapeutic Resource Center eles não teriam conseguido. “Eles não teriam permitido que fizéssemos isso na empresa inteira. Mas nós tínhamos a confiança da organização inteira: desenvolvimento, operações, finanças, clínicos. Conseguimos formar uma nova cultura”. E, segundo ele, essa é a parte mais importante do Scrum: a cultura na qual as pessoas trabalham muda, e isso pode ser assustador para algumas pessoas. “De fato, a empresa teve de se livrar de alguns funcionários que não conseguiram fazer a mudança”, disse ele. “Não porque fossem incompetentes, mas porque escondiam informações e conhecimento em benefício próprio, para se assegurarem de que eram indispensáveis, em vez de ajudar as equipes e a empresa. Mudar aquela cultura, porém, foi o que permitiu o surgimento da verdadeira excelência”.

PONTOS PRINCIPAIS

O mapa não é o terreno. Não se apaixone pelo seu plano. É quase certo que ele esteja errado.

Planeje apenas o necessário. Não tente projetar tudo com anos de antecedência. Planeje o suficiente para manter as equipes ocupadas.

Raças de cachorro. Não faça estimativas usando termos absolutos, como horas — já foi provado inúmeras vezes que os seres humanos são péssimos nisso. Defina o tamanho dos problemas de forma relativa, de acordo com uma raça do cachorro ou o tamanho de uma camiseta (P, M, G, GG, XG), ou, o que é mais comum, usando a sequência de Fibonacci.

Pergunte ao oráculo. Use uma técnica cega, como o método Delphi, para evitar preconceitos de segurança, como os efeitos de manada ou halo, ou apenas um modo burro de pensamento em grupo.

Pôquer do planejamento. Use essa técnica para estimar rapidamente o trabalho que precisa ser feito.

O trabalho é uma história. Pense primeiro sobre quem vai obter valor com algo, então, pense no que é, e, então, por que eles precisam daquilo. Os seres humanos pensam em uma estrutura narrativa, então dê isso a eles. Por exemplo: “como X, eu quero Y, para conseguir Z”.

Conheça a sua velocidade. Toda equipe deve saber exatamente quanto trabalho consegue realizar a cada Sprint. E eles devem saber o quanto podem melhorar a própria velocidade ao trabalhar de forma mais inteligente, removendo as barreiras que os tornam mais lentos.

Velocidade x Tempo = Entrega. Uma vez que você saiba a velocidade com que trabalha, saberá quanto tempo levará para chegar lá.

Defina objetivos audaciosos. Com o Scrum não é difícil dobrar a produção ou cortar o cronograma de entrega pela metade. Se você fizer as coisas da forma certa, a sua receita e o preço das suas ações também devem dobrar.

34 Mike Cohn, *Agile Estimation and Planning*, Upper Saddle River, NJ, Prentice Hall, 2005.

35 Sushil Bikhchandani et al., A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades, *Journal of Political Economy* 100.5, 1992, pp. 992-1026.

36 E. L. Thorndike, A Constant Error in Psychological Ratings, *Journal of Applied Psychology* 4.1, 1920, pp. 25-29.

37 N. Dalkey e O. Helmer, An Experimental Application of the DELPHI Method to the Use of Experts, *Management Science, Vol* 9(3), 1963, pp. 458-67.

CAPÍTULO 7

Felicidade

As pessoas querem ser felizes. Não felizes de um modo complacente e tímido, mas de uma maneira mais ativa. Thomas Jefferson, entre muitos outros, exaltava o tipo de felicidade resultante de uma busca. As buscas parecem ser o que nos torna felizes. O Scrum feito da *maneira certa* torna os trabalhadores, os clientes, os gerentes e os acionistas felizes — em geral, nesta ordem.

Não é fácil conquistar a verdadeira felicidade. Certa vez conheci um alpinista que me vendeu uma fotografia que mostrava do topo do Himalaia ao crepúsculo. Ele a tirou logo depois de ter chegado ao topo do Everest tarde demais naquele dia. Parecia impossível voltar ao acampamento na base antes de escurecer. Se ele não tivesse chegado, certamente teria morrido congelado. A aflição da foto refletia os seus sentimentos enquanto escrevia o que ele achou que talvez fosse a sua última carta, que estava feliz por ter chegado ao pico, apesar do fato que, quem quer que lesse aquelas palavras, o encontraria já morto.

Se você conversar com alpinistas sobre uma expedição, eles não passarão muito tempo falando sobre o topo. Em vez disso, comentarão algo sobre as temperaturas gélidas, as bolhas dolorosas, a comida ruim, as péssimas condições e o equipamento. E dirão que, depois do júbilo de chegar ao topo, geralmente eles têm uma sensação de desapontamento (a não ser que a experiência de quase morte perdure). Eles conseguiram. O esforço deles conquistara algo. Mas se você perguntar a eles quando se sentiram mais felizes, então ouvirá que foi naqueles momentos de dificuldade — de levar o corpo, a mente e o espírito ao limite. *Foi nesse momento* que se sentiram mais felizes, quando tiveram a experiência de verdadeira alegria. E é isso que querem experimentar novamente. Diante daquelas condições, nenhuma pessoa em sã consciência se colocaria voluntariamente naquele tipo de aventura *duas vezes*. Ainda assim, os alpinistas parecem ser incapazes de parar, e desafiam picos e mais picos, em busca do júbilo no próximo cume.

É fascinante que a maioria das culturas não são organizadas para recompensar e encorajar esse tipo específico de felicidade. O Professor Tal Ben-Shahar dava aulas no curso mais popular na Universidade Harvard, “Psicologia Positiva”. Em seu livro *Seja mais feliz,* ele escreve: “nós não somos recompensados por apreciarmos a jornada em si, mas por um *término* bem-sucedido. A sociedade recompensa resultados, não processos; chegadas, não jornadas”.

Contudo, nossa vida cotidiana é quase toda feita de jornadas. Não atingimos picos mais altos todos os dias, ou conseguimos uma pontuação alta, nem obtemos um grande bônus. A maior parte dos nossos dias é dedicada a seguirmos *em direção* aos nossos objetivos, independente de quais sejam. Em uma empresa, a meta pode ser a entrega de um grande produto, ou tornar a vida das pessoas um pouco melhores com tal produto, ou resolver algum problema que prejudica o mundo. Mas se nós somos apenas recompensados pelos resultados, e não pelos processos, seremos extremamente infelizes.

Assim que deixei a vida acadêmica e entrei no mundo corporativo, no início da década de 1980, fui nomeado responsável por dezenas de programadores de computador que eram totalmente infelizes. Seus projetos estavam sempre atrasados e acima do orçamento — isso quando o projeto funcionava. O humor deles se tornou tão negativo que a energia na sala deixava todo mundo deprimido. Os processos que usavam eram tão falhos, que era impossível ser bem-sucedido. Passei os últimos trinta anos tratando desse tipo de problema.

A importância da felicidade realmente me atingiu quando eu estava organizando a minha primeira equipe Scrum. Percebi que eu tinha de lidar não só com o estado emocional, mas também com o estado mental da equipe. Para um piloto de guerra treinado em West Point, isso era algo a ser ajustado, afinal eu estava acostumado a coisas já decididas. Como pesquisador e cientista, levei algum tempo para perceber que, para dar autonomia às pessoas, para mudar suas vidas para melhor, eu tinha de mudar também. No decorrer daquele primeiro esforço Scrum, me dei conta de que a verdadeira grandeza é profundamente enraizada na alegria. E que para ser alegre é preciso dar o primeiro passo em direção ao sucesso.

Se isso soa meio Nova Era ou se estou prestes a dizer para você se sentar perto de uma fogueira e cantar “Kumbayah”, você deve saber que no início da minha carreira de consultoria para start-ups, os capitalistas de risco com quem eu trabalhava achavam que eu era um hippie de São Francisco. Dar poder de decisão às pessoas nunca funcionaria na visão de mundo deles. É claro que

hoje em dia eu sou um consultor sênior em empresas de capital de risco e costumo ser tratado como um oráculo — quando as pessoas têm um problema difícil, elas pedem a solução a um oráculo. Resumindo: elas não esperam necessariamente que a resposta faça sentido; apenas tentam e, para sua surpresa, quase sempre funciona.

Isso porque a felicidade é crucial para o seu negócio e, na verdade, é um indicador da receita melhor do que a maioria das medidas que o seu CFO fornece. Neste capítulo, descreverei como a felicidade é importante para o resultado final, e como consegui-la, mensurá-la e aplicá-la. Trata-se de felicidade com rigor.

Eu talvez tenha me tornado uma pessoa melhor enquanto desenvolvia o Scrum — o que tornou a minha família e a mim mais felizes. Mas como um homem de negócios e um cientista, gosto de dados sólidos.

Felicidade é sucesso

A pesquisa é surpreendentemente clara. As pessoas felizes simplesmente se saem melhor — em casa, no trabalho, na vida. Elas ganham mais dinheiro, têm empregos melhores, formam-se na faculdade e vivem mais tempo. É realmente notável. Quase sem exceção, elas são melhores no que fazem.

As pessoas felizes vendem mais, ganham mais dinheiro, custam menos, apresentam uma probabilidade menor de pedir demissão, são mais saudáveis e vivem mais. Ou, nas palavras retiradas de um artigo publicado em 2005, que realizou uma metanálise de cerca de 225 artigos com mais de 275 mil participantes:

> A felicidade leva ao sucesso em quase todos os domínios das nossas vidas, incluindo casamento, saúde, amizade, participação na comunidade, criatividade e, principalmente, no trabalho, na carreira e nos negócios.[38]

A metanálise mostrou que as pessoas que se sentiam felizes tinham mais chances de conseguir entrevistas de emprego, ser avaliadas de forma mais positiva pelos supervisores, mostrar desempenho e produtividade maiores; e ser melhores gestores

Aqui está a parte mais interessante. Intuitivamente faz sentido que pessoas felizes se saiam melhor — isso *porque* é em decorrência do seu sucesso que

são felizes, certo? Errado. Da mesma metanálise: “Um grande número de estudos mostra que a felicidade *precede* resultados importantes e são indicadores de prosperidade”.

Isso mesmo. As pessoas não são felizes porque são bem-sucedidas; elas são bem-sucedidas porque são felizes. Felicidade é uma medida *indicadora*. E o desempenho melhora mesmo se a pessoa for só um pouquinho mais feliz. Não é necessário mudar drasticamente a vida de alguém para torná-lo mais feliz, pelos menos por um tempo. Até mesmo um pouco de felicidade resulta em resultados marcadamente melhores. As pessoas não precisam ser felizes da forma delirante como no dia de seu casamento, mas apenas um pouco mais felizes do que eram. É claro, torná-las ainda mais felizes tem um efeito maior. No entanto, a mensagem que eu quero deixar com isso é simples: *até mesmo gestos pequenos podem ter um grande impacto.* O Scrum tem como foco pegar essas coisas pequenas e construí-las, de forma sistemática, como degraus para o sucesso. Apenas uma coisa de cada vez, e você pode, na verdade, mudar o mundo.

Vou dar a você uma caixa de ferramentas para medir a sua felicidade e a da sua equipe, empresa e família, assim como de qualquer organização com a qual você pode estar envolvido. É isso o que o Scrum faz. Esqueça os exercícios de construção de confiança; em vez disso, construa a confiança todos os dias. E eu quero que você a meça. Não é suficiente *achar* que as pessoas são felizes, eu quero que você seja um cientista em relação a isso, que quantifique e compare com desempenho. Se algo não se equiparar, existe um problema. É ótimo ir a um bar para criar laços com a sua equipe, mas isso não ajudará muito a empresa se esses laços não se traduzirem em um desempelho melhor. Existe um monte de gente com quem eu saio só para me divertir. Com a minha equipe, eu quero que aquele aspecto social afete diretamente o desempenho. E é isso que acontece.

Quantificando a felicidade

Como fazer nossos funcionários e colegas de equipe, e nós mesmos, mais felizes? Como canalizar tal felicidade para melhorar a produtividade e a receita? Existe alguma forma melhor de fazer isso do que distribuir balas todos os dias?

Existe sim. Mas, primeiro, preciso voltar à Toyota e à cruzada de Taiichi

Ohno para eliminar o desperdício. Esse objetivo o levou à ideia de melhoras contínuas; a meta não é chegar a determinado nível de produtividade e se manter ali, mas analisar constantemente seu processo para melhorá-lo cada vez mais. É claro que a perfeição jamais será alcançada, mas cada incremento nessa direção conta muitos pontos.

Assim como o trabalho e o tempo precisam ser divididos em partes gerenciáveis, a melhora precisa ser fatiada de forma a galgarmos um degrau de cada vez. Os japoneses usam a palavra *kaizen*, ou "aperfeiçoamento". Qual é o pequeno aprimoramento que pode ser feito imediatamente para tornar as coisas melhores?

No Scrum, isso é capturado ao final de cada Sprint, no que chamamos de "Retrospectiva do Sprint". Depois que a equipe mostra o que conseguiu — tudo que foi "Feito" e pode ser enviado para os clientes para obtenção de feedback —, eles se sentam e pensam sobre o que deu certo, o que poderia ter sido melhor e o que pode ser aperfeiçoado. Qual é o aprimoramento no processo que eles, como uma equipe, podem aplicar de forma imediata?

Saiba que, para ser eficaz, é necessária uma dose de maturidade emocional e uma atmosfera de confiança. O mais importante a lembrar é que você não está procurando culpados; você está olhando para um processo. Por que algo aconteceu de determinada maneira? Por que não percebemos aquilo? O que poderia ter nos tornado mais rápidos? É essencial que as pessoas, *como uma equipe*, assumam a responsabilidade pelos seus processos e resultados, e procurem soluções, da mesma maneira. Ao mesmo tempo, precisam ter coragem de levantar as questões que realmente as incomodam, de forma que sejam orientadas à solução, em vez de soar como uma acusação. Ao mesmo tempo, devem ter a maturidade de ouvir o feedback, absorvê-lo e buscar uma solução, em vez de assumirem uma postura defensiva.

A Retrospectiva do Sprint corresponde à "verificação" do ciclo PDCA de Deming. A chave é chegar à etapa "aja", àquele *kaizen*, que vai realmente mudar o processo e torná-lo melhor da próxima vez. Não é suficiente compartilhar como você se sente; você precisa ser capaz de agir.

A melhor maneira que encontrei de capturar tudo isso é com o que eu chamo de "A Métrica da Felicidade". Trata-se de uma forma simples, mas bastante eficaz, de chegar ao que o *kaizen* deve ser, mas também a qual *kaizen* tornará as pessoas mais felizes. E eu já usei isso com resultados consideravelmente notáveis.

Eis como funciona. No final dos Sprints, cada pessoa na equipe responde a

estas poucas perguntas:

1. Em uma escala de 1 a 5, como você se sente em relação ao papel que desempenha na empresa?
2. Usando essa mesma escala, como você se sente em relação à empresa como um todo?
3. Por que você se sente assim?
4. O que tornaria você mais feliz no próximo Sprint?

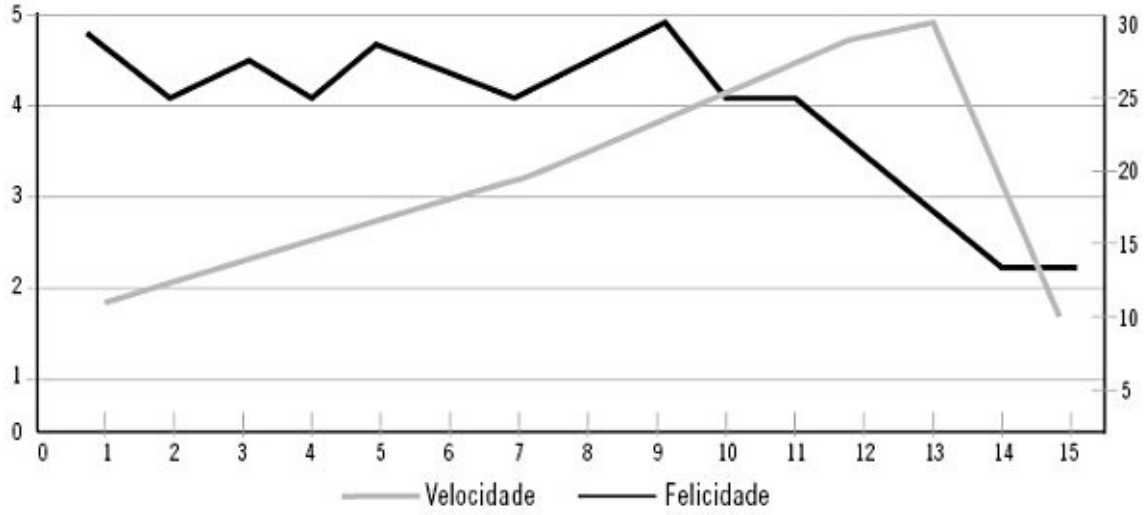
Só isso; leva apenas alguns minutos. Cada pessoa na equipe tem sua vez, e isso inspira conversas bastante reveladoras. Juntos, os membros costumam chegar ao *kaizen* bem mais rápido. O método expõe o que é mais importante para cada um, e o que eles acreditam ser o melhor para a empresa.

E aqui está um elemento crucial: a equipe pega o aprimoramento mais importante e o transforma na questão principal a ser alcançada no Sprint seguinte — *com testes de aceitação.* Como você pode provar que atingiu tal aprimoramento? Você precisa definir o que é o sucesso de uma forma concreta e contestável, de modo que na próxima Retrospectiva do Sprint seja realmente fácil verificar que chegou ao *kaizen*.

Alguns anos atrás, decidi expandir a minha empresa, a Scrum Inc., transformando-a em uma consultoria prestando um serviço Scrum completo. Rastreamos a nossa velocidade e descobrimos que conseguíamos concluir cerca de quarenta pontos de histórias a cada Sprint de uma semana. Quando implementei a Métrica da Felicidade, a primeira questão levantada foi que as nossas histórias não eram boas o suficiente. Elas ainda não estavam prontas, não tinham uma definição de Concluída e eram vagas demais. Trabalhei nisso, e começamos a ter histórias melhores. No decorrer do Sprint seguinte, elas ainda não eram boas o suficiente. Os números da nossa Felicidade não eram ótimos. No terceiro, surgiu outra questão, e nós a resolvemos. Em apenas algumas semanas, nossa velocidade aumentou de 40 pontos para 120 pontos por Sprint. Triplicamos a produtividade simplesmente ao perguntar o que traria mais felicidade às pessoas. Como resultado, nossos cientes ficaram mais felizes, e a nossa receita aumentou incrivelmente. Tudo que eu tive de fazer foi começar a perguntar à equipe: “o que tornaria você mais feliz?” e, então, cumprir a tarefa inteira.

Fizemos um gráfico com esses dados e percebemos algumas coisas extraordinariamente interessantes. Como um CEO, o meu foco está no que vai acontecer com a nossa receita no futuro, além de nosso crescimento e produtividade. Diferente de métricas financeiras, descobri que a Métrica da

Felicidade pode ser *profética*. A contabilidade financeira olha o que aconteceu no passado, mas quando você pergunta às pessoas qual é o seu nível de felicidade, elas, na verdade, fazem uma projeção do futuro. E quando pensam no nível de felicidade em relação à empresa, começam a projetar como pensam que a empresa está se saindo. Como resultado, é possível obter indícios de que um problema está por vir, *antes* que ele realmente apareça. E se você prestar bastante atenção ao que a sua equipe está contando, é possível agir e resolver a questão antes que ela se torne um problema. Neste gráfico, por exemplo, uma queda na felicidade precede uma baixa na velocidade ou na produtividade em semanas. Se você estivesse olhando só para a produtividade, não saberia que estava diante de um problema até que despencasse do penhasco. No entanto, se você vir que o nível de felicidade da equipe está caindo, mesmo que a produtividade esteja aumentando, saberá que existe alguma questão que precisa ser resolvida o mais rápido possível.



Torne tudo visível

O que *realmente* faz as pessoas felizes? São as mesmas coisas que fazem grandes equipes: autonomia, maestria e objetivo. Ou, para dizer de forma mais específica: é a habilidade de controlar o próprio destino, a sensação de se aperfeiçoar cada vez mais e saber que você está fazendo algo maior do que si mesmo. Contudo, existem também alguns passos concretos e fáceis que a gerência pode dar para fazer com que a cultura da empresa desenvolva tais qualidades.

Um elemento do Scrum que costuma ser o prelúdio para autonomia, maestria e objetivo é a transparência. A ideia é que não deve haver nenhuma conspiração secreta ou planos ocultos, nada atrás das cortinas. É muito comum que não esteja claro no que cada um está trabalhando e como a atividade diária dos funcionários resulta em avanços nos objetivos da empresa.

Quando eu estava começando, dediquei muito tempo pensando sobre as leis que um amigo meu tentava incluir na legislação do Colorado — as leis "Sunshine". Essas leis exigiam que todas as reuniões públicas fossem abertas, que todos os registros estivessem disponíveis para o público, e que nada acontecesse atrás de portas fechadas — não haveria nada escondido. É por isso que, no Scrum, qualquer pessoa pode participar de qualquer reunião. Qualquer *stakeholder* pode observar a Reunião Diária ou assistir a uma Revisão.

O que eu queria, então, era tornar tudo visível. E isso pode ser assustador para algumas pessoas. A PatientKeeper é uma empresa que desenvolve aplicações portáteis para hospitais e médicos. Quando fui contratado, fiz de todo o departamento de engenharia uma oficina Scrum. Disse para os desenvolvedores que todos saberiam tudo. Eles estavam tão acostumados a ver as medições serem usadas contra eles que o novo nível de transparência os deixou temerosos de que acabariam sofrendo ainda mais abusos.

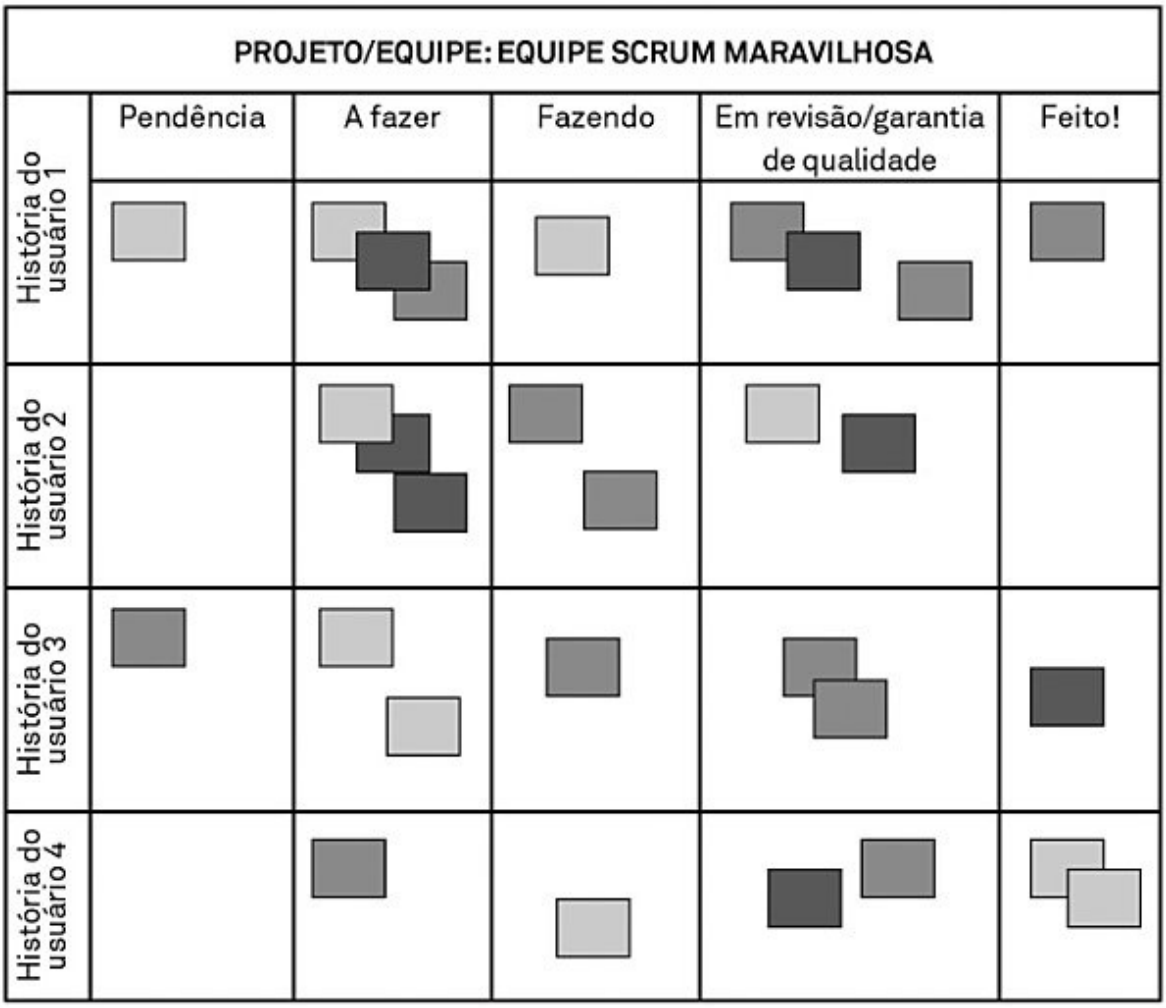
"Confiem em mim", pedi. "Isso não será usado para magoar nem para punir vocês. Isso será usado para melhorar as coisas".

Conforme eu disse antes, não estou muito interessado em desempenho individual; apenas no desempenho da equipe. Eu consigo dobrar a produtividade de uma equipe em um mês, mas um indivíduo? Isso levaria um ano. E um grupo completo de indivíduos? Um departamento inteiro? Uma empresa inteira? Isso levaria uma eternidade. Então, uso a transparência para colocar o foco no aprimoramento da equipe. Acho que a equipe em si, em geral, consegue lidar com questões individuais de desempenho. Eles realmente sabem o que as pessoas estão fazendo, quem está ajudando, que está atrapalhando, quem torna a equipe ótima, quem dificulta o trabalho.

Então, no Scrum, tudo é visível. Nas minhas empresas, todos os salários, toda contabilidade financeira e cada gasto, tudo está disponível para todo mundo. Eu nunca entendi por que alguém deseja manter esses dados em segredo, a não ser para favorecer os próprios interesses individuais ou para infantilizar os relacionamentos. Eu quero que o assistente administrativo seja capaz de ler uma demonstração de lucros e perdas e que compreenda *precisamente* de que forma o trabalho dele contribui para isso. Quero que todos

na empresa estejam alinhados com um objetivo unificado. Pulverizar as pessoas em feudos informativos simplesmente as torna mais lentas; e desperta desconfiança e suspeita. Além disso, divide a empresa entre os poderosos, que sabem das coisas, e os peões, que apenas executam segmentos de algum objetivo misterioso o qual não conseguem compreender. Papo furado. Se você não pode confiar nas pessoas que está contratando para subirem a bordo com você, então está contratando as pessoas erradas e estabeleceu um sistema no qual o fracasso já está incorporado.

A representação visual mais dramática dessa ideia, e uma imagem que você verá na sala de toda equipe Scrum do mundo, é o quadro Scrum.



Agora, existe um tal software capaz de mensurar todo tipo de dados, fornecer diversas modalidades de métrica e análise, mas um quadro Scrum não passa de um monte de *post-its* colados em um quadro branco. Existem três níveis de status para as tarefas: “Pendências”, “Fazendo”, “Feito”. Quando alguém

assume uma tarefa, todo mundo sabe quem está trabalhando nela, e quando ela será concluída. E porque o quadro tem *post-its* que representam tudo que precisa ser feito em um único Sprint, todo mundo conhece a sua evolução. Qualquer pessoa pode entrar na sala, olhar para o quadro e saber exatamente como a equipe está se desenvolvendo.

Como todas as pessoas da equipe sabem tudo o que já foi “Feito” e o que ainda precisa ser feito, elas podem se autorregular. Sabem o que precisam fazer, conseguem perceber se um colega precisa de ajuda, se uma história está há muito tempo na coluna “Fazendo”. Quando tudo está claro, a equipe pode se auto-organizar para superar os problemas que se tornaram óbvios.

Na PatientKeeper, a transparência a qual aqueles desenvolvedores temiam inicialmente funcionou muito bem. Como todo o trabalho ficou “aberto”, fomos capazes de coordenar as tarefas entre várias equipes. Todos sabiam exatamente no que cada um estava trabalhando, o tempo todo. Eles poderiam apoiar uns aos outros se alguém se deparasse com um obstáculo. Um desenvolvedor talvez já tivesse descoberto uma solução para um problema que outro estava enfrentando. Mesmo que não estivessem na mesma equipe! A produtividade na PatientKeeper mais que quadriplicou. Não estamos falando de uma atualização do Angry Birds; estamos falando de algo usado nos maiores hospitais, e a vida das pessoas depende disso. Mas como éramos transparentes com tudo, conseguimos colocar o produto no mercado mais rápido do que qualquer pessoa do mundo. É isso que o “Sunshine” pode fazer.

Depois que saí da PatientKeeper, uma nova equipe de gestores decidiu que o Scrum não era mais a melhor forma de gerenciar as coisas. O resultado? O lançamento de produtos caiu de 45 por ano para dois; a receita de US$ 50 milhões para US$ 25 milhões; o desgaste que fora menos de 10% subiu para mais de 30%. Eles deixaram de ser uma grande empresa e voltaram a apresentar um desempenho medíocre ao retornar ao comportamento corporativo tradicional.

Proporcionando felicidade

Uma empresa que coloca a felicidade como o centro de sua cultura é a Zappos. Esse website muito bem-sucedido convenceu as pessoas a fazerem algo que muita gente achou que não poderia ser feito: comprar sapatos pela Internet. O CEO Tony Hsieh escreveu um livro sobre isso, *Satisfação*

garantida, no qual fala sobre a cultura ímpar da Zappos, que se baseia na criação de momentos "Uau!" para os clientes. Acontece que, para conseguir isso, você precisa de pessoas felizes do outro lado do telefone.

Ao conversar com os executivos da organização, uma das palavras que você mais ouve é *conexão*. A pesquisa que fizeram demonstra que quanto mais as pessoas se conectam com as pessoas com quem trabalham, mais felizes elas são — e, ao que parece, mais produtivas e inovadoras também. Então, os executivos da empresa decidiram criar tais conexões de forma deliberada — não apenas em uma equipe, ou em um departamento, mas em toda a empresa. E não somente entre pessoas em um nível, mas entre níveis diferentes, todos, desde VPs até o responsável por contas a receber.

Eles fazem isso usando tanto meios simples, quanto complexos; por exemplo, encorajam fisicamente encontros ao acaso. O prédio da empresa tem muitas saídas, mas elas ficam todas fechadas, a não ser uma, o que força as pessoas a entrar e sair por aquele mesmo lugar. A ideia é que, ao fazer com que as pessoas se encontrem, tenham mais chances de criar e cultivar tais conexões.

Outro exemplo é o modo como as pessoas entram na cultura da Zappos. Cada funcionário, desde um trabalhador do depósito até um diretor, tem de passar pelo que Christa Foley, a Gerente Sênior de Recursos Humanos, chama de "campo de treinamento". Durante quatro semanas, cada funcionário recebe um treinamento rápido de como a empresa funciona, mas também sobre como a cultura da empresa funciona. Trata-se, na verdade, da segunda triagem do processo de contratação da Zappos. Mesmo depois de receber a oferta de trabalho, você precisa provar que consegue absorver a cultura.

Foley afirma que os resultados são extraordinários. "As conexões [que os funcionários] fazem durante o treinamento de campo se mantêm por toda a sua carreira". O treinamento em campo é intenso propositalmente — as pessoas têm de chegar às sete horas da manhã, trabalhar duro, cumprir prazos e passar em testes. Mas funciona. As pessoas que passam pelo campo de treinamento continuam se relacionando não apenas por meses, mas por anos, organizando os próprios encontros e churrascos para manterem contato.

"As conexões se tornam uma extensão da família", opina Rachel Brown, uma executiva da Zappos. "Você leva os seus amigos de trabalho para sua casa. Você passa tempo com eles".

Outra maneira que a Zappos usa para manter as pessoas felizes é dar aos funcionários uma chance de aprender e crescer. A empresa quase sempre prefere contratações internas. Digamos que abra uma vaga no RH, e alguém da

contabilidade, que sempre achou que talvez fosse gostar daquele tipo de trabalho, fique sabendo da vaga. Aquela pessoa interessada no departamento de RH entra em um tipo de "programa de aprendizado". Isso dá ao funcionário a chance de ver se ele realmente gosta daquele trabalho, e ao gerente de avaliar se aquele funcionário se encaixará bem na equipe. A empresa também oferece aulas gratuitas ministradas por outros funcionários — Introdução à Contabilidade, Codificação para Iniciantes etc. A Zappos quer que as pessoas cresçam na empresa e com ela.

Conforme mencionei no Capítulo 3, as pessoas querem crescer; querem ficar melhores do que estão e encontrar outras coisas nas quais podem melhorar também; a ideia de dominar o trabalho as motiva. Dar a chance de descobrir onde se encaixar melhor ajudou a Zappos a manter os funcionários felizes, animados e comprometidos.

Para muitas pessoas que seguiram uma carreira muito tradicional, essa cultura pode ser um sopro de ar fresco. "Em toda a minha carreira antes da Zappos, meu maior foco era no recrutamento", conta Foley. "Era um trabalho de repetição" e ela estava exausta. Trabalhar na Zappos a revigorou. "Foi a cultura", disse ela. "É isso que me anima a vir para o trabalho".

É isso o que a Zappos deseja e o que qualquer empresa deveria desejar. É o que eu quero: pessoas que amem vir trabalhar. Estou falando de uma mudança de pensamento: de "trabalhar para uma empresa" para "trabalhar com a *minha* empresa". Esse é um estado mental que algumas pessoas têm dificuldades em adotar, por isso a Zappos mantém o foco em promoções internas. Eles descobriram que as pessoas que vêm de fora, principalmente de níveis mais altos, têm dificuldades para se adaptar. "Somos uma mistura de empreendedores e inovadores", explica Foley. "Mas isso é só metade da história. A outra metade é a colaboração". A empresa quer pessoas que trabalhem juntas em seus relacionamentos na organização. Isso às vezes não se encaixa na cultura corporativa padrão. Um gerente sênior me disse: "eu não tenho um cargo. Acreditamos que como um grupo conseguimos nos sair muito melhor".

Em geral, nas empresas você vê gestores que querem coordenar as próprias áreas sem transparência ou colaboração. Eles criam uma dinâmica "nós *versus* eles". Territórios são definidos, e você quase consegue enxergar as diferentes divisões conspirando umas contra as outras, como uma corte medieval maquiavélica. Imagine como uma empresa seria mais produtiva se todos trabalhassem juntos, em prol de um objetivo comum. Imagine uma empresa à

qual todos se referem como a *minha* empresa, na qual cada dia representa uma chance de tornar-se melhor, de fazer algo maior, de aprender algo novo. Em vez disso, a maioria das corporações estabelece um ambiente no qual as pessoas estão mais envolvidas em questões políticas do querem aumentar os lucros.

Na Zappos, se você não se encaixar na equipe e na cultura, você não se encaixa na empresa. A taxa anual de atrito da empresa é de 12%, e a maior rotatividade de funcionários, dizem eles, acontece na central telefônica. Isso porque eles despedem as pessoas que não têm paixão ao atender os clientes. A Zappos considera esses funcionários o rosto da empresa, e os seus padrões são altos. As pessoas na Zappos são flexíveis em relação a um monte de coisas, mas não a isso.

Eu já vi essa mesma dinâmica acontecer nas equipes. Uma pessoa em uma equipe pode ter algum conhecimento ou habilidade específicos — conhecimentos que vão juntando como pessoas avarentas e sovinas, sem compartilhar. Eles consideram esses conhecimentos uma posse que assegura seus empregos. O Scrum, por meio das Retrospectivas e transparência, elucida esse tipo de comportamento de forma quase imediata. Torna-se óbvio onde os obstáculos se encontram no mesmo lugar onde está o desperdício. Quando dirijo uma empresa, digo para as pessoas com hábitos “sovinas” que elas não podem se dar o luxo de manter a equipe e a empresa como reféns. Elas podem mudar o estado mental ou ir trabalhar para outra pessoa.

A Zappos descobriu que, quanto mais alto o cargo do novo funcionário, mais enraizado é o seu pensamento e, consequentemente, precisa se esforçar mais para abandonar as maneiras antigas de fazer as coisas. O Scrum dá às pessoas uma estrutura em que essa mudança aconteça. Fornece uma base para a empresa inteira seguir na direção do objetivo comum. Seus pilares são a transparência, o trabalho em equipe e a colaboração. Muitas empresas agora adotam essa filosofia, e é inevitável que as que não adotam perderão para as que o fizeram.

A Zappos deu um salto de US$ 1,6 milhão em vendas em 2000 para mais de US$ 1 bilhão em 2008. Essa é uma taxa de crescimento de 124% ao ano durante oito anos seguidos. Não sei quanto a você, mas eu acho que esse é um argumento bastante convincente para fazer as pessoas felizes. E o Scrum é a caixa de ferramentas que você pode usar para chegar lá.

Estoure a bolha da felicidade

Uma das coisas que a felicidade não é — pelo menos não o tipo a que *eu estou* me referindo — é complacente. Na verdade, ela é exatamente o contrário disso: um compromisso positivo e passional. Conforme Christa Foley, da Zappos, diz, a felicidade é o que há de mais distante da passividade. “Eu amo vir trabalhar. Em vez de [encorajar a] complacência, a nossa cultura positiva e enriquecedora faz com que você trabalhe com mais afinco”. Eles precisam retirar as pessoas que acham que trabalhar em um local feliz significa não trabalhar; querem sim é pessoas que usem a alegria como uma força motriz.

E, saiba você, eles não estão sozinhos nisso. A *Harvard Business Review* dedicou toda a edição de janeiro-fevereiro de 2012 à felicidade. O que descobriram foi:

> [...] que a única rota para a felicidade do funcionário que também beneficia os acionistas acontece por meio de um senso de completude resultante de um trabalho importante bem feito. Devemos almejar não apenas tornar os funcionários “felizes”, mas fazer isso ajudando-os a fazer grandes conquistas. Em outras palavras, devemos conquistar o apoio passional dos nossos funcionários à missão e ao sucesso da empresa ao ajudá-los a conquistar o apoio passional dos clientes.[39]

E esse apoio passional traz benefícios tangíveis. Funcionários felizes não faltam ao trabalho, eles trabalham com mais afinco, e não apenas não deixam a empresa, como também atraem outros que apresentam o mesmo ímpeto que eles. Em seu artigo para a edição da HBR, Gretchen Spreitzer e Christine Porath decidiram não chamar essas pessoas de “felizes”, por causa das conotações de complacência. Em vez disso, eles usaram o termo “afortunados”. Eles descobriram que essas pessoas apresentavam um desempenho 16% melhor do que seus pares, tinham uma taxa de esgotamento 125% menor, eram 32% mais comprometidas, e estavam 46% mais satisfeitas com os seus respectivos trabalhos. Elas faltavam menos por questões de saúde, tinham menos consultas médicas, e mais chances de ser promovidas.[40]

O que esses “afortunados” têm em comum é o assunto que estou abordando neste capítulo — cada afortunado é passional e cheio de vida, e está tentando aperfeiçoar sua arte, seja ele parte de uma tripulação de um avião ou um ajudante de garçom em um restaurante. O que as empresas podem fazer para criar uma atmosfera em que as pessoas possam florescer? Os gestores podem encorajar a autonomia ao permitir que as pessoas tomem as próprias decisões

sobre como fazer seu trabalho. E podem se certificar de que os funcionários saibam tudo o que está acontecendo, porque, como eles dizem, “trabalhar em um vácuo de informações é tedioso e monótono”. Os gestores também devem ter tolerância zero em relação a grosserias e jamais permitir que um funcionário envenene a cultura corporativa por meio do abuso ou desrespeito. E, por fim, eles devem dar um feedback rápido e direto.

O Scrum dá às pessoas tudo isso — foi desenvolvido para *fazer* com que eles possam acontecer, principalmente o feedback direto, que acontece todos os dias nas reuniões diárias, e é para isso que a Retrospectiva do Sprint e a Métrica de Felicidade foram criadas para elucidar.

Existe, porém, um cuidado que eu gostaria de descrever. É possível — droga, isso acontece vezes o suficiente para fazer com que eu dedique um tempo significativo estudando isso — que uma “bolha de alegria” se desenvolva. Isso costuma acontecer depois que a equipe conquistou um grande sucesso ou aumentou enormemente sua produtividade usando o Scrum. Eles se auto-organizaram e se sentem orgulhosos com o progresso que conquistaram; e é nesse momento que a complacência pode se infiltrar. Eles dizem para si mesmos: “ei, nós melhoramos tanto e não precisamos mais melhorar”. Eles atingem um platô de produtividade e, logo depois, deixam de realizar um grande trabalho. Mas eles são bons o suficiente e, por um tempo, vivem nessa bolha isolante de felicidade. Não se dão conta de que um aprimoramento contínuo significa exatamente isso: não devem parar nunca, em momento algum. Quando eu era piloto de combate, costumávamos dizer que, depois de três mil horas na cabine, você precisa se aposentar porque pode se tornar complacente e isso poderia matar você. Embora uma equipe complacente seja menos arriscada nos negócios, o desempenho contínuo da equipe está em risco.

Tal atitude complacente, em geral, se revela em comentários como: “nós *merecemos* pegar leve; conquistamos esse direito”. Ou membros individuais da equipe valorizam o seu espírito de equipe e felicidade de forma tão intensa que não querem se arriscar, ou eles temem a mudança em si, sentindo que se o que eles têm está funcionando, por que mudar?

Porque é aí que as formas do Scrum podem sucumbir, “bolhas de felicidade” constituem uma das minhas maiores preocupações. Eu já vi isso repetidas vezes: uma equipe pode estar fazendo tudo que o Scrum ensina — priorizar, realizar tarefas individuais, usar interfuncionalidade e fazer rituais de revisão —, mas pararam de se aprimorar. Em geral, elas estão muito melhores do que eram antes de aprender o Scrum, e têm seus sucessos para provar isso, mas elas

relaxam em seus louros, dizendo: “não *precisamos* melhorar mais”.

Isso me faz lembrar do time americano de basquete das Olímpiadas de 2004. Aquele time contava com alguns dos melhores jogadores do país — LeBron James, Tim Duncan e Allen Iverson, para citar apenas alguns —, e os Estados Unidos tinham um histórico de não apenas vencer, mas também dominar o esporte, principalmente depois que jogadores profissionais passaram a poder participar. Os jogadores americanos de basquete sabiam que eram os melhores. Só que *não eram*. Eles perderam mais jogos do que qualquer outro time americano de basquete em uma Olímpiada. Eles perderam para a *Lituânia*. O orgulho e a complacência foram sua queda. Eles viviam na bolha da felicidade.

Então, como você estoura a bolha antes que os seus jogadores envergonhem o país, ao vivo na TV, diante de bilhões de pessoas? O primeiro passo é estar ciente do problema, e é por isso que eu quero que as equipes meçam a sua velocidade a cada Sprint. Eu quero saber qual é a taxa de mudança deles. Se não houver crescimento positivo, eu sei que temos de nos debruçar sobre o assunto. E a pessoa de quem dependo para isso é o Mestre Scrum. Ele precisa ser capaz de detectar o problema e abordá-lo com a equipe. É essencial que ele faça perguntas difíceis. O que você realmente quer é um “tolo sábio”.

> Eu gostaria de entender que espécie de parentesco existe entre você e suas filhas: elas ameaçam me espancar porque digo a verdade;
> tu mandas me açoitar porque minto; e algumas vezes apanho por não falar nada.[41]
>
> — *O Rei Lear*, ato 1, cena 4

Um “tolo sábio” é uma pessoa que faz perguntas desconfortáveis ou levanta verdades desconfortáveis. Nem sempre é fácil ter esse tipo de pessoa por perto, já que eles podem ser considerados encrenqueiros ou alguém que não faz parte da equipe, mas precisam ser cultivados e usados.

Talvez o melhor exemplo seja um que todos já conhecemos — do conto clássico de Hans Christian Andersen, *A Roupa Nova do Imperador*. Conforme você deve se lembrar, havia um imperador que tanto adorava roupas finas, que tinha um casaco diferente para cada hora do dia. Se você quisesse saber onde eles estavam, o primeiro lugar que deveria procurar era no quarto de roupas. Um dia, alguns caloteiros foram até o imperador e juraram que tinham um tecido secreto que era tão fino que aqueles que não eram dignos não conseguiriam enxergá-lo. Eles exigiram uma seda finíssima, mas apenas fingiam tecê-la e, na verdade, “teciam” apenas o ar, e o material foi para as

suas bolsas. Um dia, o imperador veio para verificar o progresso do trabalho deles e não viu nada. Lembrando-se de que o tecido só era visível para aqueles que fossem dignos, ele elogiou o tecido como o mais lindo que ele jamais vira. Ele pediu a opinião de seus conselheiros, mas todos também juraram que se tratava do material mais maravilhoso de todos. No dia da entrega, os caloteiros vestiram, cuidadosamente, o imperador com absolutamente nada, e ele recebeu elogios exagerados de toda a corte. Então, o imperador decidiu desfilar pela cidade para exibir para o povo aquele tecido miraculoso.

Você se lembra como o conto termina: ninguém disse nada sobre a nudez do imperador, pois ninguém queria ser considerado indigno. Assim, a procissão real continuou pelas ruas até que uma criança gritou: “mas ele não está vestindo nada!”. A princípio, o pai da criança mandou que ela ficasse quieta, mas, então, começando com um sussurro que foi aumentando até um grito, e as pessoas começaram a gritar: “ele está pelado!”. O imperador, mesmo temendo que eles estivessem certos, continuou a procissão. E sua corte o seguiu, segurando uma cauda que não existia.

O tolo sábio foi aquela criança — a pessoa que consegue ver que a verdade aceita é simplesmente uma ilusão consensual, e que o imperador realmente estava sem roupas. Então, se você tem um tolo sábio ou dois, trate-os com carinho.

Existem outras maneiras de estourar a bolha da felicidade — por exemplo, ao trazer sangue novo e uma intervenção da gerência —, mas, no final das contas, é a mesma coisa: fazer a equipe encarar a realidade que talvez não queira ver. Felizmente, com o Scrum tudo é transparente — a produtividade da equipe, a qualidade do trabalho, o nível de satisfação do cliente. Uma das virtudes do Scrum é que ele torna o desconfortável visível de forma bastante rápida. Em comparação, as equipes e organizações tradicionais podem caminhar alegremente até um penhasco e se perguntarem o que pode ter saído errado. Eles esperam muito tempo para ter um feedback prático do mercado e uns dos outros.

Feliz hoje, feliz amanhã

Psicólogos, incluindo Ben-Shahar, de Harvard, dizem que uma das maneiras de analisar de que forma as pessoas tentam conhecer o mundo é perguntando se o que elas fazem as deixa felizes hoje, ou se as tornará felizes amanhã.

Descobri que isso é uma lente muito útil para observar as pessoas em ambientes de trabalho.

Os seres humanos tendem a se encaixar em quatro tipos, de acordo com Ben-Shahar. O primeiro, o "hedonista", é alguém que faz aquilo que o torna feliz agora. "Amanhã? *Deixe que o amanhã se preocupe com o amanhã. Eu só vou aproveitar o hoje"*. Vejo muito esse tipo de comportamento em start-ups: um monte de gente reunida em uma garagem figurativa apenas fazendo coisas porque elas são legais e divertidas, sem dar muita atenção a criar um produto sustentável. Uma quantidade muito pequena de energia mental é canalizada para como as coisas estarão funcionando dali a um mês, que dirá no período de um ano.

E o que costuma acontecer é que os investidores desses caras começam a ficar preocupados. Então, eles contratam um banco de gestores para cuidarem do rebanho de trabalhadores competentes. E, de repente, aqueles trabalhadores descobrem que o mundo que tanto adoravam é uma droga, pois agora existem todos os tipos de regras, testes e relatórios; é uma droga hoje, e eles acham que será uma droga para sempre. Vamos chamá-los "niilistas".

Então, tem aqueles caras que foram trazidos para gerenciar o lugar. São os dispostos a trabalhar oitenta horas por semana (e desejosos de obrigar os *outros* a fazerem o mesmo), porque acreditam que serão promovidos depois e, por isso, serão mais felizes. Então, quando isso finalmente acontece, eles precisam se contentar apenas com novas dores de cabeça, que exigirão ainda mais tempo de trabalho. É. Eles curtem uma "competição exagerada".

O quarto tipo de pessoa é aquele que o Scrum tenta identificar e encorajar — o indivíduo que está trabalhando em coisas que são legais hoje, mas mantendo um olhar em direção a um futuro melhor, e que está convencido de que o trabalho será divertido para sempre. Esse tipo de pessoa raramente passa por um esgotamento ou uma desilusão. Ele não tem os sentimentos negativos em relação ao trabalho sofrido pelos hedonistas, os niilistas e os gerentes viciados em competições exageradas que lutam para deixar todo mundo na linha.

O que o Scrum faz é promover um único estado mental estimulante. Ao fazer com que todos trabalhem juntos, a equipe ajuda o hedonista a olhar para o futuro, convence o niilista de que há um futuro sem sofrimento e diz para os gerentes presos na competição sem fim que, na verdade, existe uma forma melhor de fazer as coisas.

Foi por isso que adotei a Métrica da Felicidade na minha empresa. Ela possibilita que a equipe ajude seus membros a se tornarem pessoas melhores,

removendo as causas de infelicidade de maneira sistemática, cuidadosa e incremental. Ou seja, dá autonomia para que as pessoas mudem, e incentiva para que o façam.

Lembra do Erro Fundamental de Atribuição? Quando você está cercado por pessoas escrotas, não procure por pessoas más; procure por sistemas ruins que recompensam essas pessoas por agirem dessa forma. Então, você aplica a Métrica da Felicidade para consertar esse tipo de problema.

No Ensino Médio ou na faculdade, muitos americanos estudam a hierarquia de necessidades do psicólogo americano Abraham Maslow. Ela organiza, em forma de pirâmide, as necessidades de que os seres humanos cuidam primeiro e, depois, aquelas que se tornam mais prementes quando as mais básicas já estão satisfeitas. Na base da pirâmide estão as necessidades fisiológicas: ar, água, alimentos, roupas e abrigo. Se não tivermos isso, não poderemos nem começar a pensar em qualquer outra coisa. No nível seguinte está a segurança — não apenas física e financeira, mas também a garantia de saúde; é importante ter acesso a serviços de saúde. Surpreendentemente, muitas pessoas param aí, mesmo que o nível seguinte contenha algo de que como seres humanos certamente precisamos, mas que a sociedade costuma ignorar: amor e um senso de pertencer a algum lugar — aquela conectividade da qual a Zappos fala. Acima disso, está a necessidade de autoestima e respeito dos outros. E, no topo da pirâmide, está a necessidade de atingir o potencial completo.

É nessa camada superior que Maslow está mais interessado, e é nela que o Scrum se concentra: as pessoas conquistarem crescimento pessoal e realização. Quem está no topo daquela pirâmide não é apenas mais feliz e mais realizado, é mais eficaz e inovador; e, sim, é capaz de alcançar a grandiosidade.

Eu quase posso ver você assentindo agora, porque todos conhecemos aquela pirâmide em algum nível intuitivo, mesmo que alguns de nós nunca a tenha visto desenhada. O truque é chegar aos cumes elevados e, depois, encontrar uma maneira de aferir *precisamente* qual impacto você está tendo. Se você está dirigindo uma empresa, talvez a sua medida de grandiosidade seja a receita e o crescimento. Se está tentando curar as pessoas, é provável que seja o número de pacientes que não morrem. Se está empenhado em mudar o mundo, talvez seja *o quanto* já avançou em tal projeto. E se você está tentando realizar as tarefas que sua esposa passou para você, talvez a sua medida de grandiosidade seja quantas tardes de fim de semana você tem livres para ir pescar.

Não é suficiente apenas ser feliz. A felicidade precisa servir como um equipamento para produzir resultados. Todos os elementos do Scrum se unem

para ajudar uma pessoa a fazer exatamente isso. O verdadeiro truque? Prioridades. E é sobre isso que vamos falar no próximo capítulo.

PONTOS PRINCIPAIS

É a viagem, não o destino. A verdadeira felicidade é encontrada no processo, não no resultado. Em geral, só somos recompensados pelos resultados, mas *realmente* queremos recompensar as pessoas que se empenham para chegar à grandiosidade.

Felicidade é o novo pretinho básico. A felicidade o ajuda a tomar decisões mais inteligentes. Além disso, quando você está feliz, torna-se mais criativo, menos propenso ao emprego e mais propenso a conquistar muito mais coisas que imaginou.

Quantificar a felicidade. Não basta apenas sentir-se bem; você precisa mensurar tal sentimento e compará-lo ao desempenho real. Outras métricas olham para o passado; a felicidade é uma métrica que olha para o futuro.

Fique melhor a cada dia — mensure isso. Ao final de cada Sprint, a equipe deve escolher um pequeno aprimoramento, ou *kaizen*, que tornará mais feliz. E essa deve ser a coisa mais importante que conseguirá fazer no próximo Sprint.

O sigilo é um veneno. Nada deve ficar em segredo. Todo mundo deve saber tudo, e isso inclui informações financeiras e salários. A obscuridade só serve para as pessoas que só se preocupam consigo mesmas.

Torne o trabalho visível. Tenha um quadro que mostre todo o trabalho que precisa ser feito, e o que já foi feito. Todos devem vê-lo e atualizá-lo todos os dias.

Felicidade é autonomia, maestria e propósito. Todos querem controlar o próprio destino, ficar melhores no que fazem e ter um objetivo maior do que si mesmos.

Estourar a bolha de felicidade. Não fique tão feliz a ponto de acreditar nas suas próprias besteiras. Certifique-se de que a felicidade seja mensurada em relação ao desempenho, e, caso haja uma desconexão, esteja preparado para agir. A complacência é inimiga do sucesso.

38 Sonja Lyubomirsky et al., The Benefits of Frequent Positive Affect: Does Happiness Lead to Success?, *Psychological Bulletin*, Vol 131(6), 2005, pp. 803-55.

39 Gretchen Spreitzer e Christine Porath, Creating Sustainable Performance *Harvard Business Review,* Jan./Fev. 2012, pp. 3-9.

40 *Ibid*.

41 William Shakespeare, *King Lear*. [Em português, *O Rei Lear,* Trad. Millôr Fernandes, Porto Alegre, L&PM, 1997, Fala do Bobo, ato 1, cena 4, p. 33.]

CAPÍTULO 8

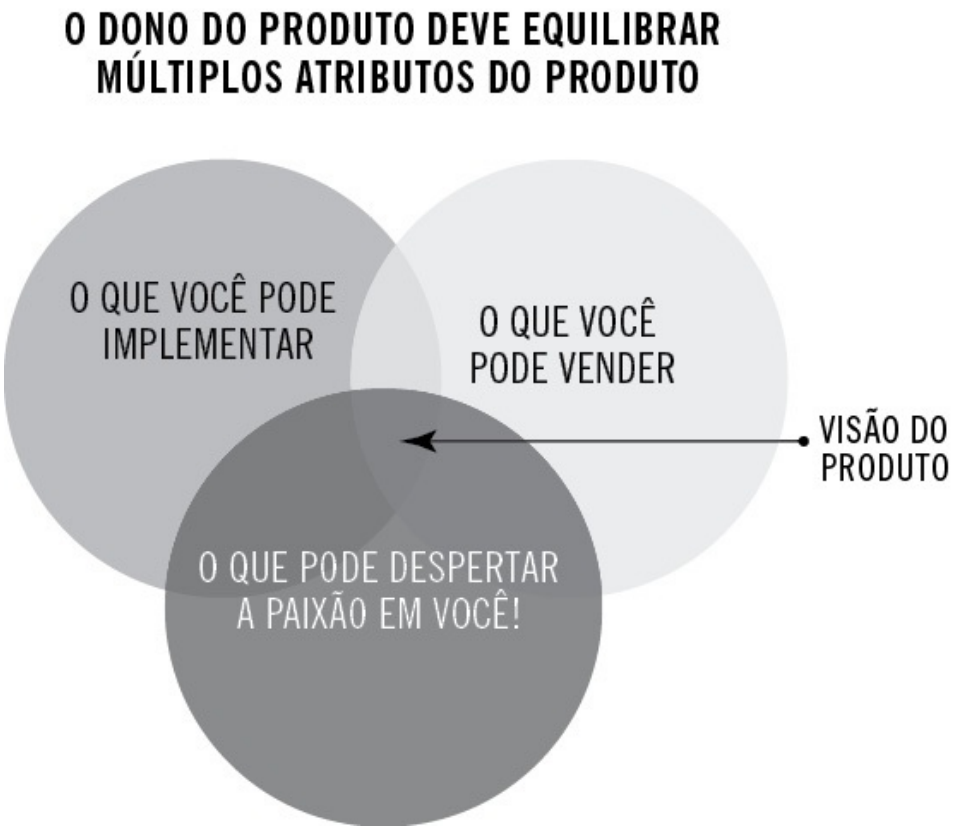
Prioridades

Alguns anos atrás, conheci Scott Maxwell, na Johnny’s Luncheonette, em Newter Center. Já falei sobre ele antes. Ele é o fundador da OpenView Venture Partners, e foi o homem que descobriu que trabalhar por mais horas gera mais trabalho do que conquistas. Eu já trabalho com a OpenView e as empresas de seu portfólio há alguns anos e, em cada uma delas, observei aumentos drásticos na produtividade. Mas o Scrum não é apenas sobre fazer as equipes trabalharem mais rápido; é sobre impulsionar o impacto, que, no caso dos caras da VC, tinha uma forma muito simples: receita. Se a empresa não estiver ganhando dinheiro, você não tem um empreendimento bem-sucedido, você tem um *hobby*.

Eu nem sei dizer quantas empresas já vi com grandes ideias e produtos muito legais — algo que deixa todo mundo animado, que parece preencher um nicho de mercado, e que parece que *deveria* ser um sucesso. É tão legal. Mas apesar das doses enormes de imaginação, inspiração e trabalho árduo, as pessoas que fizeram o produto nunca conseguem descobrir como ganhar dinheiro com ele.

Qual é a diferença entre uma Pets.com e uma Zappos? Ambas viram um segmento de mercado no qual as pessoas gastam bilhões de dólares por ano, e enxergaram uma maneira de entregar produtos de forma mais fácil e barata on-line. Uma se tornou um símbolo dos excessos de uma empresa “ponto com” e do desperdício de milhões e milhões de dólares; a outra é uma empresa que vale mais de US$ 1 bilhão. Ambas tinham uma visão — o que a Pets.com *não* tinha era um senso de prioridades. Eles não sabiam o que fazer, nem quando.

Gosto de mostrar para as pessoas este diagrama de Venn:



Toda empresa precisa pensar sobre este diagrama. Se você se concentrar apenas no que pode construir, é possível que acabe fazendo algo que ninguém, na verdade, quer, mesmo que esteja entusiasmado com a ideia. Se você só se concentrar no que pode vender, talvez prometa coisas que não conseguirá construir. E se você só construir o que consegue vender, mas sem qualquer entusiasmo, acabará trabalhando duro para construir algo medíocre. Mas, no centro, naquele ponto ideal, há uma visão calcada na realidade — uma visão com uma possibilidade real de se tornar algo grandioso. Neste capítulo, vou mostrar a você como chegar lá. Os capítulos anteriores se concentraram em como fazer as coisas de forma melhor e mais rápida. Este capítulo é sobre como fazer com que o “melhor e o mais rápido” funcionem para você atingir a grandiosidade.

Scott Maxwell afirma que a verdadeira força do Scrum está no seu catálogo de pendências pronto, priorizado e com tamanho devidamente estimado do que fazer. Foi por isso que ele implementou o Scrum no grupo de risco e é por isso que ele acredita que esse método constitui uma vantagem competitiva crucial.

Pendências: o que fazer e quando fazer

A primeira coisa que você precisa fazer quando está implementando o Scrum é criar uma lista de pendências. Ela pode conter centenas de itens, ou apenas algumas poucas coisas que você precisa fazer primeiro. É claro que você precisa ter uma ideia clara do que deseja no final do trabalho, que pode ser um produto, um casamento, um serviço, uma nova vacina ou a pintura de uma casa. Pode ser qualquer coisa, mas, uma vez que você tenha uma visão, precisa pensar no que será necessário para fazer aquilo acontecer.

Comecei a trabalhar recentemente com uma empresa que desenvolve sistemas de automação para prédios — aquecimento, refrigeração, eletricidade, encanamento e todo o resto. Um de seus novos produtos é um sistema de automação do lar. Eles estão desenvolvendo um software capaz de controlar todos os aspectos da sua casa, desde abrir a porta da frente, passando por controlar os custos com aquecimento, até acender as luzes — tudo gerenciado a partir de seu dispositivo móvel. Então, reuniram-se e fizeram uma lista de tudo que precisariam para fazer aquilo acontecer: computadores, controles, interfaces, sensores, protocolos de comunicação e tudo mais. Não eram as regras e partes específicas, na verdade, mas todas as "histórias" de que precisariam.

Então, escreveram coisas do tipo: "como dono da casa, quero poder ver quem está à porta para que eu possa abri-la apenas para as pessoas que eu quero que entrem". Eles escreveram histórias sobre abrir a porta da garagem, ligar o sistema de aquecimento e refrigeração e controlar as luzes. Continuaram escrevendo até que tivessem uma lista de todas as coisas que achavam que o sistema precisaria ter para atrair compradores.

Ao final, a lista contava com centenas de itens. Trata-se de um sistema grande e complexo. A ideia subjacente à Lista de Pendências é que ela deve conter tudo que poderia ser incluído no produto. Você nunca vai realmente desenvolver aquilo tudo, mas quer uma lista de tudo que *poderia* ser incluído na visão daquele produto.

A chave, porém, é decidir o que será feito primeiro. As perguntas que você deve fazer são: quais os itens que terão maior impacto para a empresa, que são mais importantes para o cliente, que possam resultar em um ganho maior de dinheiro e são mais fáceis de fazer? Você precisa perceber que existe um monte de coisas na lista as quais você nunca chegará a fazer, mas, primeiro, vai querer aquelas que gerem mais valor e tenham o menor risco. Com o desenvolvimento e entrega por incrementos do Scrum, você quer começar com os itens que criarão uma receita imediata, tornando o projeto menos "arriscado" e eficaz. E

você quer fazer isso no nível dos atributos, ou seja, gerando valor para seus clientes o mais rápido possível. Deseja algo que esteja completamente Feito — que você possa mostrar; pode ser apenas uma pequena parte de um projeto maior, mas deve estar Feito, concluído, de uma forma que possa ser demonstrável. Se você está pintando uma casa, por exemplo, talvez a primeira coisa a ser concluída sejam as vigas da sala de estar.

No desenvolvimento de produtos, existe uma regra dura e rápida que já foi comprovada milhares de vezes: 80% do valor está em 20% dos atributos. Pense nisso por um tempo. Em qualquer coisa que você compre, a maior parte do valor — a maior parte do que as pessoas querem — só está em um quinto do que foi desenvolvido. No caso dessa empresa, eles olharam para aquela imensa lista de coisas que poderiam ser incluídas no sistema de automação do lar que estavam desenvolvendo e sabiam — sabiam *com certeza* — que os clientes só queriam mesmo 20% deles. O truque do Scrum é descobrir como você desenvolve esses 20% primeiro. No desenvolvimento tradicional de produtos, as equipes não sabem o que isso constitui até entregarem o produto completo. Isso significa que 80% do esforço deles foi desperdiçado. E você já sabe como me sinto em relação ao desperdício.

E se você pudesse começar entregando as coisas cinco vezes mais rápido do que a sua concorrência, com cinco vezes mais valor? Essa, sim, seria uma jogada e tanto.

Então, essa empresa de automação se reuniu para avaliar aquela enorme lista de atributos e se perguntou: “tudo bem, então, o que vamos fazer amanhã? O que é mais importante para o cliente? Como geramos valor para eles mais rápido do que qualquer outra pessoa?”. Conforme Scott Maxwell diz, a parte mais difícil não é descobrir *o que* você quer conseguir, é descobrir como você *pode conseguir* isso. Isso é verdadeiro se você estiver construindo uma casa ou um carro, escrevendo um livro ou desenvolvendo um *video game*, encobrindo um crime ou descartando o lixo. Imagine onde o maior valor pode ser gerado com o menor esforço, e faça isso logo de cara. Então, identifique o incremento de valor seguinte, e o seguinte. Mais rápido do que você imagina, terá criado ou entregado algo com resultados reais e demonstráveis. A chave é priorizar o trabalho.

Como fazer isso? Bem, primeiro você precisa descobrir qual é a visão e onde o valor está. No Scrum, chamamos essa pessoa de Dono do Produto (Product Owner).

O Dono do Produto

Existem apenas três papéis no Scrum. Ou você faz parte da equipe e está fazendo o trabalho — é o Mestre Scrum, que ajuda a equipe a descobrir como trabalhar melhor; ou, então, você é o Dono do Produto, que decide que trabalho deve ser feito. Ele (ou ela) é o dono da Lista de Pendências, do que está ali e, o mais importante, em que ordem.

Quando comecei a primeira equipe Scrum, em 1993, eu não tinha um Dono de Produto. Eu fazia parte da equipe de liderança e tinha várias outras responsabilidades, além de descobrir o que cada equipe devia fazer nos Sprints. Eu era responsável pela gerência e pelo marketing, lidava com clientes, traçava estratégias. Contudo, naquele primeiro Sprint, achei que poderia lidar com as Pendências. Somente precisava me certificar de que tinha "histórias" e atributos suficientes para a equipe trabalhar durante o Sprint seguinte. O problema foi que, depois do segundo, nós incluímos as Reuniões Diárias. A velocidade subiu 400% no Sprint *seguinte*, e a equipe concluiu em uma semana o que acreditávamos que levaria um mês. Não havia mais Pendências para serem trabalhadas! Eu achei que teria um mês para criar mais "histórias". Tínhamos um grande problema, de fato, mas um que tinha de ser resolvido. Então, pensei sobre esse papel de Dono do Produto e que qualidades a pessoa deveria ter para desempenhá-lo de forma adequada.

A minha inspiração para o papel veio do Engenheiro Chefe da Toyota. Esse profissional é responsável por toda uma linha de produção, por exemplo, do Corolla ou do Camry. Para isso, eles têm de aproveitar o talento dos grupos especializados em engenharia da carroceria, chassis, elétrica etc. O Engenheiro Chefe precisa escolher profissionais de todos esses grupos para criar uma equipe interfuncional capaz de fazer um carro. Fora da Toyota, todos acham que esses lendários Engenheiros Chefes (ou *Shusas*, como eram originalmente chamados) são líderes todo-poderosos do "Modo Toyota". E, de certo modo, são. Mas o que eles não têm é autoridade. Ninguém se reporta a eles — ao contrário: são eles que se reportam aos seus grupos. As pessoas podem dizer aos Engenheiros Chefes que eles estão errados, e então eles têm de se certificar que estão certos. Eles não fazem avaliações de desempenho, não promovem funcionários nem dão aumentos. Mas decidem sobre o carro e como ele será feito — por persuasão, não coerção.

Era essa ideia que eu queria incorporar no Scrum. John Shook, do Lean

Enterprise Institute, uma vez começou sua descrição do cargo de Engenheiro Chefe citando o manual de liderança do Corpo de Fuzileiros Navais dos Estados Unidos:

> A responsabilidade de um indivíduo para a liderança não depende da autoridade... A suposição enraizada de que a autoridade deve ser equivalente à responsabilidade é a raiz de grande parte dos infortúnios corporativos. Acredito que o equívoco acerca dessa questão é desmedido, problemático e que está tão arraigado às nossas consciências que nem percebemos.[42]

Refletindo sobre o tempo que passei em West Point e no Vietnã, eu me descobri concordando que liderança não tem nada a ver com autoridade. Na verdade, tem muito mais a ver — entre outras coisas — com conhecimento e em ser um líder-servo. O Engenheiro Chefe não pode simplesmente *dizer* que algo tem de ser feito de uma maneira específica. Ele tem de persuadir, convencer e demonstrar que aquela maneira é a *melhor*. Em geral, precisa-se de alguém com trinta anos de experiência para assumir o papel. Eu queria aquilo no Scrum, mas também sabia muito bem que havia pouquíssimas pessoas com aquele nível de habilidade e experiência. Então, dividi o papel em dois, dando ao Mestre Scrum o *como* e ao Dono do Produto o *o quê*.

Mesmo naqueles dias iniciais do Scrum, sabia que eu precisava de alguém com uma profunda conexão com o cliente. O Dono do Produto precisava ser capaz de dar à equipe o feedback do cliente em cada um dos Sprints. Eles precisavam passar metade do tempo conversando com as pessoas que estavam comprando o produto (coletando suas opiniões sobre o último lançamento ampliado e como aquilo agregava valor) e a outra metade com a equipe, criando as Pendências (mostrando a eles o que os clientes valorizaram e o que não valorizaram).

Lembre-se de que o “cliente” deve ser o consumidor em geral, um grande banco, o seu marido ou alguém que precisa da vacina contra o rotavírus e depende de você para conseguir isso. O cliente, enfim, é qualquer pessoa que possa obter valor do que você está produzindo.

Mas eu não queria um gerente, e sim alguém em quem a equipe acreditaria e confiaria quando ele priorizasse as Pendências. Então, procurei o cara mais inteligente no departamento de Marketing de Produto — não na Engenharia, notem bem, mas no Marketing. E foi assim que Don Rodner se tornou o primeiro Dono do Produto. Ele conhecia o produto que estávamos desenvolvendo não do ponto de vista técnico — embora ele entendesse o

suficiente dessa parte para se comunicar com os engenheiros —, mas do ponto de vista de um cliente. Do que as pessoas que realmente usam o produto precisam? Ao escolher um Dono de Produto, escolha alguém que consiga se colocar no lugar de quem quer que seja e conseguir obter valor do que você está fazendo. Nas palavras de um amigo meu: “a minha esposa é uma excelente Dona de Produto; ela sabe exatamente o que ela quer. Eu apenas implemento”.

O Dono do Produto não apenas precisa ter uma variação mais ampla de habilidades do que o Mestre Scrum, mas também deve ser responsável por um conjunto diferente de padrões. O Mestre Scrum e a equipe são responsáveis pela rapidez com que estão produzindo e como podem aumentar a velocidade. O Dono do Produto é responsável por traduzir a produtividade da equipe em valor.

Ao longo de todos esses anos de trabalho, consegui resumir a quatro as características essenciais de um Dono de Produto:

1. Precisa ser bem-informado em relação ao setor. Com isso, eu quero dizer duas coisas: o Dono do Produto deve entender o processo que a equipe está executando o suficiente para saber o que pode ser feito e, tão importante quanto isso, o que não pode. Mas ele também precisa entender bem o *o quê* o suficiente para saber como traduzir o que pode ser feito em um valor verdadeiro e significativo. Pode ser um sistema de computador que irá ajudar o FBI a capturar terroristas ou um método de ensino que amplie o desempenho dos alunos nas escolas públicas. Esse profissional precisa conhecer o mercado a ponto de saber o que fará a diferença.

2. O Dono do Produto deve ter o poder de tomar decisões. Assim como a diretoria não deve interferir na equipe, ele precisa ter autonomia para decidir sobre a visão de como o produto será, e o que precisa ser feito para chegar até lá. Isso é importante porque o Dono está sob pressão de diversos *stakeholders*, tanto internos quanto externos, e precisa ser capaz de se manter firme. Ele deve ser responsável pelos resultados, mas permitir que a equipe tome as próprias decisões.

3. Ele tem de estar disponível para a equipe, para explicar o que precisa ser feito e o porquê. Embora o Dono do Produto seja basicamente o responsável pela definição das Pendências, é necessário que haja um diálogo constante com a equipe. Em geral, o especialista da equipe informa as decisões que o Dono do Produto precisa tomar. Então, essa figura precisa ser confiável, consistente e estar disponível. Sem acesso a ele, a equipe não saberá o que fazer, e muito menos em que ordem fazer. Eles dependem do Dono do Produto para obter “a visão” e, também, para que a inteligência de mercado saiba o que é importante. Se o Dono não estiver disponível para a equipe, todo o processo pode desmoronar. Eu raramente recomendo que CEOs ou outros altos executivos ocupem esse cargo. Eles não têm o tempo de que a equipe precisa.

4. Ele precisa ser responsável pelo valor. Em um contexto empresarial, o que importa é a receita. Eu avalio o Dono do Produto de acordo com a receita que gera por “ponto” de esforço. Se a equipe está produzindo quarenta pontos por semana, eu quero medir quanta receita é gerada para cada ponto. Mas a medida de valor poderia ser quantos sucessos a equipe obteve. Conheço uma equipe de polícia que

media o valor pelo número de prisões de criminosos procurados por semana, conheço igrejas que usam o Scrum e medem o sucesso avaliando o serviço que estão prestando à congregação e se ela está crescendo. O importante é decidir qual medida de valor será usada, e cobrar do Dono do Produto que ela seja entregue em maior quantidade. No Scrum, esse tipo de métrica é fácil de observar, por causa da incrível transparência do método.

Isso é muita coisa para pedir a uma pessoa, e é por isso que, em grandes projetos, costuma haver uma *equipe de Donos de Produto* para cuidar de todas as necessidades. Entrarei em mais detalhes sobre isso mais adiante no capítulo. Mas primeiro, como uma maneira de visualizar o que o Dono do Produto precisa fazer, quero que você se imagine na cabine de um F-86 Sabre com o "Mad Major", John Boyd, e prestes a entrar em um combate aéreo sobre a Península da Coreia.

Observe, Oriente, Decida, Aja

O combate aéreo durante a Guerra da Coreia aconteceu principalmente entre os F-86 Sabres americanos e os MiG-15 fabricados pelos Russos. Os MiG eram mais rápidos, mas maneáveis, tinham razão, impulso e peso maiores, e eram uma aeronave superior em tudo. No papel, os MiG-15 deveriam ter varrido dos céus todos os pilotos americanos. Em vez disso, eles foram derrubados a uma taxa de 10 por 1. A luta para descobrir como isso foi possível moldou o futuro da arte da guerra e tornou-se um elemento crítico no desenvolvimento do Scrum.

John Boyd é simplesmente o melhor piloto de combate que já viveu, embora ele nunca tenha derrubado um inimigo em combates. Ele voou em apenas 22 missões na Coreia antes do armistício e, naquela época, era necessário ter trinta missões como piloto antes que se pudesse assumir a liderança como "atirador". Foi depois da guerra, dando aulas na USAF Weapon School, na Base Aérea de Nellis, no sul de Nevada, que ele deixou a sua marca. Na vida militar, que valoriza revezamentos frequentes de pessoal, ele passou seis anos lá como instrutor; algo sem precedentes.

Pilotos de combate não são humildes. Quando eles chegam a Nellis, já são considerados os melhores pilotos na USAF, e demonstram certa arrogância. Boyd tinha um jeito perfeitamente seguro de desmantelar o ego de um piloto para que este realmente conseguisse aprender o que ele tinha a ensinar. Ele os desafiava para um combate aéreo e dava a vantagem aos alunos de voarem na

posição de “seis horas”, ou seja, diretamente atrás dele — a melhor posição para se estar em um combate aéreo. Então, ele diria para o aluno iniciar o ataque. Sem nunca falhar, em quarenta segundos, ele teria se alinhado na posição “seis horas” do aluno para um tiro certeiro na retaguarda da aeronave do aluno, enquanto gritava, o tempo todo, pelo rádio: “Tiro! Tiro! Tiro!”. Isso foi antes do advento dos *lasers* e computadores que simulavam tiros. Era o grito dele que avisava ao aluno que ele estava morto. O sucesso infalível de Boyd lhe deu um segundo apelido que o acompanharia para sempre: Boyd Quarenta Segundos.

Sua outra alcunha era Mad Major, um apelido que conquistou por causa de suas declarações, digamos, enérgicas. Elas eram quase sempre feitas a uns sete centímetros do rosto de quem quer que o contradissesse, e ele fazia isso enquanto usava dois dedos para cutucar o oponente. Certamente, entre aqueles dois dedos, haveria um cigarro aceso. Reza a lenda que, de vez em quando — por acidente, tenho certeza —, ele ateava fogo na gravata do adversário. Nesses momentos, ele não demonstrava o mínimo arrependimento, usando qualquer arma no seu arsenal para vencer uma discussão.

Boyd tinha a capacidade de ver todo o campo de batalha. Nas suas palavras:

> Eu me veria em uma bola vasta — eu estaria *dentro* da bola — e conseguia visualizar toda a ação acontecendo ao redor da bola [enquanto], o tempo todo, é claro, eu estava pilotando... Eu conseguia visualizar dois pontos de referência. Quando eu estava em combate no ar, *eu conseguia me ver como se eu fosse um observador separado olhando para mim mesmo e para os outros à minha volta*.[43]

Esse tipo de consciência, a capacidade de ver a esfera completa do céu e observar os eventos se desdobrarem, formou suas teorias militares e acabou reescrevendo o modo como os Estados Unidos entraram na guerra.

Quando Boyd deixou a Weapons School, decidiu estudar engenharia e, ao fazer isso, criou um modelo de desempenho de aeronave que descrevia o combate aéreo em termos de relações de energia. A teoria de Energia-Manobrabilidade (*Energy Maneuverability* — EM) leva em consideração as energias cinética e potencial em qualquer situação, sua altitude, velocidade do ar, direção e a rapidez com que consegue mudar qualquer uma dessas variáveis. A teoria acabou sendo inserida no modo como a maior parte dos aviões de combate é modelada, levando diretamente ao desenvolvimento do F-15 e F-16, os aviões de combate mais dominantes nos últimos quarenta anos.

O problema era que, de acordo com a teoria de Boyd, os MiG-15 deveriam

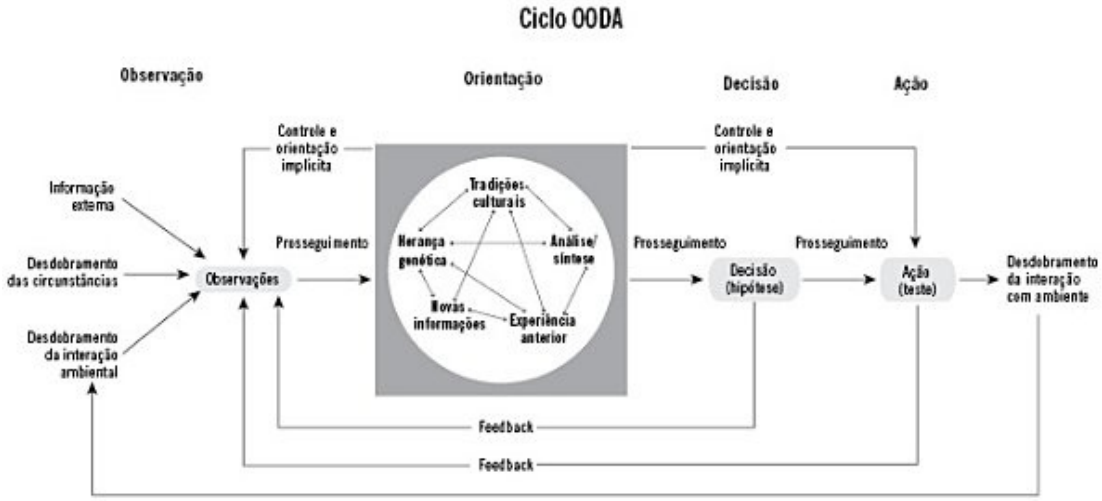
ter derrotado completamente os F-86 Sabre. Aquilo simplesmente não fazia sentido. Segundo a biografia de Robert Coran, Boyd entrava em transes frequentes, por dias, enquanto tentava entender aquilo. Ele tinha certeza de que sua teoria estava certa, mas qual teria sido o motivo para uma taxa de 10 por 1 a favor dos pilotos americanos? Treinamento? Isso poderia explicar apenas em parte. Tática? Talvez, mas, novamente, esse fator não levaria a um resultado tão desequilibrado. Então, ele encontrou a resposta. Os pilotos americanos conseguiam ver melhor e agir mais rápido. Não devido a qualquer qualidade inerente, mas por meio de simples escolhas de design. O Sabre tinha uma carlinga em bolha, ao passo que a do MiG era formada por múltiplos painéis de vidro e suportes que bloqueavam a visão do piloto. O F-86 também contava com controles de voo completamente hidráulicos, ao passo que o MiG tinha apenas controles *auxiliados* hidraulicamente. Os pilotos do MiG-15 eram conhecidos por levantar peso para lhes dar uma força superior de corpo para manobrar a aeronave.

Como resultado, os pilotos americanos conseguiam ver os MiGs primeiro e, então, crucialmente, podiam agir sobre tal informação mais rápido do que os pilotos chineses e norte-coreanos. A batalha não foi vencida pelo que a máquina podia fazer, mas sim pela rapidez com que a observação se traduzia em ação. O MiG fazia uma ação, o americano responderia e, enquanto o piloto do MiG ainda estava tentando responder àquela resposta, o americano poderia realizar outra ação. Ele respondia a cada novo alinhamento do MiG com uma rapidez tão superior que o avião tecnologicamente mais avançado se tornou um alvo fácil.

O mesmo fenômeno aconteceu no Vietnã quando eu estava lá. Naquela época, havia aeronaves diferentes, o MiG-21 e o F-4. No entanto, uma vez mais, a visibilidade superior do F-4 superou a maior capacidade de manobras do avião fabricado pelos soviéticos. Conforme Boyd diria, a sua inovação mais famosa coloca os pilotos “dentro do ciclo de decisão do inimigo”.

Esse *insight* se tornou fundamental na forma como as guerras são travadas. E foi exatamente assim que eu desenvolvi o Scrum — para permitir que o Dono do Produto tome decisões rápidas, com base em feedback de tempo real. Ao obter informações constantes de quem quer que esteja obtendo valor do que você está fazendo — seja o cliente clicando no botão de compras na Amazon, os fiéis de uma igreja, as crianças na sala de aula ou alguém experimentando um vestido —, você estará em uma posição de fazer ajustes constantes na sua estratégia e tornar-se bem-sucedido de forma mais rápida.

A ideia recebe aquele nome um tanto ridículo de ciclo OODA — acrônimo de Observe, Oriente, Decida, Aja. E, embora soe um pouco engraçado, é mortal na guerra e nos negócios. Entrar no ciclo de decisão de alguém provoca confusão e dúvida, com a reação exagerada e mínima ao mesmo tempo. Nas palavras de Boyd, em um relatório que entregou a outros comandantes: “aquele que conseguir lidar mais rápido com a taxa de mudança, vence”.[44]



“Observar” parece bastante óbvio — trata-se de enxergar claramente a situação à medida que ela se desdobra; mas não é tão simples quanto parece. Boyd descreveu isso como sair de dentro de si para conseguir ver a situação como um todo — e não apenas a partir de seu próprio ponto de vista.

“Orientar” não é somente a respeito de onde você está; também é sobre quais desfechos você é capaz de ver — o menu de alternativas que você cria para si mesmo. Segundo Boyd, os elementos dessa criação são: herança genética, tradições culturais e, é claro, as circunstâncias que estão se desdobrando no momento. Assim, a “orientação” reflete não apenas a forma com você vê o mundo e o lugar que ocupa nele, mas o mundo que você é *capaz* de ver.

A combinação da “Observação” com a “Orientação” leva a uma “Decisão”, que, por sua vez, leva a uma “Ação”. Então, o ciclo recomeça com a “Observação” dos resultados das suas ações e as de seu oponente — ou, no mundo empresarial: a observação da reação do mercado.

Ao entregar um incremento funcional, o que o Scrum faz é dar ao Dono do Produto a capacidade de ver quanto valor tal incremento cria e como as pessoas reagem a ele. Então, com base em tal informação, ele pode mudar o que a equipe vai fazer no próximo Sprint. Isso forma um ciclo constante de feedback que acelera a inovação e a adaptação, além de fazer com que o Dono do

Produto seja capaz de medir quanto valor foi entregue (nos negócios, essa medida é feita em dinheiro. Se eu estou pintando o interior de uma casa, talvez possa medir isso pela quantidade de aposentos concluídos). Assim, o Dono do Produto tem a capacidade de se ajustar, em tempo real, a um mundo em constante mutação.

Pode ser difícil imaginar a liberação incremental de produtos ou projetos que não parecem ter qualquer valor a menos que estejam concluídos. Por exemplo, como você pode fazer uma liberação incremental de um carro? Ou de um jogo de *video game* de US$ 100 milhões? O importante é procurar as partes que realmente detêm valor — valor suficiente para que você consiga um feedback verdadeiro e possa reagir em tempo real.

Vamos pegar carros como exemplo. A Toyota desenvolveu o Prius, do conceito à entrega, em 15 meses — mais rápido do que qualquer outra entrega que já tivessem feito antes. Embora a equipe que o projetou e construiu não tenha começado a vendê-lo antes que estivesse pronto, eles começaram rapidamente a construir protótipos para que o Engenheiro Chefe pudesse fazer testes a fim de testar se a equipe estava no caminho certo. Esse tipo de protótipo rápido, com veículos totalmente funcionais antes do lançamento, que depois são aprimorados repetidas vezes até que se tenha o produto que se deseja vender, pode impulsionar alterações incrivelmente rápidas. O importante não é ter um projeto totalmente estabelecido no início, mas construir um protótipo funcional e, depois, ver o que é possível aprimorar. E, então, depois de fazer tal aprimoramento, construir o protótipo seguinte e aprimorá-lo novamente. A ideia é que o quanto antes você tenha um feedback real, mais rápido você será capaz de construir um produto melhor.

A WIKISPEED, sobre a qual escrevi no Capítulo 4, constrói protótipos completos dos seus carros toda *semana;* e eles os vendem. Essas transações não ocorrem no mercado de venda de massa — a equipe da WIKISPEED não está pronta para isso ainda — mas *existem* compradores iniciais que estão dispostos a investir US$ 25 mil para adquirir um desses protótipos preliminares. Quando você pensar em construir algo, não presuma que não vai conseguir entregar algo de valor até o final do projeto. Em vez disso, tente pensar sobre o produto minimamente viável. *Qual é o mínimo absoluto que posso construir e ainda entregar como valor a um cliente?*

Os jogos de *video game* são outro bom exemplo. Hoje um número cada vez maior de desenvolvedores permite que jogadores iniciais paguem para ter um acesso “alfa” adiantado. Desse modo, eles conseguem obter feedback dos fãs

mais dedicados antes que o jogo realmente funcione. Isso permite que observem como as pessoas realmente reagem, em vez de tentar adivinhar essa reação.

Dependendo do setor em que você trabalha, ou organização que dirige, pode ser difícil compreender essa ideia de liberações incrementais. Uma alternativa, se você não conseguir colocar algo diante de um cliente *externo*, é identificar um cliente *interno* — por exemplo, o Dono do Produto —, que possa substituir público. Mostre para o seu cliente interno o que quer que seja que possa evocar um feedback útil: pequenas partes de um plano de expansão em empreendimentos imobiliários, um projeto de modernização de uma fábrica, a reconstrução de um sistema de frenagem, uma campanha de trabalho voluntário — o que for. A ideia é criar para você mesmo uma oportunidade de inspecionar e adaptar. Uma empresa ou organização que não consiga reagir às condições variáveis, à concorrência ou aos gostos, ficará em sérios apuros.

Boyd explica desta forma:

> Queremos entrar no ritmo ou no tempo do outro cara, onde poderemos derrubá-lo... Temos de ter uma imagem ou fotografia na nossa mente, que possamos chamar de orientação. Então, temos de tomar uma decisão em relação ao que vamos fazer e, então, implementar tal decisão... Então, olhamos para a ação [resultante], mais a nossa observação, e nos envolvemos com os novos dados, a nova orientação, a nova decisão, a nova ação, *ad infinitum*... A orientação não é apenas um estado no qual você se encontra; é um processo. Você está sempre se orientando...
>
> Um pequeno mundo agradável onde não existem mudanças... [criaturas que vivem nesse mundo são] dinossauros; eles vão morrer. A chave do negócio é não se tornar um dinossauro. Se você se encontra em uma condição de equilíbrio, você está morto... A mensagem subjacente é simples: não há escapatória... É assim que as coisas são, rapazes.[45]

É assim que as coisas são. Conforme eu disse no Capítulo 1, você tem uma decisão bastante difícil para tomar: mude ou morra. Se você não entrar no ciclo de decisão do seu concorrente, ele vai entrar no seu. Conforme disse Boyd: “O que eu quero é prender o meu adversário dentro dele mesmo... Então, eu poderei confundi-lo, desorganizá-lo e deixá-lo paralisado”. Eu não sei quanto a você, mas eu prefiro muito mais estar fazendo do que recebendo isso.

Primeiro, as coisas mais importantes

Então, você tem um Dono de Produto que atualiza constantemente as

Pendências, determinando a sequência em que as coisas serão produzidas e entregues. Quando você tem algumas centenas de itens, esse processo de sequenciamento rapidamente se torna algo complexo. O importante é descobrir como entregar mais valor mais rapidamente. Talvez existam milhões de maneiras de fazer uma sequência das Pendências, mas a que você deseja é aquela que entrega 20% das características que possuem 80% do valor o mais rápido possível. É quase certo que seu primeiro palpite para o primeiro Sprint não seja o certo, mas será o seu melhor palpite naquele momento.

Mas aquele foi apenas o seu *primeiro* palpite. Depois do primeiro Sprint, uma vez que você concluiu o ciclo OODA e entregou alguma parte do produto para os clientes, pode mudar a sequência, percebendo que, na verdade, determinada sequência é melhor que outra.

E, então, você continua fazendo isso, atualizando e mudando constantemente a prioridade das coisas nas Pendências a cada Sprint, seguindo na direção da sequência que entregará valor mais rapidamente. É provável que você nunca consiga encontrar a sequência absolutamente perfeita, mas deseja seguir nessa direção em todos os passos do caminho.

O importante é lembrar que a sequência sempre constitui um fluxo. A sequência certa em uma semana não será a mesma na semana seguinte; seu ambiente terá mudado. Você terá aprendido novas coisas; terá descoberto que algumas coisas são mais fáceis e outras, mas difíceis. Então, essa mudança constante no sequenciamento das Pendências acontece a cada Sprint. A chave é reconhecer a incerteza, aceitar completamente que a imagem atual da sequência e do valor só é relevante naquele instante específico. Tudo vai mudar de novo. E de novo. E de novo.

Um hábito ruim que uma empresa pode adquirir por causa das constantes mudanças das necessidades do mercado e porque os gestores não sabem exatamente onde o valor reside é tornar tudo prioridade. Tudo passa a ser *prioridade máxima*. O ditado que me vem à mente é o de Frederico II da Prússia, posteriormente conhecido como Frederico, o Grande: “Aquele que tudo defende, nada defende”. Ao não concentrar seus recursos tanto quanto sua energia mental exige, você acaba diluindo-os à irrelevância.

Alguns anos atrás, comemorei meu aniversário de setenta anos na Normandia, França. Fui para lá a fim de visitar a famosa praia onde meu pai desembarcara durante a invasão do Dia D. Na maré baixa, a praia de Omaha parece se estender em um declive de quilômetros até mergulhar no mar distante — uma extensão aparentemente infinita de areia. Subir tudo aquilo para

enfrentar as armas alemãs deve ter exigido uma coragem que só podemos imaginar; caminhar pelos túmulos dos milhares que morreram naquele dia exige silêncio e respeito. Mas, quando comecei a ler sobre as defesas dos alemães, percebi que um dos motivos de o desembarque dos americanos ter sido bem-sucedido foi porque os alemães se esqueceram da advertência de Frederico, o Grande. Eles ficaram tão confusos com as ações dos aliados que espalharam suas forças ao longo de toda a costa da França. Por isso, os aliados conseguiram isolar as unidades alemãs separadamente, derrotando cada uma sucessivamente. Os alemães não definiram prioridades de forma adequada, e, felizmente, perderam tudo.

Liberação

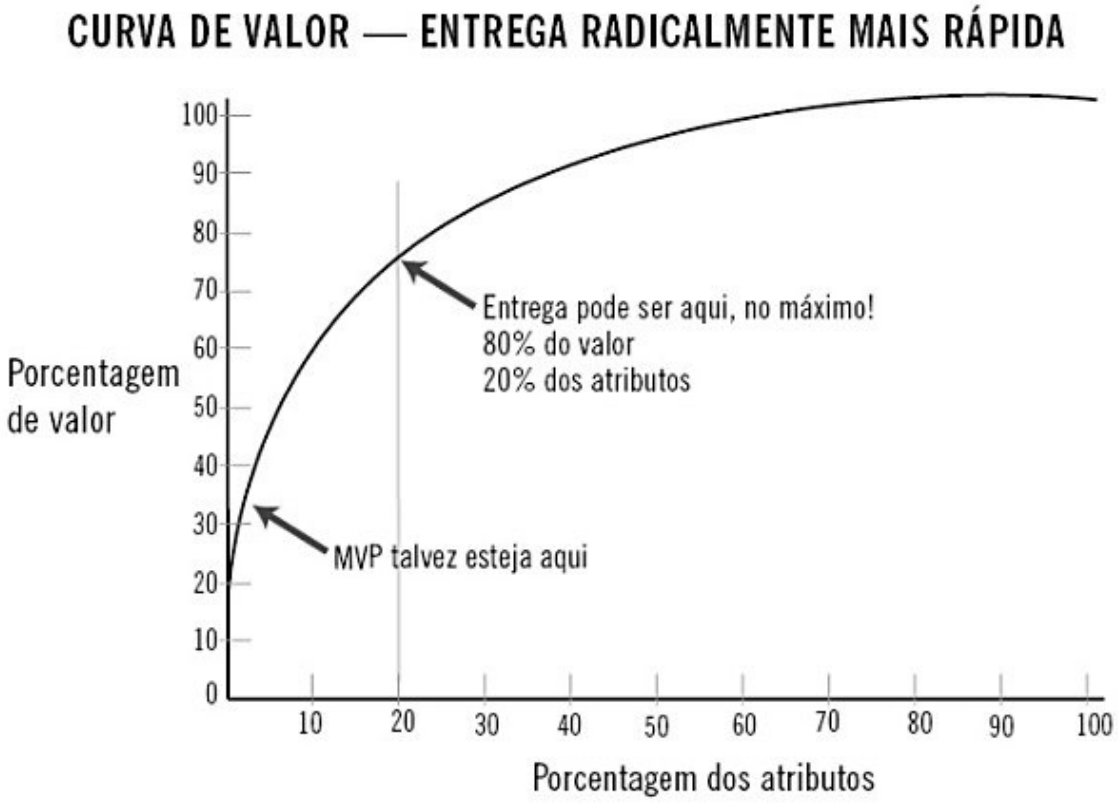
Então, você tem as prioridades. Sabe onde os 80% de valor estão. Mas, quando você entrega o seu produto? É aqui que o Scrum pode agregar valor extremamente rápido. Sempre que estiver fazendo algo, você quer colocar seu produto nas mãos de quem vai realmente usá-lo o mais rápido possível. Você quer fazer isso mesmo antes de ter 20% das funcionalidades. Você quer fazer isso com algo que entregue, pelo menos, *um pouquinho* do valor. Chamo isso de Produto Mínimo Viável (*Minimum Viable Product* — MVP). Isso poderia ser o que você mostra para o público pela primeira vez. Qual é o nível de eficácia que o produto deve ter? Bem, ele deve realmente funcionar, embora para uma pessoa que esteve trabalhando nele isso pode parecer meio desconcertante. Você precisa levar o produto até o público *o quanto antes possível!* Isso lhe dará o feedback de que precisa para energizar seu ciclo de decisão e priorização. Esta é a versão 0.5; é uma câmera que pode tirar uma foto, mas não encontra o foco; é uma mesa de jantar com apenas duas cadeiras; uma distribuição de uma vacina para cinco vilas entre as cem que você está tentando ajudar. Chega a ser engraçado de tão incompleto que está.

Mas é aí que você consegue o seu *feedback*. É bem estranho segurar essa máquina porque o botão do obturador está em um lugar esquisito; se você reparar com atenção, verá que a madeira das cadeiras não combina com as mesas; você também ofendeu as pessoas mais idosas das vilas por causa de uma gafe social totalmente evitável. Cometa esses erros nos estágios iniciais, causando o menor prejuízo possível.

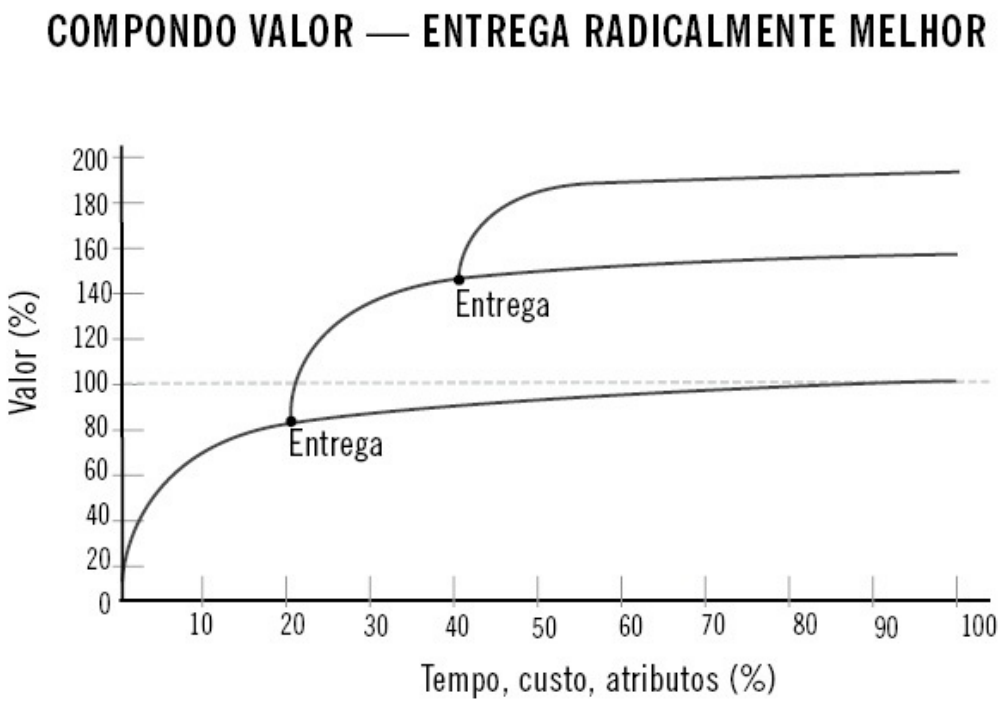
Então, quando você fizer o lançamento oficial, ou uma apresentação pública

de um grande programa, já terá feito ajustes e descoberto o que as pessoas realmente valorizam. No exemplo da nossa câmera, talvez os usuários *tenham dito* que ter um modo paisagem e poder compartilhar as fotos no Facebook era igualmente importante em termos de valor, mas quando realmente começaram a *usá-la*, nunca aplicaram o modo paisagem, mas sempre queriam compartilhar as fotos na rede social.

Isso permite que você faça primeiro as funcionalidades que eles valorizam, e lance o produto quando tiver completado apenas 20% do trabalho. Você sabe que não está perfeito, mas está bem próximo. A cada hora que você passa polindo a maçã, perde uma oportunidade para obter valor.



O que é ótimo em relação a esse processo é que ele é interativo: apenas "enxágue e repita". Quando as pessoas tiverem o seu produto ou serviço ou alteração em suas vidas, elas dirão a você o que valorizarão *em seguida*. Então, você desenvolve 20% disso e faz uma nova entrega de novo. E de novo.



Com o processo de liberação por incrementos, no tempo que você levaria para criar metade dos atributos do seu produto ou projeto inicial, você liberou 200% do valor, em metade do tempo. Esse é o verdadeiro poder do Scrum. É assim que você pode mudar fundamentalmente não apenas a forma como trabalha, mas a forma como vive a sua vida. Não se concentre em entregar uma lista completa de coisas — tudo o que precisar e mais um pouco —; concentre-se, sim, em entregar o que tem mais valor: o que realmente as pessoas querem ou precisam.

Isso me faz lembrar de histórias do Iraque e do Afeganistão. Acontecia assim: um pelotão americano chegava a uma cidade, olhava em volta e dizia: “Essas pessoas estão criando galinhas. Vamos construir para elas uma fábrica de processamento de galinhas com o que há de mais moderno”. Então, eles gastavam milhões de dólares para construir tal fábrica. Eles não levavam em consideração o fato de não haver uma rede elétrica estável ou de a grande maioria do povo da cidade ser analfabeto, e que não seria fácil treiná-los para usar os equipamentos. Então, alguém chegava à cidade e perguntava: “O que *realmente* ajudaria vocês?”, e eles respondiam: “Sabe? Uma passarela sobre o rio seria ótimo, pois assim não teríamos de gastar metade do dia para chegarmos até a travessia mais próxima para irmos ao mercado”. Aquela passarela custaria algumas centenas de dólares. Isso parece bem menos impressionante do que uma fábrica enorme. Não soa tão dramático quando

você fala sobre isso com seus chefes em Washington. Mas, para o povo daquela cidade, tem um valor infinitamente maior do que um prédio elegante com um monte de máquinas que vão acabar enferrujando.

Outro ponto a lembrar é que, às vezes, você termina mais cedo. Digamos que você está construindo um relógio despertador de última geração para a fábrica Despertadores Ltda. Você tem uma lista com dezenas de atributos: um relógio, um botão "soneca", um cronômetro, um alarme alto, um rádio, um suporte para iPhone, um GPS — e mais qualquer outra coisa. Mas, sendo um Dono de Produto experiente, você prioriza o que as pessoas realmente querem: um alarme fácil de programar, com um bom volume, um rádio e um mostrador brilhante o suficiente para ser visto mesmo quando o quarto estiver escuro ou claro demais. E quando a sua equipe concluiu tudo, você se dá conta de que eles, na verdade, criaram o relógio despertador mais elegante que já existiu. Trata-se do iPod da Apple dos despertadores. É lindo e faz apenas uma coisa, mas muito, muito bem feita. Em vez de pedir para a equipe acrescentar atributos, você lança o relógio e começa a trabalhar no próximo projeto. A equipe conseguirá mais valor ao trabalhar em outro projeto.

Pagamento por nada e alterações gratuitas

No início deste livro, eu contei a história do projeto Sentinel, do FBI. Se você se lembrar, um fornecedor externo gastou centenas de milhões de dólares construindo um software que não funcionava. Uma das maiores causas de se ficar acima do orçamento previsto como naquele caso — e na maioria de todos os contratos, seja para construir uma aplicação de software, carros ou prédios — são as taxas de alteração. Cobrar taxas excessivas de alteração constitui, na verdade, o modelo de negócios de vários fornecedores que prestam serviços para o governo. Eles oferecem um preço menor para o projeto, sabendo que terão lucro por causa dos pedidos de alteração. Quando um contrato para um projeto com anos de duração é redigido, todos os requisitos são definidos naqueles diagramas bonitinhos; assim, é tentador dizer: "bem, isso cobre tudo". Então, os fornecedores afirmam: "eu concordo em fazer isto, e apenas isto. *Se você deseja quaisquer alterações, isso terá um custo adicional*". Esse faturamento depois do fato se tornou o centro de tantos custos, que as empresas e agências começaram a organizar Conselhos para o Controle de Alterações. A partir de um ponto de vista de custos, isso faz sentido. Limite o número de

alterações, e você limitará o custo associado a eles.

O que os contadores de feijão não percebem é que estão formando um sistema projetado para não dar às pessoas o que elas realmente querem. Em lugar disso, estão tentando limitar os custos, mas, ao fazê-lo, limitam a aprendizagem, a inovação e a criatividade. Se você começa um projeto e pouco depois percebe que o valor verdadeiro, aqueles 20%, não está nos atributos que você definiu — está em um conjunto diferente que você descobriu no decorrer do trabalho —, então o gerenciamento de projetos tradicional foi projetado para impedir isso. Ele foi projetado para impedir a entrega de valor rápido demais.

Além disso, o esforço para "exercer um controle firme" nem sempre funciona! Mesmo com os Conselhos de Controle de Alterações tentando limitar as mudanças, a necessidade para isso é tão grande que, em geral, eles são voto vencido. Sem as alterações, não haveria qualquer valor no projeto. Assim, relutantemente, o conselho permite a alteração, e os custos do projeto sobram. E, então, há outra alteração que precisa ser feita. E, opa, mais outra. E, rapidamente, o projeto está milhões de dólares acima do orçamento, e um ou dois, ou cinco anos acima do prazo.

Foi por isso que eu criei o conceito de "Alterações Gratuitas". Em um contrato-padrão com preço fixo, digamos que as alterações sejam feitas de graça. Liste todas as funcionalidades que você espera; por exemplo, você está construindo um tanque, e quer que ele atinja uma velocidade de 120 km/h e atire dez salvas de tiro por minuto, acomode quatro pessoas, tenha ar-condicionado etc. — tudo o que consiga pensar que precise para tal tanque. O construtor olha para a descrição e diz: "hum, acho que a construção desse motor são 100 pontos; o mecanismo de carregamento, 50; os assentos, 5; e assim por diante em todos os itens da lista. Ao final, existe determinado número de pontos para cada atributo. Então, a cada Sprint, o cliente, que, neste cenário está obrigado, contratualmente, a trabalhar junto com o Dono do Produto, pode mudar completamente as prioridades. Qualquer item ou atributo das Pendências pode ser colocado em qualquer outro lugar. E quanto a novos atributos? Sem problemas: basta retirar atributos de tamanho equivalente do produto. Ah, você quer agora um sistema guiado por *laser*? Bem, isso são 50 pontos de trabalho — para compensar esse acréscimo, vamos retirar 50 pontos de atributos de baixa prioridade do final das Pendências.

Poucas empresas levaram essa ideia a um novo nível entregando a um cliente apenas os atributos de alto valor. Alguns anos atrás, ouvi uma história sobre um desenvolvedor Scrum que conseguiu um contrato de US$ 10 milhões para

entregar um software para uma grande construtora. As duas partes concordaram com um prazo de vinte meses, mas a empresa Scrum inseriu uma cláusula no contrato: se a construtora quisesse terminar o contrato a qualquer momento, eles teriam apenas de pagar 20% do valor remanescente do contrato. Basicamente, se o software funcionasse da maneira que a construtora desejava, eles poderiam fazer com que a empresa Scrum continuasse desenvolvendo mais funcionalidades.

O desenvolvedor de software começou seus Sprints; eles trabalhavam com Sprints de um mês. Depois do primeiro mês, o cliente redirecionou alguns dos esforços do desenvolvedor para conseguir mais valor. No segundo mês, aconteceu o mesmo. Depois do terceiro, o cliente terminou o contrato, pegou o software e o colocou para funcionar. Eles tinham o valor de que precisavam.

Vamos fazer algumas contas aqui para ver como todos saíram ganhando. Nos primeiros três meses de contrato, o cliente pagou US$ 1,5 milhão para a empresa Scrum. Encerrar o contrato antes do tempo exigia que eles pagassem 20% dos US$ 8,5 milhões restantes — ou seja, US$ 1,7 milhão. Eles pagaram US$ 3,2 milhões por um software que acreditavam que fosse custar US$ 10 milhões, e conseguiram isso 17 meses mais cedo do que esperavam.

E não foram só eles que saíram ganhando. A empresa Scrum entrou na concorrência trabalhando com uma margem de lucro esperada de 15%. Então, gastaram US$ 1,3 milhão no desenvolvimento naqueles três primeiros meses. Mas receberam US$ 3,2 milhões. A margem de lucros deles passou de 15% para 60%. Isso representa um aumento de 400% nos ganhos. E agora, com seus desenvolvedores disponíveis, eles poderiam entrar na concorrência de outros projetos. Não se trata apenas de um bom negócio, mas de uma estratégia para a antecipação da aposentadoria.

Eles conseguiram fazer isso por causa dos blocos de construção do Scrum. Ao se gerenciarem como uma unidade interfuncional, a equipe conseguiu se acelerar rapidamente, entregando valor mais rápido. Ao final de cada Sprint, um incremento do produto estava *Feito*. Funcionava. Poderia ser distribuído imediatamente. A cada Sprint, o Dono do Produto pôde reorganizar as prioridades das Pendências com base no feedback do cliente; e, quando criaram valor suficiente, todos pararam de trabalhar. Desse modo, o Scrum alinha os interesses de todos: os membros da equipe, o Mestre Scrum, o Dono do Produto, o cliente e a empresa. Todos trabalham em direção ao mesmo objetivo e com a mesma visão: *entregar valor de verdade o mais rápido possível*. Acredito fortemente em situações ganha-ganha; e ganhar mais dinheiro e

entregar produtos melhores com preços mais baixos me parece ser um ótimo negócio.

Risco

O gerenciamento do risco está no centro de qualquer empreendimento bem-sucedido. O Scrum permite que você reduza os riscos de fracasso. Os três tipos mais comuns de risco são o risco de mercado, o risco técnico e o risco financeiro. Ou, em outras palavras: as pessoas querem o que estou construindo? Elas podem realmente construir o que quero? Nós realmente conseguiremos vender o que produzimos?

Já escrevi muito sobre risco de mercado. O Scrum ajuda você a minimizá-lo ao enfatizar a entrega em incrementos. Isso permite que você coloque um produto na frente dos clientes de forma mais rápida. E, ao obter um feedback antecipado de maneira frequente, você pode fazer pequenas alterações na hora, em vez de ser forçado a fazer grandes alterações depois que já tiver investido milhões de dólares e perceber que o que você está construindo não é realmente o que o cliente deseja. Em geral, é o que o cliente *disse* que queria no início do processo, mas, na verdade, as pessoas não sabem o que querem até que possam testar algo. Muitos trabalhos de consultoria giram em torno de falhar rapidamente. Eu prefiro pensar em entregar rápido.

O risco técnico é interessante. A questão de se, de fato, é possível construir o que o cliente quer é bastante complexa, principalmente se você está fazendo algo físico, que exige plantas, ferramentas e um investimento adiantado.

Você se lembra da empresa com o sistema de automação do lar? Bem, eles adotaram uma abordagem chamada “engenharia simultânea baseada em conjuntos”. Isso significa “construiu alguns protótipos diferentes para ver qual funciona melhor antes de começar a produção plena”. Por exemplo, eles sabiam que precisavam de uma câmera para que os clientes pudessem ver quem estava batendo na porta antes de deixar o visitante entrar. A parte mais cara da câmera, e que exigia mais tempo de espera, eram as lentes. Elas deviam ser de plástico? Vidro? Cristal? O que apresenta maior durabilidade em qualquer condição climática? Qual exibe a melhor imagem? Qual é mais fácil de se arranhar? Qual existe a imagem mais nítida? Quanto cada tipo custa para ser fabricado?

Em vez de tomar a decisão antecipadamente e começar a produção, eles

construíram três lentes diferentes e totalmente funcionais e as compararam. Já que estavam apenas tentando tomar uma decisão quanto às lentes, e precisavam fazer isso primeiro por causa do tempo de espera na fabricação, eles testaram cada uma das lentes usando uma armação de câmera de um notebook. O resultado foi que o vidro atendeu melhor aos critérios. No entanto, eles só foram capazes de fazer tal julgamento depois de *verem* algo que realmente funcionava. A decisão não foi tomada com base em construtos teóricos; eles tinham algo que podiam olhar e tocar. Com essa questão resolvida, puderam iniciar o projeto da estrutura na qual as lentes entrariam e os processadores que controlariam as imagens. Ao priorizar a decisão do tipo de lente, a empresa economizou potencialmente milhões de dólares. A Apple é famosa por fazer isso com todos os seus produtos; ela constrói diversos protótipos completamente funcionais antes de organizar um teste de campo para ver qual é o melhor. Isso permite que diferentes ideias sejam expressas de forma rápida sem um investimento volumoso.

O risco financeiro é o responsável pelo fracasso da maioria das empresas. Eles construíram algo interessante, mas não conseguem vendê-lo por um valor suficiente para ter lucro. Um exemplo clássico disso é o jornalismo on-line e a morte do jornal impresso. Quando a Web estourou pela primeira vez, nos anos 1990, os jornais estavam ávidos para levar seu conteúdo para a internet. Alguns diretores de jornais acharam que, fosse off-line ou on-line, as pessoas pagariam para anunciar seus produtos, então eles tornaram o conteúdo gratuito. O problema, é claro, foi que os anunciantes estavam dispostos a pagar bem menos por anúncios on-line do que por anúncios impressos. Ainda assim, o custo de produzir conteúdo continuou o mesmo. Outros tentaram cobrar pelo conteúdo, mas havia tantos sites fornecendo notícias gratuitas que eles acabavam sendo obrigados a seguir a tendência. Mandar repórteres para lugares para que pudessem testemunhar histórias é caro. Você pode ver os resultados no encerramento das atividades de várias redações.

Essa ideia de fornecer conteúdo ou um serviço de graça e, então, ganhar dinheiro com a propaganda, ainda é prevalente em start-ups de tecnologia até hoje. Empreendedores olham para o Facebook ou para o Google e dizem “eu posso fazer isso”. O problema é que não existem tantos Facebooks e Googles. Nos primórdios da Internet, quando o espaço on-line permitiu pela primeira vez que as empresas mirassem em segmentos específicos de cliente, a “hiperconcentração” era considerada algo valioso. No entanto, à medida que mais plataformas surgiram para facilitar isso, tal capacidade passou a não ter

mais valor comercial.

Outra forma de uma empresa ruir em termos financeiros é pagar demais para conquistar clientes. Um exemplo disso são as empresas de compra em grupo, como Groupon e Living Social. No início, elas conquistaram clientes de forma rápida e fácil. Contudo, à medida que expandiram seu alcance e aumentaram o quadro de funcionários, ficou cada vez mais caro atrair anunciantes adicionais e mais pessoas que quisessem comprar cupons. Você pode ver os resultados disso nas avaliações dessas empresas.

O que o Scrum faz para os negócios é responder a esta pergunta-chave de forma rápida: nós vamos conseguir ganhar dinheiro fazendo isso? Ao fazer liberações por incrementos para os clientes de forma rápida, você descobrirá o que os seus clientes valorizam e o que estão dispostos a pagar. E se suas estimativas iniciais estiverem erradas, você pode fazer alterações. O máximo que você pode perder é o custo em tempo e a energia que já investiu em alguns Sprints — em oposição ao custo multimilionário de construir uma infraestrutura complexa apenas para descobrir que, embora as pessoas amem o seu produto, elas não o amam o suficiente para parar e pensar o que custa para produzi-lo.

Aqui está o que você vai fazer amanhã

Muito bem, o que você vai fazer amanhã para implementar o Scrum onde você trabalha? O primeiro passo é apenas organizar uma lista de Pendências e uma equipe. Pense na visão que você tem para o seu produto ou serviço ou seja qual for o seu projeto, e comece a subdividir as coisas que precisa fazer para realizar a sua visão. Você não precisa de muita coisa, apenas de uma semana de Pendências. E, enquanto os membros da equipe estão fazendo as Reuniões Diárias e executando o primeiro Sprint, você será capaz de organizar uma lista de Pendências para manter a equipe ocupada pelos dois Sprints seguintes. Mantenha sempre a atenção nas Pendências, porque, à medida que as equipes ficam mais rápidas, elas começam a entregar mais do que você acreditava ser possível.

Então, como Dono no Produto, faça um mapa de para onde você acha que as coisas estão se encaminhando. O que você acha que consegue fazer neste trimestre? Onde você quer estar neste ano? É importante se lembrar de que isso é apenas uma projeção no tempo, então não planeje demais, apenas faça

estimativas. Você não está criando um contrato de entregas obrigatórias, está apenas delineando os seus pensamentos sobre onde você estará por um tempo. Pode acreditar: essa imagem *vai* mudar. Talvez radicalmente.

O motivo para fazer esse tipo de planejamento é criar uma transparência na organização. Se você tem uma equipe de vendas, o pessoal precisa conhecer os atributos nos quais você está trabalhando para que possa começar a fazer o marketing do produto. E a liderança precisa saber de onde a receita virá — além de quando virá e de quanto será. A mensagem mais importante é que tudo que está sendo feito é de conhecimento de todos. Qualquer um pode ver em que fase o produto está a qualquer momento. Eles podem ver as histórias avançando no quadro Scrum até serem concluídas; podem fazer um gráfico de pontos das histórias em relação ao tempo e observar uma linha estável se mover em direção ao zero; podem saber quantos pontos fizeram no último Sprint e qual é a sua estimativa para o próximo. Saiba que, como Dono do Produto, você será avaliado em relação à receita e aos custos.

O que você vai descobrir bem rápido, principalmente se estiver trabalhando em um lugar com diversas equipes, é que precisará fazer uma equipe de Donos de Produto para que consiga gerar Pendências suficientes para as equipes trabalharem. Você pode ter um Dono de Produto focado mais na estratégia e na interação com o cliente, e outro que seja mais tático, decidindo no que as equipes trabalharão em cada Sprint.

O mais importante, porém, é começar. Apenas comece. O Scrum foi projetado de forma que você possa formar uma equipe em dois dias. Faça uma lista de Pendências, planeje o seu primeiro Sprint e, assim, siga adiante. Você não precisa dedicar muito tempo em planejamento, reflexões, meditações, declaração da missão, ou projeções para daqui a cinco anos. Deixe isso tudo a cargo da concorrência, e deixe-os comer a poeira que você vai deixar atrás de si. E, durante o percurso, por que você não torna o mundo um lugar melhor? No próximo capítulo, mostrarei como você pode fazer isso.

PONTOS PRINCIPAIS

Faça uma lista. Verifique-a duas vezes. Crie uma lista de tudo que é possível fazer em um projeto. Depois, classifique-a por ordem de prioridade. Coloque os itens com mais valor e menos risco no topo das Pendências, então o próximo item, e assim por diante.

Dono no Produto. É a pessoa que traduz a visão em itens da lista de Pendências. Ela precisa compreender o negócio, o mercado e o cliente.

Um líder não é um chefe. Um Dono de Produto define *o que* precisa ser feito e por quê. *A forma como*

a equipe vai fazer isso e *quem* vai fazer é ela mesma que decide.

O Dono do Produto. É a pessoa que detém o conhecimento do domínio e que tem a última palavra. Está disponível para responder às perguntas e é a responsável por entregar valor.

Observe, Oriente, Decida, Aja (OODA). Tenha uma visão completa do quadro estratégico, mas aja de forma rápida e tática.

Medo, incerteza e dúvida. É melhor causar esses sentimentos do que recebê-los. Entre no ciclo OODA do seu concorrente e faça com que ele se enrole na própria confusão.

Ganhe dinheiro por nada e faça alterações de graça. Crie coisas novas apenas quando elas geram valor. Esteja disposto a trocá-las por algo que exija um esforço equivalente. O que você pode ter acreditado ser necessário no início pode ser algo de que você, na verdade, nunca precisou.

42 John Shook, *The Remarkable Chief Engineer*, Lean Enterprise Institute, 3 de fevereiro de 2009.

43 Daniel Ford, *A Vision So Noble: John Boyd, the OODA Loop, and America's War on Terror,* CreateSpace Independent, 2010.

44 John Boyd, *New Conception*, 1976.

45 *Ibid*.

CAPÍTULO 9

Mude o mundo

O Scrum tem origem no mundo do desenvolvimento de software. Agora, está tomando uma grande quantidade de outros lugares onde o trabalho é feito. Diversos negócios estão usando esse método para tudo, desde a construção de foguetes espaciais, passando pelo gerenciamento da folha de pagamento, até a expansão de recursos humanos; e está aparecendo em todos os setores, desde o financeiro até o de investimentos, do entretenimento ao jornalismo. Fico maravilhado de saber que um processo que criei em 1993 para ajudar o desenvolvimento de software tenha provado ser universalmente aplicável. O Scrum acelera o esforço humano — não importa que esforço seja esse.

Na verdade, comecei a vê-lo se desenvolver em lugares bastante improváveis, lidando com os problemas mais espinhosos da humanidade. Pense em alguns desses problemas; por exemplo, viver na pobreza, que não é somente algo degradante, mas que também gera uma série de males sociais, desde o crime e a corrupção até a guerra e a destruição. E há o sistema de educação, o qual está fracassando com alunos no mundo inteiro. Em vez de ensinar habilidades do século 21, estamos mergulhando os jovens em métodos de ensino e aprendizagem criados no século 19. E o outro elemento confuso que vem à mente é o governo, que está amarrado de várias formas, baseando-se em ideias formadas centenas de anos atrás que não parecem mais se encaixar no modo como vivemos as nossas vidas.

É fácil erguer as mãos em derrota ao ouvir as últimas notícias sobre pessoas morrendo na África, a violência cometida em nossas escolas ou o exibicionismo infinito das pessoas no poder. Às vezes, isso parece ser algo pesado demais para lidarmos. No entanto, esses problemas, os problemas difíceis, são exatamente os projetados para o Scrum tratar. Em cada um desses casos, as pessoas agora estão se voltando para o Scrum para ajudar a resolvê-los e, assim como no mundo corporativo, estão atingindo um sucesso notável.

Educação

De certa forma, cidades-dormitório são iguais no mundo inteiro. Estabelecidas a alguns poucos quilômetros de distância de uma metrópole, é para lá que as pessoas se mudam para ter moradia mais barata, formar uma família e colocar os filhos em uma escola sem os muitos problemas enfrentados nas cidades grandes.

Alphen aan den Rijn é um exemplo bem típico nesse sentido. Está localizada na região oeste da Holanda, entre Leiden e Utrecht, talvez a 45 minutos de Amsterdã. Quando você se aproxima dela pela estrada em um dia útil, todo o trânsito está concentrado na outra direção — para empregos em outros lugares. Fazendas de gado leiteiro e moinhos antigos e novos cobrem a paisagem rural.

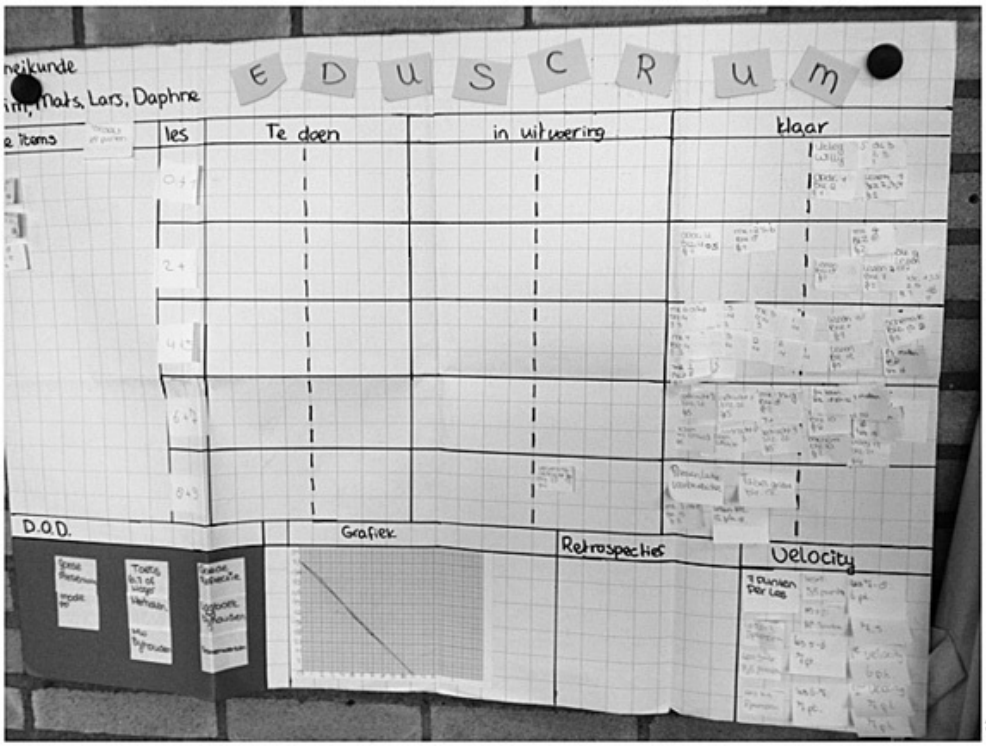
O trânsito é praticamente todo de bicicletas. Centenas e centenas delas se dirigem para a escola pública de Ensino Fundamental e Médio, a Ashram College, que tem cerca de 1.800 alunos com idades que variam de 12 a 18 anos. A Holanda acompanha seus estudantes desde cedo, dividindo-os da seguinte forma: programas vocacionais técnicos, cujo objetivo é formar profissionais que variam desde cabelereiro, passando por mecânicos até secretárias; programas vocacionais de ensino superior que os levam para carreiras como enfermagem, administração e engenharia; e programas universitários para aqueles que desejam estudar medicina, direito ou pesquisa. Os alunos nos grupos inferiores podem entrar no mercado de trabalho aos 16 anos, ao passo que aqueles dos grupos superiores podem ficar na universidade e na educação profissional até a faixa dos vinte anos. Cada um dos diferentes grupos precisa fazer algumas aulas em comum, embora estudem essas matérias em classes separadas. Ashram tem turmas dos três grupos; e uma dessas matérias essenciais é ensinada por Willy Wijnands para todas as turmas da escola: química.

Quem cursou o Ensino Médio nos Estados Unidos certamente se lembra das aulas de química na escola: mesas de laboratório em fileiras voltadas para o professor que ficava na frente da sala, talvez uma semana de matéria teórica, seguida por alguns dias trabalhando na solução de algum problema prático com um colega de laboratório, o qual foi escolhido de forma estratégica e meio estressante. Talvez você gostasse de química, ou considerasse a matéria extremamente entediante; talvez a série Breaking Bad tenha lhe dado novas perspectivas para o potencial de remuneração para uma boa técnica laboratorial

e a importância de escolher o parceiro certo. Seja qual for a sua experiência, uma vez que o professor começava a falar sobre laços covalentes ou qualquer outro conceito obscuro, era como se um *clique* quase audível acontecesse quando você e seus colegas de turma olhavam pela janela, rabiscavam algum desenho ou ficavam pensando naquele garoto ou garota atraente na segunda fileira. Vamos ser sinceros, na sala de aula de química dos Estados Unidos, sonhar acordado é o que mais acontece.

Contudo, é assim que acontece nas aulas de Wijnands: “Vejam bem”, começa ele, enquanto os alunos entram na sala e se apressam para ocupar suas carteiras — estranhamente, não se sentam. “Eu não faço nada”. São 8h30 da manhã em uma quarta-feira em setembro, e a classe de Wijnands não se parece em nada com uma sala de aula. As carteiras não estão organizadas em fileiras. Em vez disso, estão posicionadas para formar grupos de quatro alunos que ficam um de frente para o outro.

Em vez de se sentarem no início da aula, esses alunos pegam um grande cartaz de papel coberto com *post-its*, colam-no na parede e se reúnem em volta dele. O cartaz está dividido em algumas poucas colunas largas. *Alle items*, no extremo da esquerda. Seguida por *Te doen*, e por *In uitvoering* e, por fim, *Klaar*. Como você já deve ter adivinhado, elas significam “Todos os itens”, “Pendências”, “Fazendo” e “Feito”.



Na parte inferior das colunas, há quatro títulos adicionais: DDF, ou Definição de Feito, *Grafiek*, que indica o seu Gráfico de Burn-Down, mostrando o progresso deles em direção ao objetivo; e, por fim, Retrospectiva e Velocidade, no qual medem quantos "pontos" eles conseguiram durante cada aula. Seus Sprints costumam ser de quatro ou cinco semanas, terminando com uma prova.

Em frente aos quadros Scrum — ou *flops*, como eles os chamam em holandês (uma derivação da expressão *flip chart*, em inglês) — os alunos planejam quais lições vão terminar naquele dia. Eles movem os *post-its* com as tarefas que acreditam conseguir da lista de Pendências, *Alle Items*, para *Te Doen* e começam a trabalhar. Novamente, conforme Wijnands gosta de dizer, ele não faz nada. Os alunos abrem os seus livros e começam a aprender sozinhos. Talvez o mais importante: eles começam a ensinar uns aos outros. O professor caminha pela sala de aula, olhando para os quadros Scrum e os Gráficos de Burn-Down. Às vezes, ele percebe que alguns alunos estão enfrentando um problema, ou explica rapidamente algum conceito mais complexo, ou ainda, apenas escolhe uma história aleatória na coluna *Klaar* e faz um teste oral rápido com cada aluno, certificando-se de que todos entenderam os conceitos, e então volta para *Te doen*. Parte da DDF é que todos tenham compreendido a matéria.

Os alunos têm uma parte do quadro Scrum que é única para eles: uma "Definição de Diversão". O trabalho não precisa apenas estar concluído, mas eles também precisam se divertir ao realizá-lo. Os três testes são: Confiança, Humor e uma palavra única do holandês, *Gezelligheld*. Não existe uma boa tradução para essa palavra. Ela é descrita como "aconchego", ou "companheirismo", ou "prazer", ou "reencontrar um amigo depois de um longo tempo", ou "passar tempo com as pessoas amadas", ou, simplesmente, "fazer parte de um grupo". Na verdade, isso me parece uma forma perfeita de descrever a sensação de apoio, alegria, esperança, felicidade e animação por fazer parte de uma equipe realmente boa.

"Você não precisa policiar", explica Wijnands. "Agora nós temos outras formas de lidar com a orientação dos alunos. Eles fazem tudo, até mesmo definem o próprio dever de casa!". Cada equipe sabe onde está em relação à matéria, as datas quando precisam concluir os passos intermediários, e se precisam fazer algum trabalho fora da sala de aula para aprender todo o conteúdo a tempo. "Eles se auto-organizam; desenvolvem maneiras mais inteligentes e rápidas de estudar. Uma equipe começou com um teste e foi trabalhando de trás para frente. Um grupo de alunos com 11 anos de idade. Eu

disse ‘não está bom’. Eles ficaram tristes”, conta Wijnads, com um sorriso contagiante. “Então, eu disse: ‘está excelente!’”

O Scrum ou eduScrum, conforme Wijnands o chama, é apresentado aos alunos no primeiro dia de aula. A primeira coisa que fazem é escolher equipes — equipes interfuncionais. Cada aluno se autoavalia em diversas categorias, em quesitos que abordam tudo, desde coragem até o gosto por matemática, desde levar em consideração os sentimentos dos outros até “seguir em direção ao objetivo”. Então, os alunos recebem a informação que as equipes são interfuncionais, que devem ter todas as capacidades necessárias para aprender a matéria. Wijnands explica que isso ensina a eles algo tão importante quanto química — ensina a eles como trabalhar com pessoas e valorizá-las por terem talentos diferentes dos seus.

Tim Jansen é um garoto de 17 anos de idade que está no último ano do Ensino Médio. Ele já usa o Scrum há três anos e está prestes a entrar na faculdade, onde quer estudar química. Ele parece um *geek* típico: inteligente e talvez com as habilidades de socialização não tão desenvolvidas. “Eu aprendo mais rápido do que os outros”, diz ele. “Mas trabalhando em equipe, você consegue ser ainda melhor. Eu aprendo mais ao ensinar aos outros”. Ele se volta para Gudith Zwartz, que está em frente a ele na mesa. “Ela sabe que pode fazer perguntas sobre o conteúdo. Eu posso perguntar a *ela* sobre organização. Ela consegue fazer isso melhor do que eu”.

Gudith é bem diferente de Tim: esbelta, bonita, loura. “Você consegue conhecer seus colegas de turma melhor. Você sabe quem é bom em quê”.

“O Scrum ajuda os novos alunos a se conectarem com os antigos”, opina sua amiga igualmente bonita e cheia de estilo, Maneka Bowens. “Às vezes, você escolhe uma equipe e, às vezes, você é escolhido. Você aprende que eles são melhores do que você em algumas qualidades”.

Esse tipo de lição, explica Wijands, é parte da ideia. Tornar habilidades inconscientes, conscientes. Habilidades que possam ser testadas em uma prova estão longe de ser as únicas importantes. Ajudar os alunos a identificar e valorizar pontos fortes diferentes em si mesmos e nos outros é uma habilidade do século 21. Isso é algo que todos precisam aprender.

Depois de escolher as equipes, os alunos aprendem a estimar as tarefas, não em horas ou dias, mas em pontos. Então, avaliam cada parte da matéria que precisam aprender usando o método de avaliação relativa da sequência de Fibonacci e jogando Pôquer de Planejamento. Willy explica a ideia dos pontos de forma bastante simples: “ignore todas as medidas que conhecemos. Elas são

medidas absolutas. Se eu disser que o meu peso é cinquenta pontos", continua ele, apontando para uma menina esbelta, "quantos pontos você pesa?"

"Hã, quarenta?", adivinha ela.

"Ora, obrigado! Mas eu diria algo mais próximo de vinte".

Ao final de cada lição, as equipes fazem uma retrospectiva, perguntando a si mesmos: "o que deu certo?", "o que poderia ter sido melhor?" e "como a equipe pode melhorar?".

"Esse foco em equipes", diz Wijnands, "às vezes surpreende os pais". Ele conta a história de uma mãe que ligou e disse que a filha tinha feito todo o trabalho e perguntava por que ela estava sendo forçada a carregar todos nas costas.

"Eu disse que a garota tinha de ter a coragem de dizer aos outros para fazer mais. Ela disse, e os resultados nas provas foram melhores. A mãe me ligou para agradecer. Os alunos precisam aprender não apenas a trabalhar sozinhos, mas a trabalhar juntos".

A energia nas aulas em Ashram é notável e se traduz em resultados. No sistema holandês de ensino, as notas de avaliação variam de um a dez, e 5,5 é considerada uma nova aceitável para se passar de ano. Nas aulas de Willy, um sete é aceitável. E os alunos atendem a essa linha de corte. No decorrer do ano passado, conta Wijnands, as notas nas provas tiveram um aumento de mais de 10%.

Willy conheceu o Scrum com seu genro, que trabalha em uma grande empresa técnica da Holanda a qual usa esse método. Willy tem quase quarenta anos de magistério, e diz que isto era o que buscara a vida inteira: uma abordagem que ensinasse às crianças a ensinarem a si mesmas e a valorizar as próprias habilidades e as dos outros. Além disso, divertirem-se ao aprenderem.

Uma coisa importante sobre o Scrum é que ele raramente permanece algo exclusivo por muito tempo — ele foi projetado para *escalonar*. Nas escolas da Holanda, por exemplo, o eduScrum não depende apenas de uma pessoa, mesmo que seja um grande professor como Wijnands. Embora tenha começado com ele, e ele tenha convencido alguns dos outros professores de química de Ashram a o experimentarem, agora está crescendo. Com o apoio da comunidade empresarial, agora existe uma Fundação eduScrum na Holanda, que treina professores e dá informações às escolas sobre o Scrum. Eles já treinaram 74 professores até agora, de todas as matérias, em 12 escolas; e pretende-se ter um crescimento de sessenta professores e 15 escolas por ano. Em cinco anos, isso significará trezentos professores a mais e 75 escolas

adicionais. Um bom começo. Eu conheci alguns desses professores pelo país, e eles me disseram que esse é o novo Montessori[46]. Eles veem isso como um *movimento*.

Isso não está acontecendo apenas na Holanda. No Arizona, há uma escola autônoma[47] na zona rural e pobre para índios americanos que usa o Scrum. Em algumas poucas universidades, eles estão começando a ensinar. Na Havard Business School (HBS), construíram uma nova sala de aula chamada Laboratório de Inovação, onde todo o ensino se baseia em equipes. E, conforme o Professor Takeuchi, da HBS, me disse, quando você ensina para equipes, a maneira de se fazer isso é usando o Scrum.

Enquanto estive em Ashram, conversei com alguns alunos lá. Quando perguntei que dúvidas eles tinham, um menino levantou a mão. “Não consigo acreditar que você criou isso para desenvolvimento de software”, disse ele. “Parece que foi criado para o ensino fundamental”.

Meus olhos se encheram de lágrimas enquanto eu olhava para aquele rapaz. Eu soube mais tarde que ele era autista. Antes do Scrum, ele não se integrava e era resignado. O Scrum deu a ele uma maneira de seguir em frente, de realmente gostar da escola, de se tornar uma pessoa mais completa e melhor. Alguns anos atrás, quando eu estava tentando consertar algumas empresas de software, eu não tinha me dado conta de que eu também estava criando algo que poderia ajudar a consertar a vida das pessoas.

Mas assim foi. E talvez em nenhum lugar de forma mais intensa do que na zona rural de Uganda.

Pobreza

Uganda é um dos países mais pobres do mundo. Mais de um terço do povo vive com menos de US$ 1,25 por dia. A grande maioria dos ugandenses vive na zona rural, onde a pobreza é endêmica, e as pessoas lutam para subsistir trabalhando em pequenas lavouras familiares. Muitos desses lugares são bastante remotos — são dias de viagem a pé até a cidade mais próxima onde haja um mercado. As famílias enfrentam imensas dificuldades para mandar as crianças para a escola, já que precisam da ajuda delas no trabalho nas fazendas. As meninas costumam parar de estudar bem cedo. A expectativa de vida é de 53 anos. A mortalidade infantil é de mais de 5% dos nascidos vivos, e cerca de seis mil mulheres morrem todos os anos como resultado de complicações na

gravidez. A vida dos lavradores em Uganda não é nada fácil.

A Grameen Foundation foi desenvolvida pelo ganhador do Prêmio Nobel, Muhammad Yunus, do Grameen Bank, que foi pioneiro no microcrédito para os extremamente pobres em Bangladesh. A fundação se concentra em ajudar a tirar os pobres da linha da miséria, não por meio de esmolas, mas dando-lhes apoio para fortalecer seus pontos fortes e suas capacidades subestimadas. Em Uganda, eles decidiram tentar fazer isso dando a essas pessoas a oportunidade de produzir e compartilhar conhecimento.

Para tanto, recrutaram cerca de 1.200 pessoas nas áreas rurais e pobres — pessoas a quem chamavam de Trabalhadores do Conhecimento da Comunidade (*Community Knowledge Workers* — CKWs). A fundação já havia desenvolvido aplicações móveis para microfinanciamento e pagamentos, e decidiu dar a esses trabalhadores do conhecimento não apenas informações bancárias, mas informações que pudessem usar em suas vidas, o que, no caso de Uganda, significava algo a ser aplicado na lavoura. A Grameen Foundation forneceu acesso às melhores práticas agrícolas ao dar aos trabalhadores smartphones que pudessem transmitir informações.

Steve Bell, da Lean Enterprise Institute e um Mestre Scrum Certificado, recentemente visitou duas aldeias remotas e descreveu como isso funciona. Existe uma reunião de lavradores que ficam em pé em um campo. Um deles trouxe uma planta assolada por uma doença. O CKW rapidamente visualizou as fotos em seu smartphone até encontrar a fotografia de uma planta assolada por aquela doença específica. Então, havia, instantaneamente, um tratamento científico para a doença — um tratamento que não exigia pesticidas ou produtos químicos caríssimos, mas um que o lavrador poderia começar a usar imediatamente.

Bell afirma que a transmissão rápida de informações práticas seria eficaz o suficiente, mas o aplicativo também ligou os lavradores por toda a Uganda. Usando essa conectividade, eles conseguem compartilhar exatamente quanto as colheitas estão vendendo no mercado da cidade mais próxima. Os lavradores costumavam ficar nas mãos dos intermediários, que tiravam vantagem da falta de conhecimento de mercado deles para definir os preços nos níveis que quisessem. Agora, os lavradores sabem quanto os intermediários estão ganhando.

Bell me contou a história de uma mulher que lhe disse que os dados agrícolas por si só dobraram a sua produção. Mas que os dados de mercado também fizeram com que os seus preços dobrassem. Ela costumava receber trezentos

xelins por uma tulha, mas, depois que descobriu que o produto era vendido por mil xelins, ela conseguiu negociar o preço e chegar a seiscentos xelins. Dobro de produção e dobro do preço, a mesma quantidade de trabalho. Foi para isso que o Scrum foi criado, e foi isso que o método entregou a ela.

Eric Kamara chefia o grupo de tecnologia do escritório de Kinshasa da Grameen Foundation. Seu grupo usa o Scrum para desenvolver suas aplicações. Ele diz que cada vez que um grupo pede um conjunto de atributos, sua equipe classifica-os em uma escala de um a sete, de acordo com as três perguntas a seguir:

1. Qual é a importância deste trabalho para a missão de ajudar os pobres?
2. Como este atributo vai contribuir para o trabalho dos CKWs?
3. Existe algum parceiro apoiando este atributo? (A fundação prefere trabalhar com parceiros, como a Gates Foundation, em vez de sozinha.)

Isso permite que Kamara priorize o trabalho usando critérios objetivos. Antes do Scrum, explica ele, as pessoas pediam tudo de uma vez. E, com recursos limitados de uma instituição sem fins lucrativos, não tinham como fazer tudo; então, o efeito era não fazer nada. Agora, a cada Sprint, os grupos diferentes que querem atributos entram e apresentam o que querem que seja feito, de forma transparente, para que possam ver exatamente como o seu atributo se compara aos outros. Isso ajuda um grupo com parcos e preciosos recursos determinar o que terá maior impacto.

Como já vi em outros setores, esse tipo de trabalho se espalhou rapidamente pelos outros escritórios de Kinshasa, afetando, literalmente, o modo como realizavam o trabalho das 9h às 17h. O escritório costumava ter um tipo de reunião semanal que todo mundo odiava — a atualização de status que levava horas durante as quais se declaravam os problemas e se faziam reclamações, mas muito pouco era feito. A reunião parecia não ter fim, e todos saíam insatisfeitos. Em geral, o único resultado era colocar a culpa em alguém, em vez de buscar soluções. Agora, conta Kamara, toda equipe tem um quadro Scrum. Antes da reunião, os problemas e obstáculos se tornam facilmente detectáveis. Hoje, o diretor do escritório pode simplesmente caminhar pela empresa e ver, de forma instantânea, onde estão as obstruções ou bloqueios, apenas ao verificar o estado do quadro Scrum.

Se você conversar com pessoas do mundo das ONGs, vai perceber que a reclamação comum é que há um grupo imenso que compartilha os mesmos

objetivos e tem o mesmo comprometimento, mas não tem disciplina. Diante desse quadro, o que o Scrum pode fazer é pegar a paixão das pessoas e, ao dar clareza sobre o que devem priorizar, canalizá-la.

É fácil tornar o Scrum um caso de *negócios*. Se você usá-lo, vai ganhar mais dinheiro — muito mais, porque vai conseguir fazer o dobro na metade do tempo. Mas a promessa mais brilhante para a humanidade está com as pessoas que devotam suas vidas para ajudar os mais pobres entre os pobres. Se o Scrum pode ajudar essas pessoas que trabalham às margens a conseguirem o mesmo efeito, um passo gigantesco terá sido dado para conseguir um bem social mais amplo.

Esse "bem" não apenas chegará antes, mas também será mensurável. O Scrum oferece a capacidade de medir facilmente o progresso. Na Grameen Foundation, eles têm o que chamam de "Índice de Exclusão da Pobreza" (*Progress out of Poverty Index* — PPI), que mede o nível de eficácia do programa. Com isso é possível apurar e *ver* exatamente o impacto que os CKWs estão tendo com seus smartphones nas aldeias rurais. Eles podem experimentar fazer as coisas de maneiras diferentes; e podem ajudar as pessoas a *inovarem* para sair da pobreza.

Para mim, é incrível ver o Scrum voltando às suas raízes. Logo que comecei, instituições como o Grameen Bank e outras instituições de microfinanciamento me inspiravam; achava incrível o modo como ajudavam equipes de pessoas pobres a trabalharem juntas para saírem da pobreza. Eles formavam uma equipe e cada membro fazia um plano de negócios expondo o que fariam com US$ 25. Um poderia querer comprar uma carroça para vender frutas na praça da cidade; outro poderia querer comprar uma máquina de costura para fazer vestidos para vender. Apenas quando todos os empréstimos da equipe estivessem quitados é que o grupo poderia receber mais dinheiro. Eles se reuniam semanalmente para ver de que forma poderiam ajudar uns aos outros, e os resultados foram incríveis. No início, a costureira conseguia ganhar apenas o suficiente para alimentar os filhos, mas algumas semanas depois, ela talvez conseguisse comprar sapatos para eles; e, então, poderia mandá-los para a escola. Alguns ciclos depois, ela teria um pequeno negócio e poderia começar a construir uma casa de verdade. Na época, falei para os programadores de software com quem eu estava trabalhando: "Aquelas pessoas pobres não tinham sapatos, ainda assim, elas podem sair da pobreza. *Vocês* têm sapatos, mas não têm software. Eles descobriram um modo de trabalharem juntos para sair da miséria. Vocês estão dispostos a fazer o mesmo?". E foi assim que o

Scrum nasceu.

Empresas sem fins lucrativos são apenas uma das áreas onde podemos inovar o bem social. E quanto ao modo como nos organizamos? E quanto ao governo?

Governo

O governo não é apenas como organizamos a esfera pública — como conseguimos estrada, policiamento, cortes e o departamento de veículos motorizados — é também o modo como formalizamos quem somos como pessoas. É uma codificação de quem acreditamos que somos. Nos Estados Unidos, os anseios fundamentais do povo estão registrados em um documento assinado por rebeldes que certamente teriam sido enforcados separadamente se não tivessem se mantido unidos — A Declaração da Independência. Redigida por um aristocrata, idealista, latifundiário e dono de escravos, o documento capturou, de forma surpreendente, um conceito radical de que tipo de pessoas os americanos da época da revolução queriam ser.

> Consideramos estas verdades evidentes por si mesmas, que todos os homens são criados iguais, que são dotados pelo Criador de certos direitos inalienáveis, que entre estes estão a vida, a liberdade e a busca da felicidade. Que para assegurar esses direitos, governos são instituídos entre os homens, derivando seus justos poderes do consentimento dos governados.

É difícil calcular hoje o nível de afastamento das normas que tais palavras representavam. Embora as ideias do Iluminismo tivessem começado a se espalhar, não existiam democracias naquela época. As leis eram impostas de cima, do direito divino dos reis e a força de armas. Grandes impérios governavam grande parte do mundo — não apenas a Grã-Bretanha, mas também a França, a Áustria, a Rússia e a Turquia. A ideia de que indivíduos são *dotados* de direitos, em vez de recebê-los dos poderosos, era uma ideia revolucionária, para dizer o mínimo.

A “república” era uma forma de governo que emergiu desses ideais. Assim como o robô de Rodney Brooks aprendendo a andar, os Estados Unidos se levantou sobre os pés, tropeçou, caiu e, algumas vezes, tomou o caminho errado. No entanto, aqueles ideais inspiraram revoluções por todo o mundo, e hoje a maioria das principais potências é governada, pelos menos em parte, pelas pessoas que eles pretendem representar.

O problema, obviamente, são os duzentos e tantos anos do crescimento burocrático — interesses permanentes embutidos na própria estrutura do governo que dificultam que a voz do povo seja ouvida. A corrupção — seja em pequena escala de burocratas que aceitam subornos por serviços, ou em grande escala de grandes bancos acumulando riqueza ao privatizar os lucros e dividir as perdas — é o resultado da falta de transparência e da centralização do poder nas mãos de poucos.

Na maioria das capitais mundiais, desenvolveu-se uma classe de cortesãos que constitui o governo permanente. Contratos são concedidos, dinheiro é ganho, e o poder é conferido com base em "quem indica" e não em "o que você acrescenta". Em nenhum lugar isso é mais evidente do que no modo como políticos, generais e burocratas poderosos alternam entre o governo e o setor privado repetidas vezes. É inacreditável o número de generais de quatro estrelas que se tornam fornecedores de serviços de defesa ou senadores que se tornam lobistas, ou ex-funcionários públicos da administração que entram para grupos comerciais.

No entanto, como enfatizei no Capítulo 3, é inútil procurar pessoas ruins; precisamos procurar sistemas ruins. A culpa só serve para tonar as coisas mais lentas, então, não vamos cair na armadilha fácil do Erro Fundamental de Atribuição. Em vez disso, vamos fazer uma pergunta que tem uma chance de realmente mudar as coisas: "Qual é o conjunto de incentivos que estimula o comportamento ruim?". Eu realmente duvido que qualquer um que trabalhe em uma empresa *Beltway Bandits*[48] se veja como uma pessoa má, e aposto que a maioria delas é cheia de boas intenções. Foi o *sistema* que fracassou com eles, e conosco. Mas como podemos mudar isso? Como podemos encorajar transparência, prioridade e responsabilidade? Você já sabe a resposta: Scrum.

Comecemos a alguns milhares de quilômetros a oeste de Washington, D.C., na capital do estado de Washington, Olympia. Ali, os dois últimos governos — primeiro um Republicano e, depois, um Democrata — abraçaram o que chamaram de "governo Enxuto". O atual governador, Jay Inslee, disse em uma entrevista de campanha no outono de 2012: "Muito do que o estado faz é tomar decisões. Queremos encontrar uma forma de deixar menos papel em uma mesa".[49]

O plano do governador tem cinco pontos que poderiam ter sido tirados de qualquer plataforma de campanha: 1) um sistema de educação de "padrão mundial", da educação infantil até o ensino superior; 2) uma "economia próspera"; 3) tornar Washington um líder nacional em energia sustentável e

meio ambiente limpo; 4) comunidades saudáveis e seguras; e 5) um governo eficiente, eficaz e responsável.

Esses objetivos não são revolucionários. Isso é o que as pessoas *devem* esperar de seu governo. Só o fato de elas soarem como um clichê já é um indicador de sua importância. Afinal, um clichê é apenas uma verdade repetida vezes o suficiente para se tornar trivial. Mas o que é *diferente* sobre a administração de Inslee é como ele vai fazer isso. Eles vão aplicar a nova abordagem SMART — *Specific* [específica], Mensurável, Alcançável, Relevante, e com Tempo determinado. Em outras palavras: eles querem usar o Scrum. E, na verdade, estão fazendo isso.

O gabinete do CIO do Estado de Washington é responsável não apenas pela tecnologia comprada, mas por como ela é feita. Ele é formado por vinte pessoas que devem assegurar que fracassos maciços de TI, custando dezenas de milhões de dólares, não aconteçam. Enquanto isso, o departamento cuida das atualizações de TI para as partes do governo que fazem de tudo, desde emitir carteiras de habilitação, passando pela distribuição de auxílio desemprego, até a regulamentação da pesca e da vida selvagem. Em 2012, eles supervisionaram oitenta pedidos, totalizando US$ 400 milhões. E publicam padrões e guias para diversas agências sobre como implementar uma política pública.

Para fazer isso, eles usam o Scrum; na verdade, derrubaram as divisórias das salas e formaram equipes Scrum. Michael De Angelo, o CIO suplente, diz que eles tentam entregar políticas práticas e implementáveis para os departamentos públicos a cada semana.

“Estamos atualizando os nossos processos em relação ao modo como as repartições públicas enviam os planos para investimento. Definimos o objetivo de que a cada semana vamos mudar uma coisa; estamos aplicando uma abordagem incremental. Temos um produto potencialmente entregue semanalmente e isso pode ser sentido pelas repartições. Elas recebem, na verdade, algo tangível”. “Produto potencialmente entregue”, no caso deles, significa mudanças práticas na política. Não precisa ser uma coisa; precisa apenas ser algo, qualquer coisa que crie valor.

Em vez de tentar criar um documento grandioso e predominante antecipando cada uma das peças do processo de fundos, eles decidiram criá-lo considerando uma peça de cada vez. Eles querem entregar aprimoramentos na forma como o estado governa toda semana.

De Angelo diz que a reação foi mista. Existe um grande temor de não conseguirem um produto perfeito. Falando sobre agosto de 2013, ele disse:

> Na semana passada, fizemos uma mudança no modo como os clientes nos telefonam. Mas ainda existem vários documentos nos quais as maneiras antigas estão explicadas — no nosso website, documentos entre outras coisas. Então, havia todas essas outras coisas que teríamos de mudar [primeiro]. Decidimos não esperar e resolver logo isso. No próximo Sprint, atualizaremos toda a documentação. De outra forma, não daremos a eles uma maneira melhor por meses... Estaremos roubando deles o valor.

Outra coisa que o gabinete do CIO está fazendo é implementar o Scrum em toda a burocracia do estado. Foi por isso que eles mudaram todos os seus processos para o Scrum — para se tornarem um exemplo, a fim de serem capazes de falar por experiência. Os benefícios são grandes demais para não fazê-lo.

Mas existem alguns obstáculos. De Angelo diz que uma coisa que percebeu é que, em alguns casos, o método em cascata faz realmente parte da lei estadual. Mudar isso pode ser difícil. O Estado de Washington investe recursos em ciclos de dois anos. “Você precisa pedir grandes somas. Não podemos dizer que vamos agregar valor até você nos mandar parar”, explica De Angelo. “O governo quer saber que [uma coisa] vai custar aquele valor, e que [aquela coisa] vai gerar esse valor dentro de determinado período. Isso acontece para que eles possam falar sobre isso com os cidadãos. Mesmo que saibamos que isso é muito mais ineficiente”.

Parte do problema é que nos Estados Unidos, tanto na esfera federal quanto na estadual, a assembleia legislativa é dividida em comitês. Um grupo de legisladores olha para a educação, outro para o crime, outro para o orçamento, e outro para o serviço social. “Eles são fragmentados. Nunca olham a para o todo”, explica Rick Anderson. Ele é um consultor de agências estaduais e municipais em Washington, Oregon, Califórnia e Havaí; já trabalhou com assembleias legislativas e diz que, embora a mudança possa demorar um tempo, ela vai acontecer. Ele acha que eles deveriam começar definindo objetivos baseados no desempenho.

“Tudo bem, Agência X, aqui estão os seus objetivos e aqui estão os resultados esperados. Uma vez que tiverem isso, vocês poderão começar e escrever leis que são baseadas no resultado”, diz ele.

Em um mundo reformado e movido pelo Scrum, em vez de aprovar um plano específico para construir uma ponte sobre um rio, o corpo legislativo diria para o departamento de estradas: “Nós queremos que X número de pessoas sejam capazes de viajar sobre esse rio em um tempo Y com um custo Z. Como vocês

farão isso é com vocês". Isso abriria as portas para a descoberta e a inovação.

Em vez disso, a norma hoje é que os projetos de construção ultrapassem em centenas de milhões de dólares o orçamento. O motivo? À medida que as equipes trabalham no projeto, eles descobrem novos problemas e novas formas de resolvê-los. Em vez de sufocar esse tipo de inovação com Conselhos de Controle de Alterações e relatórios gigantescos, deveríamos encorajá-la.

Mas e quanto àqueles ideais com os quais comecei esta seção? Aqueles que diziam que a sociedade se forma através de um documento? Digamos, uma constituição? Bem, um país decidiu que a forma de se desenvolver uma constituição que realmente represente a vontade do povo era usar o Scrum.

Em 2008, uma crise financeira que poderia ter sido evitada atingiu o mundo. Grandes bancos dilataram os preços de forma incontrolável, alavancando-se repetidas vezes ao assumirem mais dívidas duvidosas que jamais seriam pagas de volta. Um dos países que mais sofreu foi a Islândia. Bancos privatizados lá foram desmembrados pelo governo e tiveram de correr grandes riscos no mercado financeiro. Como se costuma dizer em Wall Street: "se você não sabe quem é o trouxa da sala, você é o trouxa". Nesse caso, a Islândia foi a trouxa. A quantia de dinheiro que tomaram em empréstimo para um país tão pequeno foi inacreditável. Consequentemente, os bancos tiveram avaliações 12 vezes maiores do que o orçamento nacional do país. Quando tudo ruiu, o "milagre econômico" islandês estava destruído.

Em uma expressão de ultraje, os cidadãos de Reykjavík tomaram as ruas e bateram panelas e frigideiras do lado de fora de Alþingi, o parlamento do país. O governo que se descuidara das práticas financeiras caiu no que ficou conhecido como "revolução das panelas". O governo deixou o poder e a nova liderança prometeu uma nova constituição.

Para escrever tal constituição, alguns funcionários públicos decidiram conversar com o povo. Então, formaram um comitê constituinte que decidiu usar o Scrum. A cada semana, o grupo se reunia, decidia sobre uma seção do documento, e a entregavam para o público toda quinta-feira. Então, eles recebiam o feedback popular através do Facebook e do Twitter. Em apenas alguns meses tinham um novo documento, que recebeu um apoio impressionante dos islandeses. Tratava-se de uma nova expressão de como eles se viam.

Infelizmente, os poderes que se beneficiaram com a fraude financeira reagiram. Depois de entrarem com pedidos de mais prazo repetidas vezes — depois de atordoarem, reclamarem e agirem contra a vontade do povo —, um

novo parlamento, formado pelos mesmos partidos que negligenciaram a destruição da economia da Islândia, decidiu, simplesmente, ignorar a nova constituição. A principal exigência da revolução foi negada. Por ora, pelo menos.

O mundo está mudando, e aqueles que lucram com o sigilo e a fraude logo descobrirão que têm poucos lugares para se esconderem. O Scrum está mudando o mundo à volta deles e, embora eles possam lutar na retaguarda, a mudança é inevitável. A estrutura do Scrum é tão mais rápida, transparente e responsiva aos desejos das pessoas que, no final das contas, derrotará os políticos que ficarem em seu caminho.

Mude ou morra.

Como todos trabalharemos um dia?

Neste livro, discutimos o conceito das artes marciais, chamado *Shu Ha Ri*. As pessoas no estado *Shu* seguem exatamente as regras, para que possam aprender as ideias por trás delas. No estado *Ha, os indivíduos* começam a criar o próprio estilo dentro das regras, adaptando-as às suas necessidades. Já no estado *Ri, eles* existem além das regras: incorporam os ideais. Observar um verdadeiro mestre no estado *Ri* é como olhar uma obra de arte em movimento. Suas ações parecem impossíveis, mas isso é porque o mestre se tornou a filosofia em carne e osso, tornando-a real.

Tudo isso para prefaciar o fato de que *existem* algumas regras no Scrum, e você deve se esforçar para aprendê-las e transcendê-las. Eu as inclui em um Apêndice ao livro. Além disso, escrevi os capítulos explicando por que essas regras existem, encorajando você, espero, a aplicá-las na sua vida, na das pessoas ao seu redor e na sua comunidade. O paradoxo dessas regras, porém, é que elas eliminam fronteiras, criam liberdade — e, para muitos, a liberdade pode ser aterrorizante.

Uma empresa que tenha aprendido a como libertar seus funcionários *e* otimizar a inovação foi a Valve. Olhar para a empresa é ver como todos nós inevitavelmente nos organizamos, seja para desenvolver um software melhor, tirar as pessoas da pobreza, preparar um casamento ou reformar a casa.

Fundada nos anos 1990 como uma empresa de *video game* especializada em jogos revolucionários, tais como Half-Life e Portal, a Valve é completamente autofinanciada e detém a propriedade intelectual de todos os seus produtos.

Quase todos os seus mais de trezentos funcionários ficam em um único escritório localizado em um arranha-céu em Bellevue, Washington. A empresa conta com mais de cinquenta milhões de clientes e fatura centenas de milhões de dólares por ano. E não existe uma pessoa realmente no comando.

A origem da Valve é, entre todos os lugares, a Microsoft. Hoje a Microsoft é uma empresa muito diferente, mas, nos anos 1990, ela era o exemplo de uma corporação hierarquizada de cima para baixo. Todos se definiam de acordo com o lugar que ocupavam na pirâmide corporativa e o quanto estavam abaixo do fundador e CEO, Bill Gates — na época, o homem mais rico do mundo, e atualmente ainda um dos mais ricos.

Greg Coomer é um dos fundadores da Valve. Ele trabalhava para Gabe Newell, que chefiava um grupo de desenvolvimento na Microsoft. Greg descreve como aquela hiperatenção ao prestígio se refletia nas ferramentas que as pessoas usavam.

> Na Microsoft, existia um plug-in do Outlook chamado “Org Chart”. E a cada e-mail que alguém recebia, eles clicavam para ver onde o remetente se encaixava no diagrama organizacional da empresa. A quantos cliques de distância estavam de Bill Gates, quantos reportes diretos, eles eram amigos ou inimigos — tudo isso poderia ser descoberto a partir da posição de cada um no Diagrama Organizacional.

Greg diz que, se você afastasse um pouco o ponto de vista, poderia ver que existia aquela pirâmide gigantesca com Bill Gates no topo; e, se observasse mais de perto, veria várias pirâmides menores. “Eram pirâmides até a base”.

A não ser pelo grupo de Gabe Newell. Ele era composto por algumas poucas centenas de pessoas, e todas se reportavam diretamente a ele. “Era impressionante ver aquilo no ‘Org Chart’”, conta Greg. “Era algo que não se encaixava, e começou a causar problemas políticos, porque ele não tinha o número adequado de gerentes nem a estrutura correta”. A resposta da empresa foi quase como aquela dos glóbulos brancos do sangue atacando uma infecção. Agora, é claro, a Microsoft já tem três mil pessoas trabalhando em equipes Scrum e seu objetivo é ter vinte mil pessoas organizadas assim. Mas, na época, aquela “infecção” tinha de ser removida.

Assim, Gabe, Greg e alguns outros deixaram a Microsoft e fundaram a própria empresa, a Valve. Alguns anos atrás, Greg tentou escrever um guia do funcionário explicando como a Valve funciona. O documento não exibe níveis salariais ou a conta de despesas. Em vez disso, era uma tentativa de representar

o *éthos* da Valve.

"Eu percebi que as pessoas estavam levando nove a 16 meses para internalizar a maneira como a Valve faz as coisas. Levava muito tempo para elas sentirem que tinham poder", explica Greg. O objetivo do documento era tranquilizá-las mais rapidamente, mas Greg e os outros fundadores lutaram com as palavras, porque não queriam que a explicação parecesse estar vindo de cima. A primeira seção é "Bem-vindo à planície":

> Trata-se do nosso modo resumido de dizer que não temos qualquer gerência, e ninguém "se reporta" a ninguém. Temos um fundador/presidente, mas nem mesmo ele é o nosso gerente. Essa empresa é sua para guiar — em direção a oportunidades e afastá-la dos riscos. Você tem o poder de dar o sinal verde para projetos. Você tem o poder de entregar produtos.
>
> Uma estrutura plana remove todas as barreiras organizacionais entre o seu trabalho e o cliente que aproveitará tal trabalho. Todas as empresas dirão que "o cliente é quem manda", mas aqui essa declaração tem um peso. Não há nenhuma corda vermelha impedindo que você descubra, por sim mesmo, o que os clientes querem e, então, dê isso a eles.
>
> Se você está pensando consigo mesmo: "Uau, isso parece muita responsabilidade", você está certo.[50]

Eis como um projeto começa na Valve: alguém decide começá-lo. Só isso. Eles decidem o melhor uso de seu tempo e energia, o que servirá melhor para a empresa e os clientes, e fazem isso. Como conseguem outras pessoas para trabalharem no projeto? Convencem-nas. Se aquela outra pessoa achar a ideia boa, ela entra na equipe, ou "conspiração", como chamam na Valve. Todas as centenas de mesas na empresa têm rodinhas. Quando as pessoas começam a trabalhar juntas em um projeto, elas literalmente votam com as suas mesas, mudando-as para uma nova configuração.

Greg descreve como isso funcionou para um produto chamado Big Picture. Um dos produtos mais importantes da Valve é a plataforma Steam. Ela entrega *video games* e softwares para os usuários. Tanto jogos desenvolvidos pela Valve, quanto jogos de terceiro estão nessa plataforma; é a principal forma como os jogos de PC são entregues hoje. Mas, como Greg bem se lembra, em determinado ponto, ele e alguns outros temiam que já tivessem alcançando o número máximo de clientes que conseguiriam: mais de cinquenta milhões.

> Começamos a pensar sobre o modo como a nossa empresa estava crescendo e como o Steam estava crescendo, e olhamos para o que achamos que poderia ser o limite numérico de clientes que poderíamos atingir. Queríamos chegar a pessoas em outros lugares, na sua sala de estar, em seus dispositivos móveis ou o que quer que fosse.

Então, começamos a conversar com as pessoas — alguns designers e outros colegas. Nós os convencemos de que seria uma boa ideia inventar algo que poderia ser funcional em televisões, telefones e tablets, e eles criaram o Big Picture — uma forma de entregar *video games* para aquelas plataformas. Mas as pessoas que Greg convencera não tinham todas as habilidades necessárias para construir aquilo. Eles sabiam como queriam que o produto fosse, mas não tinham a habilidade para implementá-lo.

> Então, começamos a fazer modelos de como ele seria e, depois, fizemos um filme de como ele seria legal, e o usamos para recrutar pessoas para o projeto. Nós não tínhamos como criar os códigos para aquilo, por isso precisamos recrutar as pessoas que podiam fazê-lo.

E foi o que fizeram. Foi lançado um ano depois. Quem decidiu quando lançar? As pessoas que trabalharam no projeto. Quem decidiu se já estava bom o suficiente? As pessoas que trabalharam no projeto.

"Quando uma empresa se otimiza em torno da inovação, ela costuma mudar de maneira fundamental, eliminando estruturas internas e a hierarquia ou qualquer outra estrutura interna", explica Greg. A Valve funciona assim o tempo inteiro, ela não espera ser forçada a mudar por uma crise, mas sim, muda *constantemente*. De acordo com o seu guia:

> A Valve não é avessa a todas as estruturas organizacionais — ela surge de diversas formas o tempo todo, mas temporariamente. Mas os problemas aparecem quando divisões hierárquicas ou codificadas de trabalho não tenham sido criadas pelos membros da equipe ou quando tais estruturas persistem por longos períodos. Acreditamos que é inevitável que tais estruturas comecem a se preocupar com as próprias necessidades, em vez de com as necessidades dos clientes. A hierarquia começa a reforçar a própria estrutura ao contratar pessoas que se encaixam à sua forma, acrescentando pessoas para assumirem o papel de apoio de subordinação. Seus membros também são estimulados a assumir comportamentos que visam apenas conseguir mais renda e que tiram vantagens da estrutura do poder em vez de se concentrarem simplesmente em entregar valor aos clientes.[51]

Pode parecer que a Valve ficaria vulnerável aos parasitas — àquelas pessoas que querem tirar vantagem do sistema —, mas a revisão dos pares é constante. É claro que as pessoas decidem no que vão trabalhar, mas se eles não conseguirem convencer ninguém mais de que aquela é uma boa ideia, talvez ela não seja mesmo. Greg diz que, em vez de se *dar o luxo* de ter alguém para dizer a você o que fazer, você tem um grupo de pares dizendo a você o que eles

acham do que você decidiu fazer.

Não é um sistema perfeito. Nenhuma organização humana o é. Em geral, porém, na Valve, preocupações com pessoal costumam ser levantadas primeiro pelos membros da equipe conversando entre si. Eles podem chamar outras pessoas para fazerem uma consulta. Talvez resulte em feedback, uma ação corretiva forte ou, até mesmo, a demissão. Mas trata-se de uma decisão da equipe.

A exceção ocorreu em 2013, quando a Valve desenvolveu um problema o qual o seu sistema não conseguiu resolver. Pela primeira vez, eles contrataram um grupo grande de pessoas de uma vez só, pois tinham decidido expandir para hardware e dispositivos móveis, e simplesmente não tinham a capacidade para fazer isso.

No entanto, contratar tanta gente de forma simultânea, que não estava acostumada com o modo da Valve de resolver as coisas, começou a causar problemas. Havia diversos funcionários que não tomavam as decisões do modo tradicional da Valve. *Eles estavam mandando as outras pessoas fazerem as coisas* e não apresentavam os altos padrões da Valve. Normalmente, os outros membros da equipe não tolerariam esse tipo de comportamento. Mas, como todos no grupo eram novos, os pares não tinham confiança o suficiente no modo Valve de trabalhar para fazerem alguma coisa a respeito.

“Então, um grupo experiente de funcionários que já trabalhavam na empresa havia um tempo entrou em ação para proteger o *éthos* da Valve. Mesmo que tivesse de agir *fora* do *éthos* para isso”, conta Greg. A empresa demitiu algumas dezenas de pessoas de uma vez. Ao conversar com Greg, dá para perceber que ele ainda considera a experiência um fracasso. Ele a descreve quase como uma reação biológica, que era curiosamente parecida com a forma como a Microsoft agira com os fundadores da Valve: organismos atacando invasores estranhos para proteger o todo.

“Temos falado muito sobre o que significa para os nossos objetivos declarados o fato de termos agido fora deles”, reflete Greg. “E como podemos evitar isso no futuro. E não ter de depender de um grupo de pessoas que já está na empresa há muito tempo”. Ele para por um minuto e, então, diz com confiança: “Mais ou menos nessa época, no ano que vem, teremos descoberto”.

Existe uma fé no que eles fizeram; eles buscam constantemente maximizar a liberdade, a capacidade e a criatividade humanas. Embora tenha havido alguns soluços ocasionais, trata-se de uma forma poderosa demais de funcionamento para não ser copiada repetidas vezes.

“Esta é uma inovação capitalista tão poderosa quanto muitas inovações industriais que mudaram a natureza do trabalho”, afirma ele. “É tão útil e bem-sucedida que não tem como não se tornar uma força de mudança no mundo”.

Eles usam Scrum? “Bem”, responde Greg, “você caminha pelo corredor e vê um monte de quadros brancos presos em rodinhas cobertos com *post-its*”. Mas eles não *forçam* as pessoas a usá-los, e sim permitem que decidam qual processo julgam mais adequado. Assim como na maioria dos assuntos, Greg e os demais fundadores evitam dizer às pessoas o que devem fazer. Mas vários funcionários da Valve decidiram que, diante da escolha de fazer qualquer coisa, eles escolhem o Scrum. E isso é o suficiente.

Você ainda não vê muitas empresas como a Valve. Mas, a cada dia que passa, surgem novas empresas. E não apenas no setor de desenvolvimento de software. A Morning Star, a empresa líder de processamento de tomates, não tem nenhum chefe. Cada funcionário negocia com os outros funcionários como assumir papéis e responsabilidades, seja em relação às vendas, seja dirigindo caminhões ou trabalhando com engenharia sofisticada. Com qualquer empresa, primeiro você tem que fazer com que os funcionários se libertem e, então, tem de fazer com que aceitem a responsabilidade que vem com isso.

Ou, como Funkadelic disse em 1970: “Liberte a sua mente... E o seu rabo vai seguir logo atrás”.

O que não podemos fazer?

O cinismo talvez seja a resposta racional para a falta de esperança. No entanto, esse é um dos estados humanos mais corrosivos que existem. Os anos iniciais deste século tiveram uma abundância de elementos que cultivam o cinismo: guerras motivadas pelo patriotismo; terrorismo niilista mascarado de fé; ganância disfarçada por um manto de justiça ideológica; bajuladores políticos ambiciosos em busca dos próprios objetivos egoístas.

O cínico vai suspirar sabiamente e dizer: “É assim que o mundo funciona. Os seres humanos são, em essência, corruptos e egoístas — fingir que as coisas são diferentes é apenas ingênuo”. Desse modo, ele justifica as restrições e racionaliza os limites.

Nas duas últimas décadas, eu mergulhei na literatura e pesquisei a fundo o que torna algo ou alguém grandioso. A resposta surpreendente foi que, basicamente, os humanos *querem* ser grandes. As pessoas querem fazer algo

significativo — para tornar o mundo, mesmo que de forma restrita, um lugar melhor. O importante é se livrar do que está no caminho, retirar os obstáculos para que elas se tornem as pessoas que são capazes de ser.

É isso que o Scrum faz. Ele define objetivos e, de forma sistemática, passo a passo, descobre como chegar lá. E, ainda o mais importante, identifica o que está nos impedindo de chegar lá.

O Scrum é o código do "anticínico". Scrum não é desejar um mundo melhor, ou se render ao que já existe. Em vez disso, é, na prática, uma forma objetiva de implementar mudanças. Conheço projetos Scrum cuja meta é entregar vacinas para crianças em risco, e outros com o objetivo de construir casas mais baratas, eliminar a corrupção de baixo escalão, capturar criminosos violentos, acabar com a fome, e enviar pessoas para outros planetas.

Nenhum dos objetivos listados acima são desejos ilusórios — ao contrário: são planos práticos. Não se engane, esses planos terão de ser avaliados, adaptados e alterados a cada passo do caminho, mas eles estão em *movimento.* Em todo o mundo, interações rápidas estão acontecendo, impulsionando-nos em direção a um mundo melhor.

É isso que eu espero que você leve com este livro: o conhecimento de que é possível — que você pode mudar as coisas, que você não precisa aceitar as coisas como elas são.

> Todos os homens sonham, mas não de forma equivalente. Aqueles que sonham à noite, nos recessos empoeirados de suas mentes, acordam de dia e descobrem que tudo aquilo eram futilidades, mas os sonhadores do dia são homens perigosos, pois eles podem agir de acordo com os seus sonhos, de olhos abertos, para torná-los possíveis.
>
> — T. E. Lawrence, *Os sete pilares da sabedoria*[52]

Não ouça os cínicos que dizem a você o que não pode ser feito. Surpreenda-os com o possível.

PONTOS PRINCIPAIS

O Scrum acelera todos os empreendimentos humanos. Não importa o tipo de projeto ou de problema, o Scrum pode ser usado em qualquer empreendimento, para aprimorar desempenho e resultados.

Scrum para escolas. Na Holanda, um número crescente de professores está usando o Scrum. Eles conseguem ver um aumento quase imediato de mais de 10% nas notas dos alunos; e aplicam o método em todos os tipos de estudantes, das classes técnicas às avançadas.

Scrum pela pobreza. Em Uganda, a Grameen Foundation está usando o Scrum para fornecer dados agrícolas e de mercado para os lavradores pobres da zona rural. O resultado? O dobro da colheita e da receita para algumas das pessoas mais pobres do planeta.

Rasgue seus cartões de visita. Livre-se de todos os cargos, gerentes e estruturas. Dê às pessoas a liberdade de fazerem o que acham melhor e a responsabilidade de serem cobradas por isso. Você ficará surpreso com os resultados.

46 Montessori é um modelo educacional italiano no qual a educação se desenvolve com base na evolução do educando, e não o contrário (N.T.)

47 Escola autônoma (em inglês, *charter school*) é um tipo de escola que recebe recursos públicos, mas é administrada de forma independente. (N.T.)

48 *Beltway bandit* é um termo usado para se referir a empresas do setor privado localizadas nas proximidades de Washington, D.C. e que prestam serviços de consultoria para o governo norte-americano. A alcunha foi criada originalmente como um leve insulto, mas já perdeu a conotação pejorativa e costuma ser usada apenas de forma neutra. (N.T.)

49 Brad Shannon, McKenna, Inslee Outline Plans to Bring Efficiency to Government *The Olympian*, 6 de outubro de 2012.

50 *Valve Handbook for New Employees,* Valve Press, 2012.

51 *Ibid*.

52 T. E. Lawrence, T. E., *Seven pillars of wisdom: a triumph*, London, Cape, 1973. [Em português, *Os sete pilares da sabedoria*, Rio de Janeiro, Record, 2000.]

Apêndice

Implementando o Scrum: como começar

Agora que terminou de ler o livro, aqui está um resumo sobre a melhor forma de começar um projeto Scrum. Esta é uma descrição bem ampla do processo, mas deve ser suficiente para você começar. O livro foi escrito para explicar o porquê por trás do Scrum. Esta parte irá, de forma resumida, explicar o *como.*

1. Escolha um **Dono do Produto.** Essa pessoa é a responsável pela visão do que você vai fazer ou conseguir. Ela leva em consideração os riscos e os benefícios, o que é possível, o que pode ser feito e o que desperta a paixão na equipe. (Consulte o Capítulo 8, **Prioridades**, para obter mais informações)
2. Escolha uma **equipe**. Quem serão as pessoas que realmente trabalharão no projeto? Essa equipe precisa ter todas as habilidades necessárias para pegar a visão do Dono do Produto e transformá-la em realidade. As equipes devem ser pequenas; entre três e nove pessoas é o princípio básico. (Consulte o Capítulo 3, **Equipes**, para obter mais informações)
3. Escolha um **Mestre Scrum.** Essa pessoa vai orientar o restante da equipe em relação à estrutura do Scrum, além de ajudar a eliminar qualquer obstáculo que os esteja deixando mais lentos. (Consulte o Capítulo 4, **O Desperdício é Um Crime**, para obter mais informações)
4. Crie e priorize uma lista de **Pendências do Produto**. Trata-se de uma lista detalhada de tudo que precisa ser feito ou construído para transformar a visão em realidade. Essas Pendências existem e evoluem durante o desenvolvimento do produto; elas são o mapa dele. Em qualquer fase do projeto, são a única e definitiva visão de "tudo que precisa ser feito pela equipe a qualquer momento, em ordem de prioridade". Só existe uma lista de Pendências; isso significa que o Dono do Produto precisa tomar decisões em relação às prioridade durante todo o processo; ele deve consultar todos os *stakeholders* e a equipe para se certificar de que elas representam tanto o que as pessoas querem, quanto o que pode ser construído. (Consulte o Capítulo 8, **Prioridades**, para obter mais informações)
5. Aperfeiçoe e faça estimativas para as **Pendências do Produto.** É crucial que as pessoas que irão realmente concluir os itens da lista façam as estimativas de quanto esforço eles exigirão. A equipe deve olhar para cada item das Pendências e ver se aquilo é factível. Existem informações suficientes para conclui-lo? Ele é pequeno o suficiente para ser estimado? Existe uma definição de "Feito"? Ele cria valor visível? Cada item deve poder ser mostrado, demonstrado e, esperançosamente, ser enviado. Não estime as Pendências em horas, porque as pessoas são péssimas nesse tipo de previsão. Faça isso usando uma classificação relativa por tamanho: Pequeno, Médio ou Grande. Ou, melhor ainda, use a sequência de Fibonacci e faça estimativas de pontos para cada item: 1, 2, 3, 5, 8, 13, 21 etc. (Consulte o Capítulo

6, **Planeje a Realidade, Não a Fantasia**, para obter mais informações)

6. **Planejamento do Sprint.** Esta é a primeira das reuniões Scrum. A equipe, o Mestre Scrum e o Dono do Produto se reúnem para planejar o Sprint, que sempre tem uma duração definida de tempo menor que um mês. A maioria das pessoas define Sprints de uma ou de duas semanas. As equipes olham para as tarefas no topo das Pendências e estimam o quanto podem fazer naquele Sprint. Se a equipe já está trabalhando a alguns Sprints, ela deve pegar tarefas que totalizem o mesmo número de pontos do Sprint anterior. Esse número é conhecido como a Velocidade da equipe. O Mestre Scrum e a equipe devem tentar aumentar o número de pontos a cada Sprint. Essa é outra chance para a equipe e o Dono do Produto se certificarem que todos entendem como os itens vão satisfazer a visão. Além disso, durante essa reunião todos devem concordar com um Objetivo do Sprint.

Um dos pilares do Scrum é que, uma vez que a equipe se comprometeu com o que acredita ser capaz de fazer em um Sprint, é isso. Ele não pode ser mudado, nada pode ser acrescentado. A equipe deve trabalhar de forma autônoma durante o Sprint para concluir o que previu que conseguiu. (Consulte o Capítulo 6, **Planeje a Realidade, Não a Fantasia**, para obter mais informações.)

7. **Torne o trabalho visível.** O melhor jeito para se fazer isso no Scrum é criar um **Quadro Scrum** com três colunas: A fazer, Fazendo, Feito. *post-its* representam os itens que precisam ser concluídos e a equipe os move pelo Quadro Scrum à medida que forem concluídos, um a um.

   Outro modo de tonar o trabalho visível é criar um **Gráfico de Burn-Down**. Um eixo é o número de pontos que a equipe definiu para o Sprint, e o outro é o número de dias. Todos os dias, o Mestre Scrum soma o número de pontos concluídos e os marca no gráfico. O ideal é que haja uma ladeira descendo pelo gráfico até chegar ao zero no último dia do Sprint. (Consulte o Capítulo 7, **Felicidade**, para obter mais informações)

8. **Reuniões Diárias** ou **Scrum Diário.** Este é o ritmo do Scrum. Todos os dias, no mesmo horário, durante não mais do que 15 minutos, a equipe e o Mestre Scrum se reúnem para responder a três perguntas:

   • O que você fez ontem para ajudar a equipe a concluir o Sprint?
   • O que você vai fazer hoje para ajudar a equipe a concluir o Sprint?
   • Existe algum obstáculo impedindo você ou a equipe de alcançar o objetivo do Sprint?

   Isso é tudo. A reunião inteira. Se ela levar mais do que 15 minutos, você está fazendo alguma coisa errada. Isso serve para ajudar a equipe inteira a saber exatamente em que ponto estão no Sprint. Todas as tarefas serão concluídas a tempo? Existem oportunidades para ajudar os outros membros da equipe a superarem os obstáculos? Não há designação de tarefas vindas de cima — a equipe é autônoma; são eles que fazem isso. Não há qualquer relatório detalhado para os gestores. O Mestre Scrum é responsável por resolver qualquer obstáculo ou impedimento para o progresso da equipe. (Consulte o Capítulo 4, Tempo, e o Capítulo 6, **Planeje a Realidade, Não a fantasia,** para obter mais informações)

9. **Revisão** ou **Demonstração do Sprint.** Trata-se da reunião na qual a equipe mostra o que conseguiu fazer durante o Sprint. Qualquer pessoa pode participar, não apenas o Dono do Produto, o Mestre Scrum e a equipe, mas também os *stakeholders*, os gestores, os clientes, e qualquer outra pessoa. Esta é uma reunião aberta na qual a equipe demonstra o que conseguiu colocar na coluna Feito.

   A equipe só deve demonstrar o que satisfaz a Definição de Feito. O que está total e completamente

concluído e pode ser entregue sem qualquer trabalho adicional. Pode não ser o produto completo, mas deve ser um atributo concluído do produto. (Consulte o Capítulo 4, **Tempo**, para obter mais informações)

10. **Retrospectiva do Sprint.** Depois que a equipe mostrou o que conseguiu fazer no Sprint anterior — aquilo que está "Feito" e pode ser entregue para clientes para obtenção de feedback —, eles se reúnem e pensam no que deu certo e o que poderia ter sido melhor, e o que podem melhorar no próximo Sprint. Qual é o aprimoramento no processo que eles, como uma equipe, podem implementar de forma imediata?

    Para ser eficaz, essa reunião requer certa dose de maturidade emocional e atmosfera de confiança. O importante é lembrar-se sempre de que você não está procurando culpados; está olhando para o processo. Por que aquilo aconteceu assim? Por que você não percebeu aquilo? O que poderia ter acontecido para sermos mais ágeis? É essencial que as pessoas na equipe assumam a responsabilidade pelo processo e seus respectivos resultados, e que busquem soluções como uma equipe. Ao mesmo tempo, elas têm de ter coragem de levantar as questões que realmente as incomodam, de forma que a solução seja orientada, em vez de acusadora. E o restante da equipe precisa ter maturidade para ouvir o feedback, absorvê-lo e procurar uma solução, em vez de assumir uma postura defensiva.

    No final da reunião, a equipe e o Mestre Scrum devem chegar a um acordo sobre um aprimoramento no processo que será implementado no Sprint seguinte. Tal aprimoramento no processo, às vezes, é chamado *kaizen*, e deve ser colocado nas pendências do próximo Sprint, acompanhado de testes de aceitação. Desse modo, será fácil para a equipe verificar se o aprimoramento realmente foi implementado, e que efeito ele teve sobre a velocidade. (Consulte o Capítulo 7, **Felicidade**, para obter mais informações)

11. Comece imediatamente o próximo Sprint, considerando a experiência da equipe com os impedimentos e os aprimoramentos no processo.

Agradecimentos

Qualquer projeto não é o resultado do trabalho de uma única pessoa; é o produto do trabalho de uma equipe, e este livro não é exceção.

Primeiro, gostaria de agradecer ao meu filho, J. J. Sutherland. Ele sugeriu que escrevêssemos um livro juntos sobre a jornada realmente extraordinária para a qual o Scrum me levou há alguns anos. Ele queria uma pausa na década de viagens partindo de uma guerra ou desastre para outro no seu trabalho para a NPR, e achou que a história sobre como o Scrum foi criado — por que funciona e como mudou o mundo — não era apenas importante de ser contada, mas também seria divertido fazê-lo. O livro que está em suas mãos, embora seja a minha história, é o resultado de muitas horas juntos, mas foi ele quem botou as palavras no papel.

Howard Yoon, o mais experiente agente literário, nos pediu para pensarmos em um projeto maior, mais amplo e mais completo. Seus conselhos, sabedoria e simples *know-how* não apenas tornaram este livro possível, como o alavancou para uma escala bem diferente.

Não é sempre que temos a chance de trabalhar com um verdadeiro mestre em sua arte, e eu tive essa inacreditável sorte com Rick Horgan, do Crown Publishing Group. Seu toque hábil e perfeito apenas *melhorou* o trabalho. E ele faz isso parecer tão *fácil*. Tiro o meu chapéu e agradeço de verdade.

O Dono do Produto, Alex Brown, Joe Justice, e o restante da equipe na Scrum, Inc. compartilharam ideias críticas, energia sem limites e profunda experiência.

Também gostaria de agradecer:

Aos professores Hirotaka Takeuchi e Ikujiro Nonaka, cujo trabalho foi a centelha da ideia do Scrum, e que hoje são bons amigos.

Ao meu amigo e cocriador, Ken Schwaber, cuja teimosia ajudou a formar o Scrum e torná-lo a força que é hoje.

Acima de tudo, à minha mulher, Arline. Ela esteve comigo do início ao fim e, como uma ministra do Unitário-Universalismo, introduziu o Scrum em muitas igrejas. Ela tornou o mundo um lugar melhor quando nos mostrou como aplicar

o Scrum em uma organização inteira.

E, por fim, gostaria de deixar o meu muito obrigado para os inúmeros Mestres Scrum, Donos de Produto e Equipes em todo o mundo que realmente vivem o Scrum todos os dias. Vocês o tornam ainda mais uma força vibrante e positiva, e nunca deixam de me surpreender com o que conseguem usando esse método.