

Jez Humble, Joanne Molesky & Barry O'Reilly

# LEAN ENTERPRISE

Como empresas  
de alta performance  
inovam em escala



Casa do  
Código

O'REILLY®

**Título original:** *Lean Enterprise*

Authorized Portuguese translation of the English edition of *Lean Enterprise*,  
ISBN 9781449368425

Copyright © 2015 Jez Humble, Joanne Molesky e Barry O'Reilly.

© 2016 Editora Casa do Código

Todos os direitos reservados.

Impresso no Brasil

Publicado pela Editora Casa do Código  
Rua Vergueiro, 2135 — 8º andar  
04101-300 — Vila Mariana — São Paulo — SP  
[www.casadocodigo.com.br](http://www.casadocodigo.com.br)

**Editores:** Mary Treseler e Angela Rufino

**Coordenador de produção:** Kara Ebrahim

**Preparação de texto:** Dmitry Kirsanov

**Revisão:** Alina Kirsanova

**Índice remissivo:** Dmitry Kirsanov e Alina Kirsanova

**Editoração:** David Futato

**Capa:** Ellie Volckhausen

**Ilustrações:** Rebecca Demarest e Peter Staples

**Tradução:** Gabriel Brigidi, Gabriel Gavasso, Leonardo Campos, Lourenço Soares,  
Olivia Janequine e Paulo Caroli

O logo da O'Reilly é uma marca registrada da O'Reilly Media, Inc. *Lean Enterprise*, a imagem da capa e qualquer comércio relacionado são marcas registradas da O'Reilly Media, Inc.

Embora a editora e o autor tenham se esforçado para assegurar que as informações e instruções contidas neste livro sejam precisas, a editora e o autor não se responsabilizam por qualquer erro ou omissão, incluindo quaisquer danos resultados do uso e confiança deste livro. O uso das informações e instruções presentes aqui é de seu risco. Se quaisquer exemplos de códigos ou outras tecnologias que este livro contém ou descreve estão sujeitos a licenças open source ou a direitos de propriedade intelectual de outros, é de sua responsabilidade garantir que o seu uso obedeça tais licenças e/ou direitos.

# ISBN

Impresso e PDF: 978-85-5519-207-4

EPUB: 978-85-5519-208-1

MOBI: 978-85-5519-209-8

Você pode discutir sobre este livro no Fórum da Casa do Código: <http://forum.casadocodigo.com.br/>.

Caso você deseje submeter alguma errata ou sugestão, acesse <http://erratas.casadocodigo.com.br>.

# AUTORES

## **Jez Humble**

Jez Humble é coautor do premiado livro *Continuous Delivery*. Começou sua carreira em uma startup, e trabalhou na ThoughtWorks por dez anos construindo produtos. Fez consultoria em mais de 500 empresas globais para ajudá-las a alcançar excelência técnica no desenvolvimento de software, e implantar uma cultura de experimentação e aprendizagem. Ele trabalha na 18F e leciona na UC Berkeley.

## **Joanne Molesky**

Joanne Molesky é um membro-chave da consultoria da ThoughtWorks, com responsabilidades internas em riscos e compliance, e também presta serviços a clientes na área de entrega contínua e melhoria do processo. Sua paixão está em encontrar formas de reduzir riscos relacionados à entrega de software, sem comprometer a criatividade, a experimentação e a aprendizagem. Possui certificações CISA e CRISC pela ISACA.

## **Barry O'Reilly**

Barry O'Reilly trabalha na ThoughtWorks, prestando consultoria a empresas líderes globais em melhoria contínua, usando práticas e princípios Ágeis e Lean. Sendo um empreendedor e consultor, depois de várias startups, ganhou foco em empresas onde pôde explorar a interseção entre inovação do modelo de negócio, desenvolvimento de produto, design organizacional e transformação na cultura empresarial. É fundador e CEO da ExecCamp, cuja missão é ajudar negócios de TI a inovar em escala.

# O QUE DIZEM SOBRE LEAN ENTERPRISE

"Este livro está reorganizando a corporação para a era digital. Está destinado a ser a referência clássica e oficial sobre como empresas planejam, organizam, implementam e medem seu trabalho. *Lean Enterprise* descreve como empresas podem vencer no mercado enquanto aproveitam e desenvolvem as capacidades dos funcionários. Qualquer líder de negócio que se preocupe com criar vantagem competitiva por meio da tecnologia e construir uma cultura de inovação precisa ler este livro." — **Gene Kim, coautor de *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*, fundador e ex-CTO da Tripwire, Inc.**

"Este livro é uma dádiva para qualquer pessoa que tentou mudar sua empresa e ouviu "Isso é OK para quem é pequeno, mas somos muito grandes/regulados/complexos para trabalhar assim aqui." Humble, Molesky e O'Reilly escreveram um guia de fácil leitura que desmistifica o sucesso de empresas Lean, de modo que todo mundo possa entender e aplicar. *Lean Enterprise* fornece uma caixa de ferramentas pragmática com estratégias e práticas para estabelecer empresas de alta performance. Deveria ser uma leitura obrigatória para todo executivo que entenda que todos estamos no negócio da tecnologia agora." — **Stephen Foreshew-Cain, COO, Serviço Digital do Governo do Reino Unido**

"Para prosperar no mundo digital, a transformação deve ser mais que voltada à tecnologia — todo mundo dentro da empresa deve trabalhar coletivamente para se adaptar. Este livro fornece um guia essencial para todos os líderes poderem mudar a forma como entregam valor aos consumidores." — **Matt Pancino, CEO da Suncorp Business Services**

"Este é o livro pelo qual eu estive esperando — aquele que trata das questões mais complicadas sobre trazer a abordagem Lean para a empresa. Os autores fornecem soluções que são valiosas mesmo em ambientes de baixa confiança." — **Mark A. Schwartz**  
(@schwartz-cio)

"Em uma narrativa cativante, este livro integra os melhores pensamentos atuais sobre como criar ótimos softwares - produtos e serviços intensos. A abordagem deste livro é tanto desafiadora quanto disciplinada, e algumas empresas nem ao menos poderão imaginar seguir este caminho. Mas aqueles que seguirem a jornada acharão impossível pensar em voltar atrás — e se eles forem uma concorrência, estarão bem posicionados para tomar tanto seu mercado como suas pessoas. Ignore este livro em seu próprio risco."  
— **Mary Poppendieck, coautora de *The Lean Mindset* e da série *The Lean Software Development***

"Meu trabalho é ajudar as pessoas a praticar um padrão científico, ajudar a remodelar seus pensamentos e hábitos de trabalho, tanto nos negócios, na política, na educação, quanto no dia a dia. O século 21 está cada vez mais exigindo um modo de trabalho cognitivamente complexo, interpessoal, iterativo, e até empreendedora. Com *Lean Enterprise*, Jez Humble, Joanne Molesky e Barry O'Reilly explicam como software pode e está liderando o caminho para transformar nossos modos de trabalho, o que pode mudar nosso modo de pensar e nos ajudar a adaptar ao mundo emergente à nossa volta." — **Mike Rother, autor de *Toyota Kata***

"Quase todas as indústrias e instituições estão quebrando devido ao rápido avanço da tecnologia, guiado pela visão inspirada de times e indivíduos. Este livro claramente explica como as disciplinas de Lean, Agile, Kata, Startup Enxuta, e Design Thinking convergem por meio da união de princípios de uma empresa de aprendizado adaptativo." — **Steve Bell, membro do Lean Enterprise Institute,**

## **autor de *Lean IT* e *Run Grow Transform***

"Construir software da maneira certa é uma tarefa desafiadora por si só, mas *Lean Enterprise* vai além das considerações sobre tecnologia para guiar empresas a rapidamente construir o software certo para entregar os resultados de negócio esperados e com pouco risco. É uma leitura obrigatória para qualquer empresa que fornece software baseado em serviços para seus consumidores." — **Gary Gruver, VP de Release, QE, e Operações na Macy's.com**

"Para competir no futuro, as empresas precisam ser hábeis em entender seus clientes e levar os aprendizados validados ao mercado o mais rápido possível. Isso requer um novo tipo de empresa adaptativa e em constante aprendizagem — a empresa Lean. A jornada começa aqui neste livro!" — **John Crosby, Chefe de Produtos e Tecnologia na lastminute.com**

"Rápidos avanços na tecnologia estão criando taxas nunca vistas de disruptão. As regras do jogo de disruptão mudaram, e muitas empresas estão pensando em como competir conforme novos gigantes emergem com uma abordagem diferente para servir a seus clientes. Este livro fornece um guia essencial para aqueles que perceberam que é necessário mudar para reconquistar uma vantagem competitiva inovadora, mas estão inseguros sobre como começar." — **Jora Gill, Chief Digital Officer na The Economist**

"*Lean Enterprise* foi o livro que dei ao meu time de liderança para que todos estivessem com a mesma visão sobre como podemos desafiar o *status quo*, remover obstáculos e inovar nossa competição. Pela alavancagem contínua de insights que temos ao cocriar com os consumidores, nossas pessoas e dados, agora nós temos muitas maneiras adicionais novas para fazer nosso negócio crescer." — **Don Meij, CEO na Domino's Pizza Enterprises Ltd.**

"Enquanto métodos Ágeis e Lean têm um grande impacto na

entrega de software, o real potencial deles só aparece quando eles têm um impacto maior em empresas de todos os tamanhos. Neste livro, Jez, Joanne e Barry apresentam como são essas mudanças — uma visão realista de como as empresas do futuro farão as de hoje se parecerem com tocadores de fita cassete." — **Martin Fowler, Diretor Científico na ThoughtWorks**

"Este é um livro importante. Ele dá um olhar estudado e informativo sobre os fundamentos que precisam mudar para começar a criar organizações capazes de aprendizagem e melhoria contínuas. Ele vai muito além do técnico para o organizacional. *Lean Enterprise* é uma leitura obrigatória para líderes existentes e emergentes, que buscam assegurar que o sucesso de sua empresa se mantenha." — **Jeff Gothelf, autor de *Lean UX* e Diretor da Neo Innovation.**

"Eu estive falando que todos deviam comprar este livro um ano antes de ele estar finalizado. Ele documenta o trajeto de empresas Lean em liderança, e os retardatários serão varridos pelas empresas Lean nos próximos anos." — **Adrian Cockcroft (@adrianco)**

# PREFÁCIO

*“O software está dominando o mundo.” — Marc Andreessen*

*“Em uma companhia industrial, evite o software por sua conta e risco... uma empresa de software poderia acabar com a GE um dia, e é melhor ficarmos paranoicos com isso.” — Jeff Immelt*

*“Você é um bobo se faz apenas o que eu digo. Você é mais bobo ainda se você não faz o que eu digo. Você deve pensar por si mesmo e ter ideias melhores que as minhas.” — Taiichi Ohno*

Neste livro, mostraremos como fazer crescerem organizações que podem inovar rapidamente em resposta a mudanças nas condições de mercado, necessidades do consumidor e tecnologias emergentes.

As empresas vivem e morrem em sua capacidade de descobrir novos negócios e de criar valor contínuo para seus clientes. Isso sempre foi verdade, mas não mais do que nos últimos anos. A pressão competitiva é crescente, abastecida por mudanças rápidas na tecnologia e na sociedade. Como mostra o Shift Index da Deloitte, a expectativa de vida média de uma empresa na lista Fortune 500 caiu de 75 anos há meio século para 15 anos hoje. O professor Richard Foster, da Universidade de Yale, estima que: “*em 2020, mais de três quartos da lista S&P 500 será de empresas das quais ainda não ouvimos falar*” (GITTLESON, 2012). A sobrevivência a longo prazo de qualquer empresa depende da sua habilidade de entender e utilizar as forças técnicas e culturais que continuam a acelerar os ciclos de inovação.

Primeiro, a internet e as mídias sociais geraram consumidores com ferramentas poderosas para informar as decisões que eles tomam. Essas ferramentas também dão para as organizações modernas novas maneiras de descobrir e se engajar com usuários e consumidores. Prosperam as empresas que usam o design thinking e experiência do usuário (UX) estrategicamente para encantar consumidores a cada passo de sua interação: pesquisas mostram que empresas que aplicam UX dessa maneira experimentam crescimento mais rápido e maiores receitas (DANISH DESIGN CENTER, 2006).

Segundo, avanços na tecnologia e no processo tornaram possível construir, evoluir e escalar produtos e serviços disruptivos rapidamente e com pequeno investimento de capital. Times pequenos pelo mundo fazem protótipos de novos produtos de software em dias ou semanas, usando serviços e infraestrutura baratos ou gratuitos, e depois desenvolvem rapidamente aqueles que ganham tração.

Em um futuro próximo, a ubiquidade de dispositivos baratos e poderosos incorporados em rede nos permitirão fazer protótipos e desenvolver uma variedade de produtos maior, de maneira mais barata e em ciclos curtos de maneira similar. À medida que as impressoras 3D se tornam mais baratas e rápidas, e começam a trabalhar com uma variedade maior de materiais, criaremos e entregaremos uma enorme gama de produtos customizados sob demanda.

O software tem três características que tornam capaz esse tipo de inovação rápida. Primeiro, é relativamente barato fazer um protótipo e desenvolver ideias no software. Segundo, podemos usar tais protótipos desde o estágio inicial. Finalmente, no curso da criação desses protótipos, podemos descobrir muito sobre o que consumidores acham valioso e incorporar isso em nosso projeto —

acelerando nossa taxa de teste de novas ideias com usuários, coletando feedback e usando-o para melhorar nossos produtos e negócios.

Enquanto isso, a incansável marcha da miniaturização (personificada na Lei de Moore) tem permitido que computadores incrivelmente poderosos se tornem minúsculos e encontrem seu caminho em tudo, com o software no centro de tudo. Em um artigo da Forbes, intitulado “*Now Every Company Is A Software Company*” (Agora toda empresa é uma empresa de software), David Zanca, vice-presidente sênior de Tecnologia da Informação na FedEx, se descreve como dirigindo uma “empresa de software dentro da FedEx”. Venkatesh Prasad, líder técnico sênior na Ford, descreve sua empresa como uma fabricante de “sofisticados computadores sobre rodas”. Ben Wood, da CCS Insight, diz que a Nokia “passou por sua incrível década de inovação em hardware, mas o que a Apple viu foi que tudo o que você precisava era de um retângulo com uma tela e o resto ficava por conta do software” (LEE, 2013) - a nosso ver, esse é o insight-chave por trás da aquisição da Nokia pela Microsoft.

#### **LEI DE MOORE**

Em 1965, Gordon Moore, cofundador da Intel, previu que a densidade de circuitos integrados dobraria a cada dois anos, aproximadamente.

Como resultado dessa mudança no pensamento sobre software, as empresas, incluindo as pioneiras em terceirização de TI, como GE e GM, estão trazendo o desenvolvimento de software para dentro de casa. Como discutiremos no *Capítulo Fique onde está*, o governo do Reino Unido tem feito o que todo mundo fez. Como foi

reportado pela *The Economist* (2013):

*"Os motivos da GM para fazer isso podem se aplicar a muitas outras empresas. 'A TI se tornou mais difundida em nosso negócio, e agora a consideramos uma grande fonte de vantagem competitiva', diz Randy Mott, diretor de informação da GM, que tem sido responsável pela mudança na estratégia de terceirização. Enquanto o trabalho estava sendo feito por pessoas de fora, ele disse que a maioria dos recursos que a GM estava dedicando à TI era gasta mantendo as coisas funcionando como sempre em vez de pensar em novas maneiras de fazê-las. A empresa avalia que ter seu trabalho de TI feito quase todo dentro de casa dará mais flexibilidade e velocidade, além de encorajar mais inovação".*

O mundo dos negócios está mudando, passando a tratar a TI como uma utilidade que melhora as operações internas para usar software veloz e ciclos de inovação movidos a tecnologia como uma vantagem competitiva. Isso tem consequências no longo prazo. Os modelos tradicionais de programa e gerenciamento de projeto que usamos para TI não combinam com ciclos rápidos de inovação. Contudo, eles estão profundamente incorporados na maneira como gerenciamos tudo, de operações e serviço ao consumidor a orçamento, governança e estratégia.

Nos últimos 10 anos, apareceram os elementos de um paradigma adequado, centrado no produto, que funcione em larga escala, mas não foram ainda conectados e apresentados de maneira sistemática. Este livro quer preencher essa lacuna, fornecendo inspiração de organizações que adotaram essas ideias com sucesso. E, mais importante, fizemos uma detalhada investigação pela cultura de alta performance, que é um fator crítico para tornar capaz a inovação rápida em larga escala.

## **Por que escrevemos este livro?**

Todos os autores têm experiência de trabalho em grandes empresas e startups, e definimos que apresentaríamos uma abordagem pragmática e sistemática sobre inovação e transformação que funcionasse de maneira eficaz no contexto empresarial. Abordamos não apenas como organizações de alta performance desenvolvem produtos, mas como companhias que estão buscando a alta performance podem adotar essas técnicas de maneira incremental, iterativa e com baixo risco.

Escrevemos o livro por causa de nossa frustração com o estado da indústria. As técnicas e práticas que descrevemos não são novas e sabe-se que elas funcionam. Contudo, elas não são ainda *mainstream* e, frequentemente, são implementadas de maneira fragmentada, levando a melhorias locais em vez de melhorias sistêmicas. Como resultado, as empresas trabalham duramente — a um alto custo — construindo produtos, serviços e negócios que não entregam o valor esperado para os clientes.

Quando os livros *Continuous Delivery* (HUMBLE; FARLEY, 2010) e *The Lean Startup* (RIES, 2011) foram publicados, vimos uma demanda enorme de pessoas trabalhando em empresas que queriam adotar as práticas descritas nesses livros. Um grande número de empresas alcançou um benefício mensurável ao usar as práticas das quais falamos, resultando em uma entrega mais rápida para o mercado de produtos de alta qualidade, aumento da satisfação do consumidor e maiores retornos sobre o investimento. Isso com custo e risco reduzido, assim como funcionários mais felizes que não trabalham mais em horários insustentáveis e têm a oportunidade de usar sua criatividade e paixão no trabalho.

Contudo, todos acham difícil implementar essas ideias com sucesso. Na maioria dos casos, era impossível perceber mais do que melhorias incrementais, porque apenas parte da organização havia mudado — e aquela parte ainda precisava trabalhar com o resto da

organização, que esperava que eles se comportassem do jeito tradicional. Dessa maneira, descrevemos como empresas bem-sucedidas repensaram tudo, do gerenciamento financeiro e governança ao risco e *compliance*, a sistemas de arquitetura, ao programa, portfólio e gerenciamento de requisitos, na busca por uma performance radicalmente melhorada.

Este livro apresenta um conjunto de padrões e princípios projetados para lhe ajudar a implementar essas ideias. Acreditamos que toda organização é diferente e que teremos diferentes necessidades, então não fornecemos regras sobre como implementar essas práticas. Em vez disso, descrevemos uma abordagem heurística da implementação que enfatiza a importância da experimentação para aprender como sua organização pode adotar essas ideias e, assim, melhorar. Essa abordagem leva mais tempo, mas possui vantagens em mostrar benefícios mensuráveis mais rápido e reduzir o risco de mudanças. Ela também permite que sua organização e as pessoas nela aprendam por si mesmas o que funciona melhor.

Esperamos que você ache este livro valioso. A atitude mais perigosa seria: “*Estas são boas ideias, mas elas podem não funcionar na nossa organização*”, como disse Taiichi Ohno, o pai do Sistema de Produção Toyota (OHNO, 2012):

“*Seja na alta gerência, média gerência ou os trabalhadores que realmente colocam a mão na massa, todos somos humanos, então somos como equívocos ambulantes, acreditando que a maneira que fazemos as coisas é a melhor maneira. Ou, talvez, você não ache que é a melhor maneira, mas você está trabalhando dentro do senso comum de que ‘não dá para evitar, é como as coisas são’*”.

Você enfrentará obstáculos ao adotar as ideias neste livro. Quando você ler os estudos de caso, provavelmente verá motivos pelos quais a abordagem descrita pode não funcionar na sua

organização. Não transforme obstáculos em objeções. Trate o que você lê como uma inspiração para seus esforços, não como receitas a serem seguidas ao pé da letra. Procure constantemente por obstáculos, e trate-os como oportunidades de experimentar e aprender. Para citar Ohno (2012) novamente:

*“As oportunidades de kaizen (melhoria) são infinitas. Não pense que você fez as coisas melhores que antes e pode ficar satisfeito com isso... Isso seria como o estudante que fica orgulhoso porque superou seu mestre duas vezes em três tentativas na esgrima. Uma vez que você entendeu o cerne das ideias kaizen, é importante ter a atitude em seu trabalho diário de que logo abaixo de uma ideia kaizen há outra”.*

As oportunidades para melhorar estão em todo lugar — não apenas nos produtos ou serviços que construímos, mas na maneira como nos comportamos e interagimos, e mais importante, na maneira como pensamos.

## **Quem deve ler este livro?**

Escrevemos este livro primeiramente para líderes e gerentes. O livro foca nos princípios e padrões que podem ser aplicados em qualquer domínio, em qualquer tipo de organização.

Nosso público pretendido inclui:

- Executivos interessados em estratégia, liderança, cultura organizacional e boa governança;
- Diretores de TI, tanto em aplicações como infraestrutura e operações;
- Qualquer pessoa trabalhando em gerenciamento de programa ou projeto, incluindo membros do PMO;

- Pessoas em finanças ou contabilidade ou em governança, regulação e compliance que estão envolvidas na entrega;
- CMOs, gerentes de produto e outros envolvidos em projetar produtos e serviços que envolvam desenvolvimento de software.

Qualquer pessoa trabalhando em times de entrega deve também achar este livro valioso — mas não espere qualquer discussão aprofundada sobre práticas de engenharia, como, por exemplo, como escrever testes de aceitação funcionais sustentáveis, implementação automatizada ou gerenciar configuração. Esses tópicos são discutidos em profundidade no livro *Continuous Delivery*.

Este livro é particularmente dirigido a pessoas que trabalham em organizações médias e grandes e percebem que precisam pensar diferente sobre estratégia, cultura, governança e o modo como gerenciam produtos e serviços para que possam obter sucesso. Isso sem contar que organizações menores não considerarão o livro útil — apenas porque o material pode não ser aplicável para elas no estágio em que se encontram.

Um de nossos objetivos era manter o livro relativamente curto, conciso e prático. Para fazer isso, decidimos não gastar muito tempo discutindo modelos teóricos que guiam os princípios e práticas que descrevemos. Em vez disso, apresentamos alguns princípios fundamentais destes campos para que você possa entender os fundamentos teóricos básicos; então, descrevemos as aplicações práticas dessas teorias. Também fornecemos referências para leituras futuras, para aqueles que se interessarem.

Nós também fomos cuidadosos em não oferecer direcionamento detalhado sobre quais ferramentas de software usar e como usá-las,

por duas razões. Primeiro, achamos que a escolha da ferramenta não é, na verdade, uma decisão tremendamente importante (desde que você evite as más escolhas). Muitas organizações que mudam para metodologias ágeis gastam muito tempo na escolha da ferramenta, com a esperança que ela vá magicamente resolver seus problemas. Mas a maneira mais comum de se falhar para tais organizações é sua incapacidade de mudar sua cultura organizacional, não suas boas ferramentas disponíveis.

Em segundo lugar, a informação sobre ferramentas e processos específicos rapidamente fica desatualizada. Há muitas boas ferramentas (incluindo muitas que são open source) e literatura sobre como usá-las. Neste livro, focamos em estratégias para ajudar sua organização a obter sucesso, não importando quais ferramentas você escolha.

## Sinopse

A *Parte I* do livro apresenta os seus principais temas: cultura, estratégia e o ciclo da vida das inovações. Na *Parte II*, discutimos como investigar novas ideias para reunir dados, para que você rapidamente avalie quais lhe fornecerão valor ou para que tenha um entendimento suficientemente rápido. A *Parte III* cobre como explorar ideias validadas em grande escala — aquelas que emergem da investigação —, e também apresenta uma abordagem sistemática para melhorar a maneira como executamos grandes programas de trabalho. Ao final, a *Parte IV* mostra como empresas podem criar um ambiente que encoraje o aprendizado e a experimentação, com foco em cultura, governança, gerenciamento financeiro, TI e estratégia.

Todos devem ler a *Parte I*. Os leitores devem então se sentir livres para mergulhar nos capítulos que os interessam. Contudo, é válido ler o *capítulo Modele e meça o risco do investimento*, o

*capítulo Implemente entrega contínua e o capítulo Identifique valor e amplie o fluxo* antes de proceder para a *Parte IV*, já que ela se baseia nos conceitos apresentados nesses capítulos.

## AGRADECIMENTOS

Muitas pessoas contribuíram para este livro. Particularmente, somos profundamente agradecidos às seguintes pessoas, que nos deram reviews detalhadas dos primeiros rascunhos ou de alguns capítulos individualmente (em ordem alfabética): Adrian Cockcroft, Amy McLeod, Andy Pittaway, Bas Vodde, Ben Williams, Bjarte Bogsnes, Brett Ansley, Carmen Cook, Charles Betz, Chris Cheshire, Courtney Hemphill, Dan North, Darius Kumana, David Tuck, Don Reinertsen, Gary Gruver, Gene Kim, Ian Carroll, James Cook, Jean-Marc Domaingue, Jeff Gothelf, Jeff Patton, Jim Highsmith, Joe Zenevitch, John Allspaw, John Crosby, Jonathan Thoms, Josh Seiden, Kevin Behr, Kief Morris, Kraig Parkinson, Lane Halley, Lee Nicholls, Lindsay Ratcliffe, Luke Barrett, Marc Hofer, Marcin Floryan, Martin Fowler, Matt Pancino, Michael Orzen, Mike Rother, Pat Kua, Randy Shoup, Ranjan Sakalley, Salim Virani, Steve Bell, Tom Barker, Tristan Kromer, and Will Edelmuth. Muito obrigado.

As ideias que apresentamos vieram de várias fontes, e foram semeadas e refinadas por meio de inumeráveis workshops, conversas e discussões com pessoas trabalhando em uma enorme variedade de organizações pelo mundo. Obrigado a todos vocês que participaram dessas discussões e nos deram o benefício de suas experiências e seu feedback.

Gostaríamos de agradecer especialmente ao nosso fabuloso time editorial e de produção na O'Reilly: Mary Treseler, Angela Rufino, Allyson MacDonald, Kara Ebrahim e Dan Fauxsmith. Agradecimento especial também para Peter Staples, por criar quase todos os incríveis infográficos do livro. Steve Bell, John Kordyback, Scott Buckley e Gareth Rushgrove forneceram estudos de caso para

este livro: muito obrigado por suas contribuições e insights. Finalmente, Dmitry Kirsanov e Aline Kirsanova, que, minuciosamente, detalhadamente e com muita qualidade, editaram, revisaram e indexaram o livro — obrigado.

Jez começou a trabalhar neste livro como uma desculpa para ficar em casa depois que sua segunda filha, Reshma, nasceu. Reshma e sua irmã, Amrita, haviam ensinado a ele a alegria da disruptão completa ao fazerem pegadinhas e cocriar muitas novas aventuras que provocavam tanto novos insights como risadas incontroláveis. Rani, sua linda e brilhante esposa, manteve tudo calmo mesmo quando tudo parecia incessante, e por isso ele tem por ela infinita gratidão, amor e admiração. Ele agradece à sua mãe pelo encorajamento e suporte, especialmente quando ele tinha de escrever enquanto ela o visitava. Jez gostaria de agradecer a seus coautores Joanne e Barry por moderarem suas tendências de comando e controle, e fazer deste livro um verdadeiro exercício de colaboração. Teria sido um livro muito diferente — e muito pobre — sem vocês. Ele gostaria de agradecer a seus colegas do Cehf por darem inspiração e apoio, e por viverem o sonho e fomentarem a alegria na busca por um produto e experiência do consumidor de primeira classe. Ele também quer agradecer à empresa em que trabalhou anteriormente, a ThoughtWorks, por fornecer uma casa única e atenciosa para inovadores e experimentadores, cujas ideias povoam estas páginas. Finalmente, agradecimento especial para Chris Murphy, Chad Wathington, David Rice, Cyndi Mitchell, Barry Crist e Adam Jacob por apoiarem este livro.

Joanne não havia entendido direito com o que ela havia concordado quando Jez Humble e Martin Fowler a convenceram a colaborar com um livro sobre os próximos passos para Entrega Contínua. Conforme o tempo passou (mais de dois anos e meio) e o livro evoluiu para o que é hoje, muitas pessoas deram apoio, encorajamento e confiaram totalmente na capacidade dela para

terminar este trabalho. John, o marido de Joanne, parceiro e melhor amigo, deu coragem e compreensão sem fim durante aqueles fins de semana e noites povoados pela culpa, quando “o livro” a distraía das diversões. Seus colegas e o time de liderança na ThoughtWorks forneceram tudo o que ela precisava para pesquisar e escrever este trabalho, especialmente David Whalley, Chris Murphy e o time de liderança da ThoughtWorks Austrália, que a contratou, pois entenderam como é importante algo tão comando e controle como segurança, risco e compliance para se encaixar nas práticas ágeis e de entrega enxuta. Por último, mas não menos importante, ela gostaria de agradecer a seus coautores e amigos Barry e Jez, que a ensinaram sobre perseverança, colaboração e confiar uns nos outros.

Barry não teria escrito este livro sem Qiu Yi, sua editora, parceira e esposa. Sua paixão, persistência e paciência suavizam suas arestas. Sua compaixão não tem fim. Seus pais, Niall e Joan, sempre acreditaram nele, dando suporte e fazendo sacrifícios para que ele atingisse seus objetivos. Ele não poderia querer melhores exemplos; seus princípios e valores moldaram o que ele é hoje, e por isso ele é grato. Ele sente saudade de seus irmãos e irmãs. O tempo que passam juntos é sempre precioso e muito curto. Sua família toda está em seu coração e em seus pensamentos. Ele foi inspirado por muitos amigos, colegas e contadores de histórias em sua vida e carreira; suas conversas, lições e conhecimento estão capturados aqui. Obrigado por expô-lo a isso. Quando ele escreveu seu primeiro blog e clicou em publicar, nunca imaginou que o resultado o traria até aqui. O encorajamento, colaboração e o calibre de Jez e Joanne o ensinaram muito mais do que transformar ideias em palavras — ele cresceu com sua orientação.

# SOBRE OS TRADUTORES

Assim como os autores, os tradutores têm experiência de trabalho em grandes empresas e startups, e também se sentem frustrados com o estado atual da indústria. Por isso, decidiram traduzir este livro, com o intuito de ajudar a compartilhar conhecimento em sua língua materna, levando tal conteúdo aos leitores da língua portuguesa.

Segue um breve parágrafo sobre cada tradutor.

## **Gabriel Brigidi**

**Gabriel Brigidi** é um entusiasta de projetos inovadores. Gosta de transformar ideias em negócios de sucesso, utilizando tecnologia como disruptão e geração de vantagem competitiva. Um generalista curioso, atua nas intersecções: estratégia e execução, teoria e prática, negócio e tecnologia. Atualmente, é responsável por descobrir e explorar oportunidades emergentes, estabelecendo modelos de negócio que criem novos valores para os clientes e novos vetores de crescimento para a ThoughtWorks.

## **Gabriel Gavasso**

**Gabriel Gavasso** trabalha na ThoughtWorks e tem mais de 10 anos facilitando a comunicação entre o negócio e times de desenvolvimento de software. Como este livro mesmo fala, esta divisão hoje existe de forma muito tênue. Hoje Gavasso passa grande parte do seu tempo potencializando times para alcançarem melhores resultados de negócio através do uso da tecnologia. Tem experiência morando e trabalhando no Brasil, Chile, Cingapura e Estados Unidos.

## **Leonardo Campos**

**Leonardo Campos** é consultor *lead* da ThoughtWorks Brasil com foco e experiência em gestão de projetos ágeis e transformações digitais. É um dos fundadores do Lean Coffee São Paulo e do blog sobre agilidade Kudoos (<http://kudoos.com.br>). Foi tradutor de diversos artigos e minilivros da InfoQ Brasil.

## **Lourenço Soares**

**Lourenço Soares** acredita que software pode ser usado para fazer um mundo melhor. Ele vem tentando fazer isso através de Entrega de Produtos e Análise de Sistemas, mas hoje em dia se envolve também em gerência de contas e coaching de agilismo. Suas áreas de interesse incluem aprendizado (individual e organizacional), sistemas (pensamento sistêmico e sistemas complexos) e desenvolvimento (econômico, de software ou de produtos).

## **Olivia Janequine**

**Olivia Janequine** é antropóloga e tecnologista. Trabalha na ThoughtWorks facilitando o trabalho e removendo obstáculos para equipes de tecnologia e negócios em contextos complexos. Gosta de problemas difíceis e cenários incertos, em que as pessoas têm a oportunidade de coletivamente transformar as organizações e o mundo em lugares mais justos e plenos. Sua experiência inclui governo, empresas e organizações sociais de pequeno, médio e grande porte em diversas áreas, entre elas comunicação, minas e energia, agricultura e varejo.

## **Paulo Caroli**

Consultor principal da Thoughtworks Brasil e cofundador da

AgileBrazil, **Paulo Caroli** possui mais de vinte anos de experiência em inovação e transformação digital, com passagem em várias corporações no Brasil, Índia, EUA e América Latina. Em 2000, conheceu o Extreme Programming e, desde então, tem mantido seu foco em processos e práticas de gestão e desenvolvimento ágil e Lean. Autor dos livros *Direto ao ponto: criando produtos de forma enxuta* e *Fun Retrospectives: activities and ideas for making agile retrospectives more engaging*, além de vários e-books, artigos e palestras. Leia mais em <http://www.caroli.org>.

## Rafael F. Buzon

**Rafael F. Buzon** é Agile Coach e consultor em gestão ágil de projetos e produtos. É cofundador do Kudoos (<http://kudoos.com.br>) e do evento Lean Coffee São Paulo. Já trabalhou na Editora Abril, Walmart.com, e tem escrito e criado vídeos para troca de conhecimento com a comunidade em seu site pessoal, em <http://rafaelbuzon.com>.

## Agradecimentos

Muito obrigado a **Hellen Guareschi** e **Tulius Lima** pelo auxílio a este trabalho.

# Sumário

<b>Oriente</b>	<b>1</b>
<b>1 Introdução</b>	<b>4</b>
1.1 A Lean Enterprise é essencialmente um sistema humano	11
1.2 Comando de Missão: uma alternativa ao comando e controle	
1.3 Criando alinhamento em escala seguindo o Princípio da Missão	15
1.4 Suas pessoas são sua vantagem competitiva	23
<b>2 Administrando a dinâmica do portfólio da empresa</b>	<b>29</b>
2.1 Explorando novas ideias	35
2.2 Aproveitando modelos de negócio validados	45
2.3 Balanceando o portfólio da empresa	50
2.4 Conclusão	59
<b>Investigue</b>	<b>62</b>
<b>3 Modele e meça o risco do investimento</b>	<b>64</b>
3.1 Modelo de risco de investimento	65
3.2 Aplicando o método científico em desenvolvimento de produto	69
3.3 Princípios para exploração	78

Sumário	Casa do Código
3.4 Gerenciamento científico versus método científico	84
3.5 Conclusão	86
<b>4 Explorar a incerteza para detectar oportunidades</b>	<b>88</b>
4.1 Descoberta	89
4.2 Em que negócio estamos?	94
4.3 Acelere a experimentação com MVPs	107
4.4 Conclusão	121
<b>5 Avalie a relação produto-mercado</b>	<b>123</b>
5.1 Contabilidade para inovação	124
5.2 Faça coisas que não escalem	133
5.3 Mecanismos de crescimento	140
5.4 Transição entre horizontes para crescer e transformar	143
5.5 Conclusão	150
<b>Explore</b>	<b>152</b>
<b>6 Implemente entrega contínua</b>	<b>157</b>
6.1 Estudo de caso: HP LaserJet Firmware	159
6.2 Diminua os custos através da inovação contínua de processos usando Melhoria Kata	163
6.3 Como o time da HP LaserJet implementou a Melhoria Kata	172
6.4 Gerenciando a demanda	181 <sup>172</sup>
6.5 Criando a empresa ágil	184
6.6 Conclusão	186
<b>7 Identifique valor e amplie o fluxo</b>	<b>189</b>
7.1 O estudo de caso Maersk	190
7.2 Ampliar o fluxo	194
7.3 Custo de Atraso: uma estrutura de trabalho para	

descentralizar decisões econômicas	210
7.4 Conclusão	219
<b>8 Adote práticas de engenharia Lean</b>	<b>221</b>
8.1 Os fundamentos da entrega contínua	223
8.2 Integração contínua e automação de teste	227
8.3 O pipeline de entrega	233
8.4 Dissociar deployment e release	240
8.5 Conclusão	242
<b>9 Realize uma abordagem experimental para desenvolvimento de produto</b>	<b>246</b>
9.1 Criando hipóteses para a próxima iteração	249
9.2 Pesquisa de usuário	255
9.3 Experimentos controlados online	258
9.4 Um exemplo de Teste A/B	262
9.5 Pré-requisitos para uma abordagem experimental em desenvolvimento de produto	266
9.6 Conclusão	270
<b>10 Implemente o Comando da Missão</b>	<b>272</b>
10.1 A abordagem da Amazon para o crescimento	274
10.2 Crie velocidade em escala através do Comando da Missão	279
10.3 Evoluindo sua arquitetura através do padrão de estrangulamento.	289
10.4 Conclusão	296
<b>Transformar</b>	<b>298</b>
<b>11 Cultive uma cultura de inovação</b>	<b>302</b>
11.1 Modele e mensure sua cultura	303
11.2 Transforme sua cultura	309

Sumário	Casa do Código
11.3 Não há escassez de talentos	316
11.4 Conclusão	334
<b>12 Adote o pensamento Lean para governança, risco e conformidade</b>	<b>336</b>
12.1 Entendendo governança, riscos e conformidade	338
12.2 Aplique princípios Lean aos processos da GRC	343
12.3 Mapeie a cadeia de valor, crie o fluxo e estabeleça um sistema puxado	349
12.4 Conclusão	359
<b>13 Evoluindo a gestão financeira para guiar a inovação de produto</b>	<b>361</b>
13.1 Introdução	361
13.2 Dançar no ritmo do tambor financeiro reduz a velocidade da inovação	363
13.3 Libertando-nos do ciclo anual de orçamento	364
13.4 Evite usar orçamentos como base para medir desempenho	
13.5 Pare de basear decisões de negócio em receita versus custo operacional	377
13.6 Altere seus processos de aquisição para ganhar maior controle sobre a entrega de valor	383
13.7 Conclusão	388
<b>14 Torne a TI em uma vantagem competitiva</b>	<b>391</b>
14.1 Repensando a mentalidade da TI	392
14.2 Liberdade e responsabilidade	397
14.3 Criando e evoluindo plataformas	401
14.4 Gerenciando sistemas existentes	407
14.5 Conclusão	416
<b>15 Fique onde está</b>	<b>419</b>
15.1 Princípios da mudança organizacional	420

15.2 O serviço digital do governo do Reino Unido	428
15.3 Comece sua jornada	436
15.4 Conclusão	439
<b>16 Referências bibliográficas</b>	<b>441</b>

Versão: 19.9.11

# Oriente

*“O propósito de uma organização é permitir que seres humanos ordinários façam coisas extraordinárias.” — Peter Drucker*

*“Valor para o acionista é a ideia mais idiota do mundo... [é] um resultado, não uma estratégia... Seus principais constituintes são seus empregados, seus clientes e seus produtos.” — Jack Welch  
(<http://on.ft.com/1zmWBMD>)*

Começamos mostrando nossa definição de um empreendimento: “um sistema complexo e adaptativo composto de pessoas que compartilham de um propósito comum”. Dessa maneira, incluímos companhias sem fins lucrativos e do setor público, bem como corporações. Entraremos em mais detalhes sobre sistemas complexos e adaptativos no *capítulo 1*. Contudo, a ideia de um propósito comum conhecido por todos os funcionários é essencial para o sucesso de uma empresa.

O propósito de uma companhia é diferente de sua visão (que descreve o que uma organização deseja se tornar) e sua missão (que descreve o negócio em que a organização está). Graham Kenny, diretor administrativo da consultoria Strategic Factors, descreve o propósito de uma organização como o que ela faz por outra pessoa, “colocando os gerentes e empregados no lugar dos clientes” (KENNY, 2014). Ele cita como exemplos a empresa Kellogg (“Nutrir famílias para que elas floresçam e perseverem”) e a empresa de seguros IAG (“Ajudar as pessoas a gerenciar o risco e se recuperar da dificuldade de uma perda inesperada”), e às quais nós

adicionamos nosso exemplo favorito: a SpaceX, “fundada em 2002 por Elon Musk para revolucionar o transporte espacial e, em última instância, tornar possível para as pessoas viver em outros planetas”.

Em seu abundante tempo livre na SpaceX, Musk cofundou a Tesla Motors com “um grupo de engenheiros intrépidos do Vale do Silício, que quiseram provar que carros elétricos podem ser incríveis”.

Criar, atualizar e comunicar o propósito da companhia são as responsabilidades dos executivos da empresa. Suas outras responsabilidades incluem criar uma estratégia pela qual a companhia atingirá seu propósito, e criar a cultura necessária para que essa estratégia tenha sucesso. Tanto estratégia como cultura vão evoluir em resposta a mudanças no ambiente, e os líderes são responsáveis por direcionar essa evolução e garantir que a cultura e a estratégia se apoiem para alcançar o propósito. Se os líderes fazem um bom trabalho, a organização conseguirá se adaptar, descobrir e sanar as necessidades do cliente, além de permanecer resiliente a eventos inesperados. Isso é a essência da boa governança.

No contexto de corporações, a ideia de um propósito comum que não seja a maximização de lucros pode parecer estranha. Por muitos anos, a sabedoria popular prega que executivos corporativos devem focar em maximizar o valor para os acionistas, e este objetivo foi reforçado ao se recompensar executivos com ações (JENSEN; MECKLING, 1976). Contudo, essas estratégias têm muitas falhas. Elas criam uma predisposição para resultados de curto prazo (como ganhos trimestrais) à custa de prioridades a longo prazo, como desenvolver as capacidades dos empregados e relação com os clientes. Elas também tendem a sufocar a inovação ao focar em

ações táticas para reduzir custos em curto prazo à custa de estratégias mais arriscadas que têm o potencial de fornecer uma recompensa maior ao longo da vida da organização, como pesquisa e desenvolvimento ou criar novos produtos e serviços disruptivos. Finalmente, elas frequentemente ignoram o valor dos intangíveis, como as habilidades dos empregados e propriedade intelectual, e a externalidade, como o impacto no meio ambiente.

Pesquisas mostram que focar apenas na maximização de lucros tem um efeito paradoxal de reduzir a taxa do retorno sobre investimento (KAY, 2010). Em vez disso, organizações têm sucesso no longo prazo por meio do desenvolvimento de sua capacidade de inovar e adaptar a estratégia dita por Jack Welch na epígrafe anterior: focar em empregados, clientes e produtos. A *Parte I* deste livro mostra como conseguir isso.

*Obliquity*, de John Kay (2010), fornece pesquisa e análise detalhadas que embasam o que ele descreve como “o paradoxo da busca do lucro”.

## CAPÍTULO 1

# INTRODUÇÃO

*"Às vezes, é possível que pessoas boas, em sistemas perversamente projetados, cometam atos que causem grande dano a desconhecidos sem nem perceber" — Ben Goldacre*

Em 1º de abril de 2010, a única fábrica de automóveis da Califórnia, a New United Motor Manufacturing, Inc. (NUMMI), fechou. A NUMMI, aberta em 1984, foi uma *joint venture* entre a GM e a Toyota. Ambas as empresas tinham o que ganhar com a parceria. A Toyota queria abrir uma fábrica nos EUA para evitar as restrições à importação que o Congresso norte-americano impunha como reação à queda contínua da participação de fabricantes norte-americanos no mercado de automóveis. Para a GM, era uma chance de aprender a construir carros pequenos de maneira rentável e estudar o Sistema de Produção Toyota (Toyota Production System — TPS), o qual permitia fabricantes japoneses de carros entregarem consistentemente a qualidade mais alta da indústria com custos mais baixos que os dos fabricantes dos EUA.

A história da fábrica da NUMMI foi contada em detalhes no podcast *This American Life*, episódio 403: <http://www.thisamericanlife.org/radio-archives/episode/403/>. Todas as citações diretas foram retiradas do vídeo.

A GM escolheu como local para essa *joint venture* sua antiga fábrica em Fremont. A fábrica de Fremont era uma das piores em termos da qualidade dos carros produzidos e da relação entre gerentes e trabalhadores. Quando a fábrica fechou em 1982, as relações de trabalho haviam se deteriorado completamente, a ponto de funcionários beberem e jogarem no local de trabalho. Surpreendentemente, a Toyota concordou com a exigência do Sindicato dos Trabalhadores da Indústria Automobilística (*United Auto Workers*), representado por Bruce Lee, de recontratar os antigos líderes sindicais da fábrica de Fremont para liderar a força de trabalho na NUMMI.

Os trabalhadores foram enviados para Toyota City, no Japão, para aprender o TPS. Em três meses, a fábrica da NUMMI estava produzindo carros quase impecáveis — entre os melhores da América em termos de qualidade, tão bons quanto os que vinham do Japão — a um custo muito menor do que a fábrica de Fremont havia conseguido. Lee tinha razão em sua hipótese de que "era o sistema que trazia problemas, não as pessoas".

Muito se escreve sobre o TPS, mas, ao ouvir os trabalhadores da fábrica de Fremont que migraram para a NUMMI, um tema recorrente é o trabalho em equipe. Pode parecer banal, mas foi uma experiência incrivelmente poderosa para muitos dos funcionários do sindicato. O TPS prevê que a mais alta prioridade é embutir qualidade durante a construção dos produtos. Por isso, após um

problema ser descoberto, ele deve ser corrigido o mais rapidamente possível e, em seguida, o sistema deve ser melhorado visando impedir que isso ocorra novamente.

Trabalhadores e gestores cooperam para tornar isso possível. No momento em que um trabalhador descobre um problema, ele pode chamar o gerente, puxando uma corda (a famosa corda *andon*). O gerente então chega para ajudar a tentar resolver o problema. Se o problema não pode ser resolvido dentro de um prazo inicial disponível, o trabalhador pode parar a linha de produção até que o problema seja corrigido. A equipe depois vai experimentar — e implementar — ideias para evitar que o problema ocorra novamente.

Esses conceitos — de que a principal tarefa dos gestores é ajudar os trabalhadores; que os trabalhadores devem ter o poder de parar a linha de produção; e que eles devem ser envolvidos para decidir como melhorar o sistema — foram revolucionários para os trabalhadores do sindicato. John Shook, o primeiro americano a trabalhar em Toyota City, encarregado da formação dos trabalhadores da NUMMI, pondera que "eles tiveram uma experiência emocional extremamente poderosa ao aprender uma nova forma de trabalhar, uma forma em que as pessoas poderiam realmente trabalhar em conjunto de forma colaborativa — como uma equipe".

A forma que o TPS funciona é radicalmente diferente das práticas de gestão tradicionais dos EUA e da Europa, baseadas nos princípios de Frederick Winslow Taylor, o criador da *administração científica*. De acordo com Taylor, a gestão deve analisar o trabalho e dividi-lo em tarefas separadas. Essas tarefas são realizadas por trabalhadores especializados, que precisam somente entender como fazer a sua tarefa específica da forma mais eficiente possível. O Taylorismo fundamentalmente encara as organizações como

máquinas que podem ser analisadas e compreendidas através da divisão em partes menores que as compõem.

Em contrapartida, no coração da TPS está a criação de uma cultura de alta confiança, em que todos estão alinhados em seu objetivo de construir um produto de alta qualidade, sob demanda, e onde trabalhadores e gerentes colaboram para melhorar constantemente o sistema — e, às vezes, reprojetar radicalmente. Esses conceitos do TPS — uma cultura de alta confiança focada na melhoria contínua (*kaizen*), empoderada pelo alinhamento e autonomia a todos os níveis — são essenciais para a criação de uma organização que possa se adaptar rapidamente às novas condições do ambiente.

Parte fundamental do sucesso do TPS está em seu efeito sobre os trabalhadores. O Taylorismo transforma os trabalhadores em engrenagens de uma máquina, pagos simplesmente para executar ações pré-definidas o mais rápido possível. O TPS, ao contrário, pressupõe que trabalhadores busquem maestria por meio da melhoria contínua, e oferece a eles um propósito maior: a busca por níveis cada vez mais elevados de qualidade, valor e serviço ao cliente. Fornece também autonomia, permitindo-os experimentar ideias de melhoria e implementar aquelas que são bem sucedidas.

Décadas de pesquisa têm demonstrado que esses *motivadores intrínsecos* geram uma alta performance em tarefas que requerem criatividade e tentativa e erro — onde o resultado desejado não pode ser alcançado simplesmente seguindo receitas. Na verdade, motivadores extrínsecos, como bônus e avaliações de desempenho, *diminuem* o desempenho das pessoas em trabalhos criativos. Décadas de estudos têm repetidamente demonstrado estes resultados. Para um excelente resumo, veja Pink (2009).

Cientistas comportamentais normalmente classificam o trabalho em dois tipos: tarefas de rotina, onde há um único resultado correto que pode ser alcançado seguindo uma regra, conhecidos como *algorítmicos*; e aqueles que necessitam de criatividade, tentativa e erro, chamados de *heurísticas*.

Rick Madrid, que trabalhou na fábrica de Fremont tanto antes quanto durante a era NUMMI, diz que a TPS "mudou a minha vida — antes deprimido, entediado — e, como meu filho disse, mudou a minha atitude. Ele tornou-me muito melhor". Dar às pessoas orgulho pelo seu trabalho em vez de tentar motivá-los com "cenouras e chicote" é um elemento essencial de uma cultura de alto desempenho.

Com efeito um dos "Quatorze Pontos para a transformação da gestão" de W. Edwards Deming é "Remover as barreiras que roubam das pessoas na gestão e na engenharia do direito ao orgulho da sua obra. Isto significa, entre outras coisas, a abolição da avaliação anual ou por mérito e da administração por objetivos" (DEMING, 2000, p. 24).

Embora os princípios centrais da TPS possam parecer relativamente simples, eles são muito difíceis de se adotar. Na verdade, a GM fracassou completamente em capturar o que havia alcançado na NUMMI e reproduzir em suas outras fábricas. Um dos maiores obstáculos foi a mudança necessária na hierarquia organizacional. O TPS desfaz o conceito de senioridade pelo qual trabalhadores sindicalizados recebem postos de trabalho com base

em quantos anos de serviço possuem, oferecendo os melhores empregos para os mais antigos. No TPS, todos precisam aprender todas as tarefas de sua equipe e rotacionar entre elas. O TPS também elimina benefícios e privilégios dos gerentes. Ninguém usava gravata na fábrica NUMMI — nem mesmo terceiros — para enfatizar o fato de que todo mundo fazia parte da mesma equipe. Gerentes não recebiam regalias concedidas a eles em outras fábricas da GM, como cafeteria e estacionamento separados dos demais.

Por fim, as tentativas de melhorar a qualidade esbarraram em fronteiras organizacionais. No TPS, fornecedores, engenheiros e trabalhadores colaboravam para melhorar continuamente a qualidade dos insumos e para garantir que os trabalhadores tivessem as ferramentas de que precisam para fazer seu trabalho. Isso funcionou na NUMMI porque os engenheiros estavam dentro da própria fábrica, e os insumos vinham de fornecedores japoneses que possuíam uma relação de colaboração com a Toyota. Na cadeia de suprimentos dos EUA, as coisas eram diferentes. Se os insumos entregues para a GM eram de má qualidade — ou não serviam —, simplesmente não havia mecanismo para corrigir o problema.

Ernie Schaefer, gerente da fábrica da GM em Van Nuys, a qual enfrentou muitos dos mesmos problemas de Fremont, descreve o que havia de diferente na NUMMI: "Você pode ver muitas coisas diferentes. Mas a única coisa que você não vê é o sistema que apoia a fábrica da NUMMI. Eu não acredito que, naquela época, alguém tenha entendido a amplitude deste sistema. A General Motors era uma espécie de organização onde as coisas eram jogadas por cima de muros. Em cada departamento, éramos muito compartmentados; você projetava o veículo, e então jogava o projeto por cima do muro para os caras da fabricação". Este é o legado de uma abordagem de gestão Taylorista. O TPS existe — e só consegue ter sucesso — dentro de um ecossistema com cultura organizacional, relações com fornecedores, gestão financeira,

recursos humanos e governança projetados em torno de sua filosofia.

A GM tentou implementar o TPS em Van Nuys, mas não conseguiu. Trabalhadores e gerentes se rebelaram frente às mudanças de status e comportamento necessárias, apesar da ameaça de fechamento da fábrica (que acabou de fato ocorrendo). De acordo com Larry Spiegel, um veterano da NUMMI havia sido enviado a Van Nuys para ajudar a implementar o TPS, as pessoas na fábrica simplesmente não acreditavam no risco de fechamento: "Havia pessoas demais convencidas de que não precisavam mudar".

Essa falta de senso de urgência mostrou-se uma barreira para a adoção ampla do TPS na GM — e talvez seja o maior obstáculo para qualquer mudança organizacional (KOTTER, 2012). A divisão da GM nos EUA levou cerca de 15 anos para decidir que precisava priorizar seriamente a implementação do TPS, e mais 10 anos para realmente implementá-lo. A essa altura, qualquer vantagem competitiva que poderiam ter ganhado foi perdida. A GM foi à falência e acabou socorrida pelo governo dos EUA em 2009, momento em que a empresa deixou a NUMMI. A Toyota fechou a fábrica da NUMMI em 2010.

John Kotter, autor de *Liderando Mudanças*, diz que "a maioria dos funcionários, talvez 75 por cento da gestão global e praticamente todos os altos executivos, precisa acreditar que a mudança considerável é absolutamente essencial" (KOTTER, 2012, p. 51).

A história da NUMMI é relevante, pois ilustra a principal preocupação deste livro: criar uma empresa enxuta, como a Toyota, e quais são os obstáculos mais comuns. A Toyota sempre foi muito

aberta sobre suas práticas, oferecendo visitas abertas em suas fábricas, até mesmo para os concorrentes, em parte porque sabe que o que faz o TPS dar certo não são as práticas em particular, mas sim a cultura. Muitas pessoas concentram-se nas práticas e ferramentas popularizadas pelo TPS, tais como as cordas *andon*. Um vice-presidente da GM chegou a pedir a um de seus gerentes para tirar fotos de cada centímetro da fábrica da NUMMI, para que eles pudessem copiá-la com precisão. O resultado foi uma fábrica com cordas *andon*, mas sem ninguém para puxá-las, visto que os gerentes (seguindo o princípio da motivação extrínseca) eram medidos pela quantidade — *independente* da qualidade — de automóveis produzidos.

## 1.1 A LEAN ENTERPRISE É ESSENCIALMENTE UM SISTEMA HUMANO

Com a forte aceleração das transformações sociais e tecnológicas no mundo de hoje, a abordagem *lean*, iniciada pela Toyota, se torna cada vez mais importante. Ela é uma estratégia eficaz para se ter sucesso em ambientes de incerteza, aceitando a mudança como constante. A chave para se entender uma empresa enxuta é saber que ela é essencialmente um sistema *humano*.

É comum se dar foco nas práticas e ferramentas específicas que equipes enxutas e ágeis utilizam, tais como quadros Kanban, stand-ups, programação em pares, e assim por diante. No entanto, muitas vezes elas são adotadas como rituais ou "melhores práticas", porém não são entendidas pelo seu real significado: *contramedidas* que são eficazes dentro de um contexto específico na busca de um objetivo em particular.

Em uma organização com uma cultura de melhoria contínua, essas contramedidas surgem naturalmente dentro de equipes e

depois são descartadas quando não tem mais valor. A chave para se criar uma empresa enxuta é permitir que as pessoas possam resolver os problemas dos clientes de forma alinhada à estratégia da organização. Para isso, espera-se que as pessoas sejam capazes de tomar decisões locais coerentes com a estratégia — que, por sua vez, é extremamente dependente do fluxo de informações, incluindo ciclos de feedback.

Fluxos de informação têm sido profundamente estudados pelo sociólogo Ron Westrum, principalmente em contextos de acidentes e erros humanos na aviação e hospitais. Westrum (2014) percebeu que a segurança nesses ambientes poderia ser deduzida pelo tipo de cultura organizacional, e desenvolveu um "espectro de culturas de segurança" com três categorias:

- *Organizações patológicas* são caracterizadas por grandes doses de medo e ameaças. As pessoas muitas vezes guardam informações para si por razões políticas, ou as distorcem em seu benefício.
- *Organizações burocráticas* criam proteções para departamentos. Aqueles que estão no departamento querem manter seu "território", insistem em jogar por suas próprias regras, e geralmente fazem as coisas segundo o manual — o seu manual.
- *Organizações generativas* concentram-se na missão. Como atingir a meta? Fazer a coisa certa e ter um desempenho ótimo é o que importa.

Cada uma delas processa informações de diferentes maneiras. Westrum observa que "o clima que fomenta um bom fluxo de informação também incentiva outros tipos de comportamentos cooperativos e focados na missão, tais como resolução de problemas, inovação e conexões interdepartamentais. Quando as

coisas dão errado, organizações patológicas incentivam a busca por bodes expiatórios, organizações burocráticas buscam a justiça, e organizações gerativas buscam entender problemas sistêmicos”. As características dos vários tipos de cultura são apresentadas na *Tabela 1-1*.

**Tabela 1-1 — Como organizações processam informação**

Patológicas (orientadas a poder)	Burocráticas (orientadas a regras)	Gerativas (orientadas a performance)
Baixo nível de cooperação	Nível de cooperação módico	Alto nível de cooperação
O mensageiro é o culpado	O mensageiro é ignorado	O mensageiro é treinado
Se esquivar da responsabilidade	Limitar a responsabilidade	Compartilhar o risco
Conexões são desencorajadas	Conexões são toleradas	Conexões são estimuladas
Falhas levam à busca de um bode expiatório	Falhas levam à justiça	Falhas levam ao aprendizado
Novidade é destruída	Novidade leva a problemas	Novidade é implementada

A tipologia de Westrum tem sido amplamente estudada e estendida, e tem características essenciais as quais qualquer um que tenha trabalhado em uma organização patológica (ou mesmo burocrática) vai reconhecer. No entanto, algumas de suas implicações estão longe de ser apenas acadêmicas.

Em 2013, PuppetLabs, IT Revolutions Press, e ThoughtWorks realizaram uma pesquisa com 9.200 tecnologistas em todo o mundo para descobrir o que faz as organizações de alto desempenho serem bem sucedidas. A pesquisa *State of DevOps Report* de 2014 é baseada na análise de respostas de profissionais de diversas indústrias, incluindo finanças, telecomunicações, varejo, governo, tecnologia, educação e saúde (FORSGREN; KIM; KERSTEN; HUMBLE, 2014). O principal resultado da pesquisa é que um bom desempenho de TI

é uma vantagem competitiva. A análise mostrou que empresas com alto desempenho em TI possuem *duas vezes mais chances* de ampliar sua rentabilidade, participação de mercado e metas de produtividade.

A pesquisa mediu o desempenho organizacional pedindo aos entrevistados para avaliar o desempenho relativo de sua organização em termos de alcance de metas de rentabilidade, participação de mercado e produtividade. Esta é uma escala padrão que foi validada várias vezes em pesquisas anteriores. Veja Widener (2007).

A pesquisa também se propôs a examinar os fatores culturais que influenciam o desempenho organizacional. O mais importante deles se provou ser se as pessoas estavam satisfeitas com seus empregos, com base em quanto concordavam com as seguintes afirmativas (lembram bastante da reação dos trabalhadores da NUMMI introduzidos ao Sistema Toyota de Produção):

- Eu recomendaria esta organização como um bom lugar para trabalhar.
- Eu tenho as ferramentas e os recursos para fazer bem o meu trabalho.
- Estou satisfeito com o meu trabalho.
- Meu trabalho faz bom uso das minhas habilidades e capacidades.

O fato de que a satisfação no trabalho tenha sido o principal fator preditivo do desempenho organizacional demonstra a importância da motivação intrínseca. A equipe envolvida na pesquisa queria descobrir se o modelo de Westrum era uma

ferramenta útil para prever a performance organizacional.

Por uma questão de transparência, informamos que o Jez foi parte do time por trás do *2014 State of DevOps Report*.

A pesquisa pediu então às pessoas que avaliassem a cultura de sua equipe, considerando cada um dos eixos do modelo de Westrum (conforme *Tabela 1-1*) para avaliar o grau de concordância com afirmativas como "*No meu time, uma falha causa investigação*" — este método de medição quantitativa de atitudes é conhecido como uma escala do tipo Likert. Com isso, a pesquisa foi capaz de medir a cultura dos times e organizações.

A análise estatística dos resultados mostrou que a cultura das equipes não apenas está fortemente correlacionada com o desempenho organizacional, mas também é um forte fator preditivo da satisfação no trabalho. Os resultados são claros: uma cultura generativa e de alta confiança não só é importante para a criação de um ambiente de trabalho seguro — é também o alicerce da criação de uma organização de alto desempenho.

## 1.2 COMANDO DE MISSÃO: UMA ALTERNATIVA AO COMANDO E CONTROLE

Culturas organizacionais de alta confiança muitas vezes são comparadas em oposição ao popular "comando e controle": a ideia da Administração Científica de que as pessoas no comando formulam os planos, e as pessoas no chão de fábrica os executam — também geralmente entendida como o modelo militar. Na verdade, contudo, este tipo de sistema de comando e controle não é utilizado pelos militares desde 1806, quando o exército prussiano, uma

organização tradicionalmente orientada por planejamento, foi derrotado violentamente pelas forças descentralizadas de Napoleão, compostas por soldados altamente motivados.

Napoleão usou um estilo de guerra conhecido como *guerra de manobra* para derrotar exércitos maiores, mais bem treinados. Na guerra de manobra, o objetivo é minimizar a necessidade de embate real, minando a capacidade do inimigo de agir de forma coesa por meio do uso de choque e surpresa. Um elemento-chave na guerra de manobra é ser capaz de aprender, tomar decisões e agir mais rápido que seus inimigos — a mesma capacidade que permite que startups causem disruptão nos negócios de empresas estabelecidas.

Como discutiremos no *capítulo 3*, este conceito é formalizado no ciclo OODA de John Boyd (observar-orientar-decidir-agir), que por sua vez inspirou Eric Ries no ciclo "construir-medir-aprender".

Três homens foram especialmente importantes para a reconstrução do exército prussiano depois da sua derrota para Napoleão: Carl von Clausewitz, David Scharnhorst e Helmuth von Moltke. Suas contribuições não só transformaram a doutrina militar; elas têm implicações importantes para as pessoas que lideram e administram grandes organizações. Isso se aplica em particular à ideia de *Auftragstaktik*, ou Comando de Missão, que vamos explorar aqui. Comando de Missão é o que permite que a guerra de manobra funcione em grande escala — e é chave para entender como empresas estabelecidas podem competir com startups.

Após a eventual derrota de Napoleão, o general David Scharnhorst foi promovido a chefe do recém-criado Estado-Maior

Geral do Exército Prussiano. Ele montou uma comissão de reforma que conduziu um *post-mortem* e começou uma transformação no exército prussiano. Scharnhorst observou que os oficiais de Napoleão tinham a autoridade para tomar decisões à medida que a situação da batalha mudava, sem precisar esperar por aprovações da cadeia de comando. Isto lhes permitia adaptar-se rapidamente às novas circunstâncias.

Scharnhorst queria desenvolver capacidade semelhante, de forma sistemática. Ele percebeu que isso exigia a formação de um quadro independente e inteligente de oficiais que compartilhassem valores semelhantes e fossem capazes de agir de forma decisiva e autônoma no calor da batalha. Assim, foram criadas escolas militares para treinar oficiais, que, pela primeira vez, foram aceitos vindos de qualquer origem social, com base no mérito.

Em 1857, Helmuth von Moltke — talvez mais conhecido por seu ditado "nenhum plano sobrevive ao contato com o inimigo" — foi nomeado Chefe do Estado-Maior Geral do Exército Prussiano. Sua principal inovação, com base na cultura militar estabelecida por Scharnhorst, foi tratar a estratégia militar como um conjunto de opções que deveriam ser estudadas extensivamente pelos oficiais antes da batalha. Em 1869, ele emitiu uma diretiva intitulada "Orientações para comandantes de grandes unidades" que definia como conduzir uma organização de grande porte em condições de incerteza.

Neste documento, von Moltke observa que "na guerra, as circunstâncias mudam rapidamente, e é muito raro que orientações que abrangem um longo período de tempo e com muitos detalhes sejam executadas plenamente." Ele recomenda, portanto, "não comandar mais que o estritamente necessário, nem planejar além das circunstâncias que você pode prever". Em vez disso, ele dá o seguinte conselho: "Quanto mais alto o nível hierárquico, mais

curtas e mais gerais as ordens devem ser. O próximo nível abaixo deve acrescentar quaisquer especificações adicionais que acreditar serem necessárias, e os detalhes de execução são deixados para instruções verbais ou, quem sabe, uma palavra de comando. Isso garante que todos mantenham a liberdade de movimento e de decisão dentro dos limites da sua autoridade... A regra a ser seguida é que uma ordem deve conter tudo — e apenas — o que os subordinados não podem determinar por si próprios para atingir uma finalidade específica”.

Fundamentalmente, as ordens sempre incluem um trecho que descreve a sua *intenção*, comunicando o *propósito* das ordens. Isso permite aos subordinados tomar boas decisões diante das oportunidades emergentes ou obstáculos que os impedem de seguir exatamente as ordens originais. Von Moltke observa que "existem inúmeras situações nas quais um oficial deve agir de acordo com seu próprio julgamento. Um oficial esperar por ordens nos momentos em que nenhuma pode ser dada seria um tanto quanto absurdo. Mas, via de regra, é quando ele age de acordo com a vontade de seu superior, que mais eficazmente pode desempenhar o seu papel no todo".

Essas ideias formam o núcleo da doutrina do *Auftragstaktik* (ou Comando de Missão), que, combinada com a criação de um quadro profissionalmente treinado de oficiais que entendiam como aplicar a doutrina na prática, foi adotada por várias unidades militares de elite, incluindo o Corpo de Fuzileiros Navais dos EUA, bem como (mais recentemente) a OTAN.

A história do desenvolvimento pelo exército prussiano do *Auftragstaktik* é descrito mais detalhadamente no tratado de Stephen Bungay (2010) sobre estratégia de negócios, “O Melhor Ataque é a Execução” (de onde foram extraídas as citações anteriores sobre “Orientação para comandantes de grandes

unidades"). Bungay desenvolve uma teoria sobre dirigir estratégia em grande escala que se baseia na obra de Scharnhorst, von Moltke, e outro general prussiano, Carl von Clausewitz. Com 26 anos de idade, Clausewitz já havia lutado contra Napoleão nas batalhas decisivas de Jena e Auerstadt. Em seguida, ele serviu na comissão de reforma de Scharnhorst e deixou legada sua inacabada obra prima, *Sobre a Guerra*.

Neste trabalho, ele introduz o conceito de "névoa de guerra" — a incerteza fundamental que enfrentamos como atores em um ambiente de grandes e rápidas mudanças, com conhecimento necessariamente incompleto do estado do sistema como um todo. Ele também introduz a ideia de *atrito* que impede a realidade de se comportar de uma forma ideal. O atrito se manifesta na forma de informações incompletas, efeitos colaterais não previstos, fatores humanos como enganos e mal-entendidos, e acúmulo de eventos inesperados.

## ATRITO E SISTEMAS ADAPTATIVOS COMPLEXOS

O conceito de atrito de Clausewitz é uma excelente metáfora para compreender o comportamento de sistemas adaptativos complexos como uma empresa (ou, de fato, qualquer organização humana). A característica principal de um sistema adaptativo complexo é que o seu comportamento global não pode ser entendido por meio da abordagem reducionista de Taylor, de analisar seus componentes.

Ao contrário, muitas propriedades e padrões de comportamento de sistemas adaptativos complexos "emergem" das *interações* entre eventos e componentes em vários níveis dentro do sistema. No caso de sistemas abertos (como empresas), também é preciso considerar as interações com o ambiente, incluindo as ações dos clientes e concorrentes, bem como mudanças sociais e tecnológicas mais amplas. Atrito é, por fim, uma consequência da condição humana — o fato de que as organizações são compostas por pessoas com vontades independentes e informações limitadas. Assim, o atrito não pode ser eliminado.

Para quem estiver interessado em diferentes tipos de sistemas e como entendê-los, recomendamos estudar modelo Cynefin de Dave Snowden: <http://www.youtube.com/watch?v=N7oz366X0-8>.

Bungay argumenta que o atrito cria três lacunas. Em primeiro lugar, uma *lacuna de conhecimento* surge quando planejamos ou agimos, devido ao estado necessariamente imperfeito da informação que temos à disposição e a necessidade de fazermos suposições e interpretar essa informação. Em segundo lugar, uma *lacuna de*

*alinhamento* resulta de pessoas fazendo as coisas diferente do planejado, talvez devido a prioridades conflitantes, mal-entendidos, ou simplesmente alguém esquecendo ou ignorando algum elemento do plano. Finalmente, há uma *lacuna de efeitos* devido a alterações imprevisíveis no ambiente, talvez causadas por outros atores ou efeitos colaterais inesperados que produzem resultados que diferem dos que prevíamos. Essas lacunas são mostradas na figura a seguir.

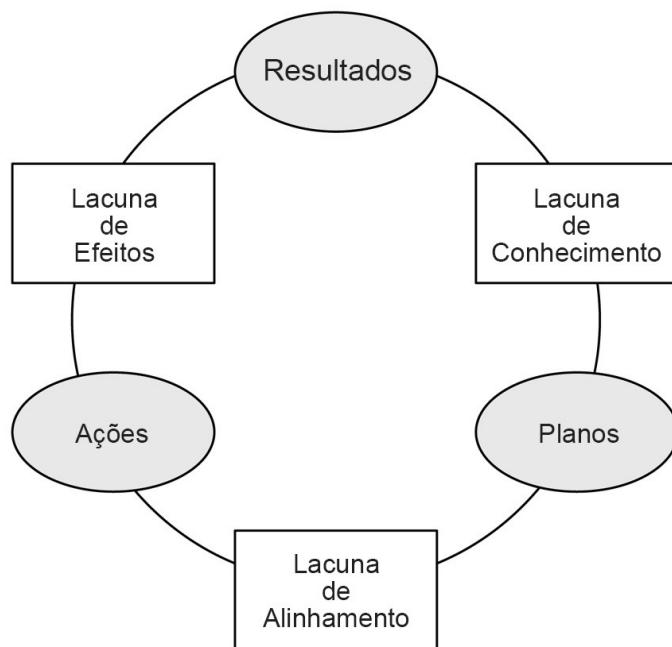


Figura 1.1: Lacunas em sistemas adaptativos complexos, de “O Melhor Ataque é a Execução, Como Transformar Planejamento Em Ações Efetivas” de Stephen Bungay (reproduzida com a autorização da Nicholas Brealey Publishing)

Bungay passa a descrever em seguida as contramedidas da Administração Científica tradicional aplicadas por empresas, a alternativa proposta pela doutrina da *Auftragstaktik*, e sua própria interpretação do Comando de Missão aplicada aos negócios, que ele denomina "oportunismo dirigido". Estes são mostrados na *Tabela 1-*

2.

**Tabela 1-2 — As três lacunas, e como lidar com elas**

	Lacuna de efeitos	Lacuna de conhecimento	Lacuna de alinhamento
<i>O que é</i>	A diferença entre em que esperamos que nossas ações resultem e o que elas de fato resultam	A diferença entre o que gostaríamos de saber e o que de fato sabemos	A diferença entre o que queremos que as pessoas façam e o que elas de fato fazem
<i>Solução na administração científica</i>	Controles mais detalhados	Informação mais detalhada	Instruções mais detalhadas
<i>Solução no Auftragstaktik</i>	"todos mantêm liberdade de decisão e ação dentro de limites"	"Não dê mais ordens do que o necessário, nem planeje mais do que pode prever"	"Comunique a cada unidade o quanto for necessário para atingir o propósito"
<i>Solução do oportunismo direcionado</i>	Dê aos indivíduos liberdade para ajustar as ações de acordo com a intenção	Limite o direcionamento a definir e comunicar a intenção	Permita que cada nível defina como vai atingir a intenção do nível imediatamente superior e apenas os explique o plano

É fundamental entender que, quando trabalhamos em um sistema adaptativo complexo em que prevalece o atrito, as soluções da Administração Científica *não podem* funcionar. Na verdade, elas pioram as coisas. Criar planos cada vez mais detalhados retarda o feedback que nos diria quais das nossas suposições são inválidas. Conjuntos complexos de regras e controles punem o inocente, mas podem ser evitadas pelo culpado, ao mesmo tempo destruindo o ânimo, a inovação e o empreendedorismo. A coleta de informações falha diante de organizações burocráticas ou patológicas que ocultam ou distorcem informações, a fim de proteger seu território. Organizações incapazes de escapar das garras da Administração Científica são alvos perfeitos para disruptão por organizações que

entendem como se mover rapidamente em larga escala.

### 1.3 CRIANDO ALINHAMENTO EM ESCALA SEGUINDO O PRINCÍPIO DA MISSÃO

A mais importante preocupação que líderes e gestores operando em um sistema adaptativo complexo encontram é esta: como podemos permitir que as pessoas dentro da organização tomem boas decisões — como agir no interesse da organização — dado que eles *nunca* terão informações e contexto suficientes para compreender as consequências de suas decisões, e dado que os eventos muitas vezes atropelam nossos planos?

Em *Princípios do Fluxo de Desenvolvimento de Produtos*, Donald Reinertsen (2009) apresenta o *Princípio da Missão*, com base na doutrina da Comando de Missão, em que se deve "especificar o estado final, o seu objetivo e as restrições mínimas possíveis." De acordo com o Princípio da Missão, criamos alinhamento não ao fazer um plano detalhado de como podemos alcançar o nosso objetivo, mas sim ao descrever a *intenção* de nossa missão e comunicando o *porquê* de a estarmos realizando.

A chave para o Princípio da Missão é criar alinhamento e permitir autonomia estabelecendo condições-alvo claras e de alto nível dentro de um prazo acordado — e que se torna mais curto em condições de maior incerteza —, deixando então aos times os detalhes de *como* alcançar as condições estabelecidas. Esta abordagem pode ser aplicada até mesmo em múltiplos níveis de hierarquia, com cada nível reduzindo o escopo enquanto provê mais contexto. No decorrer deste livro, esse princípio é aplicado em múltiplos contextos:

- *Orçamentação e gestão financeira*

Em vez do tradicional processo orçamentário que requer que todos os gastos para o próximo ano sejam planejados e alocados em projeções detalhadas e planos de negócio, definimos objetivos de alto nível em múltiplas perspectivas, como pessoas, organização, operações, mercado e financeira, que são revisados regularmente. Este tipo de exercício pode ser usado em múltiplos níveis, com recursos alocados dinamicamente conforme necessário e indicadores revisados regularmente.

- *Gestão de programas*

Em vez de criar planos detalhados no começo do trabalho a ser feito, e então dividir este trabalho em partes menores distribuídas entre times individuais, nós especificamos no nível de programa somente os objetivos mensuráveis de cada iteração. Os times então organizam-se para alcançar aqueles objetivos, inclusive colaborando com outros times, e continuamente integrando e avaliando seu trabalho para assegurar que alcançarão os objetivos no nível do programa.

- *Melhoria de processo*

Trabalhar para melhorar continuamente os processos é um fator chave do TPS e uma ferramenta poderosa para transformar organizações. No *capítulo 6*, nós apresentamos o Kata de Melhoria, no qual trabalhamos em iterações, especificando metas para processos e oferecendo às pessoas que operam os processos o tempo e recursos para realizar os experimentos que precisam para alcançar as metas estabelecidas para a próxima iteração.

Fundamentalmente, estes processos baseados em missão devem *substituir* os processos de comando e controle, e não ser operados em paralelo. Isto requer que pessoas comportem-se e atuem de uma maneira diferente e que aprendam novas habilidades. Também requer uma mudança cultural dentro da organização, como discutiremos no *capítulo 11*. Na discussão de como aplicar Comando de Missão nos negócios, Stephen Bungay reflete sobre a cultura que permite a Comando de Missão acontecer — que, não coincidentemente, tem as mesmas características encontradas em organizações generativas descritas por Westrum na *Tabela 1-1*:

*"O núcleo imutável é uma abordagem holística que afeta recrutamento, treinamento, planejamento e processos de controle, mas também a cultura e os valores de uma organização. Comando de Missão envolve uma concepção de liderança que sem sentimentalismos coloca o ser humano em seu centro. Depende fundamentalmente de fatores que não aparecem no balancete das organizações: a boa vontade das pessoas em aceitar responsabilidade; a prontidão dos seus superiores em apoiar suas decisões; e a tolerância a erros cometidos de boa-fé. Projetado para um ambiente externo que é imprevisível e hostil, se baseia em um ambiente interno que é previsível e encorajador. E no seu cerne está uma rede de confiança unindo pessoas, acima, abaixo e espalhadas na hierarquia. Alcançar e manter isso requer trabalho constante" (BUNGAY, 2010, p. 88).*

## 1.4 SUAS PESSOAS SÃO SUA VANTAGEM COMPETITIVA

A história da fábrica de Fremont não se encerra com a NUMMI. É, de fato, o local de duas mudanças de paradigma na indústria de fabricação de automóveis dos EUA. Em 2010, a fábrica da NUMMI foi comprada pela Tesla Motors e se tornou a Fábrica da Tesla. A

Tesla usa métodos contínuos para inovar mais rápido do que a Toyota, rejeitando o conceito de um modelo por ano em favor de atualizações mais frequentes e, em muitos casos, permitindo que proprietários de veículos mais antigos baixem atualizações de software para acessarem novos recursos.

A Tesla também tem promovido a transparência da informação, anunciando que não vai impor seus direitos de patente. Ao fazê-lo, ecoa uma história das origens da Toyota quando construía teares automáticos. Ao ouvir que os planos para um dos teares tinha sido roubado, dizem que Kiichiro Toyoda observou:

*"Certamente os ladrões podem ser capazes de seguir o projeto e produzir um tear. Mas estamos modificando e melhorando nossos teares todos os dias. Então, no momento em que os ladrões produzirem um tear com os planos que roubaram, já teremos avançado muito além desse ponto. E como eles não têm a experiência adquirida com as falhas que cometemos para produzir o original, eles vão perder muito mais tempo do que nós para melhorarem o seu tear. Não precisamos ficar preocupados com o que aconteceu. Precisamos apenas continuar como sempre, fazendo nossas melhorias"* (ROTHER, 2010, p. 40).

O valor a longo prazo de uma empresa não é capturado pelo valor de seus produtos e propriedade intelectual, mas sim por sua capacidade de aumentar continuamente o valor que ele fornece aos clientes — e para atrair *novos* clientes — por meio da inovação.

A premissa fundamental deste livro — apoiada pela experiência de empresas como Tesla e muitas, muitas outras — é que a flexibilidade proporcionada pelo software pode, quando corretamente alavancada, acelerar o ciclo de inovação. Software pode fornecer à sua empresa uma vantagem competitiva ao permitir buscar novas oportunidades e levar adiante oportunidades validadas mais rápido do que a concorrência.

A boa notícia é que esses recursos estão ao alcance de todas as empresas, não apenas gigantes da tecnologia. Os dados do *State of DevOps Report* de 2014 mostra que 20% das organizações com mais de 10.000 funcionários se enquadram no grupo de alto desempenho — uma porcentagem menor do que as empresas menores, mas ainda significativo.

Muitas pessoas que trabalham em grandes empresas acreditam que há alguma diferença fundamental entre elas e gigantes da tecnologia, como Google, Amazon ou Netflix, que são apontados como exemplos de tecnologia "bem feita". Muitas vezes ouvimos "isso não vai funcionar aqui". Isso pode estar certo, mas muitas vezes as pessoas procuram nos lugares errados pelos obstáculos que os impedem de melhorar. Céticos frequentemente tratam tamanho, regulamentação, percepção de complexidade, tecnologia legada, ou alguma outra característica especial do domínio em que operam como uma barreira à mudança. O objetivo deste capítulo é mostrar que, ainda que esses obstáculos realmente sejam reais, as barreiras mais sérias encontram-se na cultura organizacional, liderança e estratégia.

Muitas organizações tentam pegar atalhos para o alto desempenho fazendo laboratórios de inovação, adquirindo startups, adotando metodologias, ou fazendo reorganizações. Mas esses esforços não são necessários nem suficientes. Eles só podem ser bem sucedidos se combinados com esforços para criar uma cultura e uma estratégia generativas em toda a organização, incluindo fornecedores — e, se isso for alcançado, não haverá necessidade de recorrer a tais atalhos.

O segundo capítulo deste livro descreve os princípios que permitem às organizações terem sucesso no longo prazo equilibrando seu portfólio de produtos. Em particular, distinguimos dois tipos independentes de atividades no ciclo de desenvolvimento

de produto: *explorar* novas ideias para coletar dados e eliminar aquelas que não terão rápida adoção pelos usuários; e *aproveitar* daquelas que validamos no mercado. A *Parte II* do livro discute como estudar o domínio de exploração, enquanto a *Parte III* cobre o domínio de aproveitamento. Finalmente, a *Parte IV* do livro mostra como transformar sua organização focando em cultura, gestão financeira, governança, risco e compliance.

## CAPÍTULO 2

# ADMINISTRANDO A DINÂMICA DO PORTFÓLIO DA EMPRESA

*“A finalidade de um negócio é criar um cliente.”* — Peter Drucker

Neste capítulo, examinaremos o ciclo de vida dos negócios e como as empresas podem equilibrar a *exploração* de novos modelos de negócios com o *aproveitamento* dos que forem comprovados. Vamos precisar desta distinção para entender onde as práticas e princípios de Lean Startup podem ser aplicados em um contexto empresarial e como podem ser usados como base da gestão de um portfólio de inovação.

Em seu livro *Difusão de Inovações*, Everett Rogers (2003) descreve o ciclo através do qual todas as tecnologias e ideias bem-sucedidas progridem, mostrado na figura seguinte. Com o tempo, todas as ideias de sucesso — não importa se são tecnologias, categorias de produtos, modelos de negócio, ou até mesmo metodologias — progridem de algo escasso e desigualmente distribuído para, eventualmente, se tornarem *commodities*. Elas, então, formam as fundações para novas inovações mais valiosas, de

nível superior. Claro que o tempo que leva para inovações progredirem através das diferentes fases do ciclo pode variar substancialmente.

O trabalho de Rogers (2003) derivou, na verdade, de pesquisas sobre a adoção da tecnologia pelos agricultores em Iowa.

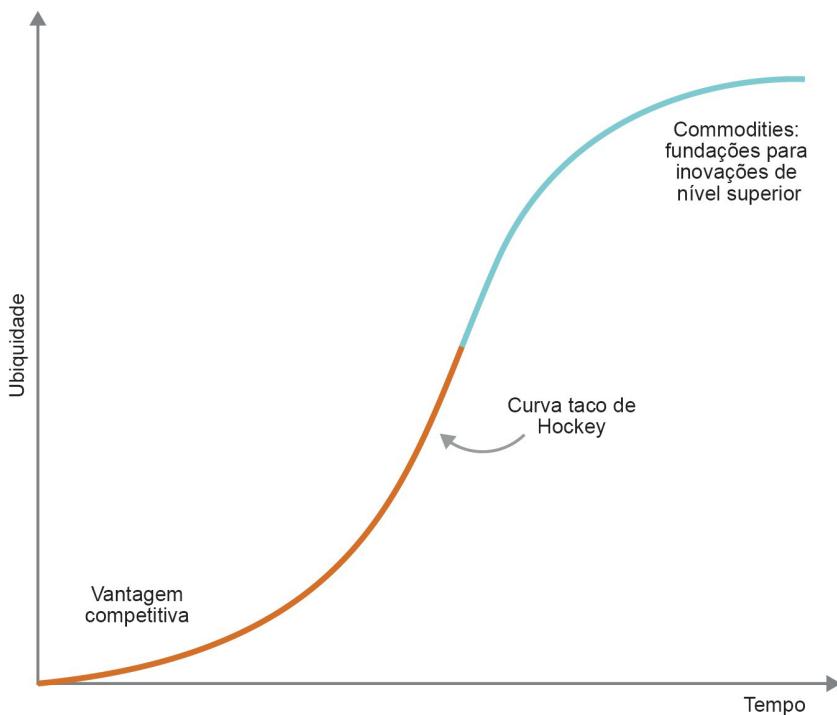


Figura 2.1: A curva em S que mostra o ciclo de vida de inovações

Rogers acreditava que pessoas poderiam ser classificadas em grupos com base em como elas respondem à inovação, conforme exibido na figura adiante. Inicialmente, novas tecnologias e ideias são experimentadas e testadas por inovadores, que formam o menor

grupo na população total. À medida que inovadores descobrem novas tecnologias que oferecem vantagem competitiva (a maioria não oferecerá), elas começam a ser adotadas pelos primeiros seguidores. Dessa maneira, sucesso em um grupo leva a ainda mais difusão pelos grupos seguintes. As ideias de Rogers foram popularizadas e aperfeiçoadas por Geoffrey Moore, que introduziu o conceito de “abismo”, uma divisão lógica entre adoção pelos pioneiros e pela maioria precoce. Esse abismo se inspirou na observação de Moore que muitas inovações fracassam quando não são mais vistas como uma fonte de vantagem competitiva por visionários, mas não estão ainda suficientemente estabelecidas para serem vistas como uma aposta segura ou prática comprovada pelos membros da maioria precoce.

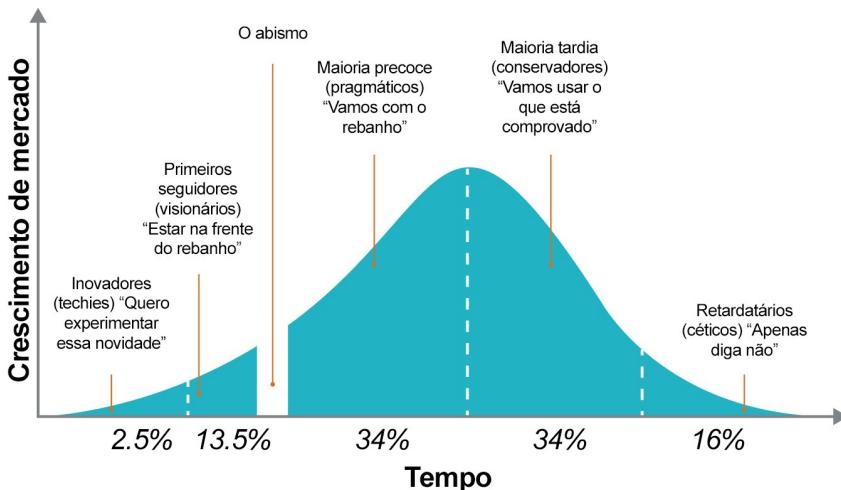


Figura 2.2: O ciclo de vida de adoção de tecnologia, de "Lidando com Darwin" (Dealing with Darwin) de Geoffrey A. Moore (2006) — utilizado com a permissão de Portfólio, um selo editorial do Grupo Penguin (EUA) LLC.

Uma vez que o mercado assimila uma tecnologia ou ideia disruptiva, gera-se uma gama de ofertas de produtos. A visão de Moore sobre o ciclo de vida de uma categoria de produto é exibida na figura adiante. Uma categoria bem-sucedida de produto

experimentará grande crescimento inicialmente, seguida por um mercado maduro (etapa C) no qual se dá a consolidação. Crescimento em mercados maduros é normalmente impulsionado por aquisição de competidores e novos consumidores, bem como ganhos de eficiência. Finalmente, categorias de produtos decaem (etapa D). A qualquer momento, uma categoria pode sofrer disruptão por alguma inovação nova — de fato, uma inovação é definida como “disruptiva” baseada no seu efeito sobre categorias de produtos e modelos de negócios. Mesmo diante de disruptão, às vezes é possível manter um nicho de mercado lucrativo; por exemplo, feature phones ainda são uma categoria importante em muitos países, e a IBM ainda possui um negócio lucrativo de mainframes.

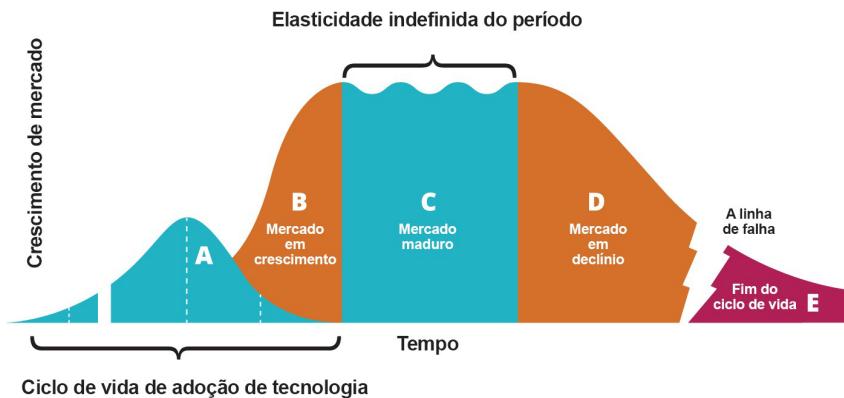


Figura 2.3: Ciclo de vida da maturidade de uma categoria, de "Lidando com Darwin" (Dealing with Darwin) de Geoffrey A. Moore (2006) — utilizado com a permissão de Portfólio, um selo editorial do Grupo Penguin (EUA) LLC.

O primeiro ponto a se observar é que produtos em etapas de maturidade diferentes variam significativamente em termos de como são administrados, desenvolvidos, promovidos e financiados. Por exemplo, em um mercado maduro, temos uma boa compreensão dos nossos clientes e do valor que estes tiram do produto; adquirir um novo cliente ou vender novos produtos a

clientes existentes é bem entendido, e novos produtos na categoria normalmente só contêm inovações incrementais. Para novas categorias, o oposto é verdade.

Se por um lado ainda existem muitos detalhes envolvidos em entender essas diferentes etapas do ciclo de vida, tais como se os nossos clientes são outras empresas ou consumidores, por outro lado, podemos chegar ao ponto importante da discussão ao simplificar drasticamente o problema olhando para duas atividades-chaves que todas as empresas vão fazer: *explorar* novas categorias de produto e modelos de negócio, e *aproveitar* os já comprovados. Esta distinção foi proposta pela primeira vez por James March (1991), em seu artigo *Exploration and Exploitation in Organizational Learning*. Steve Bleank (2005) se refere a essas atividades como “buscar” e “executar” no contexto de desenvolvimento de clientes.

Startups começam explorando novas oportunidades através de inovação no modelo de negócio: elas buscam um novo modelo de negócio que seja alinhado com a intenção e a visão dos fundadores, que entregue valor ao cliente e possa dirigir a lucratividade e o crescimento da empresa. Uma vez encontrado, o modelo de negócio é *aproveitado* crescendo e ganhando escala, encontrando formas de reduzir custos, melhorar a eficiência e aumentar a participação no mercado e base de clientes. Entretanto, todo modelo de negócio é, em última análise, transitório: em algum momento todos os modelos de negócio e as categorias de produto sofrerão disruptão causada por novidades — é apenas questão de tempo.

Explorar novas oportunidades e aproveitar as existentes são estratégias fundamentalmente diferentes, exigindo estruturas, competências, processos e mentalidade diferentes. É difícil exagerar essa questão-chave: práticas de gestão que são efetivas no domínio do aproveitamento vão conduzir ao fracasso se aplicadas ao se explorar novas oportunidades — e vice-versa. As diferenças entre os

dois domínios são listadas na *Tabela 2-1*.

**Tabela 2-1 — Explorar versus aproveitar**

	Explorar	Aproveitar
Estratégia	Inovação radical ou disruptiva, inovação de novo modelo de negócio	Inovação incremental, otimizando modelo de negócio existente
Estrutura	Equipe pequena com muitas habilidades diferentes e incluindo várias funções	Diversos times alinhados de acordo com o Princípio da Missão
Cultura	Alta tolerância de experimentação, arrisca, aceita o fracasso, foco no aprendizado	Melhoria incremental e otimização, foco na qualidade e satisfação do cliente
Gestão do risco	Maior risco é não conseguir atingir o alinhamento produto/mercado	Um conjunto mais complexo de <i>trade-offs</i> para cada produto/serviço
Objetivos	Criar novos mercados, descobrir novas oportunidades em mercados existentes	Maximizar retornos de um mercado cativo, obter resultados melhores do que os competidores
Medida de progresso	Atingir o alinhamento produto/mercado	Obter resultados que superem o plano, atingir os marcos e alvos planejados

Startups que descobrem um modelo de negócio bem-sucedido e cruzam o abismo frequentemente têm dificuldade em fazer a transição para a próxima etapa: execução e escala em um mercado em crescimento. Enquanto isso, organizações que têm sucesso em se transformar em motores de execução frequentemente perdem sua habilidade de explorar novos modelos de negócio. Eric Ries escreveu para ele mesmo um memorando imaginário descrevendo essa mudança de mentalidade:

*“Caro Eric, obrigado por teus serviços a essa empresa. Infelizmente, o trabalho que você vinha fazendo não está mais disponível. Entretanto, temos prazer em te oferecer um novo trabalho em uma empresa completamente nova, que por acaso contém todas as mesmas pessoas que antes. Esse novo trabalho começou meses*

*atrás e você já está fracassando nele. Por sorte, todas as estratégias que você desenvolveu e te fizeram bem-sucedido na empresa antiga são completamente obsoletas. Boa sorte!*” (RIES, 2008).

Um objetivo-chave para a gestão de portfólio bem-sucedida em uma empresa é entender como equilibrar a exploração de novos negócios com o aproveitamento de modelos de negócio comprovados que já existem — e como fazer com sucesso a transição de negócios de um estado para o outro. Líderes devem entender a diferença desses dois domínios e ser capazes de operacionalizar as mentalidades e estratégias muito diferentes que os governam.

## 2.1 EXPLORANDO NOVAS IDEIAS

Menos de 50% das startups estão vivas cinco anos depois de sua fundação (SHANE, 2012) — esses números variam por setor, com a taxa de sobrevivência de 5 anos para negócios de tecnologia da informação, sendo significativamente mais baixa do que a de educação ou a de saúde (SHANE, 2008). De forma semelhante, empresas desperdiçam enormes quantidades de dinheiro ao tentar desenvolver novos negócios, criando pouco ou nenhum valor para os clientes. Dados concretos são difíceis de encontrar, mas provas circunstanciais estão por toda parte.

É claro que é impossível saber com antecedência se um novo negócio será ou não bem-sucedido, mas em *A Startup Enxuta*, Eric Ries descreve um método para trabalhar em condições de extrema incerteza. A metodologia da Startup Enxuta é aplicável no contexto empresarial tal como no mundo das startups, desde que sejamos claros no seu propósito: *descobrir e operacionalizar modelos de negócios novos e potencialmente disruptivos, e para descartar rapidamente aqueles que não vão funcionar.*

Todo empreendedor, quer trabalhe em uma startup ou em uma empresa, tem uma visão do seu negócio e do impacto que isso terá sobre legiões de clientes gratos e adoradores. Para que esta visão se torne realidade, existem duas premissas principais que devem ser testadas: a hipótese de *valor* e a hipótese de *crescimento*. A hipótese de valor pergunta se o nosso negócio realmente oferece valor para os usuários através da solução de um problema real. Se assim for, podemos dizer que encontramos um ajuste *problema/solução*.

A hipótese de crescimento testa o quão rápido podemos adquirir novos clientes e se nós temos o que Steve Blank chama de um processo de vendas reproduzível e escalável — em outras palavras, se a nossa base de clientes pode rapidamente escalar o "taco de hóquei" na primeira figura do capítulo, e se temos um custo de aquisição do cliente suficientemente baixo. Se passarmos estes testes, nós temos um *ajuste de mercado/produto* e podemos avançar para as duas últimas fases de desenvolvimento do processo de desenvolvimento de clientes de Steve Blank: *criação de clientela*, na qual lançamos o nosso negócio a sério, seguido por *construção da firma*, na qual tentamos atravessar o abismo (RIES, 2008).

Na metodologia de Startup Enxuta, usamos uma abordagem sistemática, trabalhando nesse processo de forma iterativa. Começamos descobrindo o que precisamos *aprender* através da criação de uma hipótese valor. Então, decidimos o que *medir* a fim de testar essa hipótese. Daí projetamos um experimento, chamado de *produto mínimo viável* (*Minimum Viable Product* — *MVP*), que *construímos* a fim de colher os dados necessários de clientes reais para determinar se temos um ajuste *produto/mercado*.

O truque é investir o mínimo de esforço para passar por este ciclo. Já que estamos operando em condições de extrema incerteza, esperamos que a nossa hipótese de valor seja incorreta. Neste momento, nós *pivotamos*, chegando com uma nova hipótese de

valor com base no que aprendemos, e passamos pelo processo novamente. Continuamos a fazer isso até que consigamos um ajuste produto/mercado, decidamos parar de experimentar, ou nossos fundos acabem. A quantidade de tempo que temos antes que o dinheiro se esgote é conhecida como *pista*, e o objetivo é pivotar tão frequentemente quanto possível, buscando encontrar um ajuste produto/mercado, enquanto ainda temos pista sobrando.

Uma característica importante do método Startup Enxuta é que experiências são baratas e rápidas de executar comparadas com a construção de um produto completo. Em geral, somos capazes de construir um produto mínimo viável e coletar dados no espaço de horas, dias ou semanas em vez de meses ou anos, usando equipes pequenas e multifuncionais que estão focadas em executar o ciclo construir-medir-aprender de feedback mostrado na figura adiante. Esperamos que muitos experimentos vão falhar, mas alguns vão ter sucesso; no entanto, ao seguir rigorosamente os passos, cada iteração resultará em *aprendizagem validada*. Aprendizagem validada significa que testamos — no grau de precisão necessário e não mais do que isso — as premissas fundamentais por trás do nosso modelo de negócio para entender se seria ou não bem-sucedido e, em seguida, tomar a decisão de perseverar, pivotar, ou parar.

O processo da Startup Enxuta começa relativamente barato, em um contexto empresarial, podemos perseguir múltiplos modelos de negócio simultaneamente usando o Princípio da Opcionalidade.

## O QUE É UMA OPÇÃO?

A compra de uma opção nos dá o direito, mas não a obrigação, de fazer algo no futuro (tipicamente comprar ou vender um ativo a um preço fixo). As opções têm um preço e uma data de expiração. Ingressos para concertos, um acordo para sair para jantar com alguém, e uma decisão de financiar o desenvolvimento de um novo produto são exemplos de opções.

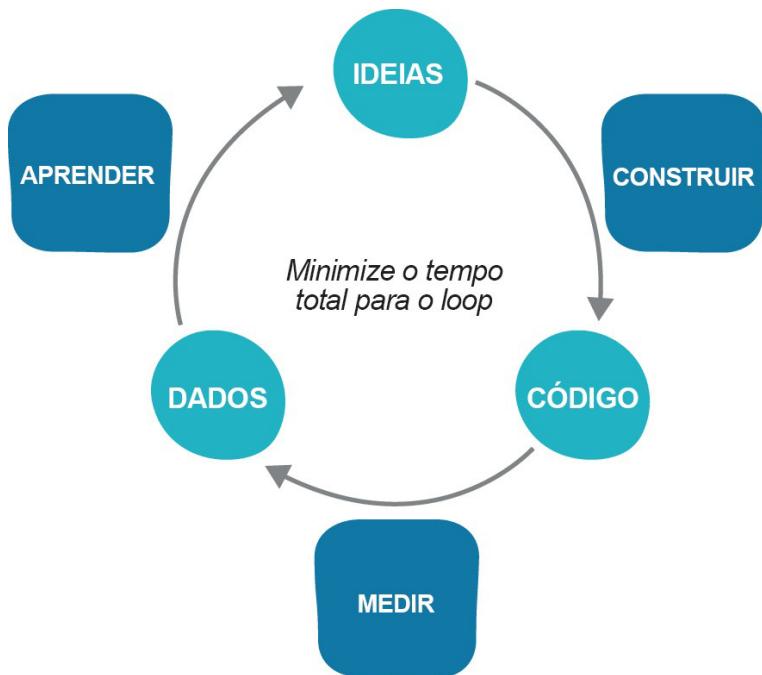


Figura 2.4: O ciclo de construir-medir-aprender

Investir uma quantia fixa de tempo e dinheiro para investigar os parâmetros econômicos de uma ideia — seja ela um modelo de negócio, produto ou uma inovação como uma mudança de processo

— é um exemplo de uso de opções para gerir as incertezas da decisão de investir mais. Limitamos a perda máxima de investimento ("downside") de qualquer ideia individual, com a expectativa de que um pequeno número de ideias vá dar um resultado extraordinário, e compensar ou anular os investimentos feitos naquelas que não o fizeram, conforme descrito em *O princípio da opcionalidade*, em *Antifragile: Things That Gain From Disorder*, de Nassim Nicholas Taleb (2012).

Essa ideia de tentativa e erro semelhantes a opções ou ajustes é explorada no livro de Nasism Taleb (2012, p. 181ff). Utilizando a linguagem do modelo Cynefin de Dave Snowden, opções são uma forma de fazer experimentos "seguros para falhar" ao projetá-los de forma a limitar os resultados negativos associados ao fracasso. Para mais sobre a aplicação de opções na gestão de TI, leia *Commitment* (MAASSEN; MATTS; GEARY, 2013).

Opcionalidade é um conceito poderoso que permite adiar decisões sobre como alcançar um resultado desejado explorando múltiplas abordagens possíveis ao mesmo tempo.

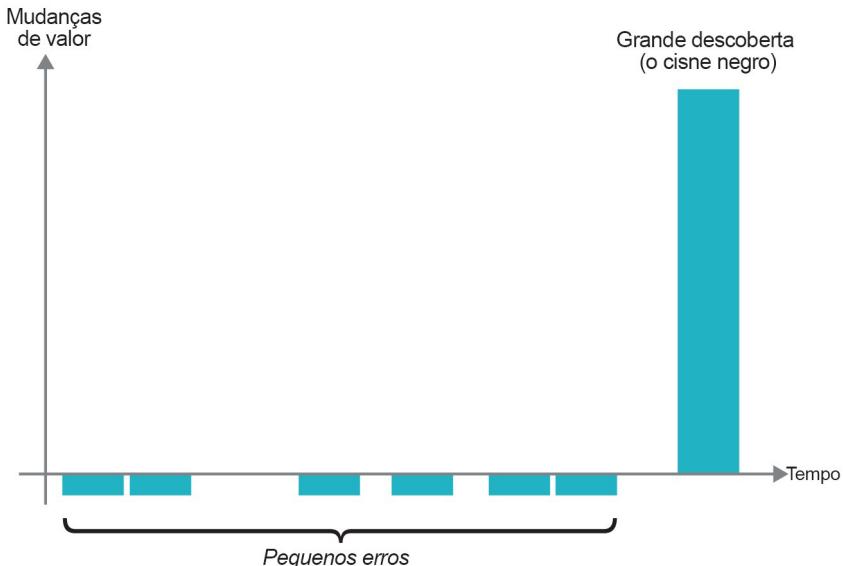


Figura 2.5: O Princípio da Opcionalidade (TALEB, 2012) — usado com permissão de Random House

## EFETIVAÇÃO

Limitando a desvantagem e garantindo que cada decisão cria, pelo menos, algumas vantagens (mesmo que seja apenas novas informações) é uma das várias técnicas que os empreendedores aplicam em situações de incerteza, onde raciocínio de causa e efeito simples (algorítmico) é um maneira inadequada de gerir risco. Em seu livro *Effectuation: Elements of Entrepreneurial Expertise*, a cientista cognitiva Dra. Saras Sarasvathy (2009) descreve uma estrutura de empreendedorismo com base em pesquisas sobre como empresários trabalham na vida real.

Para um guia sobre o modelo de efetivação, visite <http://bit.ly/1v6YjmK>.

Limitar o investimento inicial e criar escassez de recursos é essencial para gerir o risco da inovação. Dado que a maioria das ideias inovadoras que temos não será bem-sucedida, devemos conceber experimentos simples e rápidos para eliminar as más ideias de forma rápida e barata.

Consideremos o caso de a CPU ARM que está no coração de quase todos os dispositivos móveis hoje. A primeira versão deste processador foi concebida em Cambridge, Reino Unido, na década de 80 por duas pessoas, Sophie Wilson e Steve Furber. Ele foi de conceito a projeto pronto para a produção em 18 meses (BIDMEAD, 2012). Quando perguntaram ao seu chefe, Hermann Hauser, sobre como eles fizeram isso, ele disse: "Olhando para trás, quando decidimos fazer um microprocessador, eu acho que tomamos duas grandes decisões. Confiei na equipe, e lhes dei duas coisas que a Intel e a Motorola nunca tinham dado ao seu pessoal: a primeira foi nenhum dinheiro e a segunda foi nenhuma pessoa. Eles tiveram de fazer de forma simples" (TURTON, 2010).

Os conceitos centrais da Startup Enxuta são projetados para avaliar rapidamente os modelos de negócios através da identificação e testes de hipóteses de uma forma científica. Assim, eles têm aplicação para além da criação de novas empresas. Por exemplo, os princípios de restringir tempo e recursos, limitando a perda e construindo um produto mínimo viável para testar sua hipótese de valor o mais rapidamente possível com clientes reais, deveriam ser aplicados no início de qualquer empreendimento.

Devemos usar essa abordagem para explorar todas as novas ideias que têm incógnitas, incertezas e, portanto, riscos — seja para entregar novos produtos, substituindo sistemas existentes, adotando novas ferramentas, processos ou metodologias, ou implementando soluções comerciais de pacotes de software (COTS). Sempre que você ouvir falar de um novo projeto de TI começando com um

grande orçamento, equipes de dezenas ou centenas de pessoas, e um cronograma de muitos meses antes que algo realmente seja entregue, você pode esperar que o projeto vai estourar o tempo e o orçamento, e não entregará o valor esperado.

### APLICANDO A STATUP ENXUTA EM PROJETOS INTERNOS DE TI

Princípios de Startup Enxuta são igualmente importantes para projetos de engenharia de software internos, incluindo serviços e plataformas, como nuvens privadas, substituições de sistemas, e assim por diante. Iniciativas enormes, com planos de meses ou mesmo anos, surgem a toda hora para estes tipos de projetos, fazendo promessas vazias de trabalhar de forma incremental para resolver um problema real de um cliente (interno). Na verdade, as equipes que constroem estes sistemas frequentemente desdenham das necessidades e preferências de seus clientes — muitas vezes ouvimos declarações como "sabemos do que eles precisam melhor do que eles mesmos."

Projetos executados desta forma, sem entregar regularmente valor incremental para os seus clientes a fim de obter feedback, são um terrível desperdício de tempo e recursos, e raramente cumprem seus objetivos, resultados ou intenções. Mas há outras graves consequências: sistemas internos que são ruins de usar deixam funcionários frustrados, impactando a motivação e sua capacidade de fazer seu trabalho de forma eficaz. Além dos custos de um mau desempenho, as empresas criam sistemas que agregam ainda mais a complexidade de ambientes de produção já extremamente complexos. O resultado inevitável é "shadow IT" — equipes abandonando os serviços aprovados ou mantidas pelo departamento de TI em favor de algo melhor para que eles possam fazer o trabalho.

As organizações tendem a iniciar novos projetos com grandes equipes, tanto porque eles assumem (erroneamente) que isso vai ajudar a acabar antes e porque eles usam processos, tais como ciclos de orçamento anual, que estimulam a disputa pelos projetos e recursos favoritos. Na construção de sistemas complexos, no entanto, estas forças levam inevitavelmente ao inchaço do sistema, aumento da complexidade e gerenciamento de dependências, ineficiência e má qualidade. Estabelecer e tentar manter uma comunicação eficaz dentro de grandes equipes consome enormes quantidades de capacitação em projetos grandes. Enquanto isso, os sistemas criados crescem rapidamente de forma descontrolada e sem direção.

Neste ambiente, é extremamente difícil estabelecer ciclos eficazes de feedback para determinar se o que está sendo construído é valioso e alinhado à visão do produto ou projeto. Muitas vezes, não é nem mesmo possível integrar os componentes em um sistema que funcione por grande parte do projeto — e quando tentamos fazer isso, encontramos uma infinidade de problemas que devem ser resolvidos para deixar o sistema apenas em funcionamento, imagine para ser publicado? Tem sido a nossa experiência, conforme refletido na *Tabela 1-2*, que ao acrescentarmos planejamento antes de começar, o trabalho tende a fazer o resultado final pior, não melhor.

Nenhum grande empreendimento deve ser totalmente financiado antes que nós tenhamos evidência para apoiar o modelo econômico e de negócio no qual ele se baseia, e essa exploração deve ser feita com equipes pequenas e multifuncionais com orçamento limitado, como descrito na próxima parte deste livro.

## **AO EXPLORAR NOVOS MODELOS DE NEGÓCIOS, MINIMIZE O INVESTIMENTO EM DESENVOLVIMENTO DE SOFTWARE**

Uma grande rede de varejo com que trabalhamos queria abrir uma loja em um novo mercado — sua primeira expansão internacional. A equipes de TI receberam oito semanas para adaptar os seus sistemas de ponto de venda para que funcionassem no novo país, calculando um imposto sobre vendas diferente e usando uma moeda diferente. Estimamos que alterar o sistema existente para suportar diversas moedas e regimes fiscais seria um projeto de TI significativo, de muitos meses, e que necessitaria investimento significativo. Forçada a procurar opções para validar que a solução era realmente possível, a equipe codificou o novo imposto sobre as vendas diretamente no sistema de mainframe já existente e implementou um proxy simples que substituiu os símbolos de moeda em tempo real para os sistemas na nova loja. Embora a expansão internacional tenha sido cancelada por causa da crise financeira de 2008, a parte inicial de software do projeto validou a solução proposta com mínimo investimento, antes de decidir investir em uma solução totalmente funcional e robusta de longo prazo.

Uma última nota sobre exploração. Neste capítulo, nós nos concentrarmos na difusão da inovação no que diz respeito aos produtos, mas exatamente os mesmos princípios se aplicam à mudança organizacional. Muitas empresas tentam implantar novas metodologias, práticas, processos e ferramentas em toda a organização de uma só vez, ignorando o fato de que as pessoas respondem a essas inovações de maneiras diferentes e que não existe uma abordagem padrão para a adoção. É comum ver este tipo de

abordagem "big bang" não atingir os resultados esperados, e ser discretamente abandonado para dar lugar a mais uma nova iniciativa para resolver as falhas do passado.

Devemos explorar e experimentar mudanças radicais de processo — chamadas *kaikaku* na terminologia Lean — da mesma forma que exploramos potenciais novos modelos de negócio. Isto é, devemos testá-los com uma parte relativamente pequena, interfuncional da organização, com as pessoas que se enquadram na categoria de "inovadoras". Estas pessoas devem estar interessadas nos experimentos de processo propostos e ter as habilidades necessárias para executá-los. Para uma mudança que se prove valiosa, esta equipe pode ajudar na adoção em outros grupos para que ela "atravesse o abismo" até outras áreas da empresa, até que se torne a maneira padrão de se trabalhar. Entretanto, a melhoria de processos não para por aqui. Conforme discutido no próximo capítulo, todas as equipes ainda vão fazer melhorias de processo contínuas e incrementais, chamadas *kaizen*, como parte de sua atividade diária de reduzir desperdício e aumentar vazão e qualidade. Discutiremos sobre cultura organizacional mais profundamente no *capítulo 4*.

## 2.2 APROVEITANDO MODELOS DE NEGÓCIO VALIDADOS

Empresas são otimizadas para aproveitar modelos de negócio que já "atravessaram o abismo" — foram feitas para isso. Entretanto, é muito comum no trabalho de engenharia haver um gargalo quando se está evoluindo produtos existentes e introduzindo novos produtos dentro da categoria sendo explorada.

Projetos são a base do paradigma tradicional para executar o trabalho em uma empresa. Um projeto normalmente necessita que

se escreva um plano de negócios para receber orçamento, o que por sua vez leva a uma grande dose de planejamento, projeto e análise antecipados. Os diversos departamentos devem então coordenar o trabalho e executar o plano. O sucesso de um projeto é medido por completar o plano original dentro do prazo e do orçamento. Lamentavelmente, entretanto, o fato de o projeto ser "bem-sucedido" de acordo com esses critérios é irrelevante e insignificante quando comparado a se o projeto criou, de fato, valor para clientes e para a organização.

Dados obtidos ao evoluir sistemas baseados na web revelam que a abordagem de desenvolvimento de funcionalidades baseada em planos é muito fraca no que diz respeito a criar valor para os clientes e para a organização. A Amazon e a Microsoft (e muitas startups) usam uma técnica chamada teste A/B para coletar dados pesquisando se uma funcionalidade vai de fato gerar valor para o usuário antes de desenvolvê-la completamente. Ronny Kohavi, que foi diretor do grupo de Mineração de Dados e Personalização na Amazon antes de ir para a Microsoft como Gerente Geral da Plataforma de Experimentação, revela as "humildes estatísticas": entre 60% e 90% das ideias não melhoraram as métricas.

Com base em experimentos na Microsoft, 1/3 das ideias criou uma mudança positiva estatisticamente significativa, 1/3 não produziu diferença estatisticamente significativa, e 1/3 criou uma mudança negativa estatisticamente significativa (KOHAVI; CROOK; LONGBOTHAM et al., 2009). Imaginava-se que todas as ideias testadas eram boas — mas nem a intuição nem a opinião de especialistas são bons preditores do valor que uma ideia terá para os usuários.

O paradigma de projetos agrava esse problema. Projetos normalmente levam tanto tempo do início ao fim que as partes interessadas tentam forçar o máximo possível de funcionalidades

em cada um, cientes de que vai ser difícil adicionar funcionalidades uma vez que o projeto estiver completo. Além do mais, o processo de planejamento ocorre quando temos a menor quantidade de informações e entendimento dos riscos do projeto — justo no começo. Devido a um viés cognitivo conhecido como *falácia do planejamento*, executivos tendem a "tomar decisões baseadas em um otimismo delirante em vez de levar em conta ganhos, perdas e probabilidades. Eles exageram benefícios e subestimam custos. Eles vendem cenários de sucesso enquanto ignoram o potencial para enganos e erros de cálculo. Como resultado, eles perseguem iniciativas com baixa probabilidade de se manterem dentro do orçamento e do prazo, ou de entregar os resultados esperado — ou até mesmo de serem concluídos" (KAHNEMAN, 2011, p. 252).

Muitos prestadores de serviço confiam na falácia de planejamento quando apresentam propostas extremamente baixas para os serviços iniciais definidos no contrato (especialmente quando os contratos são concedidos à proposta mais baixa) e depois fazem seu lucro através de solicitações de mudança pelas quais os clientes pagam um valor maior.

À medida que realizamos um projeto, descobrimos novas informações — mas como ninguém quer ter suas funcionalidades eliminadas, novas informações geralmente levam a mais trabalho, o que é conhecido como *scope creep* (aumento gradual do escopo do projeto). Donald Reinertsen descreve isso como um ciclo vicioso de adicionar mais escopo conforme executamos o projeto e descobrimos mais informações como "a espiral do lote grande da morte" — a qual, em combinação com a falácia do planejamento, significa que projetos vão exceder tanto seu orçamento quanto o seu prazo proporcionalmente ao seu tamanho. Esse é um argumento

importante para a diminuição do tamanho dos lotes. Reinertsen (2009) tem um capítulo inteiro em seu livro para argumentar pela redução do tamanho de lote.

Todas essas funcionalidades e escopo acrescentados significam que projetos normalmente adicionam uma quantidade enorme de complexidade em ambientes de produção, os quais não são tipicamente contabilizados como parte do processo de planejamento de projeto. Essa complexidade leva a custos mais altos e trabalho não planejado no departamento de operações, e aumenta significativamente o custo e esforço necessário para executar projetos futuros.

Finalmente, uma vez que a abordagem de projetos julga pessoas de acordo com se o trabalho terminou dentro do prazo e do orçamento, não baseado no valor entregue para os clientes, a produtividade é medida pelo rendimento, não pelos resultados. Isso gera vários comportamentos nocivos. Os profissionais de produto passam a ser julgados por sua habilidade de produzir documentos de especificação abrangentes e casos de negócio bem elaborados, em vez de se os produtos e funcionalidades que eles propõe geram valor para o usuário. Desenvolvedores são recompensados por concluir código em seus próprios computadores, mas não por integrá-lo em um sistema funcionando e testado que pode suportar uso real em grande escala. Criamos uma "cultura de heroísmo" insustentável que premia o excesso de trabalho e alta utilização (garantindo que todo mundo esteja ocupado) em vez de fazer o mínimo de trabalho possível para alcançar os resultados desejados.

Alta utilização significa que o trabalho que envolve colaboração leva mais tempo para terminar, porque as pessoas que você precisa para concluir o trabalho estão sempre ocupadas com outras prioridades. Para cumprir prazos cada vez mais urgentes, as pessoas deixam de realizar manutenção e melhoria de processos de trabalho

(como a automação), que aumentaria qualidade e vazão. Isso, por sua vez, eleva o custo de fazer o trabalho no futuro, aumentando a pressão sobre a organização para "trabalhar mais" e alimentando um ciclo vicioso de excesso de trabalho.

John Seddon (2005), autor de *Freedom from Command & Control*, afirma que "o comportamento disfuncional é onipresente e sistêmico, não porque as pessoas são más, mas porque a exigência de servir à hierarquia compete com a exigência de atender os clientes. [...] A criatividade das pessoas está envolvida na sobrevivência, não na melhoria".

Como nos livramos dessa espiral descendente? Na *Parte III* deste livro, descrevemos como executar programas de trabalho em grande escala no domínio de aproveitamento, usando os seguintes princípios:

1. *Definir, medir e gerenciar resultados e não produção.*  
Aplicando o Princípio de Missão, nós norteamos nosso programa de trabalho — os resultados ideais para as partes interessadas. Então, no nível de programa, nós trabalhamos iterativamente, especificando para cada iteração os resultados mensuráveis no nível do programa que queremos obter. Fica a cargo das equipes trabalhando no programa como atingir esses resultados. Com base no feedback de consumidores reais, após cada iteração, trabalhamos para melhorar a qualidade da demanda.
2. *Gerencie para vazão em vez de capacidade ou utilização.*  
Implementamos princípios de Kanban para fazer todo o trabalho visível e limitar o trabalho em execução. Nós então buscamos parar de começar o trabalho, e começamos a terminar o mais cedo possível. Nós buscamos continuamente a melhoria de processo para reduzir o tempo de ciclo — o tempo que leva para entregar o trabalho — em todo o sistema.

Nós usamos entrega contínua e trabalhamos em pequenos incrementos para tornar barato e baixo o risco de entregar trabalho em pequenos lotes e ter ciclos de feedback mais fáceis.

3. *Garanta que as pessoas sejam recompensadas por favorecerem uma visão de longo prazo e no nível de sistema antes de buscarem objetivos funcionais de curto prazo.* As pessoas devem ser recompensadas por colaboração contínua e efetiva (ganha-ganha), por minimizarem a quantidade de trabalho necessária para alcançar os resultados desejados, e por reduzirem a complexidade dos sistemas que criamos para levar a estes resultados. As pessoas não devem ser punidas quando ocorrem falhas; em vez disso, devemos construir uma cultura de experimentação e colaboração, projetar sistemas que tornam seguro falhar, e pôr em prática processos para que possamos aprender com nossos erros e usar esta informação para tornar os nossos sistemas mais resilientes.

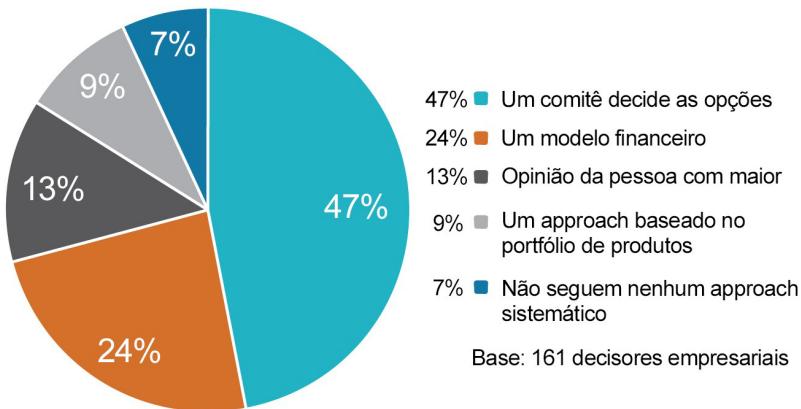
## 2.3 BALANCEANDO O PORTFÓLIO DA EMPRESA

A chave para gerenciar o portfólio de negócios de uma empresa, como qualquer outro investimento financeiro, é utilizar um modelo econômico. No entanto, essa não é uma prática amplamente difundida. Em uma pesquisa com 161 tomadores de decisão em negócios globais, exibida na figura a seguir, apenas 24% dos respondentes reportaram utilizar um modelo econômico para a tomada de decisões de investimento em produtos e serviços. Impressionantemente, 13% admitiram que a opinião da pessoa mais bem paga (conhecido como o método HiPPO — *Highest Paid Person's Opinion*) é o fator primário de decisão — termo foi criado por Ronny Kohavi, arquiteto associado na Microsoft. 47%

responderam usar o método de decisão por comitê, que é apenas um pouco menos embaraçoso.

Decisão por comitê é como a maioria das empresas desenvolve serviços de software

**“Por favor, selecione a afirmação que mais se alinha com a forma que a sua empresa decide como quais produtos vão ser construídos.”**



Fonte: estudo conduzido pela Forrester Consulting no nome de ThoughtWorks, Setembro 2012.

Figura 2.6: Como empresas tomam decisões de investimento?

Em *Escape Velocity: Free Your Company's Future from the Pull of the Past*, Geoffrey Moore (2011) apresenta um "matriz de crescimento / essencialidade" para visualizar as decisões de investimento existentes, mostrada na figura seguinte, e descreve como ela nos permite distinguir entre as empresas que têm uma estratégia de carteira eficaz e aquelas que não o fazem. O eixo Y do diagrama mede se um negócio em particular é essencial para você em relação aos outros, em que "essencial" significa que ele gera entre 5% e 10% ou mais do faturamento total ou lucro. O eixo X mede a taxa de crescimento do negócio em termos absolutos.

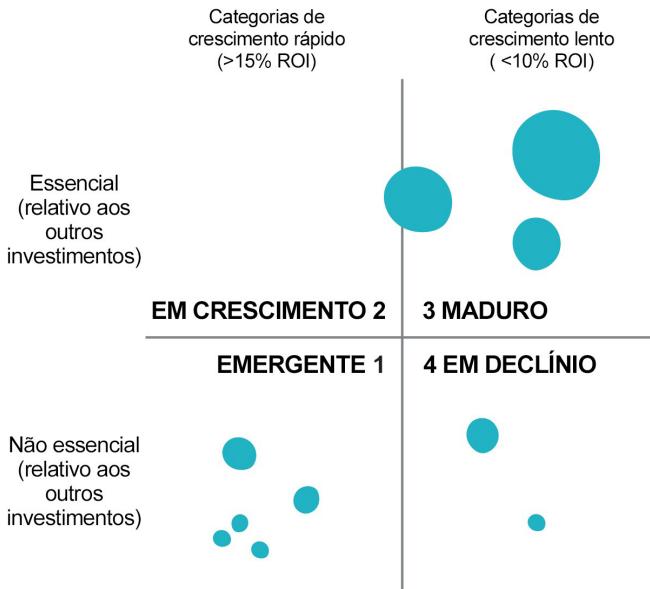


Figura 2.7: A matriz de crescimento / essencialidade, de Geoffrey A. Moore (2011) — utilizado com a permissão de HarperCollins Publishers LLC

Diversas organizações — como, em 2014, a Microsoft, IBM e HP — têm linhas que lideram o mercado no quadrante 3, que corresponde ao estágio C (um mercado maduro) na figura *Ciclo de vida da maturidade de uma categoria*, de "Lidando com Darwin". Porém, como mostrado na figura anterior, nenhuma foi capaz de desenvolver (em oposição a adquirir) uma grande linha de produtos no estágio B (correspondente ao quadrante 2) apesar de investimentos em P&D significativos que levaram a novos negócios no quadrante 1. Por outro lado, a Amazon, o Google e a Apple criaram na última década negócios que cresceram rapidamente para se tornar essenciais para a empresa.

Para entender por que tantas empresas deixam de criar negócios no quadrante 2, devemos entender a dinâmica do portfólio da empresa. Isso é descrito no modelo de 3 horizontes apresentado em Baghai, Coley e White (1999), exibido na figura a seguir. O

horizonte 1 consiste do seu conjunto de categorias de produtos e negócios centrais (correspondente ao quadrante 3 na figura anterior).

## ADMINISTRANDO UM PORTFÓLIO

Os três modelos de Horizonte



Figura 2.8: Três Horizontes, de Geoffrey A. Moore (2011) — utilizado com a permissão de HarperCollins Publishers LLC

Investimentos em negócios no horizonte 1 darão resultado no mesmo ano, e normalmente tomam a forma de desenvolver produtos existentes e lançar novos dentro das categorias existentes. Horizonte 2 é o conjunto de negócios emergentes que formarão o centro do negócio no futuro. Estes necessitam de investimento significativo e atenção das divisões de marketing e vendas para serem bem-sucedidos, mas não terão o mesmo retorno que os

investimentos no horizonte 1.

Horizonte 3 é o domínio da Startup Enxuta, onde experimentamos com novos modelos de negócio e tentamos criar alinhamento produto/mercado para novos negócios. Nós buscamos investir tempo e dinheiro o suficiente para criar pistas para descobrir o alinhamento produto/mercado antes de fazer investimentos futuros. Então, nós ou movemos a ideia para o horizonte 2 ou a arquivamos, talvez até que as condições do mercado ou avanços tecnológicos a tornem favorável novamente.

Existem três problemas significativos conspirando para eliminar negócios que chegam ao horizonte 2. Primeiro, eles necessitam de um investimento significativo em termo de recursos de pesquisa, venda e marketing sem que tragam o retorno correspondente de faturamento — a métrica pela qual esses departamentos são tradicionalmente medidos. Segundo, cada um dos três horizontes necessita de práticas muito diferentes de administração e apoio para ser bem-sucedido, conforme exibido na *Tabela 2-2*. Escolher cegamente uma abordagem consistente para cada resultará em fracasso. Finalmente, como Clayton Christensen (1997) discute em *The Innovator's Dilemma*, empresas rentáveis frequentemente são relutantes em canibalizar seu lucro e participação no mercado, lançando novos produtos disruptivos — aqueles que podem ameaçar seu resultado final e avaliação de mercado.

**Tabela 2-2 — Três horizontes**

	<b>Horizonte 1 (0-12 meses)</b>	<b>Horizonte 2 (12-36 meses)</b>	<b>Horizonte 3 (36-72 meses)</b>
<i>Objetivos</i>	Maximizar o retorno econômico	Atravessar o abismo, começar a contribuir com faturamento significativo	Criar um novo negócio
.			

	<b>Horizonte 1 (0-12 meses)</b>	<b>Horizonte 2 (12-36 meses)</b>	<b>Horizonte 3 (36-72 meses)</b>
<i>Métricas-chave</i>	Faturamento comparado ao plano, participação de mercado, lucratividade	Taxa de vendas, contas-chave	Empolgação / popularidade no boca a boca (consumidor final), clientes famosos (empresas)

Nós frequentemente vemos essas forças conspirando para impedir que negócios sobrevivam ao horizonte 2, mesmo em empresas que fazem um bom trabalho tanto de explorar quanto de aproveitar modelos de negócio. Frequentemente, outras empresas acabam trazendo eles para o mercado com resultados avassaladores. O PARC da Xerox inventou a GUI moderna (bem como diversos outros elementos da computação moderna). Mas os "cabeças de toner" na matriz da Xerox "não tinham a menor ideia do que era um computador e o que ele poderia fazer" e assim, acabou sendo a Apple e a Microsoft que trouxeram os computadores para as salas de estar da população (Steve Jobs, *The Lost Interviews*).

A gigante da fotografia Kodak, que pediu falência em 2012, de fato inventou a câmera digital. Steve Sasson e seu time do Laboratório de Pesquisa da Divisão de Aparelhos Kodak criou a dramática inovação em 1975 (JARVIS, 2008). Entretanto, o time se defrontou com gerentes confusos que não conseguiam compreender por que consumidores iriam querer ver fotografias em uma tela de computador. O negócio estava otimizado para revelar fotografias — produzindo papel, filme e outros suprimentos —, não capturando memórias.

## **POR QUE VOCÊ NÃO PODE SIMPLESMENTE CONTRATAR OU COMPRAR O RUMO PARA INOVAÇÃO**

Diversas empresas têm comprado startups em uma tentativa de escolher a tendência atual e acelerar a inovação — talvez para diversificar e equilibrar o seu portfólio, para se tornar um "laboratório de inovação", ou marcar uma caixinha no estágio A, quadrante 1, horizonte 3. Temos observado de perto os maus resultados que isso cria, com essas compras fracassando em produzir o retorno esperado e os membros mais experientes do time saindo assim que eles puderem sacar suas ações. Os problemas ocorrem quando a empresa comprada — trabalhando em um produto de horizonte 3 ou 2 — é submetida à governança horizonte 1, metas financeiras, e as estruturas de gestão da empresa adquirente, destruindo completamente a sua capacidade de inovar. Às vezes, as pessoas da organização-mãe são rodadas através do laboratório de inovação na esperança de que isso vá magicamente ensiná-los a inovar em um horizonte diferente, em vez de simplesmente dar-lhes um choque cultural.

Uma aquisição para contratar os talentos (*acqui-hiring*) frequentemente falha pela mesma razão: pegar pessoas ótimas e colocá-las em culturas patológicas ou burocráticas não muda a cultura — quebra as pessoas. A solução é fazer o trabalho duro para transformar a cultura da sua própria organização e fomentar liderança eficaz e uma gestão adequada para cada horizonte, que, aliás, remove a necessidade de contratar ou adquirir inovadores.

Nossa hipótese é de que organizações sobrevivem e crescem no

médio e longo prazo ao balancear a habilidade de explorar continuamente novos modelos de negócio em potencial com o aproveitamento efetivo dos modelos existentes. De fato, um dos traços marcantes de uma organização verdadeiramente adaptativa e resiliente é que ela continuamente causa disruptão em seus próprios modelos de negócio em busca de futuras oportunidades e novos mercados e consumidores.

Por exemplo, a busca da Amazon por livros eletrônicos e a produção do Kindle causaram disruptão no que era, então, seu principal modelo de negócios: vender livros físicos. A criação do Amazon Marketplace permitiu outros fornecedores tirarem vantagem da infraestrutura da Amazon e potencialmente vender com preços mais baixos os mesmos produtos vendidos por ela. A 3M define a sua estratégia como constante inovação de novos produtos e define metas para a porcentagem de faturamento de produtos introduzidos nos últimos 5 anos em 30%, a qual foi excedida em 2008 (GUNTHER, 2010). Inge G Thullin, Presidente e CEO da 3M, espera que esse número chegue a 40% até 2017 (CARUSO-CABRERA, 2013).

A Intuit usa um modelo simples para equilibrar horizontes 1, 2 e 3, como pode ser visto na *Tabela 2-3*. A Google segue um modelo semelhante, mas com alocações diferentes: 70% para horizonte 1, 20% para horizonte 2 e 10% para horizonte 3 (BATTELLE, 2005).

**Tabela 2-3 — Os horizontes e métricas da inovação na Intuit**  
[\(<http://bit.ly/1v6YI8Q>\)](http://bit.ly/1v6YI8Q)

	Negócios existentes	Negócios adolescentes	Ideias
Investimento	60%	30%	10% das despesas operacionais, financiada a cada quarto, baseada em aprendizado validado

	<b>Negócios existentes</b>	<b>Negócios adolescentes</b>	<b>Ideias</b>
<i>Métricas</i>	Crescimento da categoria, participação, <i>net promoter</i> , faturamento	Crescimento, eficiência crescente (levará à lucratividade)	Medidas de amor baseadas em trazer benefícios para o cliente, uso ativo do produto, boca a boca proativo
<i>Exemplo de produtos</i>	Turbo Tax, Mint	QuickBooks Online Accounting	SnapTax

O ponto mais importante a se ter em mente no que diz respeito ao equilíbrio entre horizontes é que, a menos que a liderança sênior tenha um papel ativo na gestão de investimentos, incluindo estabelecer práticas de gestão adequadas para diferentes horizontes e prestar atenção em como os gestores são incentivados, os negócios centrais sempre vão encontrar uma maneira para usar sua influência corporativa para escantejar e, finalmente, neutralizar os outros horizontes. Se as barreiras culturais e de gestão forem simplesmente fortes demais para esse tipo de abordagem "ambidestra", a alternativa é criar uma unidade de negócio o mais independente possível.

## **COMO A AETNA CRIOU NOVAS COMPANHIAS PARA DISRUPTAR SEU NEGÓCIO CENTRAL**

Assim como todos os outros participantes no mercado Norte-Americano de saúde, a Aetna sabia que a Lei do Cuidado Acessível (*Affordable Care Act*) do governo Obama representava tanto um risco sério quanto uma oportunidade significativa. Com 160 anos de idade na época que a lei foi assinada, a Aetna decidiu criar uma nova companhia chamada Healthagen, "uma organização separada, financiada separadamente, compensada separadamente, e administrada separadamente, por isso ela não está sujeita ao mesmo processo de gestão da Aetna", a fim de causar disruptão no mercado de planos de saúde com novos modelos de tecnologia e de negócios. A Healthagen tem um alvo de gerar entre \$1.5 e 2.5 bilhões de dólares em faturamento inicialmente (TECHNOMY, 2013).

A Aetna também criou outra subsidiária com ideias gerais semelhantes para criar um mercado para consumidores e gerar modelos de fundo de pensão. Mark Bertolini, Presidente do Conselho de Administração, CEO e Presidente da Aetna, afirma que seu objetivo é construir um ecossistema competitivo baseado na tecnologia que vai causar disruptão no modelo de negócio da própria Aetna.

## **2.4 CONCLUSÃO**

Cada ideia tem seu próprio ciclo de vida, com ideias bem-sucedidas criando vantagens competitivas para os primeiros seguidores (*Early Adopters*) e, por fim, se tornando a fundação para

inovação de nível mais alto. Empresas devem garantir que têm uma série de novas ideias para prover a base do crescimento no futuro. Uma gestão efetiva do portfólio da empresa requer a criação e aplicação de um modelo econômico para equilibrar investimentos entre os três horizontes. Para leitura adicional em gestão de portfólio, recomendamos o livro de Geoffrey Moore, *Escape Velocity: Free Your Company's Future from the Pull of the Past*.

Esperamos que existam diversas ideias para novos negócios sendo incubadas no horizonte 3. Como não podemos prever quais serão bem-sucedidos, nós aplicamos o Princípio da Opcionalidade e assumimos que muitos falharão, mas alguns serão bem-sucedidos. Nós aplicamos a metodologia da Startup Enxuta para girar rapidamente entre modelos de negócio para esses negócios até que as equipes explorando-os esgotem seus recursos ou descubram um alinhamento produto/mercado e ganhem impulso. A maioria das ideias nunca chega ao horizonte 2, mas aquelas que chegam necessitam de uma abordagem gerencial fundamentalmente diferente. No horizonte 3, nós nos importamos principalmente com encontrar alinhamento produto/mercado, mas no horizonte 2 precisamos identificar e administrar para um conjunto maior de riscos específicos para o nosso negócio. Em vez de inovação no modelo de negócio, mudamos para inovação incremental, que requer um conjunto diferente de habilidades.

Empresas demais matam a inovação ao tentar gerenciar investimentos de horizonte 2 e 3 usando estratégias de horizonte 1. No restante deste livro, vamos geralmente ignorar horizonte 1 (apesar de que muitos dos princípios e técnicas descritas na *Parte III* possam ser úteis de se aplicar para este domínio). A *Parte II* deste livro lida com o domínio da exploração que usaremos em investimentos no horizonte 3. A *Parte III* discute como se movimentar rápido em grande escala no domínio de aproveitamento, utilizando os mesmos princípios Lean que têm sido

aplicados em manufatura a décadas para continuamente proporcionar maior qualidade e menores custos. Na *Parte IV* discutiremos como transformar a sua empresa, iniciando por cultivar uma cultura de inovação.

## Questões para os leitores

- Qual estrutura sua organização utiliza para equilibrar o portfólio de explorar novos modelos de negócio, aproveitar os já validados e desenvolver seus negócios principais?
- Existe um lugar onde se pode ver todo esse portfólio de uma vez?
- Quais métricas de desempenho são usadas para medir a saúde das atividades em cada um desses domínios?
- Qual a porcentagem de investimentos nos horizontes 1, 2 e 3 em sua organização? Ela é intencional ou acidental? Qual você acha que deveria ser?
- Como a liderança sênior garante que investimentos em horizontes 2 e 3 sejam fomentados e que transições entre horizontes sejam administradas de forma que maximize as métricas de desempenho relevantes para cada investimento individual?

# Investigue

“Aos melhores, falta convicção, enquanto que os piores estão cheios de intensidade apaixonada.”\* — W. B. Yeats

Quando enfrentamos uma nova oportunidade ou problema a ser resolvido, nosso instinto, como humanos, é pular direto para a solução sem investigar adequadamente o tamanho do problema, testar as suposições inerentes à solução proposta ou nos desafiar a validar a solução com usuários reais.

Podemos ver esse instinto no trabalho quando projetamos novos produtos, adicionamos novas features a produtos existentes, abordamos problemas de processos e organizacionais, começamos projetos ou substituímos sistemas existentes. É a força que nos leva a comprar ferramentas caras que parecem resolver todos os nossos casos, implantar uma nova metodologia ou ânimo organizacional pela empresa toda, ou investir em programas de trabalho que põem a companhia em risco.

Pior, nós frequentemente nos apaixonamos pelas nossas próprias soluções e, em seguida, caímos na falácia do custo irrecuperável quando ignoramos evidências que deveriam nos fazer questionar se deveríamos continuar buscando-as. Quando combinadas com uma posição de poder, essas forças podem ter consequências catastróficas — um dos nossos colegas quase foi demitido por um cliente por ter a audácia de perguntar sobre o case de negócio por trás de um projeto em particular.

Se tivéssemos um superpoder, seria magicamente aparecer

sempre que um problema ou oportunidade estivesse em discussão. Nossa missão seria evitar que qualquer um iniciasse um grande programa para resolver o problema ou buscar a oportunidade até que fizesse o seguinte:

- Definisse o resultado de negócio mensurável a ser alcançado;
- Construísse o menor protótipo possível capaz de mostrar um progresso mensurável em direção àquele resultado;
- Demonstrasse que a solução proposta realmente fornece valor para o público ao qual é destinada.

Como somos meros mortais, confiamos que você manterá uma cópia deste livro à mão para consultar no momento apropriado.

Nesta parte, discutimos como investigar oportunidades e tamanhos de problemas ao usar uma abordagem científica e sistemática na solução de problemas. Ao lançar mão de uma abordagem experimental, podemos gerenciar de maneira eficaz os riscos, e permitir que os times tomem melhores decisões e façam melhores julgamentos em relação à incerteza que é inerente à inovação.

## CAPÍTULO 3

# MODELE E MEÇA O RISCO DO INVESTIMENTO

“A dúvida não é uma condição agradável, mas a certeza é um absurdo.”\* — Voltaire

Para empresas que experimentam novos modelos de negócio e produtos, como as startups, o maior risco é uma falha em criar algo que realmente entregue valor para os usuários. A estrutura Lean Startup permite que rapidamente descartemos ideias que não entreguem valor ou que não serão adotadas rápido o suficiente, não desperdiçando nossos recursos com elas. Contudo, os princípios por trás da Lean Startup podem ser aplicados a todos os tipos de atividades dentro da empresa, como melhoria de processo, mudança organizacional, substituição de sistemas e programas de GRC (governança, risco e conformidade).

Neste capítulo, apresentaremos os princípios e conceitos que nos permitem ter uma abordagem sistemática para gerenciar o risco do trabalho planejado, reunindo informações de maneira a reduzir a incerteza. Essa estrutura de trabalho forma a base para uma abordagem prática para explorar novas oportunidades que apresentamos ao longo da *Parte II*.

### 3.1 MODELO DE RISCO DE INVESTIMENTO

Normalmente, nas empresas devemos montar um plano de negócio antes de termos a aprovação para continuar com um investimento. Isso geralmente envolve um time de pessoas criando um documento detalhado que estima o valor que a iniciativa proposta criará. O plano de negócio descreve os recursos necessários, as dependências e, finalmente, um lindo conjunto de números detalhando o trabalho planejado com custos, as métricas-chaves, um plano de recursos e os prazos. Dependendo do nível de detalhes e do investimento estimado requerido, esse processo pode levar semanas ou meses para ser completado.

Um objetivo importante deste processo de planejamento é apoiar uma decisão de investimento. Para tomar essa decisão, temos de ter um bom entendimento dos riscos envolvidos no investimento. Segundo Douglas Hubbard (2010, p. 50), definimos risco como “um estado de incerteza em que algumas das possibilidades envolvem perda, catástrofe ou outro resultado indesejável”, e a medição de risco como “um conjunto de possibilidades, cada uma com probabilidades quantificadas e perdas quantificadas”. Por exemplo: “Acreditamos que há 50% de chance que o projeto será cancelado, com uma perda de \$ 2 milhões de trabalho incompleto”.

Em *How to Measure Anything*, Hubbard (2010, p. 111) fala sobre seu trabalho analisando planos de negócios para investimentos em TI:

"Cada um desses planos de negócio teve 40 ou 80 variáveis, como custos de desenvolvimento iniciais, taxa de adoção, melhoria de produtividade, aumento de receita, e assim por diante. Para cada um desses planos, rodei uma macro no Excel que computou o valor para cada variável. Usei esse valor para descobrir onde focar os

esforços de mensuração.

Quando eu rodei a macro que computava o valor da informação para cada uma dessas variáveis, comecei a ver este padrão: 1) A vasta maioria das variáveis tinha um valor de informação zero; 2) As variáveis que tinham altos valores de informação eram rotineiramente aquelas que o cliente nunca mediou; 3) As variáveis em que os clientes costumavam gastar mais tempo medindo eram normalmente aquelas com valor de informação muito baixo."

Pegue o exemplo de estimar custos de desenvolvimento para obter a aprovação do projeto. Isso normalmente envolve a análise de meses de trabalho futuro, dividindo em pequenos pedaços e estimando o esforço necessário para cada um deles. Contudo, Hubbard nota que "mesmo em um projeto com custos de desenvolvimento muito esclarecidos, descobrimos que aqueles custos não são muito significativos para a decisão de investimento... A coisa mais importante é se o projeto será cancelado... A próxima variável mais importante é a utilização do sistema, incluindo quão rápido ele é implantado e se algumas pessoas vão efetivamente usá-lo" (HUBBARD, 2007).

Sendo assim, o plano de negócio essencialmente se torna um livro de ficção científica baseado em um universo bem pouco conhecido — ou que nem deve existir! Enquanto isso, um tempo significativo é perdido em planejamento detalhado, análise e estimativas, que resultam em grandes quantidades de informação com valor extremamente limitado.

De acordo com uma pesquisa de Donald Reinertsen (2009), autor de *The Principles of Product Development Flow: Second Generation Lean Product Development*, é normal que 50% do tempo de desenvolvimento de um produto seja gasto em "atividades vagas de front-end". Naturalmente, isso leva a decisões de investimento fracas, e desnecessariamente longos ciclos de desenvolvimento de

produto. Isso cria múltiplos resultados negativos:

- Longos ciclos de desenvolvimento de produto reduzem dramaticamente o potencial retorno sobre o investimento que podemos obter de novos produtos bem-sucedidos.
- Mais fatalmente, longos ciclos de desenvolvimento atrasam o tempo que se leva para obter feedback do cliente sobre estarmos construindo algo valioso.
- Atividades de pesquisa de mercado típicas são fracas para prever a relação produto-mercado, especialmente em novas categorias de produto. Pesquisas disseram que minivans e iPods não seriam bem-sucedidos.
- Na falta de bons dados, as pessoas tendem a escolher seus projetos preferidos para serem financiados. Particularmente em TI, vemos frequentemente enormes quantias de dinheiro sendo jogadas no ralo em projetos de substituição de sistemas — até mesmo (talvez especialmente?) em organizações operando em setores altamente regulados.

Há dois fatores com que nos preocupamos em um plano de negócios. O primeiro é a sensibilidade da métrica-chave para as muitas variáveis do case de negócio. O segundo é o nível de incerteza nas variáveis para as quais a métrica é sensível. Dadas as distribuições e a extensão para as variáveis-chaves, uma abordagem simples, mas poderosa, seria a simulação Monte Carlo para ver quais seriam os possíveis resultados. Isso permitiria encontrar as variáveis às quais devemos prestar atenção para tomar boas decisões de investimento.

Para fazer uma simulação Monte Carlo, usamos um computador

para criar milhares de cenários randômicos baseados na forma de distribuição e alcance para as variáveis de entrada, e então computamos o valor da métrica na qual estamos interessados em cada cenário. O resultado de uma simulação Monte Carlo é um histograma, com o número de cenários para cada valor no eixo Y e os alcances no eixo X. Você pode fazer a simulação Monte Carlo usando o Excel, ou use uma das muitas ferramentas customizadas existentes.

Veja <http://www.howtomeasureanything.com> para um exemplo. Para uma introdução à simulação Monte Carlo para modelos de negócio, veja <http://bit.ly/1vKoXBE>.

O resultado de uma simulação Monte Carlo para um plano de negócios pode ser parecido com a figura seguinte. Como ressalta Hubbard, a incerteza do ROI para programas de TI tende a ser bem alta e aumenta com a duração do programa.

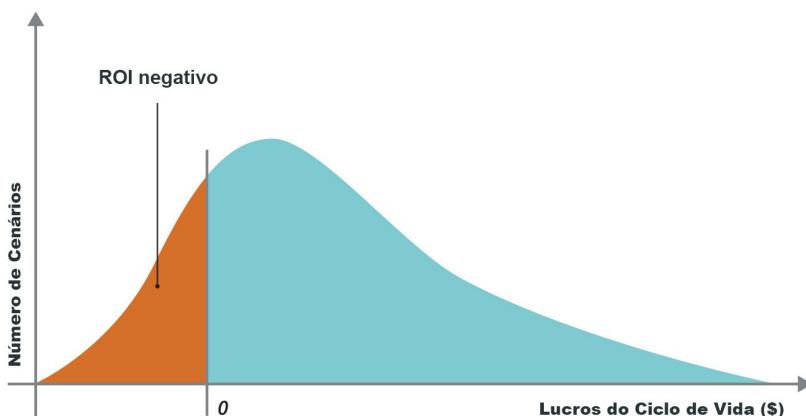


Figura 3.1: Resultado de uma simulação Monte Carlo

Como você pode averiguar ao fazer uma simulação Monte Carlo em seus próprios planos de negócios, o ROI em programas de TI não é muito sensível ao custo, mas sim ao fato de o programa ser cancelado e à utilização do sistema resultante. Essas variáveis dependem primariamente de termos construído a coisa certa. Contudo, o processo-padrão de planejamento de uma empresa fornece quase nenhuma validação disso.

Vamos ser absolutamente claros. Na maioria dos empreendimentos, cerca de 30% a 50% do tempo total para comercializar é gasto em atividades que fornecem quase zero valor em termos de mitigação de riscos de nossos investimentos. Essa atividade de valor quase nulo é principalmente guiada por gerenciamento financeiro e processos de planejamento.

Em nossa experiência, o front-end vago apresenta a maior oportunidade para uma melhoria de processo radical (*kaikaku*) em empresas. Podemos drasticamente reduzir o tempo necessário e tomar melhores decisões ao usarmos uma abordagem sistemática em gerenciamento de risco. Neste capítulo, discutimos como atacar o front-end vago para novos negócios e novos produtos. No *capítulo 7*, mostraremos como mudar a maneira que backlogs em nível de programa são gerenciados.

### 3.2 APLICANDO O MÉTODO CIENTÍFICO EM DESENVOLVIMENTO DE PRODUTO

“A maneira de o mundo lhe dizer se o que você está fazendo é valioso é se estão lhe mandando dinheiro.”\* — Donald Reinertsen

Quando há muita incerteza na métrica-chave da qual cuidamos, começamos identificando as variáveis com o maior valor de informação — as suposições mais arriscadas. São a elas que nossa métrica é mais sensível. Tanto no caso de inovação de modelo de negócio quanto no desenvolvimento de produto, Donald Reinertsen afirma que “vendas unitárias são o pulo do gato”.

A maneira mais ineficaz de testar um modelo de negócio ou uma ideia de produto é planejar e construir um produto completo para ver se o mercado previsto para ele realmente existe. Mesmo assim, é exatamente isso que fazemos, uma vez que temos um case de negócio aprovado. Parte do problema é a linguagem que usamos para descrever o processo de desenvolvimento do produto. Por exemplo, considere o termo “requisitos”. De quem são esses requisitos? São requisitos dos usuários? Em *Lean IT*, Steve Bell e Moke Orzen (2011, p. 48) comentam que “usuários são frequentemente incapazes de explicar exatamente do que precisam, porém parecem insistentes sobre o que não querem, já que o estão vendendo”.

Deveríamos parar de usar a palavra “requisitos” em desenvolvimento de produto, pelo menos no contexto de features não triviais. O que temos, em vez disso, são hipóteses. Acreditamos que um modelo de negócios particular, ou produto, ou features, será valioso para os clientes. Mas precisamos testar nossas suposições. Podemos utilizar uma abordagem científica para testar essas suposições, fazendo experimentos.

No caso de inovação do modelo de negócios e produto, o movimento Lean Startup nos fornece uma estrutura de trabalho para operar em condições de extrema incerteza. Em *Running Lean*, Ash Maurya (2010a) explica como executar o modelo Lean Startup:

- Não gaste muito tempo criando um modelo de negócio sofisticado. Em vez disso, projete um canvas de modelo

de negócio simplificado que capture e comunique as suposições-chaves de operação de seu modelo de negócio proposto.

- Reúna informações para determinar se você tem um problema que vale a pena ser resolvido — ou seja, que é solucionável e que as pessoas pagarão para tê-lo solucionado. Se houver essas duas condições, conseguimos o encaixe problema-solução.
- Então, projete um MVP (*minimum viable product*, em inglês) — um experimento feito para maximizar o aprendizado vindo de potenciais *primeiros seguidores* com o mínimo esforço. No caso de os resultados do MVP invalidarem sua hipótese de produto, pivote e comece de novo. Continue este processo até você decidir abandonar o problema inicial, ficar sem recursos ou descobrir um encaixe produto-mercado. Neste último caso, saia da fase de investigação e siga para a exploração do modelo validado.
- Ao longo desse processo, atualize o canvas do modelo de negócio baseado no que você aprendeu ao conversar com clientes e testar os MVPs.

Apresentaremos esta abordagem em detalhes no capítulo seguinte.

Há duas inovações-chaves neste modelo. Primeiro, paramos de usar planejamento detalhado como modo de gerenciar risco. Em vez disso, encontramos clientes e executamos experimentos baratos para descobrir se nosso modelo de negócio ou produto proposto é realmente valioso para eles. Segundo, mais do que criar apenas um plano, nós repetimos uma série de experimentos de maneira a descobrir um encaixe produto-mercado. Dadas as condições de

incerteza, é pouco provável que nossa primeira ideia dê frutos.

Uma objeção comum a esses princípios é que tais experimentos não podem ser representativos de um produto completo. Essa objeção é baseada em um falso entendimento de mensuração. O objetivo da mensuração não é ter certeza, mas reduzir a incerteza. O trabalho de um experimento é reunir observações que reduzam a incerteza quantitativamente (HUBBARD, 2010, p. 23). O princípio-chave para se ter em mente é este: quando o nível de incerteza de alguma variável é alto, temos muito pouca informação para reduzir essa incerteza significativamente.

### **DEFINIÇÃO DE MENSURAÇÃO**

Mensuração: uma redução de incerteza expressa quantitativamente baseada em uma ou mais observações (HUBBARD, 2010, p. 23).

Essa definição pode parecer contraintuitiva, a menos que você tenha experiência em executar experimentos em um contexto científico. Em ciência experimental, o resultado de uma mensuração nunca é simplesmente um valor único. É, preferencialmente, uma distribuição de probabilidade que representa o alcance de possíveis valores, como mostrado na figura adiante.

Qualquer mensuração que não indique a precisão do resultado é considerada praticamente sem significância. Por exemplo, uma medição da minha posição com a precisão de 1 metro é muito mais valiosa que essa mesma posição com a precisão de 500 milhas. O sentido de se investir em mensuração em um contexto científico é reduzir nossa incerteza em relação ao real valor de uma certa quantidade. Assim, em particular, se expressamos nossas estimativas em números precisos (ao

contrário de gamas), estamos nos preparando para o fracasso: a chance de encontrarmos uma data seis meses no futuro, no dia preciso, é praticamente nula.

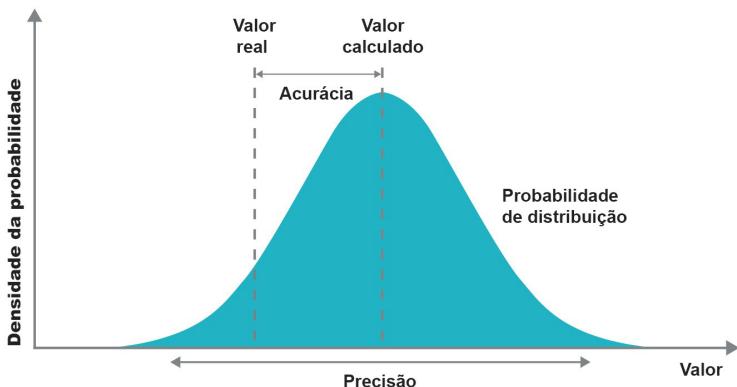


Figura 3.2: Acurácia e precisão

Um produto mínimo viável pode ser pensado como uma maneira de conduzir uma mensuração relativamente barata, de forma a reduzir nossa incerteza em relação à nossa métrica-chave. É isso que faz do MVP um investimento tão bom. Tipicamente, montar um plano de negócio e requerimentos para obter uma iniciativa significativa leva semanas ou meses dentro de uma empresa. No mesmo período, ao seguir o modelo de Lean Startup, podemos executar múltiplos experimentos, aprender com consumidores reais, e emergir com um plano superior e testado, baseado em evidência. Vamos examinar as diferenças entre essas duas abordagens quando precisamos tomar uma decisão de investimento, como mostrado na *Tabela 3-1*.

#### **Tabela 3-1 — Ciclo de vida tradicional versus ciclo de vida Lean Startup**

Pergunta	Processo de planejamento de projeto tradicional	Processo de descoberta — Lean Startup
Quais dados temos para tomar a decisão de investimento?	Um plano de negócio baseado em um conjunto de hipóteses não testadas e suposições, apoiado em estudos de caso e pesquisa de mercado.	Dados reais baseados em evidência compilados de um produto ou serviço em operação, testado com clientes reais.
O que acontece depois?	Devemos criar requerimentos detalhados, se ainda não os temos, e então começar um projeto para construir, integrar, testar e finalmente lançar o produto.	Já temos um MVP validado no qual podemos trabalhar imediatamente com novas features e melhorias baseado em feedback do cliente.
Quando descobrimos se a ideia é boa (isto é, vai ter um bom retorno sobre o investimento)?	Uma vez que o projeto está completo e o produto ou o serviço está no ar.	Já temos esta evidência, baseado em dados que coletamos.

Como discutido no *capítulo 2*, um fator importante no sucesso da abordagem Lean Startup é limitar o tamanho do time de exploração e os recursos disponíveis para eles (incluindo o tempo). Isso encoraja as pessoas a aplicarem sua criatividade e foco no aprendizado, em vez de ficarem buscando soluções “perfeitas”. Não existem prêmios para a elegância do design do software ou cobertura de teste automatizada em um MVP. Muitas das “histórias de guerra” compartilhadas pelos praticantes da Lean Startup descrevem os atalhos engenhosos que tomaram na busca por um aprendizado válido.

Claro que uma questão razoável é: uma vez que o desenvolvimento do produto é efetivamente uma forma de descoberta, quanto tempo e dinheiro devemos gastar em um aprendizado validado? A teoria dos jogos fornece uma fórmula para o Valor de Informação Esperado (*expected value of information*, ou EVI, em inglês). Uma discussão detalhada de como calcular este

número vai além do escopo deste livro, mas é tratada no livro de Hubbard (2010), *How to Measure Anything*.

O EVI nos dá um limite superior do quanto deveríamos estar preparados para pagar para reunir a informação em questão. Se o custo de fazer essa medição é muito menor que o EVI (digamos, uma ordem de grandeza menor), é claramente válido fazer a medição. Assim, quanto mais arriscado e caro o projeto em questão, mais valor você dá ao seu dinheiro ao buscar uma abordagem Lean Startup.

#### VALOR DE INFORMAÇÃO ESPERADO

Hubbard define o valor de informação como: “De modo grosseiro, o valor da informação é igual à chance de estar errado vezes o custo de estar errado. O custo de se estar errado — isto é, o que é perdido se sua decisão não funcionar — é chamado uma perda de oportunidade.

Como um exemplo simples, digamos que você está considerando investir \$1 milhão em um novo sistema. Ele promete um ganho líquido de \$3 milhões em três anos (para o nosso exemplo, será um completo sucesso ou uma falha total). Se você investir, mas o sistema falhar, seu erro custa a você \$1 milhão. Se você decidir não investir quando deveria investir, o erro custará \$3 milhões. Quando multiplicamos a perda de oportunidade pela chance de um erro, temos a perda de oportunidade esperada (*expected opportunity loss*, ou EOL, em inglês). Calcular o valor da informação se reduz a determinar em quanto será reduzido o EOL (HUBBARD, 2007).

Na realidade, o sucesso de um produto raramente é um resultado binário. Se voltarmos ao exemplo do ROI previsor para um plano de negócio ilustrado na figura *Resultado de uma*

*simulação Monte Carlo*, temos o EOL ao calcular a área sombreada da curva, que representa os cenários em que perdemos dinheiro em nosso investimento. Em outras palavras, somamos o ROI em cada ponto multiplicado pela probabilidade daquele resultado. Supondo que temos a perfeita informação no resultado exato em ROI, isso poderia valer tanto quanto o EOL que acabamos de calcular. Já que um MVP tipicamente fornecerá informação não tão perfeita, o EOL representa um limite superior do que deveríamos gastar no caminho para descobrir um encaixe produto-mercado.

No site <http://www.howtomeasureanything.com>, Hubbard fornece uma planilha que ajuda você a calcular o valor da informação.

## Aplicando a abordagem Lean Startup dentro das empresas

O modelo Lean Startup não é limitado ao desenvolvimento de um novo produto. Ele pode ser usado para qualquer tipo de novo trabalho no contexto empresarial, incluindo substituição de sistemas, construção de ferramentas e produtos internos, inovação de processo e avaliação software de prateleira comercial. Em todos os casos, começamos definindo o resultado de cliente mensurável que queremos atingir.

Podemos definir nosso objetivo em termos de cliente imediato, como nosso colega que vai usar a ferramenta, processo, ou software. Por exemplo, para uma ferramenta interna de automação de teste, nosso objetivo seria reduzir o tempo de espera para o teste total de regressão para 8 horas.

Para determinar se temos um encaixe problema-solução,

procuramos por um cliente que queria trabalhar conosco no piloto do novo sistema, ferramenta, processo ou software. Este é um passo crítico que é frequentemente ignorado pelas empresas. Realmente, para ferramentas internas é comum impor sua utilização — uma política desastrosa que frequentemente resulta em uma enorme quantidade de lixo, usuários infelizes e pouco valor para a organização.

O processo de achar clientes e descobrir um problema real pelo qual eles pagarão para você resolver (mesmo que o pagamento seja em forma de tempo e feedback, em vez de dinheiro) — dessa forma obtendo uma relação problema-solução — é essencial para desenvolver ferramentas internas, adquirir software de prateleira ou substituir sistemas internos. Impor o uso de uma solução em particular torna mais difícil a obtenção de feedback para saber se aquela solução realmente fornece valor.

Uma vez que temos um time-piloto, projetamos e executamos um produto mínimo viável. Este pode ser um protótipo de uma ferramenta projetada para ajudar apenas um time, ou uma implementação de um pacote para resolver um problema para apenas um time ou para um único processo de negócio para aquele time. A parte mais difícil aqui é limitar o escopo tanto para resolver um problema real quanto para entregar algo no espaço de dias ou semanas, em vez de meses. A pior coisa que podemos fazer é desaparecer para projetar a ferramenta ou estratégia perfeita, sem continuamente entregar valor para usuários reais e colher o feedback deles durante o processo. É essencial ser disciplinado em relação ao período de tempo desta atividade, e focar em resolver um problema real e urgente o mais rápido possível.

A medida do sucesso — e se devemos ou não prosseguir — é se nossos usuários acham o MVP bom o suficiente para usá-lo por vontade própria, e se realmente alcançamos os resultados

mensuráveis que tínhamos como objetivo. Se não, precisamos dar meia-volta e voltar ao começo.

### 3.3 PRINCÍPIOS PARA EXPLORAÇÃO

No *capítulo 1*, mostramos como forças pequenas e altamente motivadas foram capazes de vencer inimigos maiores e mais bem treinados com um estilo de guerra conhecido como guerra de manobra. Atualmente, “disrupção” (ou ruptura) é uma palavra onipresente, quase um clichê. Mas, no contexto da guerra de manobra, o principal expoente da ideia de romper o processo de tomada de decisão de seu oponente foi o coronel John Boyd, da Força Aérea Americana.

Em sua carreira como piloto de caça e instrutor, Boyd era famoso por nunca perder a aposta de que ele conseguia vencer qualquer *dogfight* — mesmo em desvantagem — dentro de 40 segundos, e também por cocriar a teoria de manobrabilidade de energia do desempenho da aeronave, que levou ao projeto do jato de caça F-16. Contudo, sua criação mais conhecida é o “loop OODA”, um modelo (mostrado na figura a seguir) de como humanos interagem com seu ambiente, que forma a base da teoria de Boyd sobre a guerra de manobra. A abreviação OODA vem de *observar, orientar, decidir e agir*, as quatro atividades que compõem o loop.

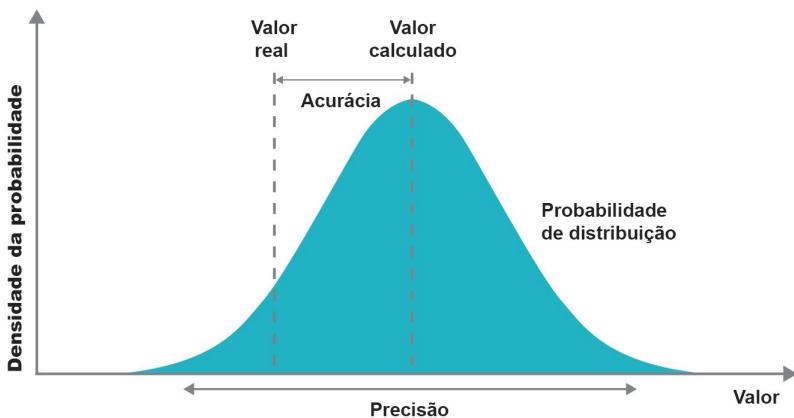


Figura 3.3: Acurácia e precisão (RICHARDS, 2012)

Um equívoco comum (principalmente com pessoas que não viram o diagrama) é que estas atividades são levadas uma seguida das outras em um loop, e que a disruptão é alcançada ao se percorrer o ciclo mais rapidamente que seu oponente. Há dois erros importantes nessa interpretação. Primeiro, na realidade tanto humanos quanto organizações estão desempenhando todas essas atividades simultaneamente, e há múltiplos feedbacks e loops *feed-forward* entre cada uma delas. Segundo, é frequentemente vantajoso atrasar a tomada de decisões até o “último minuto responsável” (que podemos analisar usando opcionalidade e custo de atraso, visto no *capítulo 7*).

Para entender realmente este diagrama, precisamos começar com orientação. O insight de Boyd aqui é que nossas observações, decisões e ações são todas contingentes em relação à nossa orientação atual, a qual é em parte determinada por uma complexa série de fatores, incluindo nossa genética, nossos hábitos e experiências e as culturas nas quais crescemos e em que estamos vivendo, assim como nas informações que temos em mãos. A segunda coisa a ser notada sobre o diagrama é que existem dois mecanismos de influência: um são os loops de feedback e de feed-

forward, e o outro são a “direção e o controle implícitos”.

A Psicologia nos diz que nossas ações podem ser formadas tanto por IGT (direção e controle implícitos) quanto por feed-forward a partir de uma decisão consciente. Direção e controle implícitos em humanos são fornecidos por um sistema na mente, chamado Sistema 1, que “opera automática e rapidamente, com pouco ou nenhum esforço e sem senso de controle voluntário”.

Decisões conscientes são feitas pelo Sistema 2, que “aloca a atenção para as atividades mentais com esforço que a demandam, incluindo computações complexas. As operações do Sistema 2 são frequentemente associadas com a experiência subjetiva de ação, escolha e concentração” (KAHNEMAN, 2011, p. 20-21). Igualmente, o IGT afeta como nós observamos as coisas, como, por exemplo, em nossa tendência a ignorar informação que contradiga nossas crenças (isso é conhecido como viés de confirmação).

Ambos os mecanismos existem no nível organizacional. Em termos de ação, organizações usam o mecanismo de direção e controle implícitos quando delegam a tomada de decisão. Elas usam comando descentralizado e o Princípio da Missão, confiando em um entendimento comum de seus objetivos com o alinhamento de toda a organização para assegurar que as pessoas ajam pensando no interesse da organização maior. Contudo, algumas ações (particularmente, aquelas envolvendo compliance) devem ser feitas usando o mecanismo explícito do feed-forward.

Direção e controle implícitos também governam como as organizações observam. Culturas geradoras criam sistemas de monitoramento com displays visíveis que permitem que as pessoas rapidamente acessem informações relevantes — o que, por sua vez, muda sua orientação. Mudanças na orientação farão com que atualizemos o que medimos e como a informação flui pela organização. Em culturas organizacionais patológicas e

burocráticas, a mensuração é usada como uma forma de controle, e as pessoas escondem informações que desafiem as regras, estratégicas e estruturas de poder existentes. Como Deming disse, “quando há medo, você terá números errados”.

Quando Boyd fala sobre “operar por dentro” de um loop OODA do oponente, ele quer dizer entender o loop de seu oponente e como ele determina suas ações. Então você pode usar esse conhecimento contra eles.

“O padrão básico é simples: uma organização usa seu melhor entendimento — sua melhor ciência — do desdobramento da situação para influenciar seu oponente, empregando ações que se encaixem nas expectativas do oponente, as quais Boyd, seguindo Sun Tzu, chamou de *zheng*. Quando a organização sente (de experiências anteriores, incluindo treinamento) que chegou a hora, ela joga o *qi*, o inesperado, extremamente rápido. A razão primária para a direção implícita quando se está ocupado com oponentes é que instruções explícitas — ordens escritas, por exemplo — tomariam muito tempo. Como diria Boyd, ‘a ideia-chave é enfatizar o implícito sobre o explícito, de maneira a ganhar uma incompatibilidade favorável em atrito e tempo (isto é, o nosso mais baixo que qualquer outro do adversário) para alcançar superioridade em formar e se adaptar às circunstâncias’” (RICHARDS, 2012).

O modelo OODA pode também ser aplicado no contexto do engajamento do cliente: “Em vez de surpresa -> choque -> exploração, como na guerra e nas artes marciais, o *zheng/qi* poderia operar como algo mais parecido com surpresa -> prazer -> fascinação -> clientes mais comprometidos. A Apple joga esse jogo, a ‘busca do uau!’, como Tom Peters descreveu muito bem” (RICHARDS, 2012).

Boyd refere-se aos caminhos de direção e controle implícitos

dento de uma organização, determinados por sua cultura, conhecimento e processos institucionais existentes, como seu repertório. Discutimos como organizações aplicam seu repertório existente para quebrar sua concorrência, mas, para melhorar a performance e evitar a disruptura, devemos criar constantemente nosso próprio repertório. Isso pode se dar em forma de melhoria de processos, evolução dos produtos existentes, ou criação de novos negócios e novos produtos. Este loop também é representado no modelo OODA, como mostrado na figura a seguir.

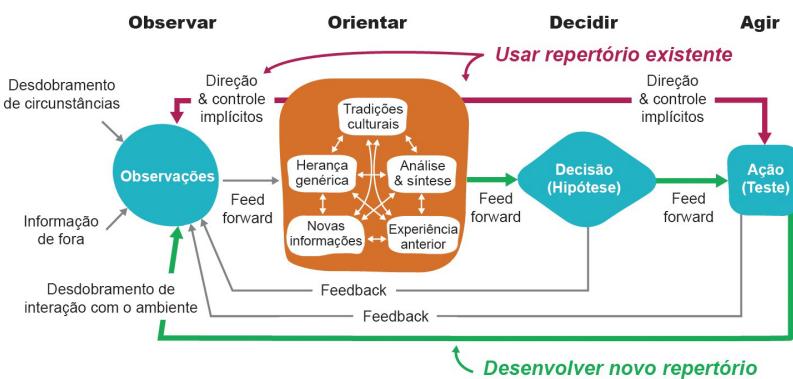


Figura 3.4: Criando novo repertório (RICHARDS, 2012)

O loop de geração de repertório é mais ou menos uma afirmação do método científico, no qual criamos novas hipóteses baseadas em observação e síntese, projetamos experimentos para testar essas hipóteses e, então, atualizamos ou descartamos nossas teorias (que formam parte da nossa orientação), baseando-nos nos resultados do experimento. Este loop, por sua vez, inspirou o loop construir-medir-aprender de Eric Ries (figura *O ciclo de construir-medir-aprender* do capítulo anterior), que mostra como criar novo repertório na forma de novos modelos de negócio, produtos e features.

O loop construir-medir-aprender parece direto, mas é difícil de

adotá-lo na prática devido à sua combinação de abordagem científica (construir para aprender) com uma mentalidade de engenharia (aprender para construir).

Para melhoria de processo (discutida no *capítulo 6*) e para mudar a cultura organizacional (abordada no *capítulo 11*), podemos usar um loop conhecido como o ciclo Deming, mostrado na figura:

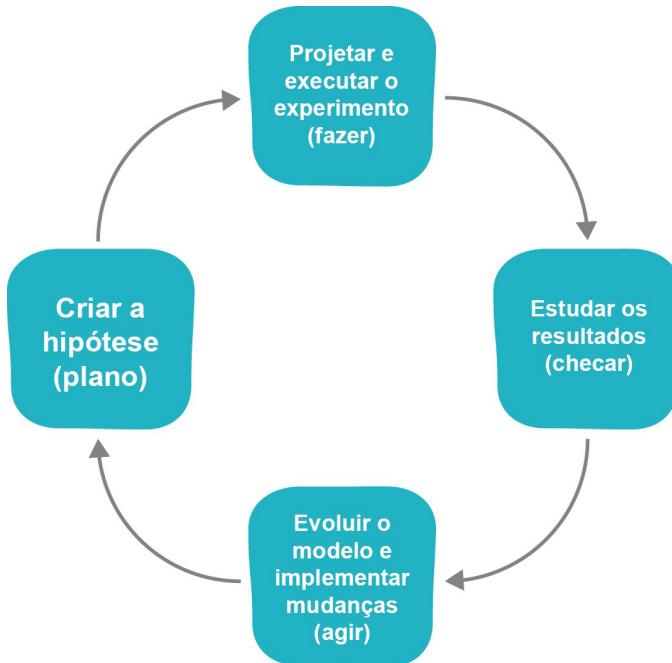


Figura 3.5: O ciclo Deming

A chave para ser bem-sucedido com esses ciclos (e com o método científico em geral) é usá-los sistematicamente e continuamente. Aplicá-los sistematicamente significa usá-los como uma ferramenta genérica para explorar todos os tipos de risco, assegurando-se de que o custo de executar um experimento é proporcional ao valor da informação que vamos descobrir. Aplicá-los continuamente significa fazer isso o mais frequentemente possível (como diz Mike Roberts, “contínuo significa muito mais

frequente do que você imagina”), com foco em passar pelo loop no tempo mais curto possível.

A questão mais importante a se fazer no contexto da geração de repertório é: quão rápido podemos aprender? Embora não possamos divulgar os resultados de nossos exercícios de aprendizado para o resto do mundo — quando lançar nosso produto é uma questão de estratégia —, devemos aprender e testar nossas suposições com usuários reais o mais frequentemente possível.

Quando todo mundo na organização for treinado para empregar a abordagem científica em inovação como parte de seu trabalho diário, teremos criado uma cultura geradora. Conseguimos isso ao praticar a abordagem experimental até que se torne um hábito, parte de nosso repertório, usando o *Improvement Kata* descrito no *capítulo 6*. É isso que permite uma organização a se adaptar rapidamente ao seu ambiente em mudança. A Toyota chama isso de “construir pessoas antes de construir carros” (LIKER, 2003).

### 3.4 GERENCIAMENTO CIENTÍFICO VERSUS MÉTODO CIENTÍFICO

É fundamental distinguir entre o gerenciamento científico de Taylor, discutido no *capítulo 1*, e a abordagem experimental que descrevemos aqui. No gerenciamento científico, a análise é feita e decisões são tomadas pela gerência, com pessoas que desempenham seu trabalho mais ou menos como autômatos. Na abordagem experimental, o trabalho da liderança e da gerência é projetar, evoluir e operar um sistema no qual as pessoas que fazem o trabalho têm as habilidades e recursos necessários para executar seus próprios experimentos — dessa forma, aprendendo, se

desenvolvendo e aumentando seu conhecimento individualmente e coletivamente.

Como mostrado na *Tabela 3-2*, aplicar o método científico em desenvolvimento de produto é fundamentalmente diferente da tradicional abordagem baseada em planejamento, e requer diferentes habilidades e comportamentos. Não que o ciclo de vida do projeto tradicional seja ruim — pode ser eficaz em projetos em que a coisa a ser construída tenha sido feita muitas vezes antes e os riscos são bem entendidos. Mas o gerenciamento de projeto tradicional é o modelo errado para condições de incerteza, como no desenvolvimento de um novo produto ou qualquer tipo de desenvolvimento de software personalizado.

**Tabela 3-2 — Planejamento de projeto tradicional versus Lean Startup**

Habilidade ou comportamento	Abordagem de planejamento tradicional	Abordagem experimental
Mudanças no plano	Mudanças no plano uma vez que foi acordado são consideradas problemáticas e indicam uma falha no processo.	Esperamos que o plano inicial não sobreviva ao contato com clientes reais, e objetivamos invalidá-lo e refazê-lo o mais rápido possível.
Habilidades necessárias	Reunião de requisitos, análise, determinação de custos, recursos e planejamento de dependência, habilidade de conseguir apoio político	Projetar experimentos e fazer medições, coletar dados e analisar, habilidade de trabalhar efetivamente em times com várias funções e comunicar-se com toda a organização
Como o sucesso é medido	Se o plano é aprovado e financiado	Quão rápido podemos passar pelos ciclos de aprendizado e sair da fase de investigação, tanto com a despriorização ou cancelamento do trabalho como seguindo para a fase de exploração
Como alcançamos compliance	Os processos apropriados foram seguidos corretamente e as aprovações necessárias	Identificamos os riscos reais para os stakeholders e reunimos as informações relevantes para

	foram recolhidas?	gerenciá-los?
--	-------------------	---------------

Os maiores obstáculos de se fazer uma abordagem científica em desenvolvimento de produto e mudança organizacional são culturais e organizacionais, como discutiremos na *Parte IV*. Na maioria dos casos, as organizações simplesmente nunca executaram uma abordagem empírica, e faltam as habilidades e a experiência para implementá-la.

No contexto de desenvolvimento de produto, entender como projetar e executar experimentos e analisar dados é tão difícil como criticamente importante — e ainda assim eles não são parte do currículo principal da maioria dos programas de MBA ou de cursos de design e análise de software. Em organizações patológicas e burocráticas, uma abordagem experimental pode também desafiar as estruturas de poder e normas culturais existentes.

## 3.5 CONCLUSÃO

Nós explicamos as fundações de uma abordagem científica para explorar um novo trabalho — seja um novo modelo de negócio ou produto, trabalho interno na empresa, como construção de novas ferramentas, ou adoção de novos processos. Quando temos um entendimento compartilhado do que queremos dizer sobre risco, mensuração e incerteza, podemos aplicar os princípios e práticas do movimento Lean Startup. Estes fornecem uma maneira melhor de gerenciar os riscos da decisão de investimento do que com atividades de planejamento tradicionais.

Nossa habilidade de competir é baseada em criar uma orientação comum na organização, e permitir que as pessoas façam seu trabalho para criar constantemente e praticar um novo repertório por meio de um processo de experimentação. Estas atividades permitem que identifiquemos e analisemos mais

efetivamente as mudanças em nosso ambiente, para adentrar nos processos de tomada de decisão de outras organizações e para agir — para melhor servir nossos clientes e moldar nosso ambiente. O modelo OODA, de Boyd, mostra que a adaptação ao nosso ambiente é um processo contínuo — tanto para organizações como para pessoas.

## Questões para o leitor

- Como o modelo de investimento de sua organização ou departamento se arrisca em seu plano de negócio? Em que dados está baseado?
- Quais são as variáveis no plano com o maior valor de informação? Quais mensurações foram feitas para reduzir a incerteza nestas variáveis?
- Quão confiante você está de que as pessoas vão achar que o trabalho que você está fazendo atualmente é valioso? Qual evidência você tem para apoiar sua decisão?
- Com que frequência você tem testado o produto em que está trabalhando com seus usuários pretendidos? O que você mudou como consequência disso?

## CAPÍTULO 4

# EXPLORAR A INCERTEZA PARA DETECTAR OPORTUNIDADES

*“Foi a escuridão que criou a lâmpada. Foi a neblina que criou a bússola. Foi a fome que nos levou à exploração.”* — Victor Hugo

Neste capítulo, veremos as práticas que apoiam os princípios discutidos no *capítulo 3*, de exploração das oportunidades em condições de extrema incerteza, especialmente quando consideramos novos modelos de negócios ou produtos. Introduzimos o conceito de Descoberta para mostrar como mapear rapidamente uma hipótese de negócio, criando um entendimento comum de um problema e engajando os *stakeholders* da organização para apoiar e se alinhar com nossa visão.

Compartilharemos ferramentas e técnicas concretas para, com segurança, criar e testar hipóteses para resolver problemas reais de negócios, identificados e validados em nosso processo de desenvolvimento de cliente. Então, descreveremos como usar uma abordagem disciplinada, científica, baseada em evidências para responder à questão fundamental — não “podemos construir”, mas “devemos construir”?

Discutiremos como testar as suposições arriscadas de nossas hipóteses e gerar dados empíricos para apoiar nossa decisão de

pivatar, perseverar ou parar ao criar experimentos seguros usando MVPs. Nosso propósito é basear as decisões em evidências para futuros investimentos, não em ficção científica. Colocaremos em prática oportunidades, construindo o produto certo no tempo certo, e pararemos de gastar o tempo das pessoas em ideias que não têm valor.

## 4.1 DESCOBERTA

Descoberta é um conjunto de atividades rápidas, de forma iterativa, que integra as práticas e princípios de design thinking e Lean Startup. Usamos estas atividades intensivamente no começo da fase de exploração de uma nova iniciativa.

Em *Lean UX: Applying Lean Principles to Improve User Experience*, Jeff Gothelf e Josh Seiden (2013) afirmam que o “*design thinking precisa de uma abordagem focada em solução para resolver problemas, trabalhar colaborativamente para iterar em um caminho sem fim, e mutante em direção à perfeição. Ele trabalha no sentido de alcançar os objetivos do produto via formação de ideias, protótipos, implementação e etapas de aprendizagem para se chegar à solução mais apropriada*”.

Ao combinar os princípios do design thinking com práticas de Lean Startup, podemos ter um loop de feedback contínuo com usuários reais e clientes, em nosso ciclo de desenvolvimento. O princípio é investir a mínima quantidade de esforço para conseguir a máxima quantidade de aprendizado, e usar os resultados de nossos experimentos como base para nossa decisão de pivotar, perseverar ou parar.

## CLIENTES E USUÁRIOS

Apesar de frequentemente usarmos os termos intercambiando-os, é bom distinguirmos entre clientes de um produto ou serviço, que pagam por isso ou investem em seu desenvolvimento, e usuários. Usuários não pagam pelo produto, mas contribuem muito para a organização que desenvolve o produto e, frequentemente, para o produto em si (redes sociais são o exemplo mais óbvio).

Em uma empresa, as pessoas devem usar sistemas específicos para fazer seu trabalho, e as organizações lidam com consequências negativas reais quando os sistemas são difíceis de usar. É essencial engajar clientes e usuários como stakeholders-chave na cocriação de produtos, serviços ou oportunidades de melhoria.

Durante a Descoberta, criamos um ambiente colaborativo e inclusivo para um time multifuncional e multidisciplinar para explorar um negócio, produto ou oportunidade de melhoria. O time deve estar totalmente dedicado e localizado junto para maximizar a velocidade de aprendizado e a efetividade da tomada de decisão em tempo real. Ele deve assumir a propriedade da entrega e ser empoderado para tomar as decisões necessárias para alcançar os objetivos da iniciativa.

Quando montamos um time, é fundamental manter o grupo pequeno, incluindo apenas as competências necessárias para explorar o problema em questão. Grandes times são mal equipados para exploração rápida, e não aprendem na velocidade necessária para serem bem-sucedidos. O grupo deve saber suas limitações, assumindo a responsabilidade de tocar e engajar aqueles de fora do

grupo para sugestões e colaborações, quando apropriado.

Outros membros importantes do time — e frequentemente esquecidos — são os clientes e usuários. É fácil cair na armadilha de vê-los como simplesmente um consumidor da solução que criamos. Na verdade, eles são stakeholders exigentes. Suas colocações são o ingrediente-chave e a medida mais objetiva do quão valiosa nossa solução é, ou pode ser. Por meio do feedback que eles dão, clientes e usuários são cocriadores de valor para qualquer solução. Suas necessidades devem sempre ser o ponto focal para tudo que fazemos.

## Criando um entendimento compartilhado

*“Quando você quer construir um navio, não comece juntando madeira, cortando tábuas e distribuindo o trabalho, mas desperte dentro do coração do homem o desejo pelo vasto e infinito mar.”* — Atribuído a Antoine de Saint-Exupéry

Quando começamos um novo trabalho, é imperativo que o grupo crie um ambiente que maximize o potencial de todos os envolvidos. Baseadas nas novas informações que estão descobrindo, as pessoas aprendem, mudam e melhoram quando estão envolvidas em um processo energizante, interativo e adaptativo.

Como Dan Pink (2009) argumenta em *Drive*, há três elementos-chave para se considerar quando montamos um time engajado e altamente motivado. Primeiro, o senso de propósito compartilhado pelo time todo. A visão precisa ser desafiadora o suficiente para o grupo ter uma aspiração, porém, clara o suficiente para que todos entendam o que precisam fazer. Segundo, as pessoas devem ser empoderadas por seus líderes para trabalhar com autonomia para que alcancem os objetivos do time. E, finalmente, as pessoas precisam de espaço e oportunidade para dominarem sua disciplina, e não apenas aprenderem a alcançar o “bom o suficiente”.

O processo de moldar a visão começa pela explanação clara do problema que o time tentará resolver. Esse passo essencial é frequentemente negligenciado, ou se assume que todos sabem qual é o problema. A qualidade da explanação do problema aumenta a habilidade do time de focar no que realmente importa — e, mais importante ainda, ignorar o que não importa. Ao desenvolver o entendimento compartilhado do time sobre os objetivos e sobre o que queremos conseguir, melhoramos nossa habilidade de gerar melhores soluções.

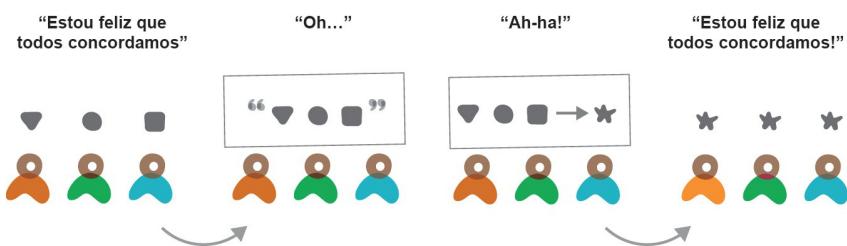


Figura 4.1: Construindo um entendimento compartilhado em equipe TIP

### FAÇA GAMESTORMING

O livro *Gamestorming*, de Gray, Brown e Macanufo (2010), e o Wiki Go Gamestorming (<http://www.gogamestorm.com>) contêm inúmeros jogos que encorajam o engajamento e a criatividade, ao mesmo tempo em que trazem estrutura e clareza à formação de ideias, inovação e workshops de melhoria.

Uma das técnicas fundamentais da Descoberta é usar artefatos visuais, modelos de informação visual para comunicar e apreender aprendizados em grupo. Usando templates gráficos e exercícios para externar ideias ajuda o time a articular, debater e evoluir conceitos e

ideias para formar consensos (ver figura anterior). Também ajuda a despersonalizar e deixar anônimas as considerações para que tenhamos um debate seguro de ideias e não de indivíduos — minimizando egos, HiPPOs (opiniões de pessoas mais bem pagas, ou em inglês, *Highly Paid People Opinion*) e as tentativas dos extrovertidos de comandar o show.

## Exploração estruturada da incerteza

*“Se você quer ter boas ideias, você deve ter muitas ideias.”* — Linus Pauling

Ao explorar a incerteza, é importante começar de um jeito amplo — para gerar o maior número de ideias possível para percorrê-las antes de afinarmos nosso foco para onde vamos ir.

O site Lastminute (<http://lastminute.com>) é um varejista de viagens na Europa, e opera em uma indústria altamente competitiva com grandes empresas e novas startups tentando inovar o mercado de viagens todos os dias. Para permanecer relevante, a companhia precisa inovar mais rápido e com mais inteligência que a concorrência. Eles convidaram seus clientes para serem parte do processo de inovação.

Durante dois dias, eles conduziram workshops de cocriação que geraram mais de 80 novas ideias para produtos online alinhados com seus objetivos de negócio. O time então montou um laboratório de inovação em um lobby de hotel por uma semana, e experimentou rapidamente cada ideia para descartá-la ou validá-la como um problema de cliente viável para ser implementado. Em alguns dias, o time identificou três ideias vencedoras para se investir futuramente em desenvolvimento — resultando em mais de 100% de aumento de conversão para seu produto (<https://www.youtube.com/watch?v=r64rrgbcEHo>).

O pensamento divergente é a habilidade de oferecer ideias diferentes, únicas ou variantes sobre um tema; o pensamento convergente é a habilidade de identificar uma potencial solução para um dado problema. Começamos a exploração com exercícios de pensamento divergente projetados para gerar múltiplas ideias para serem discutidas e debatidas. Depois, usamos o pensamento convergente para identificar uma possível solução para o problema. A partir daqui, estamos prontos para formular um experimento para testá-la (ver a figura a seguir).

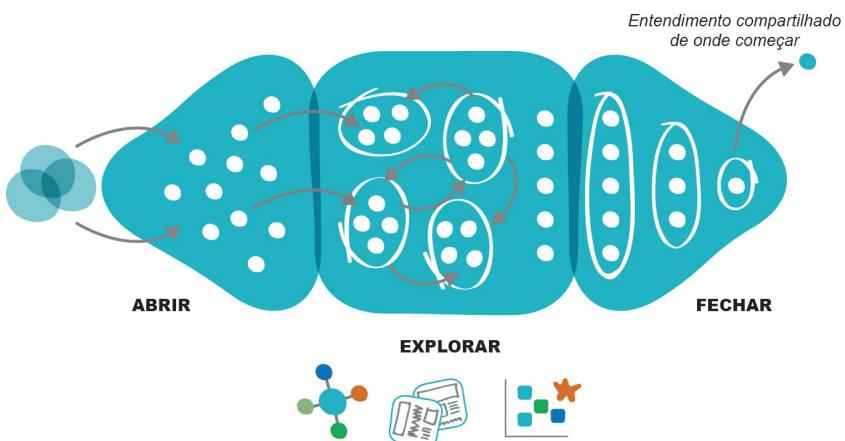


Figura 4.2: Exploração estruturada com pensamento divergente e convergente

## 4.2 EM QUE NEGÓCIO ESTAMOS?

Modelos de negócios são transitórios e inclinados à disruptão pelas mudanças no ambiente competitivo, avanços no design e na tecnologia, e mudança econômica e social mais ampla. Organizações que calculam mal seu propósito ou não conseguem sentir e se adaptar a essas mudanças se extinguirão.

Organizações podem tornar-se obsoletas por concorrentes que resolvem o mesmo problema com uma oferta alternativa ou superior para seus clientes. A definição do negócio e a identificação

de futuras oportunidades devem ser um desafio contínuo e estar sempre em evolução. Permitir a complacência devido ao sucesso de hoje é o caminho mais rápido para o fracasso de amanhã. Precisamos apenas citar exemplos, como Blockbuster *versus* Netflix ou Tower Records *versus* iTunes, YouTube e Spotify, para ilustrar que nenhum modelo de negócio ou vantagem competitiva é indefinidamente sustentável.

## **Entendendo nosso problema de negócio para montar nosso plano de negócio**

Como diz Steve Blank (2005), autor de *The Four Steps to the Epiphany* e *The Startup Owner's Manual*: “*Um plano de negócios é o documento de execução que companhias escrevem quando planejam extensões para linhas de produtos nas quais cliente, mercado e as features do produto são conhecidos. O plano é um documento operacional e descreve a estratégia de execução para abordar esses “conhecidos”.*

O objetivo primário de uma nova iniciativa de negócio é validar seu modelo de hipótese de negócio (e iterar e pivotar até que se valide). Pesquisa *versus* execução é o que diferencia uma nova jornada de uma unidade de negócio existente. Uma vez que um modelo de negócio é validado, ele deve ser então colocado no modo de execução. É neste ponto que o negócio precisa de um plano operacional, previsões financeiras e outras ferramentas de gestão bem compreendidas (BLANK, 2012).

É importante considerar vários modelos de negócio nos estágios iniciais de uma nova iniciativa. Não queremos comprometer um plano até testarmos a hipótese do modelo de negócio e termos evidência de que estamos no caminho certo. O time deve identificar as suposições mais arriscadas de nossa hipótese, bolar experimentos para testar essas suposições e aumentar a quantidade de informação

que podemos obter para reduzir a incerteza. A única suposição que sempre podemos considerar verdadeira é que nenhum plano de negócios sobrevive ao primeiro contato com clientes.

O canvas de Modelo de Negócio, mostrada na figura adiante, foi criado por Alex Osterwalder e Yves Pigneur junto com 470 cocriadores como um gerador de design do projeto de negócios simples e visual. É uma ferramenta estratégica empresarial e de gerenciamento que possibilita que os times descrevam, projetem, desafiem, inventem e pivotem modelos de negócios. Em vez de escrever um plano de negócios, o que pode se tornar um processo longo, devemos esboçar múltiplos modelos possíveis — cada um enquadrado em um tempo de 30 minutos — em um canvas.

CANVAS DO MODELO DE NEGÓCIO				
PARCEIROS	ATIVIDADES	PROPOSTA DE VALOR	RELACIONAMENTO COM CLIENTES	SEGMENTOS DE CLIENTES
	Produção Distribuição Fluxos de receita	Para cada segmento de cliente  Qual problema/necessidade estamos resolvendo?	Com nossos segmentos de clientes  Qual relacionamento existe hoje?  Como eles fazem parte do nosso negócio?	Para quem estamos criando valor?  Quais são os segmentos prioritários?
RECURSOS	Infraestrutura física Intelectuais – marca, IP, dados etc. Humanos Financeiros		CANAIS	
<b>ESTRUTURA DE CUSTO</b>  Custos fixos e variáveis Economias de escala/escopo Custos com pessoas Custos com recursos Custos com atividade			<b>RECEITA</b>  O que os clientes vão pagar Como vão pagar Fluxos de receita diferentes	

Figura 4.3: Canvas do Modelo de Negócio

O canvas do Modelo de Negócio, disponível gratuitamente em <http://www.businessmodelgeneration.com/canvas>, esboça nove componentes essenciais do modelo conceitual de negócios de uma organização:

- **Segmento de cliente**

Para qual público-alvo estamos criando valor? Quem são nossos clientes?

- **Proposta de valor**

Quais problemas vamos resolver de forma a gerar valor para nossos clientes?

- **Canais**

Por meio de quais canais vamos alcançar nosso público-alvo?

- **Relacionamento com clientes**

Qual tipo de relacionamento cada um de nossos clientes espera que criemos e mantenhamos com eles?

- **Atividades**

Quais atividades serão necessárias para apoiar nossas propostas de valor?

- **Recursos**

Quais recursos, pessoas, tecnologias e suporte serão necessários para o negócio funcionar?

- **Parcerias**

Com quem precisamos construir parcerias? Quem são nossos fornecedores-chave ou quem poderia ser necessário para fornecer recursos de apoio ou atividades para nossa proposta de valor?

- **Custo**

Quais são os custos inerentes mais importantes para o nosso negócio?

- **Receita**

Qual valor nossos clientes estão dispostos a pagar?  
Quanto e com qual frequência?

Ao preencher as sessões do canvas, somos impelidos a considerar qualquer ideia em potencial como tijolos da construção do negócio. Ao preencher todo o canvas, somos encorajados a pensar de maneira holística sobre como essas peças se encaixam para suportar uma oportunidade maior. É fundamental se lembrar de que cada componente do canvas representa um conjunto de hipóteses e suposições associadas que requerem validação para provar que nosso modelo de negócio é saudável.

Além do próprio canvas, Osterwalder também mostrou quatro níveis de domínio estratégico na comparação entre modelos de negócio para refletir a intenção estratégica de uma organização:

- **Estratégia Nível 0** — Os Alheios focam nas propostas de produto/valor por si só em vez de focar na proposta de valor e no modelo de negócio.
- **Estratégia Nível 1** — Os Iniciantes usam o canvas de Modelo de Negócio como um check-list.
- **Estratégia Nível 2** — Os Mestres superam outros em concorrência com um modelo de negócio superior em que todos os tijolos da construção fortalecem-se entre si (por exemplo, Toyota, Walmart e Dell).
- **Estratégia Nível 3** — Os Invencíveis estão em contínua autodisrupção ao mesmo tempo em que seus modelos de negócio ainda são bem-sucedidos (por exemplo, Apple e Amazon).

O objetivo primário do Canvas de Modelo de Negócio é externar a hipótese de negócio e deixar suas suposições claras para que possamos identificar e validar os principais riscos. O canvas oferece uma estrutura de trabalho para entender cada modelo de negócio, de uma maneira que é compreendida por todos. Dessa forma, é possível construir um senso compartilhado de propriedade e possibilitar a colaboração em toda a organização.

O Canvas do Modelo de Negócio difere de outros canvas listados na *Tabela 4-1*, na medida em que não pressupõe que a combinação produto/mercado é a hipótese mais arriscada, que deve ser testada em primeiro lugar. Há vários canvas criados por outras pessoas que focam em desenvolvimento do produto, como mostrado na *Tabela 4-1*.

**Tabela 4-1 — Canvas de visualização de ideias**

Nome	Propósito
O Canvas Lean ( <a href="http://www.leancanvas.com">http://www.leancanvas.com</a> )	Supõe que a relação produto/mercado é a hipótese mais arriscada que deve ser testada.
O Canvas Oportunidade ( <a href="http://comakewith.us/tag/opportunity-canvas">http://comakewith.us/tag/opportunity-canvas</a> )	Foca em discussões sobre o que estamos desenvolvendo e por que, para depois ajudá-lo a entender quão satisfatoriamente estes clientes e usuários reforçam a estratégia global da organização.
O Canvas Proposta de Valor ( <a href="http://bit.ly/1v6Z5Ae">http://bit.ly/1v6Z5Ae</a> )	Descreve como nossos produtos e serviços criam ganhos para o cliente, e como eles criam benefícios que os clientes esperam, desejam ou estariam interessados em usar.

## Entendendo nossos clientes e usuários

*“A coisa mais importante para lembrar sobre qualquer empresa é que não existem resultados do lado de dentro de suas paredes. O resultado de um negócio é um cliente satisfeito.”* — Peter Drucker

Para que qualquer produto ou solução seja bem-sucedido, as

pessoas devem querer usá-lo e, de fato, querer pagar por isso. Para que um time desenvolva uma solução que atenda um problema ou necessidade real, é essencial entender quem estamos tentando atingir e por quê.

#### DÊ UMA CARA A SEU CLIENTE E USUÁRIO

Uma persona é uma representação de problemas, necessidades, objetivos e comportamento de um grupo hipotético de clientes ou usuários. Personas são baseadas em informação relevante e conhecimento compartilhados pelos seus criadores. Essencialmente, elas são coleções de suposições que devem ser testadas e refinadas durante o processo de desenvolvimento de nosso cliente.

Quando criar uma persona, lembre-se dos seguintes pontos:

- Defina e faça um *brainstorm* da sua persona inicial rapidamente para deixar o time alinhado.
- Redefina iterativamente sua persona, baseado em evidências de pesquisa de usuário, testes e feedback durante o ciclo de desenvolvimento do cliente.
- Continuamente, realinhe a persona e a visão de negócio/produto assim que o produto começar a emergir. Personas são apenas um ponto de partida que usamos para criar entendimento compartilhado de nossos clientes ou usuários. Elas não são objetivas ou empíricas, não é este seu propósito. Usamos personas para criar empatia com os problemas de nossos grupos-alvo.

Ter empatia por clientes e usuários é muito poderoso. Quando

criamos empatia, melhoramos nossa habilidade de receber e processar informação (BATTARBEE; SURI; HOWARD, 2014). A empatia no design requer prática deliberada. Devemos projetar experimentos e oportunidades de interação para nos conectar com nossos clientes e usuários de maneira significativa e desafiar nossas suposições, pré-conceitos e preconceitos. Precisamos assumir o papel de um questionador, tentando entender os desafios que eles enfrentam.

Criar um equilíbrio entre simpatizar com uma experiência e analisar a situação permite que entendamos os sentimentos e perspectivas de nossos clientes e usuários. Podemos, então, usar esse entendimento para guiar nossa identificação de hipóteses de solução e começar o processo de experimentação.

#### VÁ, OLHE, VEJA

A empresa de design IDEO (<http://www.ideo.com>), famosa por criar o mouse da Apple, faz workshops nos quais os times fazem total imersão no contexto em que o produto ou serviço imaginado vai ser usado. Seus desenvolvedores leem tudo o que interessa sobre os mercados, observam e entrevistam futuros usuários, pesquisam ofertas que concorrerão com o novo produto e sintetizam tudo o que aprenderam em fotos, modelos e diagramas. O resultado são insights sobre clientes e usuários que são testados, melhorados ou abandonados durante o processo de desenvolvimento.

Na Toyota, o *genchi genbutsu* (vá e veja) permite que líderes identifiquem riscos de segurança existente, observem condições de maquinário e equipamento, e perguntam sobre práticas-padrão para acumular conhecimento sobre o status de trabalho e construir relações com seus empregados. O objetivo

do *genchi genbutsu* é ir ao *gemba* (local de trabalho) para entender o fluxo de valor e seus problemas em vez de revisar relatórios ou fazer comentários superficiais.

De maneira similar, sair do escritório (uma expressão popularizada pelo empresário e autor Steve Blank) é uma técnica de desenvolvimento de cliente para obter feedback e focar os esforços iniciais do desenvolvimento do produto com *early adopters*, por meio de questionários qualitativos frequentes (incluindo entrevistas) com múltiplos clientes em potencial.

Pessoas que não podem temporariamente largar sua função ou status, ou deixar de lado sua própria expertise e opiniões, falharão ao desenvolver empatia com os pensamentos conflituosos, experiências ou modelos mentais de outros indivíduos. A habilidade de ouvir e perguntar as questões certas torna-se uma aptidão poderosa e os insights que isso traz são a fundação da resolução eficaz de problemas e experimentação.

## Transformando insights e dados em vantagem

A capacidade de descobrir e alavancar insights é essencial para organizações de alta performance. Vivíamos em um universo de poucos dados com altos custos associados a arquivos, armazenamento e processamento de dados. Mas o movimento Big Data forneceu tecnologias e técnicas para revisar, processar e correlacionar grandes conjuntos de dados.

As organizações podem ganhar valor adicional de insights sobre como e por que seus clientes estão interagindo com seus produtos e soluções. Podemos detectar sinais que nos dizem o que está

funcionando — ou não está —, e usar essa informação para melhorar serviços existentes ou criar novas ofertas. Quando combinados, software, análises e dados formam um pilar-chave para o capital intelectual de nossa organização.

Acesso e compreensão de clientes existentes são uma vantagem competitiva significativa que organizações já estabelecidas têm sobre as startups. As startups enfrentam o desafio de ganhar alcance e força de mercado devido à falta de acesso a dados de clientes conhecidos. Por outro lado, organizações já estabelecidas têm dados sobre mercado e cliente que podem ser reusados e alavancados para descobrir novas oportunidades.

As organizações agora são capazes de fazer perguntas, como “Por que clientes estão cancelando suas assinaturas?” ou “Como os clientes se relacionam uns com os outros?”, e fazer experimentos rápidos e baratos para testar suas hipóteses. Essa é uma técnica poderosa para remover o viés de decisão de nosso processo de priorização e possibilitar que tomemos decisões guiadas por dados.

A análise de dados nos permite ver como os clientes estão usando serviços existentes e fazer projeções futuras para um novo modelo de negócio, produto ou oportunidades de serviço.

## **COMO EMPRESAS MINERAM DADOS PARA DESCOBRIR SEUS SEGREDOS**

No livro *O Poder do Hábito*, Charles Duhigg (2014) escreve: “Quase todo grande varejista, das cadeias de alimentos a bancos de investimento aos Correios dos EUA, tem um departamento de ‘análise preditiva’, dedicado a entender não apenas os hábitos de compra dos clientes, mas também seus hábitos pessoais, assim como as opções mais eficiente para eles”.

A rede Target usou dados dos seus clientes para identificar e comercializar para mulheres grávidas. Quando você está grávida, você precisa se preparar para o novo bebê comprando muitas coisas. A Target tentou encorajar famílias de gestantes a fazer a maioria das suas compras lá, potencialmente fazendo-as serem clientes para a vida toda. Foram analisados dados de clientes existentes para identificar mulheres no segundo trimestre da gravidez que poderiam ser alvo de ofertas.

A Target foi capaz de identificar mudanças nos padrões de compra de 25 produtos-chave, incluindo suplementos nutricionais, algodão em bolas e loções sem perfume, que previram precisamente não apenas gravidez, mas também a data do parto. Como resultado, a rede foi capaz de enviar às gestantes cupons de desconto — sutilmente disfarçados entre outras ofertas para que as mulheres não percebessem que estavam sendo alvo de promoção — para encorajá-las a fazer suas compras pré-natais na Target (HILL, 2012).

Big data é uma ferramenta, não uma solução. Crucialmente, não substitui empatia. Ainda precisamos da intuição humana e inovação para melhorar a definição do problema, e identificar necessidades e

problemas de clientes e usuários, de maneira a formar hipóteses que possam ser testadas confrontando os dados.

Times multifuncionais, personas e entrevistas com usuários são ferramentas poderosas que nos possibilitam projetar experimentos mais efetivamente e rapidamente. Precisamos aprender como ouvir e aprender com os dados por meio de uma análise imparcial — do contrário, nossos dados são inúteis: “Dados, como lanternas, são tão úteis quanto a pessoa que os maneja e a pessoa que interpreta o que eles mostram” (BERKUN, 2013).

## **Usando insights para montar hipóteses e experimentos**

Durante a descoberta, vários membros do time multifuncional terão — e devem ser encorajados a compartilhar — insights interessantes e valiosos para a organização, clientes, negócio, canais ou mercados. Ao compartilhar estes insights com o time, podemos gerar novas perspectivas e inspiração para novos produtos ou soluções.

Peça aos envolvidos na Descoberta para compartilharem quaisquer insights e dados interessantes que eles tenham para montar, criar ou desafiar a explanação do problema baseados em um número de perspectivas, usando o canvas mostrado na figura adiante. Por exemplo:

- **Cliente**

Qual informação específica o grupo tem sobre clientes existentes? Quais são seus comportamentos de uso e de engajamento? Como estes insights ajudam a moldar futuras oportunidades dentro das ofertas de produto existentes?

- **Tendências de mercado**

Tendências do mercado em que estamos tentando entrar são fundamentais para entender como e onde estão as oportunidades — por exemplo, tecnologias mobile, serviços baseados em localização, pagamentos mobile. Quais são as tendências do mercado para o produto que estamos criando? Como fazer a mensuração diante dessas tendências?

- **Organização**

Qual informação específica o grupo tem sobre nossa organização? Onde a organização está focando seus esforços? Qual o impacto desses esforços? Quanto do panorama competitivo eles cobrem? Onde a organização é mais efetiva?

- **Você não vai acreditar!**

Toda empresa tem indivíduos dispostos a compartilhar fatos interessantes e espantosos sobre o negócio e sua base de clientes. Como podemos testar se eles são verdadeiros e/ou oferecem oportunidades de criarmos novas propostas de valor como resultado?

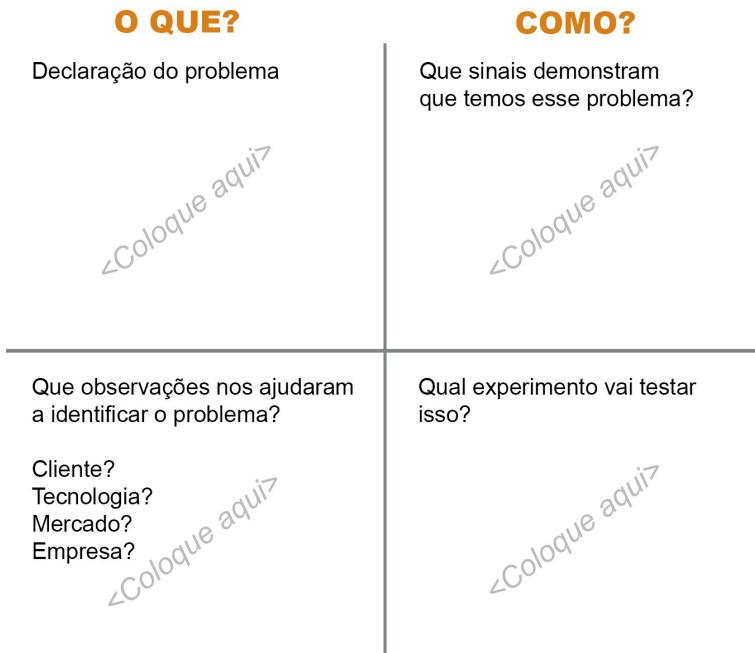


Figura 4.4: Canvas de explanação do problema

Ao tornar essa informação visível e discuti-la, podemos tentar identificar novos modelos de negócios e propostas de valor apropriadas para o negócio, dadas suas limitações atuais e explanações do problema identificadas.

### 4.3 ACELERE A EXPERIMENTAÇÃO COM MVPS

O movimento Lean Startup desafia a suposição de que clientes devem ter todas as features imagináveis disponíveis em um produto antes de começarem a usá-lo. Eric Ries cunhou o termo produto minimamente viável (MVP) para descrever a estratégia de investir o mínimo de recursos para testar suposições subjacentes de nossas hipóteses com clientes. O objetivo é eliminar o desperdício gerado

por soluções muito engenhosas e acelerar nosso aprendizado ao testar uma solução com clientes iniciais o mais rápido possível.

Um MVP nos permite usar uma mínima quantidade de esforço para gerar uma máxima quantidade de aprendizado quando experimentando com clientes. O objetivo de usar um MVP é executar um experimento que testa as suposições de nossas hipóteses o mais barato, rápido e efetivamente possível, de maneira a estudar se nossa solução se dirige ao problema do cliente que identificamos. O MVP elimina as partes da hipótese de solução que criam uma complexidade desnecessária e consumo excessivo de recursos quando experimentamos com nossos clientes-alvo iniciais. O resultado do experimento é aprendizado, o que nos permite tomar decisões baseadas em evidências para perseverar com nosso modelo de negócio existente, pivotar para explorar um novo modo de alcançar nossa visão, ou parar.

É importante distinguir entre um MVP na definição de Eric Ries e o lançamento público inicial de um produto, que cada vez mais assume a forma “beta”:

## PRODUTO MÍNIMO VIÁVEL

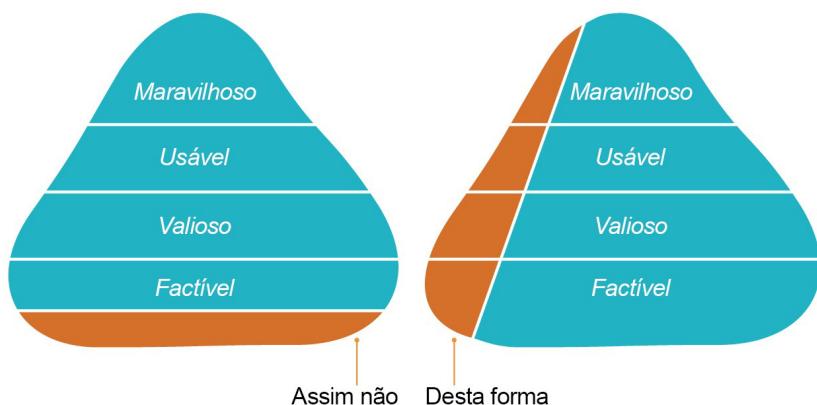


Figura 4.5: Produto Mínimo Viável: construir uma fatia com todas as etapas em vez de uma camada por vez — Diagrama inspirado por Jussi Pasanen, com reconhecimentos a Aarron Walter, Ben Tollady, Ben Rowe, Lexi Thorn e Senthil Kugalur.

De maneira confusa, as pessoas frequentemente se referem a qualquer atividade de validação em qualquer ponto deste espectro como um MVP, sobrecarregando o termo e seu entendimento na organização ou na indústria. Marty Cagan, autor de *Inspired: How to Create Products Customers Love* e ex-vice-presidente do eBay, notadamente usa o termo “teste de MVP” para se referir ao que Eric Ries chama de MVP.

Cagan (2008) define o MVP como “o menor produto possível que tem três características críticas: as pessoas escolher usá-lo ou comprá-lo; pessoas podem descobrir como usá-lo; e podemos entregá-lo quando precisamos com os recursos disponíveis — essas características também são conhecidas como valioso, usável e praticável”, às quais adicionamos “agradável”, já que design e estética são também essenciais para um MVP tanto quanto para um produto acabado, como mostrado na figura anterior (CAGAN, 2011a). Certifique-se de que seu time e seus stakeholders estejam certos em sua definição de MVP.

## **“DEVEMOS CONSTRUIR?”, E NÃO “PODEMOS CONSTRUIR?”**

JustGiving é uma plataforma online para angariar fundos que já conseguiu mais de 2 bilhões de libras para caridade. A plataforma queria explorar novos modelos de negócio para financiar iniciativas comunitárias que não são necessariamente ligadas com algum projeto de caridade.

Eles montaram um pequeno time para fazer experimentos rápidos com clientes, fazer sessões com um protótipo de uma plataforma de crowdfunding completa com projetos comunitários reais que estavam querendo apoiar. Baseados nas reações positivas dos clientes, eles procederam para criar um MVP concierge, lançando os mesmos projetos comunitários para clientes reais, ao mesmo tempo em que lidavam manualmente com tarefas administrativas (como configuração do projeto e processamento de pagamento e cobrança), para ver qual seria o desempenho do produto no mercado.

Em sete semanas, o JustGiving validou um modelo de negócios que eles puderam escalar para um negócio no sentido pleno. O produto agora se tornou o YIMBY (<https://www.justgiving.com/yimby>), com histórias de sucesso que incluem a compra de cadeiras de rodas para times de basquete em cadeira de rodas, ferramentas para expandir um jardim comunitário, e salvou o centenário time de futebol Kettering Town.

Os MVPs, como mostrado a seguir, não garantem sucesso. Eles são projetados para testar as suposições de um problema que queremos resolver sem investir demais. De longe, o resultado mais provável é que veremos que nossas suposições eram inválidas e

precisamos pivotar ou encerrar nossa abordagem. Nosso objetivo final é minimizar o investimento ao explorar solução até que estejamos confiantes de termos descoberto o produto certo — em seguida, explorar a oportunidade, adicionando complexidade e valor para construir o produto corretamente.

## **Exemplos de tipos de MVPs**

### **Papel**

#### **O que é**

Esboços descartáveis, desenhados a mão, de uma interface para usar como protótipo, ou exemplos ilustrativos de um design.

#### **Prós**

Rápido, visual, gera entendimento compartilhado.

#### **Contras**

Interação limitada, não testa usabilidade ou hipótese.

#### **Exemplos**

Diagramas, wireframes, esboços.

### **Protótipo interativo**

#### **O que é**

Um modelo clicável e interativo de um protótipo ou projeto.

#### **Prós**

Testa design e usabilidade, itera soluções com velocidade, usa entrevistas qualitativas de consumidores.

## **Contras**

Não testa hipóteses ou tecnologias suportadas.

## **Exemplos**

HTML ou modelos clicáveis, vídeos.

## **Concierge**

### **O que é**

Um serviço pessoal, em vez de um produto, que manualmente guia o cliente pelo processo usando os mesmos passos propostos para resolver o problema no produto digital. O nome deriva de *hotel concierge*.

### **Prós**

Reduz complexidade, suporta pesquisas generativas, valida premissas qualitativamente com um pequeno investimento.

## **Contras**

Escalabilidade limitada, é um recurso manual e intensivo, o cliente sabe que há envolvimento humano.

## **Exemplos**

Fundadores do AirBnb oferecendo colchões de ar para clientes durante uma Convenção Nacional do Partido Democrata dos Estados Unidos; a instalação de Collison na Stripe (GRAHAM, 2013).

## **Mágico de Oz**

### **O que é**

Produto que realmente funciona, mas nos bastidores todas as funções do produto são realizadas manualmente, o que é desconhecido ao usuário.

### **Prós**

Uma solução que funciona do ponto de vista do cliente, uma pessoa no papel do Mágico pode adquirir insights valiosos pelo envolvimento; permite pesquisas avaliadoras de pontos de preços e validação de proposta de valor.

### **Contras**

Escalabilidade limitada devido ao alto comprometimento de recursos; a pessoa no papel do Mágico deve gostar da funcionalidade da solução proposta; é difícil avaliar sistemas com um componente de interface gráfica grande.

### **Exemplos**

Tony Hsieh comprando sapatos para clientes iniciais do Zappos.com.

## **Micronicho**

### **O que é**

Reducir todas as features do produto ao mínimo, socializar e investir em dirigir o tráfico para o produto, para descobrir se clientes estão interessados ou dispostos a pagar por ele.

### **Prós**

Um teste altamente focado dedicado a qualquer tópico em específico, exige mínimo esforço.

### **Contras**

Precisa de investimento financeiro para dirigir tráfego, há uma competição por palavras-chaves e cliques do cliente.

### **Exemplos**

<http://whatkatewore.com>

## **Software em funcionamento**

### **O que é**

Um produto em total funcionamento para resolver um problema do cliente, instrumentado para medir o comportamento e interações do usuário.

### **Prós**

Testa hipóteses em um ambiente real, valida premissas qualitativamente.

### **Contras**

É caro, precisa de investimento em pessoas e ferramenta.

### **Exemplos**

Testes A/B, funis de conversão, otimização de referência.

## **Como nossa visão e MVP trabalham juntos?**

Cagan (2011b) enfatiza que visão e MVP estão intimamente relacionados, mas não são a mesma coisa. Ele define visão como o entendimento comum que “descreve os tipos de serviços que você pretende fornecer e os tipos de clientes que você pretende atender, normalmente durante um período de 2 a 5 anos”. Como tal, isso fornece um roadmap e contexto para os MVPs, e deveríamos estar preparados para criar muitos MVPs enquanto pesquisamos um

processo de desenvolvimento de cliente escalável e repetitivo, alinhado com nossa visão.

Os primeiros evangelistas, particularmente em empresas, devem acreditar em toda a nossa visão, não apenas em nosso primeiro experimento com MVP. Eles precisam ouvir o que nossa organização planeja entregar nos próximos 6 a 18 meses. Eles são atraídos pela visão do que estamos tentando alcançar e, por isso, são capazes de preencher as lacunas em nossa solução se sentem a dificuldade do problema que estamos tentando resolver. Ao oferecer uma prova da solução que estamos querendo construir, damos a eles evidências de que ela funciona e providenciamos uma oportunidade para feedback sobre a solução que estamos construindo.

Aproveitar a interação e o engajamento do negócio é muito importante nos estágios iniciais de uma iniciativa. O feedback e as evidências reunidos durante o uso de um MVP fornecem insights e aprendizados sobre comportamento do cliente do que medidas agregadas de sucesso. O MVP permite que foquemos na coisa certa a ser construída e fornece informações valiosas sobre como evoluir, adaptar ou pivotar para ir de encontro às necessidades do cliente, descobertas por meio de experimentação, como mostrado na figura a seguir.

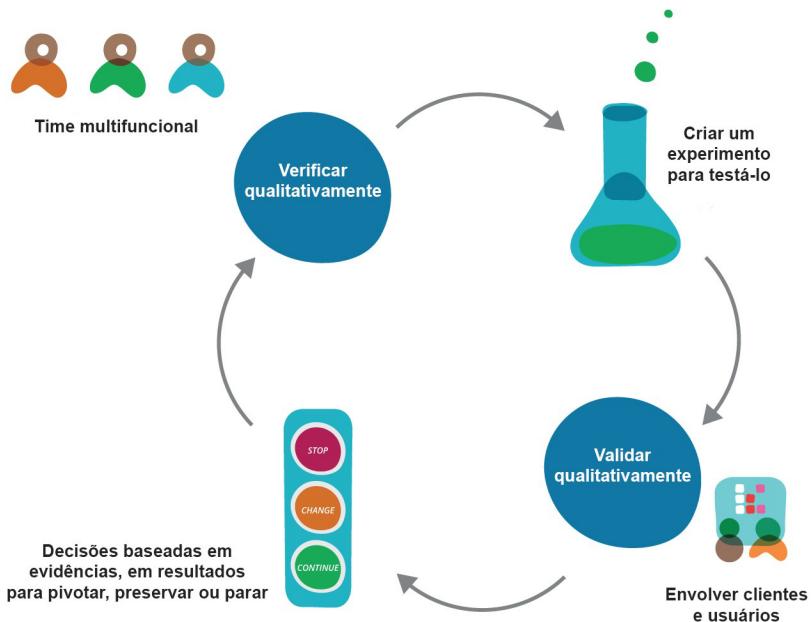


Figura 4.6: A mentalidade MVP e o loop de avaliação do experimento

Começando com a questão sobre o que queremos aprender com o experimento, podemos definir como o observaremos e mediremos, para finalmente criar o MVP mais barato, mais rápido e mais simples para testar nossas suposições, mensurar o efeito e usar este aprendizado para formular os próximos passos.

Um ponto fundamental em novas iniciativas é preservar dinheiro e iterar rapidamente enquanto os times estão testando as hipóteses para identificar uma solução repetível. Uma vez que esses conceitos estão entendidos e que alcançamos uma relação produto/mercado, preservar dinheiro se torna menos importante. Neste momento, devemos pensar em como gastá-lo de modo a criar uma solução escalável.

## A Única Métrica que Importa (UMI)

Ao projetar MVPs para experimentação, é importante identificar uma métrica-chave que nos dirá se as suposições de nossas hipóteses são válidas. Alistair Croll e Benjamin Yoskovitz, autores do *Lean Analytics*, introduziram o conceito de Única Métrica que Importa (OMTM, *Only Metric That Matters*, em inglês). A UMI é uma única métrica que priorizamos como a mais importante para guiar decisões, dependendo do estágio do ciclo de vida de nosso produto e nosso modelo de negócio. Não é uma única métrica que vamos usar durante a vida do produto: ela mudará com o tempo dependendo do problema com o qual estamos lidando.

Focamos na Única Métrica que Importa para:

- Responder as questões mais urgentes que temos, ligando-as às suposições nas hipóteses que queremos testar;
- Criar foco, conversação e reflexão para identificar problemas e estimular melhorias;
- Fornecer transparência e um entendimento comum no time e na organização;
- Apoiar uma cultura de experimentação baseando-se em taxas ou proporções, nunca médias ou totais, relevantes para nosso histórico de dados.

Não deve ser uma métrica que nos atrasa, como o retorno sobre investimento (ROI) ou rotação de clientes, que medem o rendimento depois do fato. Indicadores lentos tornam-se interessantes mais tarde, quando alcançamos a relação produto/mercado. Ao focarmos inicialmente em métricas principais, podemos ter uma indicação do que é mais provável acontecer — e lidar com uma situação mais rapidamente para tentar mudar os resultados futuros.

Por exemplo, queixas de clientes são frequentemente um

indicador principal de rotatividade. Se as queixas de clientes estão aumentando, podemos esperar que clientes nos abandonarão e a rotatividade vai aumentar. Nossa UMI deve sempre evoluir, de acordo com o que aprendemos sobre o problema que queremos resolver.

O propósito da UMI é obter evidência objetiva de que as mudanças que estamos fazendo em nosso produto estão tendo um impacto mensurável no comportamento de nossos clientes. Em última instância, estamos procurando entender:

- Estamos progredindo (o quê)?
- O que causou a mudança (por quê)?
- Como melhoraremos (como)?

O fundador do Intuit, Scott Cook, afirma que fundadores deveriam focar em “métricas do amor”. Por exemplo, o quanto as pessoas amam o produto, ou com que frequência elas voltam, ou quão encantadas elas estão nos estágios iniciais. “Se você não está obtendo alta atividade dos usuários que você já tem, é hora de pivotar”. Escolher uma UMI dá clareza, alinhamento e foco para os times, possibilitando a tomada de decisão efetiva, especialmente nas iniciativas em estágio inicial.

### USAR O PENSAMENTO A3 COMO MÉTODO PARA PERCEBER OPORTUNIDADES DE MELHORIA

O Pensamento A3 é uma ferramenta de solução de problemas para pegar informações críticas e definir o foco e as limitações do time. Num outro momento, ela se torna uma medida para testar nossos resultados. Um relatório A3 (assim chamado porque cabe em uma folha do tamanho A3), composto de sete elementos, inclui o ciclo de experimentação “Planejar — Fazer — Checar — Agir”:

- **BACKGROUND** — Pegar as informações críticas para entender a extensão e a importância do problema. Unir o background à definição do problema reduz o desperdício ao limitar as vezes em que focamos em áreas incorretas.
- **CONDIÇÃO ATUAL E DEFINIÇÃO DO PROBLEMA** — Este é o problema que o stakeholder do negócio quer resolver, em termos simples e inteligíveis; e não como uma declaração de falta de solução. Por exemplo, evite declarações como “Nosso problema é que precisamos de um sistema de gerenciamento de conteúdo”.
- **DECLARAÇÃO DO OBJETIVO** — Como saberemos se nossos esforços foram bem-sucedidos ao fim da implementação? Idealmente, precisamos de uma métrica-chave para o sucesso. Por exemplo, “Nosso objetivo é reduzir falhas do sistema comparado com os resultados anteriores de 22 questões importantes; nosso objetivo é reduzir essa quantia em 20%”.
- **ANÁLISE DA CAUSA E EFEITO** — Detalhe a hipótese e as suposições, ou um conjunto de experimentos realizados para testar causa e efeito.
- **CONTRAMEDIDAS** — Liste os passos de um experimento a ser implementado para testar a hipótese.
- **EFEITO CHECAR/CONFIRMAR** — Defina um método para avaliar se as contramedidas tiveram um efeito.

- **ACOMPANHE AS AÇÕES E REPORTE** — Identifique passos futuros e compartilhe o que aprendeu com o time e com a organização.

Para saber mais sobre Pensamento A3, leia *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*, de Durward K. Sobek II e Art Smalley (2008). Outros exemplos incluem o discurso do elevador (GRAY, 2010) e os Cinco Ws e Um H (em inglês, *Who, What, Where, When, Why, and How*).

Lembre-se, métricas foram feitas para machucar — não para nos fazer sentir que estamos vencendo. Elas devem ser açãoáveis e desencadear uma mudança em nosso comportamento ou pensamento. Precisamos considerar essas duas questões-chaves quando decidirmos qual será nossa UMI:

- **Qual o problema que estamos tentando resolver?**
  - *Desenvolvimento de produto* — Estamos tentando criar novos produtos ou serviços que envolvem clientes? Como saberemos que os estamos engajando e que eles estão interessados em nosso produto?
  - *Seleção de ferramenta* — Estamos tentando selecionar uma ferramenta para usar na organização? Como saberemos se é a melhor ferramenta para o processo?
  - *Melhoria de processo* — Estamos tentando melhorar nossa capacidade e eficiência internas? Como saberemos se nossas mudanças estão tendo o impacto desejado?

- **Em que estágio do processo estamos?**
  - *Validação do problema* — Estamos tentando identificar que um problema existe conversando com pessoas para ver se elas estão sofrendo com a questão que estamos tentando resolver?
  - *Validação da solução* — Nosso pessoal está demonstrando alinhamento e apoio para o problema que estamos querendo resolver por meio de entrevistas qualitativas?
  - *Validação do MVP* — Estamos criando experimentos para provar quantitativamente que nossa solução está funcionando para resolver o problema que identificamos?

A UMI é uma ferramenta útil para simplificar a complexidade de uma análise. Ela nos fala especificamente se nossa solução está sendo bem-sucedida ou não. Uma vez que definimos a métrica-chave em que vamos focar, podemos identificar métricas de apoio que nos fornecem insights para outras áreas e suporte para tomada de decisão.

Um bom exemplo de UMI: no LinkedIn, o time não fala sobre “total de pageviews”, mas somente “views de perfis” — o número de pessoas usando o LinkedIn que pesquisa e encontra outras pessoas, e o número de perfis do LinkedIn que elas visualizaram (ELMAN, 2012).

## 4.4 CONCLUSÃO

A Descoberta nos permite, com segurança, explorar oportunidades em condições de extrema incerteza — especialmente no desenvolvimento de um novo produto e inovação do modelo de negócio. Os conceitos e ferramentas da Descoberta nos deixam

investir a mínima quantidade de esforço para obter a máxima quantidade de aprendizado para ter um progresso mensurável ao explorar oportunidades validadas. A Descoberta cria uma visão clara e um entendimento compartilhado do problema que estamos tentando resolver dentro de nossa organização.

Devemos adotar a mentalidade na qual nossas ideias são hipóteses baseadas em suposições que devem ser testadas e que a maioria dessas suposições estarão erradas. Ao basear nossa tomada de decisão em informações recolhidas de experimentos rápidos e baratos usando MVPs, podemos tomar decisões melhores de investimento. Quanto mais cedo pudermos pivotar ou abandonar ideias ruins, menos tempo e recursos desperdiçaremos, e mais poderemos devotar a ideias que entregarão valor para nossos clientes — ou criará novos.

## Questões para os leitores

- Qual é sua atual hipótese de negócio e como você criaria um experimento usando um MVP para testá-lo?
- Você se pergunta se “devemos desenvolvê-lo” antes de prosseguir com “podemos desenvolvê-lo”?
- Quais experimentos seu time desempenharia e quais evidências eles reuniriam para decidir quando pivotar, perseverar ou parar?
- Qual é a sua Única Métrica que Importa?

## CAPÍTULO 5

# AVALIE A RELAÇÃO PRODUTO-MERCADO

*“O limite... não há uma maneira decente de explicá-lo porque as únicas pessoas que realmente sabem onde ele está são aqueles que o ultrapassaram.” — Hunter S. Thompson*

Neste capítulo, discutiremos como identificar quando uma relação produto/mercado foi alcançada e como sair da fase de investigação para começar a explorar nosso produto em seu mercado identificado. Mostraremos como usar métricas customizadas para entender se estamos alcançando resultados de negócio mensuráveis enquanto resolvemos os problemas de nossos clientes ao engajá-los em nosso processo de desenvolvimento.

Abordaremos como as organizações se estabelecem para o sucesso com a estratégia, estrutura e suporte certos, e como encontram clientes internos e externos para providenciar feedbacks e insights valiosos ao desenvolverem seu produto. Trataremos de como aproveitar capacidades, serviços e práticas existentes para escalar nosso produto, enquanto buscamos defensores internos na organização para colaborarem. Finalmente, descreveremos as métricas e os mecanismos de crescimento que nos ajudam a gerenciar a transição entre horizontes de modelo de negócios

enquanto começamos a escalar nossa solução.

## 5.1 CONTABILIDADE PARA INOVAÇÃO

*“Não é o suficiente fazer o seu melhor; você deve saber o que fazer, e então fazer o seu melhor.” — W. Edwards Deming*

Vivemos em um mundo de excesso de dados, em que qualquer argumento pode encontrar dados de apoio se não formos cuidadosos para validar nossas suposições. Encontrar uma informação para apoiar uma teoria nunca é um problema, mas testar a teoria para depois executar a ação correta ainda é difícil.

Como discutimos no *capítulo 3*, o segundo maior risco para qualquer novo produto é construirmos a coisa errada. Dessa forma, é imperativo não investirmos demais em oportunidades não comprovadas ao fazer a coisa errada do jeito certo. Devemos começar confiando que estamos realmente fazendo a coisa certa. Mas como testamos se nossa intuição está correta, especialmente quando operando em condições de extrema incerteza?

Eric Ries introduziu o termo *contabilidade para inovação* para se referir ao rigoroso processo de definir, experimentar, medir e comunicar o verdadeiro progresso de inovação para novos produtos, modelos de negócio ou iniciativas. Para entender se nosso produto tem valor e nos responsabilizarmos, focamos em obter evidências admissíveis e planejar uma trajetória razoável enquanto exploramos novos domínios.

Medidas de contabilidade financeira tradicionais, como desempenho de operação, fluxo de caixa ou indicadores de lucratividade, como retorno sobre o investimento (ROI) — que não

são projetados para inovação —, frequentemente têm o efeito de sufocar ou matar novos produtos ou iniciativas. Eles são otimizados e mais eficientes para se aproveitar domínios mais conhecidos ou produtos e modelos de negócios já estabelecidos.

Por definição, inovações têm um histórico de operação limitado, receita mínima ou nenhuma, e requerem investimento para começar, como mostrado na figura seguinte. Nesse contexto, retorno sobre investimento, análise de índice financeiro, análise de fluxo de caixa e práticas similares proporcionam poucos insights sobre o valor de uma inovação, e não permitem sua avaliação de investimento em relação ao desempenho de produtos já bem estabelecidos por meio de comparação de dados financeiros.

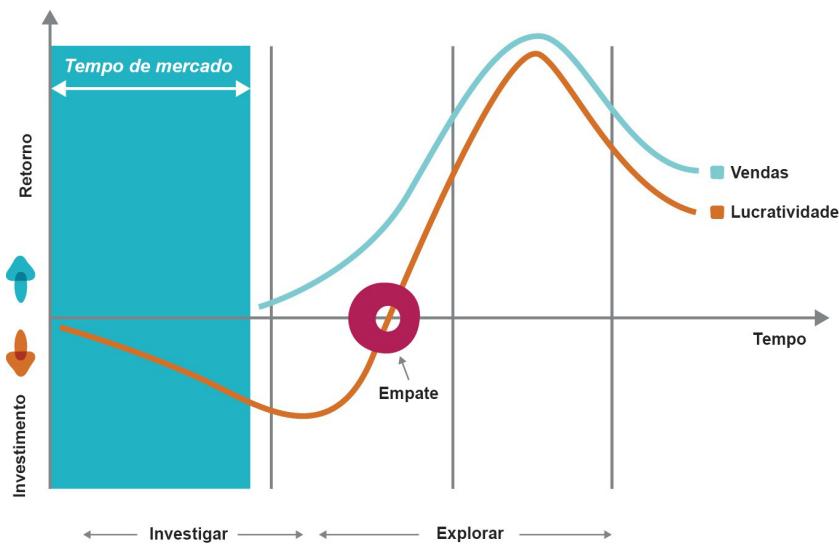


Figura 5.1: Índices de lucratividade e vendas para inovações em estágio inicial

Quando na fase de investigação, a contabilidade não deve ser ignorada ou tomada como irrelevante. Ela simplesmente precisa ser interpretada de maneira diferente para medir os resultados da inovação e de iniciativas em estágio inicial. Nossos princípios de contabilidade e mensuração para inovação devem visar os seguintes

objetivos:

- Estabelecer responsabilidade por decisões e critérios de avaliação;
- Gerenciar os riscos associados à incerteza;
- Sinalizar oportunidades e erros emergentes;
- Oferecer dados precisos para análise de investimento e gerenciamento de risco;
- Aceitar que, às vezes, precisaremos ir em frente com informações imperfeitas;
- Identificar maneiras de melhorar continuamente a capacidade de inovação de nossa organização.

#### A FALÁCIA DA MENSURAÇÃO

*“O que você mede é o que você tem.” — Kaplan e Norton (1992)*

Uma das ideias-chave do livro *The Lean Startup*, de Eric Ries (2011), é o uso de métricas acionáveis. Ele defende que devemos investir energia em juntar métricas que nos ajudem a tomar decisões. Infelizmente, frequentemente o que vemos coletado e socializado em organizações são métricas de vaidade, projetadas para nos fazer sentir bem, mas que não oferecem uma direção clara de qual ação tomar.

Em *Lean Analytics*, Alistair Croll e Benjamin Yoskovitz (2012, p. 13) afirmam: “Se você tem dados sobre os quais não pode agir, então isso é uma métrica de vaidade... Uma boa métrica muda o seu comportamento. Isso é, de longe, o critério mais importante para

uma métrica: o que você fará diferente baseado nas mudanças na métrica?”. Alguns exemplos de métrica de vaidade e as métricas açãoáveis correspondentes estão mostradas na *Tabela 5-1* (MAURYA, 2010b; MCCLURE, 2007; KOHAVI, 2013)

**Tabela 5-1 — Exemplos de métricas de vaidade versus métricas açãoáveis**

Vaidade	Açãoáveis
<b>Número de visitas.</b> É uma pessoa que visita cem vezes ou cem pessoas que visitam uma vez?	<b>Métricas de funil, análise de grupo.</b> Definimos os passos do nosso funil de conversão, então agrupamos usuários e rastreamos seu ciclo de vida de uso com o tempo.
<b>Tempo no site, número de páginas.</b> Estes são um pobre substituto para engajamento ou atividade real a não ser que seu negócio esteja atrelado a este comportamento. Eles indicam volume, mas não dão indicação se clientes podem achar a informação que precisam.	<b>Número de sessões por usuário.</b> Definimos um critério de avaliação global para quanto tempo deve levar uma sessão (ou ação) para ser completada no site, então medimos com qual frequência os usuários a fazem com sucesso.
<b>E-mails coletados.</b> Uma grande lista de e-mails de pessoas interessadas em um novo produto pode ser excitante até sabermos quantos vão abrir nossos e-mails (e reagir ao que há dentro deles).	<b>Ação de e-mail.</b> Enviar e-mails de teste para um número de assinantes registrados e ver se eles fazem o que pedimos para fazerem.
<b>Número de downloads.</b> Enquanto esse número às vezes altera seu ranking em lojas de aplicativos, downloads por si só não levam a um valor real.	<b>Ativações de usuários.</b> Identificar quantas pessoas baixaram o aplicativo e o usaram. Criações de conta e referências fornecem mais evidências de engajamento de cliente.
<b>Uso da ferramenta</b> reflete o nível de padronização e reuso na cadeia de ferramentas da empresa.	<b>Efeito ferramental</b> é tempo de ciclo do check-in à liberação para produção de uma nova linha de código.
<b>Número de pessoas treinadas</b> conta aqueles que passaram pelo treinamento Kanban e obtiveram a certificação com sucesso.	<b>Taxas de transferência mais elevadas</b> mensuram que um trabalho de alto valor seja feito mais rápido, levando uma maior satisfação do consumidor.

Em *How to Measure Anything*, Douglas Hubbard (2010, p. 37) recomenda uma boa técnica para decidir por certa métrica: “Se você pode definir o resultado que realmente quer, dê exemplos dele e

identifique como são observáveis, assim você pode projetar mensurações que medirão os resultados que importam. O problema é que, pelo contrário, gerentes estavam simplesmente medindo o que era mais simples de ser medido (isto é, apenas o que eles já sabiam como medir), e não o que mais importava”.

Ao combinar o princípio das métricas acionáveis com a recomendação de Hubbard para criar as medidas que mais importam, podemos ir além da tradicional eficiência interna e mensuração financeira para focarmos em valor da perspectiva dos stakeholders do que mais importa — nossos clientes.

As “métricas piratas” de Dave McClure (<http://slidesha.re/1v6ZL8B>) são uma maneira elegante de formular qualquer negócio orientado para o serviço, como mostrado na *Tabela 5-2* (seguimos Ash Maurya, colocando receita antes de referência). Note que, para usar as métricas piratas eficazmente, devemos sempre as medir por coorte. Uma coorte é um grupo de pessoas que compartilham uma característica comum — normalmente, o dia em que usaram seu serviço pela primeira vez. Dessa maneira, quando mostrarmos métricas de funil como as de McClure, filtramos os resultados que não são parte da coorte com a qual nos preocupamos.

**Tabela 5-2 — Métricas Piratas: AARRR!**

Nome	Propósito
Aquisição	Número de pessoas que visitaram seu serviço
Ativação	Número de pessoas que tiveram uma boa experiência inicial
Retenção	Número de pessoas que voltaram para saber mais
Receita	Número de pessoas da coorte que se engajaram em atividade criadora de receita
Referência	Número de pessoas da coorte que recomendam para outros usuários

Mensurar as métricas piratas para cada coorte permite que você meça o efeito das mudanças para seu produto ou modelo de negócio, se você está pivotando. Ativação e retenção são as métricas que servem para a relação problema/solução. Receita, retenção e referência são exemplos de métricas do amor — o tipo de coisa com que você se preocupa para avaliar a relação produto/mercado. Ash Maurya (2010b) tem um ótimo post sobre métricas piratas, coortes e relações problema/solução: <http://bit.ly/1v6ZG4L>.

Na *Tabela 5-3*, reproduzimos o efeito das métricas piratas da mudança incremental e do pivotamento para o produto de Votizen (BINETTI, 2011). Note que a ordem e o significado das métricas são ligeiramente diferentes da *Tabela 5-2*. É importante escolher métricas adequadas para seu produto (principalmente se não é um serviço). Escolha métricas acionáveis!

**Tabela 5-3 — Efeito de mudança incremental e pivôs em métricas piratas de Votizen**

Métrica	Interpretação	v1	v1.1	v2	v3	v4
Aquisição	Criar conta	5%	17%	42%	43%	51%
Ativação	Autenticidade certificada	17%	90%	83%	85%	92%
Referências	Enviou para amigos	-	4%	54%	52%	64%
Retenção	Usou o sistema ao menos 3 vezes	-	5%	21%	24%	28%
Receita	Causas apoiadas	-	-	1%	0%	11%

Para determinar uma relação produto/mercado, também precisamos reunir outras métricas de negócios, como aquelas mostradas na *Tabela 5-4*. Como sempre, é importante não apontar para uma precisão desnecessária quando reunir essas métricas. Muitas dessas métricas de crescimento devem ser mensuradas em uma base por corte, mesmo que seja apenas por semana.

**Tabela 5-4 — Métricas de crescimento Horizonte 3**

Métrica	Propósito	Exemplo de cálculo
Custo de aquisição do cliente	Quanto custa adquirir um novo cliente ou usuário?	Vendas totais e despesas de marketing divididas pelo número de clientes ou usuários adquiridos
Coeficiente viral (K)	Uma medida quantitativa da viralização de um produto	Número médio de convites que cada usuário envia multiplicado pela taxa de conversão de cada convite
Valor do cliente (CLV)	Prevê o lucro líquido total que receberemos de um cliente	O valor atual de futuros fluxos de caixa atribuídos ao cliente durante toda sua relação com a empresa (FARRIS; BENDLE; PFEIFER; REIBSTEIN, 2010)
Taxa de queima mensal	A quantidade de dinheiro necessária para funcionar o time, uma pista de quanto tempo podemos operar	O custo de pessoal e de recursos

Quais são as métricas que nos importam em um dado momento dependerá da natureza de nosso modelo de negócio e quais suposições estamos tentando validar. Podemos combinar as métricas importantes para nós em um *scorecard*, como mostrado na figura adiante (obrigado a Aaron Severs, fundador do hirefrederick.com, pela inspiração e por nos permitir usar este diagrama).

Métricas de sucesso com o cliente fornecem insights revelando se clientes acreditam que nosso produto é valioso. Métricas de negócios, por outro lado, focam no sucesso de nosso próprio modelo de negócio. Como apontamos anteriormente, coletar dados nunca é um problema para novas iniciativas; as dificuldades estão em coletar os dados açãoáveis, atingir o nível certo de precisão e não se perder em meio a todo o ruído.

Para nos ajudar a melhorar, nosso painel deveria mostrar apenas métricas que causarão uma mudança no comportamento, que são focadas no cliente e apresentam metas para melhoria. Se não estamos inspirados em agir baseados nas informações de nosso

painel, estamos medindo a coisa errada, ou não buscamos o suficiente o nível apropriado de dados acionáveis.

Stakeholder	Medida	Atual	Meta	Tendência
Cliente	% de usuários que completam o fluxo de vendas	30%	45%	▲
	% retenção	20%	25%	▲
	Net Promoter Score	44	60	▲
Negócio	% visitas para assinar o serviço	20%	25%	↔
	% conversão para clientes pagantes	15%	20%	▲
	Custos de aquisição de cliente	\$0.25	\$0.05	↔
	Valor de cliente	\$0.30	\$0.80	▼
	% desgaste	30%	15%	▼

Figura 5.2: Exemplo de scorecard de inovação

Em termos de governança, a coisa mais importante a fazer é ter uma reunião regular (semanalmente ou quinzenalmente) que inclua os líderes de produto e engenheiros do time, junto com alguns stakeholders-chave (como o líder responsável pelo portfólio do Horizonte 3 e seus representantes sêniores de produto e engenharia). Durante a reunião, vamos avaliar o estado das métricas escolhidas e talvez atualizar em quais vamos focar (incluindo a Única Métrica que Importa).

O objetivo da reunião é decidir se o time deve perseverar ou pivotar e, mais importante, decidir se o time descobriu a relação produto/mercado — ou, na verdade, se deve parar e focar em algo mais valioso. Os stakeholders precisam fazer duros questionamentos para manter o time franco sobre seu progresso.

## **ENERGIZANDO DEFENSORES INTERNOS NA EMPRESA**

A inovação em organizações grandes e burocráticas é um desafio, porque elas são inherentemente projetadas para apoiar estabilidade, conformidade e precedência sobre correr riscos. Líderes que chegaram ao topo poderiam fazê-lo porque trabalharam o sistema tal qual ele existiu até agora.

Dessa forma, precisamos ter cuidado para que qualquer crítica não se foque em indivíduos ou seu comportamento dentro do sistema. Precisamos procurar colaboradores e cocriadores pela organização sem causar alienação, para conseguir futuro apoio para nossos esforços e para cruzar o abismo para o próximo estágio da curva de adoção dentro da organização. Por final, precisaremos identificar agentes modificadores nas áreas em que precisamos mudar para sermos bem-sucedidos. A melhor munição aqui são evidências demonstráveis de que nossos esforços estão atingindo resultados de negócios mensuráveis.

Sem dúvida há pessoas em nossa organização que estão frustradas e apoiam mudanças. Contudo, elas procuram segurança, contexto e cobertura antes de terem vontade de se tornar campeões de uma iniciativa. Energizar e engajar essas pessoas é o segredo do sucesso. Ao se tornarem os primeiros seguidores de nossas ideias e iniciativas, elas fornecerão um loop de feedback, permitindo-nos iterar e melhorar nosso produto.

Elas também são nossos patrocinadores dentro da organização como um todo. Em ambientes burocráticos, as pessoas tendem a proteger sua marca pessoal e não apostar em zebras. Nossa objetivo é dar a elas a confiança, os recursos e a evidência que as encorajem a serem defensoras de nossa iniciativa na

organização.

## 5.2 FAÇA COISAS QUE NÃO ESCALEM

Mesmo quando validamos as suposições mais arriscadas de nosso modelo de negócio, é importante que continuemos a focar nos mesmos princípios de simplicidade e experimentação. A tentação, uma vez que atingimos a tração, é procurar automatizar, implementar e escalar tudo o que é identificado como “requerimento” para que nossa solução cresça. Contudo, isso não deve ser nosso foco.

Em estágios iniciais, devemos gastar menos tempo nos preocupando com o crescimento e focar na interação significante com o cliente. Podemos continuar dessa maneira apenas para adquirir clientes individualmente — muitos consumidores em um estágio muito inicial podem nos levar à falta de foco e nos atrasar. Precisamos focar em encontrar *primeiros seguidores* apaixonados para continuar a experimentar e aprender. Então, buscamos engajar clientes similares para eventualmente “cruzar o abismo” para aquisição de clientes e adoção mais abrangentes.

Isso é contraintuitivo para a maioria das iniciativas nas grandes organizações. Somos programados para mirar em um crescimento explosivo, e fazer coisas que não escalam nem combinam com o que somos treinados para fazer. Ainda, tendemos a medir nosso nível requerido de serviço, despesas e sucesso considerando a renda, tamanho e escopo de produtos mais maduros em nosso ambiente ou domínio competitivo.

Devemos lembrar de que ainda estamos no estágio de formação de nosso processo de descoberta, e não queremos investir em excesso e nos prender a uma solução muito cedo. Continuamente,

nós testamos e validamos as suposições de nosso modelo de negócio por meio de experimentos de mercado em cada passo. Se identificamos um cliente-chave com um problema e podemos agir naquela necessidade, temos então uma oportunidade viável de construir algo que muitas pessoas querem. Não precisamos engajar cada departamento, segmento de cliente ou mercado para começarmos. Precisamos apenas de um cliente focado com quem possamos cocriar.

Uma vez que os líderes virem evidências de crescimento significativo em nossa operação com processos não escaláveis, nós facilmente seremos capazes de assegurar pessoas, fundos e suporte para construir soluções robustas para lidar com o fluxo de demanda. Nossa objetivo deve ser criar um sistema de atração para clientes que querem nosso produto, serviço ou ferramentas, e não empurrar uma solução obrigatória, planejada e pronta para as pessoas, querendo “vender” ou pedindo para usarem.

## **Intimidade com o cliente**

Ao deliberadamente estreitar nosso mercado para priorizar a qualidade do engajamento e o feedback de nossos clientes, podemos construir uma intimidade, relações e lealdade com nossos primeiros seguidores. As pessoas gostam de se sentir parte de algo único e especial.

## **CRIANDO EMPATIA COM OS CLIENTES: ÀS VEZES, A RESPOSTA ESTÁ DENTRO DO ESCRITÓRIO**

A Royal Pharmaceutical Society sabia que sua base de dados de substâncias clínicas era a melhor do mundo. Também sabia que deveria haver muito mais usos para ela do que ser apenas uma pilha de livros. Mas por onde deveria começar? Em vez de tentar adivinhar ou construir uma plataforma cara para os produtos, ou tentar fechar negócio sem ter um produto, eles usaram seu outro grande bem: um prédio cheio de farmacêuticos. Por meio de uma rápida prototipagem, teste de usuário com farmacêuticos trabalhando para a sociedade e pesquisa de produto em farmácias próximas, eles rapidamente foram capazes de focar em um aplicativo para checar potenciais interações entre medicações prescritas. Há grandes oportunidades em licenciar os dados para uso internacional. Ao começar com um app que eles mesmos usariam, eles foram capazes de entender o que clientes internacionais queriam e construir uma grande ferramenta de marketing.

Ao manter nossa base inicial de clientes pequena — não buscar números de vaidade para ficarmos enormes rapidamente —, nós nos forçamos a nos manter simples e próximos de nossos clientes em todos os passos do processo. Isso permite que times tenham mais tempo com os clientes para ouvi-los, construir confiança e garantir aos primeiros seguidores que estamos prontos para ajudar. Lembre-se: alcançar grandes números não é uma grande vitória; solucionar necessidades e encantar clientes é.

## **Construa um backlog de questões, não de requisitos**

O instinto de times de produto, uma vez que um problema ou validação de solução é atingido, é começar a construir todos os requisitos para uma solução escalável, totalmente operante e completa, baseados nos gaps em seus MVPs. O perigo dessa abordagem é que ela nos impede de evoluir o produto com base no feedback de nossos clientes.

No estágio inicial, ainda estamos aprendendo. Dessa forma, é importante que não limitemos nossas opções ao comprometer tempo, pessoal e investimento construindo features que podem não gerar os resultados desejados para clientes. Devemos aceitar que tudo é uma suposição a ser testada, e continuamente buscar identificar nossa área com mais incertezas e formular experimentos para aprender mais. Para proteger nossas apostas com essa abordagem, alavanque coisas que não escalam — construa um backlog com cenários de como podemos continuar a construir nosso produto.

Nosso backlog deveria ser uma lista de hipóteses a serem testadas, não uma lista de requisitos a serem construídos. Quando recompensamos nossos times por sua habilidade em entregar requisitos, é fácil de, rapidamente, inflar nossos produtos com features desnecessárias — levando a complexidade aumentada, custos de manutenção mais altos e capacidade limitada de mudança. As features entregues não são uma medida de sucesso, os resultados de negócios são. Nosso backlog é uma série de questões que precisamos testar para reduzir a incerteza e melhorar nosso entendimento sobre oportunidades de crescimento.

## CRIE UM MAPA DE HISTÓRIA QUE CONTE A NARRATIVA DA LINHA DE NOSSA VISÃO

Mapas de história são uma ferramenta desenvolvida por Jeff Patton (2014), explicadas em seu livro *User Story Mapping*. Patton declara que “seu software tem uma espinha dorsal e um esqueleto — e seu mapa mostra isso”. Mapas de história ajudam a planejar e priorizar ao deixarem visível a solução como um todo (ver figura a seguir).

O mapeamento de história não é projetado para gerar histórias ou criar um plano de lançamento — mas sim para entender os objetivos dos clientes e os trabalhos a serem feitos. Mapas de história fornecem meios necessários para comunicar a narrativa de nossa solução para engajar o time e os stakeholders, e obter feedback. Ao seguir mapas de história e contar a história da solução, garantimos que nenhum grande pedaço está faltando. Ao mesmo tempo, maximizamos o aprendizado ao identificar a próxima hipótese mais arriscada para testar, enquanto minimizamos o desperdício e o excesso de soluções de engenharia que não se encaixam nas necessidades de nosso cliente, como definimos em nosso MVP.

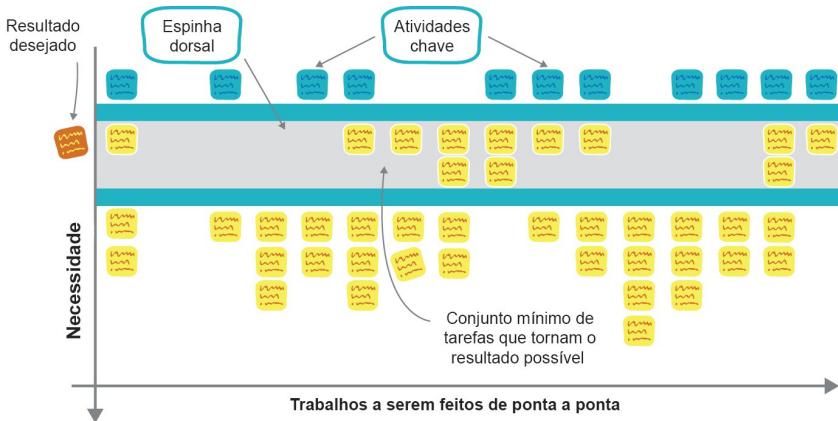


Figura 5.3: Um mapa de história de usuário

Quando começamos a consolidar, integrar e automatizar nosso produto, isso impacta nossa habilidade de rapidamente nos adaptar ao que estamos descobrindo, frequentemente limitando nossa responsividade e habilidade de mudar. Dentro do Horizonte 3, devemos continuamente trabalhar para evitar o inchaço do produto ao alavancar serviços existentes, capacidades ou processos manuais para entregar valor aos usuários. Nossa objetivo é não nos afastar de nossos usuários. Queremos garantir que estamos constantemente interagindo. Se otimizamos apenas para construir sem constantemente testar nossas suposições com nossos clientes, podemos perder “pontos de dor” fundamentais, experiências e sucessos — e é ali que geralmente estão os verdadeiros insights.

Se queremos aprender, devemos ter empatia com nossos usuários e experimentar suas dores. Quando encontramos um cliente com um problema que podemos resolver manualmente, fazemos isso o quanto for possível. Quando a qualidade de serviço do nosso cliente é comprometida, ou quando não podemos lidar com o nível de demanda, consideramos introduzir features que visem aos gargalos que apareceram com o aumento do uso do produto.

### **ALAVANQUE A INOVAÇÃO FRUGAL**

Técnicas não escaláveis não são apenas uma necessidade — elas podem ser um catalisador para mudança na cultura de uma organização. Ao provar que é possível testar nossas ideias rapidamente, de forma barata e com segurança, dá aos outros na organização a coragem e confiança de que experimentar é possível, e o resultado é uma mudança duradoura para melhor em nossa cultura.

## **Práticas de engenharia para explorar**

Em geral, somos a favor dos princípios do Sistema Toyota de Produção ao “construir qualidade” dentro do software, discutido bastante no *capítulo 8*. Contudo, quando investigando, há uma tensão entre a necessidade de experimentar construindo MVPs e construir em altos níveis de qualidade por meio de práticas como automação de teste.

Quando começamos a trabalhar na validação de uma nova ideia de produto ou uma nova feature em um produto existente, queremos tentar a maior quantidade de ideias o mais rápido possível. Idealmente, faremos isso sem escrever qualquer software. Mas, para o software que escreveremos, não queremos perder muito tempo construindo testes de aceitação e refatorando nosso código. Nós vamos (como diz Martin Fowler), deliberadamente e prudentemente, acumular débito técnico para que executemos experimentos e obtenhamos validação (FOWLER, 2009a).

Contudo, se nosso produto é bem-sucedido, chegaremos a um impasse com essa abordagem. Talvez em um ano ou dois (dependendo do nosso limite de dor), as mudanças se tornarão

onerosas e consumirão tempo, e o produto ficará cheio de defeitos e sofrendo com desempenho ruim. Podemos até chegar ao ponto em que consideramos uma Grande Reescrita.

Nosso conselho é esse. Há duas práticas que deveriam ser aderidas desde o começo que nos permitirão pagar o débito técnico mais tarde: integração contínua e um pequeno número de unidade básica e testes usuário-jornada. No momento em que um produto (se estamos no Horizonte 3) ou uma feature (no Horizonte 2) deixa de ser um experimento para ser validado, precisamos começar a pagar agressivamente o débito técnico. Normalmente, isso significa adicionar mais testes usuário-jornada, empregar boas práticas de arquitetura, como modularização, e ter certeza de que todo novo código escrito na feature usa desenvolvimento guiado por teste (bons engenheiros já usarão TDD).

Tendo nos forçado a fazer algo que deve ser antinatural para engenheiros — hackear código sem testes automatizados e subir o produto para produção (ir logo às ruas) para obter validação desde o início —, devemos então puxar a alavanca com força na direção contrária, matar o *momentum* e mudar nosso foco, de construir a coisa certa para construir a coisa da maneira certa. Não é preciso dizer que isso requer extrema disciplina.

Escolher em que ponto no ciclo de vida de nosso produto ou feature pagar nosso débito técnico é uma arte. Se você achar (como muitos acham) que foi longe demais no acúmulo de débito técnico, considere as alternativas de fazer a Grande Reescrita, descrita no *capítulo 10*.

## 5.3 MECANISMOS DE CRESCIMENTO

Em *The Lean Startup*, Eric Ries (2011) argumenta que existem três estratégias-chave para o crescimento — escolha uma:

- **Viral** — Inclui qualquer produto que faça com que novos clientes se inscrevam como um efeito colateral necessário do uso normal dos clientes existentes: Facebook, MySpace, AIM/ICQ, Hotmail, Paypal. As métricas-chave são a aquisição e referência, combinadas no agora famoso coeficiente viral.
- **Pago** — É quando usamos uma fração do valor de vida de cada cliente e fluxo que volta em aquisição paga por meio de marketing de mecanismos de busca, anúncios em banners, relações públicas, afiliados etc. A amplitude entre o valor do cliente e o custo de aquisição de cliente combinados determina sua lucratividade ou sua taxa de crescimento, e uma alta avaliação depende do equilíbrio entre esses dois fatores. Retenção é a meta-chave neste modelo. Exemplos são a Amazon e a Netflix.
- **Grude** — Significa algo que faz com que clientes se tornem viciados no produto, e não importa como adquirimos um novo cliente, a tendência é mantê-lo. A métrica para o grude é a “taxa de churn”: a fração de clientes em qualquer período que não permanece engajada com nosso produto ou serviço. Isso pode levar a um crescimento exponencial. Para o eBay, o grude é resultado de efeitos incríveis de network de seu negócio.

Para empresas, porém, existem mais opções de crescimento a serem consideradas:

- **Expandir** — É construir um modelo de negócio inicial adaptativo que poderíamos simplesmente evoluir e expandir ao nos abrir para novas geografias, categorias

e adjacências. A Amazon tem executado essa estratégia perfeitamente, passando de vendedora de livros para uma loja de e-commerce que oferece novas categorias de varejo. Com essa estratégia de crescimento, o mercado-alvo inicial deve ser grande o suficiente para arcar com múltiplas fases de crescimento com o tempo.

- **Plataforma** — Uma vez que temos um produto principal bem-sucedido, o transformamos em uma plataforma em volta da qual um “ecossistema” de produtos e serviços complementares se desenvolve por meio de fornecedores internos e externos. A Microsoft fez isso com o Windows ao criar o MS Office, Money e outros pacotes de suporte, incluindo aqueles desenvolvidos por fornecedores externos. Outros exemplos de plataforma incluem a Apple AppStore, Salesforce’s Force Market Place e Web Services da Amazon.

Grandes produtos, ferramentas e práticas, tanto internos como externos, sempre se espalharam com o boca e boca por causa de sua proposta de valor atraente e por uma marca que clientes se sentem orgulhosos de defender. Se nosso crescimento deriva de nossos clientes, então ele acontecerá sem que precisemos investir. Se não, seremos limitados pelo esforço requerido de, manualmente, descobrir, converter e manter nossos clientes.

Em última análise, nosso produto é nosso guia de crescimento. Se construímos uma solução verdadeiramente atraente que vai de encontro à necessidade do cliente e que é algo que ele realmente ama, então ele vai usá-la. E, mais impressionante, ele se tornará um defensor e encorajará outros a usarem — criando o melhor time de vendas que poderíamos imaginar para atingirmos o sucesso.

## 5.4 TRANSIÇÃO ENTRE HORIZONTES PARA CRESCER E TRANSFORMAR

No *capítulo 2*, mencionamos que organizações devem gerenciar os três horizontes ao mesmo tempo. A habilidade de reconhecer, mudar e converter iniciativas por meio desses ciclos, como mostrado na figura adiante, guarda a chave para o futuro sucesso, relevância e longevidade da organização.

Como descrevemos no *capítulo 3*, são os Horizontes 2 e 3 que precisam de um maior suporte da liderança. Eles contêm muito mais incerteza e baixa receita, então podem ser arrasados se não forem gerenciados independentemente do Horizonte 1.

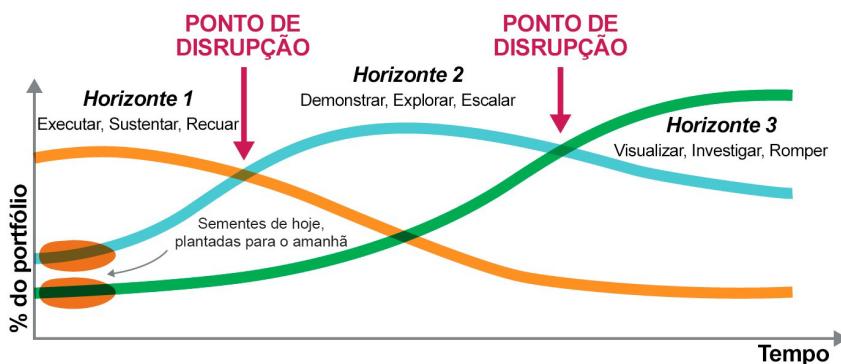


Figura 5.4: Porcentagem de portfólio de produto para os três horizontes de inovação ao longo do tempo

Devemos estar atentos às armadilhas em cada fase, incluindo fazer mudanças em hora incorreta e selecionar uma estratégia errada para cada horizonte.

### DESENVOLVIMENTO ENXUTO E OPERAÇÕES ENXUTAS, POR STEVE BELL

O pensamento lean geralmente é associado com operações, pois se originou com o Sistema Toyota de Produção e foi

amplamente adotado em configurações fabris. Apesar do conceito de “enxuto” ter origem no sistema fabril, ele evoluiu para muitos setores, incluindo o de saúde, serviços financeiros, transporte, construção, educação e setor público.

Mas o sucesso a longo prazo da Toyota é também devido à aplicação dos princípios enxutos para rapidamente e eficientemente desenvolver novos produtos de alta qualidade e custo razoável. A Toyota tem demonstrado que uma empresa que adota o pensamento lean sem entraves entre desenvolvimento e operações pode ganhar uma vantagem competitiva.

Desenvolvimento e operações enxutas são inter-relacionados e complementares, mas muito diferentes em sua natureza. Operações enxutas enfatizam a padronização e redução de desperdício, incerteza e oscilações para criar processos eficientes que geram produtos consistentes e de qualidade. Em contraste a isso, o desenvolvimento enxuto utiliza a incerteza e a oscilação bem no começo do processo de design para aprender com as experiências e, principalmente, com as falhas — que é a maneira mais efetiva de resolver problemas e guiar o processo de inovação.

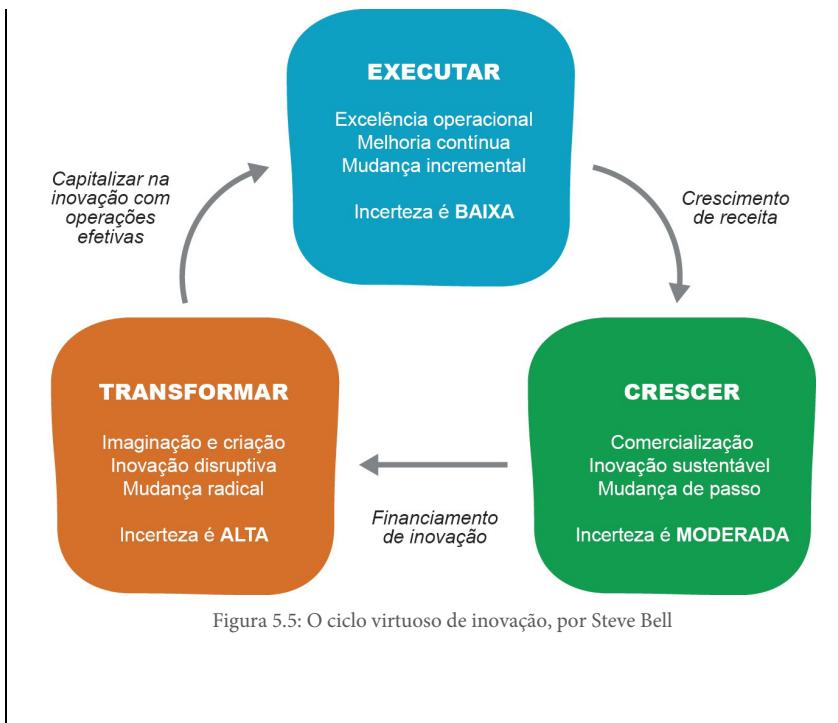
Ainda assim, aqui há o paradoxo: o desenvolvimento enxuto, que requer oscilação e incerteza, depende de métodos de trabalho padronizados para formular hipóteses para a inovação, e para executar experimentos consistentes e repetíveis que minimizem o desperdício e o tempo, ao mesmo tempo em que maximiza a criatividade e o valor.

Por exemplo, o desenvolvimento enxuto rigorosamente e continuamente envolve a Voz do Consumidor no *gemba* (o lugar físico e virtual onde o trabalho é feito), utilizando design iterativo e frequentemente *set-based* para acelerar o

aprendizado e rapidamente criar um produto com uma hipótese de valor válida. Outras práticas enxutas padronizadas, como solução de problemas A3, gerenciamento visual e mapeamento de fluxo de valor são úteis em um ambiente de desenvolvimento, melhorando a velocidade para o mercado enquanto reduz o custo de R&D e risco na empresa.

Uma vez que há um novo produto ou serviço viável, a empresa pode utilizar suas habilidades em operações enxutas para levar o produto ou serviço para o mercado com rapidez e eficiência, validando a hipótese do crescimento. Aqui é quando muitas startups enxutas perdem o mercado para similares rápidos ou são compradas por empresas maiores com capacidade de operar de maneira enxuta para rapidamente alcançar o mercado, explorar lucros no início e atingir a domínio de marca. Enquanto adquirir startups com uma fonte de inovação é certamente uma estratégia viável para grandes empresas, a maioria também gostaria de aumentar sua capacidade de inovação interna.

O desenvolvimento enxuto assim cria produtos e serviços inovadores que fluem pela operação enxuta e para as mãos do consumidor como um fluxo de valor contínuo, no mesmo espírito em que entrega contínua (DevOps) faz no contexto do software. Quando uma empresa é capaz de integrar e explorar esse rápido fluxo de ideias ao valor, os lucros de produtos maduros podem financiar a inovação contínua, criando um ciclo virtuoso, ilustrado pela figura:



Quando tentamos cruzar o Horizonte 3, os indicadores de satisfação do consumidor e engajamento contínuo são sinais importantes para monitorar, visando o futuro crescimento. Uma vez que encontramos clientes, aprendemos como satisfazer suas necessidades e estamos confiantes que atendemos sua demanda, devemos procurar expandir o mercado consumidor geograficamente, por cana ou por oferta.

Ao investigarmos, estamos testando uma combinação entre produto e mercado, normalmente por meio de soluções sugeridas por nossos clientes iniciais. Explorar é encontrar uma oferta e modelo de negócio atrativo para um ambiente consumidor mais amplo.

Os cinco facilitadores importantes para o crescimento durante a mudança de investigação para exploração são:

- **Mercado** — É imperativo selecionar o mercado certo. Idealmente, há muitos clientes em potencial que apoiarão nossas aspirações de crescimento; devemos identificar os elementos que nos tornaram bem-sucedidos com nossos primeiros seguidores, e então procurar encontrar grupos similares, porém maiores. Os insights que tivemos ao trabalhar com nossos primeiros seguidores são fundamentais para abastecer essa decisão. Primeiros seguidores também podem espalhar suas experiências com nosso produto de maneira boca a boca, eventualmente levando o produto a “cruzar o abismo” para uma adoção mais abrangente.
- **Modelo de monetização** — Devemos decidir qual é a melhor maneira de capturar o valor criado por nossa oferta, já que ela essencialmente define o que guiará a receita em nosso modelo de negócio. É algo difícil de mudar mais tarde.
- **Escolha de consumidor** — Como vamos angariar consumidores com o produto? Devemos ter cuidado para não fazer grandes concessões de produto ou preço para qualquer grupo individual para conquistar uma grande conta. Devemos permanecer fiéis à nossa visão de produto, gerenciar a tensão e demandar de qualquer grupo de consumidor que possa limitar nosso futuro crescimento.
- **Esqueça de lançamentos do tipo “big bang”** — Jogue com segurança: continue a testar e validar o produto, apague incêndios, trabalhe com amostras menores de clientes. Construa *momentum* por meio de lançamentos de produto alfa e beta com segmentos específicos de clientes. Ao ganhar mais confiança, entendimento e

sucesso, podemos ampliar nossa base de clientes. Idealmente, queremos que clientes cheguem até nós com problemas para resolver para que não tenhamos de empurrar novos produtos para eles.

- **Engajamento do time** — Devemos fazer tudo o que pudermos para manter o time unido para proteger a cultura, velocidade de aprendizado e conhecimento adquirido. Não queremos construir um muro entre os times de inovação e operação. A colaboração, direcionada para o aprendizado e o desenvolvimento da organização, é a chave para consolidar uma cultura de inovação enquanto começarmos a escalar e contratar novos membros para o time.

Quando considerar melhorias de processo e seleção de ferramentas, princípios similares se aplicam para identificar usuários-alvo, avaliação e captura de benefício, adoção de usuário, evitando lançamentos do tipo “big bang”, e engajamento do time.

### **INOVAÇÃO LEVA TEMPO: DO LEILÃO AO MERCADO**

Os leilões da Amazon (conhecidos como zShops) foram lançados em março de 1999 em resposta ao sucesso do eBay. O site foi promovido fortemente a partir das páginas home, categoria e produto individual. Apesar da promoção, um ano depois do lançamento ele havia atingido 3,2% do share do mercado de leilões online, comparado com 58% do share do eBay, e posteriormente diminuiu.

Em novembro de 2000, o zShops foi renomeado para Amazon Marketplace, oferecendo preços competitivos para produtos disponíveis por vendedores terceirizados junto com listas de produtos padrão. A estratégia, inicialmente guiada pela necessidade de competir com o eBay, foi ajustada para se alinhar com o foco estratégico da Amazon em preços baixos.

Estendendo o modelo ainda mais, a Amazon começou a vender produtos usados por meio de feiras de vendedores, fornecendo outro fluxo de receita sem qualquer impacto em sua cadeira de fornecedores. Propaganda, embalagem e envio são feitos exclusivamente pelos vendedores, com a Amazon ficando com uma fatia da transação por fornecer o canal de vendas com custo mínimo.

Em 2012, o serviço Amazon Marketplace gerou 12% de receita (MERINO, 2013) com total de vendas unitárias crescendo 32% em relação ao ano anterior (BROHAN, 2013).

Ao reconsiderar como definimos e medimos o aprendizado validado, podemos começar a testar, e comunicar se e quando nossas iniciativas estão ganhando força. Ao experimentarmos

continuamente com nossos clientes e movendo nossa Única Métrica que Importa o mais rápido e barato possível, podemos limitar nosso investimento, reduzir riscos associados e maximizar o aprendizado. Uma abordagem de desenvolvimento de produto baseada em evidência fornece segurança, contexto e dá cobertura para stakeholders — e é um catalizador para a mudança em grandes organizações.

## 5.5 CONCLUSÃO

A contabilidade para inovação fornece uma estrutura de trabalho para medir o progresso no contexto do Horizonte 3 — ou seja, sob condições de extrema incerteza. Ela é projetada para reunir os principais indicadores do crescimento futuro da ideia, para que possamos eliminar aqueles que não serão bem-sucedidos no Horizonte 2.

Nós identificamos as três áreas-chaves a serem consideradas nesse estágio. Primeiro, devemos achar clientes que atuem como cocriadores de valor. Usamos seu feedback para experimentar e refinar nossa proposta de valor antes de mirar em um mercado maior. Segundo, focamos em aprendizado em vez de receita ao estreitarmos nosso foco no cliente e validarmos cada suposição de nossa solução. Não precisamos construir requisitos; precisamos responder questões sobre as funcionalidades desejadas de nosso produto.

Finalmente, focamos em engajamento do usuário em vez de ganho financeiro — com mais usuários satisfeitos, virá mais receita (ou qualquer outro valor que esperamos obter para nossa organização). Enquanto melhorarmos nosso entendimento sobre nossos usuários e oportunidade de produtos, podemos decidir em um modelo de monetização para assegurar o sucesso contínuo para o produto.

A maioria das ideias não atingirá a relação produto/mercado. Para aquelas que atingirem, uma metamorfose é necessária. Os comportamentos e princípios de gestão necessários para o sucesso no Horizonte 2 são fundamentalmente diferentes daqueles que governam o Horizonte 3. A *Parte III* apresenta como fazer crescer uma organização focada em criar o produto corretamente, agora que temos confiança que estamos criando o produto certo.

## Questões para os leitores

- Que métricas de clientes e de negócios estariam em seu *scorecard* de inovação?
- Quem são os stakeholders-chaves, e qual sua influência para cada de estágio da curva de adoção de seu produto? Como você planeja engajá-los e criar alinhamento?
- Quais experimentos você planeja executar para testar e validar sua hipótese de negócio com clientes? Como você vai visualizá-las e priorizá-las?
- Como você pode reunir dados para testá-los em seu mercado identificado o mais barato e rápido possível?
- Quais são seus critérios para mover um produto do Horizonte 3 para o Horizonte 2?

# Explore

*“Previsão é difícil, especialmente sobre o futuro” — Niels Bohr*

Na *Parte II*, mostramos como investigar novas oportunidades — seja de potenciais produtos ou de serviços e ferramentas internas. Nesta parte, veremos como explorar ideias validadas. Como discutido no *capítulo 2*, esses dois domínios demandam uma abordagem completamente diferente para gerenciamento e execução. Contudo, ambos são necessários — e, na verdade, complementares — se quisermos equilibrar o portfólio de nossa empresa de maneira eficaz e nos adaptar a um ambiente de negócios em constante mudança.

Esperamos que você esteja lendo esta parte porque saiu com sucesso do domínio da investigação, mas é provável que você esteja aqui porque participou de um grande programa de trabalho em uma empresa que se estabeleceu da maneira tradicional. Assim, esta parte do livro descreve principalmente como mudar a maneira com a qual conduzimos e gerenciamos tais programas de trabalho em larga escala, de modo que empodere os empregados e aumente dramaticamente a taxa pela qual entregamos produtos valiosos e de alta qualidade para os consumidores. Mas, antes de começarmos, devemos entender nossa condição atual.

No contexto empresarial, o trabalho planejado é normalmente priorizado por meio de um planejamento por departamento ou centralizado e processo de orçamento. Projetos aprovados então passam pelo processo de desenvolvimento antes de entrarem no ar

ou serem liberados para produção. Mesmo em organizações que adotaram métodos de desenvolvimento ágeis, o fluxo de valor necessário para entregar um projeto geralmente lembra a figura adiante, que descrevemos como “water-scrum-fall”.

O termo “water-scrum-fall” foi cunhado por Forrester Research. Um fluxo de valor é definido como “a sequência de atividades que uma organização faz para entregar um pedido de cliente” (MARTIN; OSTERLING, p. 2). Falamos sobre fluxos de valor no *capítulo 7*.

Nos casos em que uma ou mais destas fases são terceirizadas, devemos também passar pelo processo de aquisição antes de proceder para as fases de projeto e desenvolvimento depois da aprovação. Devido a esse processo ser oneroso, tendemos a dividir o trabalho em lotes, criando grandes programas que agravam ainda mais os problemas com o paradigma do projeto.

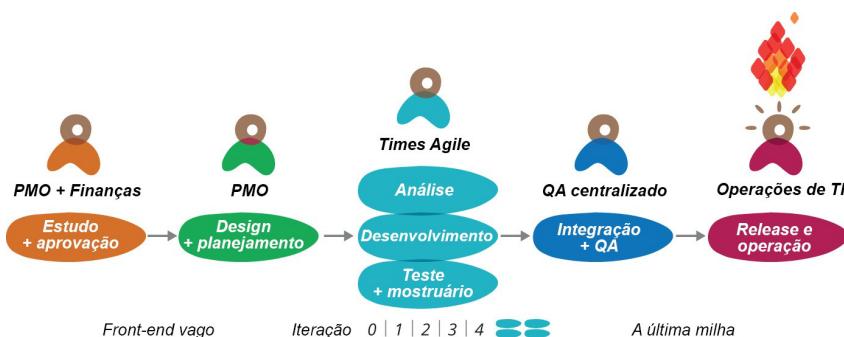


Figura 1: Water-scrum-fall

Este paradigma baseado em projeto para executar desenvolvimento de software em larga escala tem sua origem no complexo militar-industrial dos Estados Unidos, pós-Segunda

Guerra Mundial, em que o software foi crucial para construir uma nova geração de aviões, sistemas de mísseis e aeronaves que tinham essencialmente um cliente: o governo americano. Não é coincidência que o termo “engenharia de software” foi cunhado em uma conferência da OTAN em 1968, que foi convocada para resolver como formalizar o desenvolvimento de software em larga-escala (<http://homepages.cs.ncl.ac.uk/brian.randell/NATO>).

O paradigma tradicional centralizado de projeto *stage-gate* foi projetado em uma época mais simples. Produtos não entregavam valor até que estivessem completamente produzidos, e eles não precisavam mudar substancialmente durante seu ciclo de vida. Também tínhamos um alto grau de confiança em que não precisaríamos mudar a especificação significantemente em resposta a novas informações descobertas durante a construção do produto.

Nenhum desses critérios se aplica a sistemas de software hoje em dia, e o poder do software deriva do fato de que é barato fazer protótipo e alterá-lo. Em especial, já que frequentemente estamos errados sobre o que os usuários de nossos produtos e sistemas vão achar valioso, planejar grandes programas de trabalho com meses de antecedência leva a grandes desperdícios e animosidade. Em vez de tentar melhorar em prever o futuro, deveríamos melhorar nossa capacidade de nos adaptarmos rapidamente e efetivamente a novas informações (esta é uma reivindicação-chave da obra de Nassim Taleb).

O paradigma moderno, enxuto e ágil, que apresentamos para executar programas de trabalho em larga escala discutidos nesta parte é resultado de trabalhar e estudar um monte de organizações que precisam tirar o desenvolvimento de software do caminho crítico. Elas querem se mover rapidamente em grande escala, detectar sinais fracos no mercado e explorá-los rapidamente. Isso é o que permite que elas forneçam melhor serviço para o consumidor,

reduzam custo de criar e desenvolver produtos, e aumentem a qualidade e estabilidade de seus serviços.

Há várias estruturas de trabalho que lidam com escalonamento de métodos ágeis de desenvolvimento de software. Em geral, essas estruturas pegam pequenos times que praticam Scrum e adicionam mais estruturas para coordenar seu trabalho. Contudo, esses times ainda estão incorporados ao um programa *stage-gate* e ao processo de gerenciamento de portfólio, que é mais ou menos a mesma coisa que o paradigma tradicional do gerenciamento de projeto. Eles ainda usam o pensamento de cima para baixo e tendem a lotear o trabalho em releases com tempos de ciclo longos, limitando o uso da informação coletada para guiar futuras decisões. Nossa abordagem difere em vários aspectos importantes dessas estruturas, assim como das estruturas *stage-gate* mais tradicionais.

A diferença mais importante é que, em vez de apresentar um conjunto particular de processos e práticas para implementar, focamos na implementação da melhoria contínua no nível de liderança sênior para guiar a evolução de sua organização e dos processos que você usa. A melhoria contínua não pode estar nas extremidades de nosso “grande diagrama”: nós a colocamos na frente e no centro.

Isso reflete o fato de que não há uma solução única e que toda organização enfrenta circunstâncias diferentes. Toda organização construirá seu próprio caminho para abordar mudanças, alinhado a seus próprios objetivos de negócios. Para criar resultados duradouros, devemos permitir que os times tentem as coisas, e aprendam o que funciona e o que não funciona para eles.

Nos capítulos seguintes, apresentaremos os seguintes princípios para desenvolvimento de produto enxuto e ágil em escala:

- Implementar um processo iterativo de melhoria

contínua no nível de liderança com resultados concisos para criar alinhamento em grande escala, seguindo o Princípio da Missão.

- Trabalhar cientificamente em direção a objetivos desafiadores, que o levará a identificar e remover — ou evitar — atividades que não agregam valor.
- Usar a entrega contínua para reduzir o risco em releases, diminuir o tempo de ciclo e tornar econômico trabalhar com pequenos lotes.
- Desenvolver uma arquitetura pouco acoplada que suporte times voltados para o cliente que têm autonomia em como trabalhar para conseguir os resultados no nível de programa.
- Reduzir os tamanhos dos lotes de trabalho e usar uma abordagem experimental no processo de desenvolvimento do produto.
- Aumentar e amplificar os loops de feedback para tomar decisões menores e mais frequentemente, baseados em informações que temos da performance de nosso trabalho para maximizar o valor para o cliente.

Também forneceremos vários exemplos de empresas que influenciaram esses princípios para criar uma vantagem competitiva duradoura, e descreveremos como elas se transformaram no processo.

## CAPÍTULO 6

# IMPLEMENTE ENTREGA CONTÍNUA

*“O paradoxo é que quando os gerentes focam em produtividade, melhorias de longo prazo raramente são feitas. Por outro lado, quando gerentes focam em qualidade, a produtividade melhora continuamente” — John Seddon*

Na maioria das empresas já estabelecidas, há uma distinção entre pessoas que constroem e operam os sistemas (frequentemente chamados de “TI”) e aqueles que decidem o que o sistema deve fazer e fazem as decisões sobre investimento (frequentemente chamados de “o negócio”). Estes nomes são relíquias de uma era passada em que a TI era considerada um custo necessário para melhorar eficiências do negócio, e não um criador de valor para clientes externos através da criação de produtos e serviços.

Estes nomes e a separação funcional têm sobrevivido em muitas organizações (assim como também o relacionamento entre eles, e a mentalidade que frequentemente vem com este relacionamento). Por fim, nós visamos remover esta distinção. Em organizações atuais de alto desempenho, pessoas que projetam, constroem e operam produtos baseados em software são parte integral do negócio; são dadas a elas — e elas aceitam — responsabilidades por

resultados de cliente. Mas chegar a este estado é difícil, e é muito fácil voltar às maneiras antigas de se fazer as coisas.

Alcançar alto desempenho em organizações que tratam software como uma vantagem estratégica depende do *alinhamento* entre o departamento de TI e o resto da organização, junto com a habilidade de TI para *executar*. Vale a pena. Em um relatório do *MIT Sloan Management Review, Avoiding the Alignment Trap in Information Technology*, os autores pesquisaram 452 empresas e descobriram que aquelas com alto desempenho (7% do total) gastam um pouco menos que a média em TI enquanto alcançam consideravelmente taxas mais altas de crescimento de faturamento (SCHPILBERG; BEREZ; PURYEAR; SHAH, 2007).

Entretanto, *como* você se move de baixo desempenho para alto desempenho importa. Empresas com baixo alinhamento e departamentos de TI têm uma opção de escolha. Elas devem buscar alinhamento primeiro, ou tentar melhorar sua habilidade de execução? Dados mostram que empresas em que as capacidades de TI eram ruins alcançavam piores resultados quando elas buscavam alinhamento com as prioridades de negócio antes de execução, mesmo quando colocavam um investimento adicional significativo em trabalho alinhado. Por outro lado, empresas em que os times de engenharia fazem um bom trabalho entregando o seu trabalho no prazo e simplificando os seus sistemas alcançavam melhores resultados de negócio com bases de custo muito menores, mesmo se os seus investimentos de TI não estivessem alinhados com o negócio.

Os pesquisadores concluíram que, para alcançar um alto desempenho, empresas que dependem de software devem focar-se primeira e principalmente na sua habilidade de execução, construir sistemas confiáveis, e trabalhar continuamente para reduzir complexidade. Somente então a busca de alinhamento com as

prioridades de negócio terá êxito.

Porém, em todo time nós estamos sempre balanceando os esforços que fazemos para melhorar nossas habilidades com o trabalho de entrega que provê valor aos clientes. Para fazer isso de maneira efetiva, é essencial gerenciar ambos os tipos de trabalho, tanto nos níveis de programa como de fluxo de valor. Neste capítulo, nós descrevemos como alcançar isto através de um framework chamado *Melhoria Kata*.

Este é o primeiro passo que devemos fazer para levar melhoria contínua à execução de programas de grande escala. Uma vez que tenhamos alcançado isso, nós podemos usar as ferramentas dos capítulos seguintes para identificar e remover atividades que não agregam valor no nosso processo de desenvolvimento de produtos.

## 6.1 ESTUDO DE CASO: HP LASERJET FIRMWARE

Nós vamos começar com um estudo de caso do time de Firmware LaserJet da HP, que encontrou problemas tanto de alinhamento como de execução (GRUVER, 2012). Como o nome sugere, este time estava trabalhando em software embarcado, em que os clientes não tinham nenhum desejo de receber atualizações frequentes do software. Entretanto, este caso provê um excelente exemplo de como os princípios descritos no restante da *Parte III* funciona em escala para um time distribuído, assim como os benefícios econômicos em adotá-los.

A divisão de Firmware LaserJet da HP constrói os softwares embarcados que rodam em todos os equipamentos como scanners, impressoras e multifuncionais. O time consiste em 400 pessoas distribuídas entre Estados Unidos, Brasil e Índia. Em 2008, esta divisão tinha um problema: eles estavam movendo-se muito

devagar. Eles estavam no caminho crítico do lançamento de novos produtos por anos, e estavam incapazes de entregar novas funcionalidades: “Marketing vinha até nós com milhões de ideias que deslumbrariam o cliente, e nós simplesmente respondíamos ‘Da sua lista, escolha as duas coisas que você gostaria de ter nos próximos 6-12 meses’”. Eles tentaram aumentar os gastos, contratar e terceirizar para resolver o problema, mas nada funcionou. Eles necessitavam uma nova abordagem.

O primeiro passo foi entender o problema com maior profundidade. Eles o fizeram através de *contabilidade por atividade* — alocar gastos para as atividades que o time está fazendo. A Tabela 6-1 mostra o que eles descobriram.

**Tabela 6-1 — Atividades do time de Firmware LaserJet da HP em 2008**

% dos custos	Atividade
10%	Integração de código
20%	Planejamento detalhado
25%	Portando o código entre <i>branches</i> do sistema de controle de versões
25%	Suporte ao produto
15%	Teste manual
~5%	Inovação

Isto revelou muito sobre as atividades que não agregavam valor, como portando o código entre *branches* e planejamento detalhado de maneira antecipada. A grande porção de tempo gasta em suporte ao produto também indicava um problema na qualidade do software sendo produzido. Dinheiro gasto em suporte está geralmente servindo a *demand-falha*, ao contrário de *demandavaliosa*, que estava gerando apenas 5% dos custos do time.

A distinção entre demanda-falha e demanda-valiosa vem de John Seddon, que notou que, quando bancos terceirizavam o seu atendimento ao cliente para centrais de atendimento, o volume de chamados crescia enormemente. Ele mostrou que até 80% dos chamados eram “demandas falhas” de pessoas chamando o banco porque os seus problemas não foram solucionados corretamente na primeira vez (SEDDON, 1992).

O time tinha o objetivo de melhorar a proporção de gastos em inovação por um fator de 10. Para alcançar este objetivo, eles tomaram a corajosa, mas arriscada, decisão de construir do zero uma nova plataforma de software embarcado. Havia dois objetivos arquiteturais principais para a nova plataforma, batizada de “FutureSmart”. O primeiro era melhorar a qualidade ao mesmo tempo em que se reduz a quantidade de testes manuais necessária para lançar novas versões de firmwares (um ciclo manual de testes completo tomava seis semanas). O time esperava que este objetivo poderia ser alcançado por meio de:

- Prática de integração contínua (que nós descrevemos no *capítulo 8*).
- Investimento significativo em automação de testes.
- Criação de simuladores de hardware, para assim testes poderem ser realizados em uma plataforma virtual.
- Reprodução das falhas em testes nas estações de trabalho dos desenvolvedores.

Três anos depois de começar o desenvolvimento do novo software, milhares de testes automatizados foram criados.

Segundo, eles queriam remover a necessidade dos times

gastarem tempo portando o código entre *branches* (25% do custo total do sistema existente). Isto era causado pela necessidade de criar um *branch* — em realidade, uma cópia de toda a base de código — para cada nova linha de equipamentos em desenvolvimento. Se uma funcionalidade ou correção de falha adicionada a uma linha de equipamentos era também necessária por outras, estas mudanças necessitariam ser combinadas (copiadas de volta) nos *branches* de código relevantes para o equipamento alvo, como mostrado na figura a seguir.

Mover-se para longe de desenvolvimento baseando em *branches* e em direção ao desenvolvimento baseado em *trunk* também foi necessário para implementar integração contínua. Assim o time decidiu criar uma plataforma única e modular que podia operar em qualquer equipamento, removendo a necessidade de usar *branches* no controle de versão para gerenciar as diferenças entre os equipamentos.

O objetivo final do time era reduzir a quantidade de tempo que os seus membros gastavam em atividades de planejamento detalhado. As áreas responsáveis pelo marketing de várias linhas de produtos insistiam em planos detalhados, porque elas simplesmente não podiam acreditar que o time entregaria. A maior parte deste tempo era gasta em replanejamentos detalhados depois de não seguir os planos originais.

Além disso, o time não sabia como implementar a nova arquitetura, e nunca antes tinha usado desenvolvimento baseado em *trunk* ou integração contínua nesta escala. Eles também entendiam que automatização de testes exigiria um grande investimento. Como eles avançariam?



Em desenvolvimento baseado em trunk (acima), os desenvolvedores fazem as mudanças diretamente no trunk. Em um típico estilo de desenvolvimento não baseado em trunk (abaixo), os desenvolvedores tipicamente fazem suas mudanças em branches que sobrevivem por muito tempo, que então são estabilizadas antes de serem lançadas.

Linha de produto A

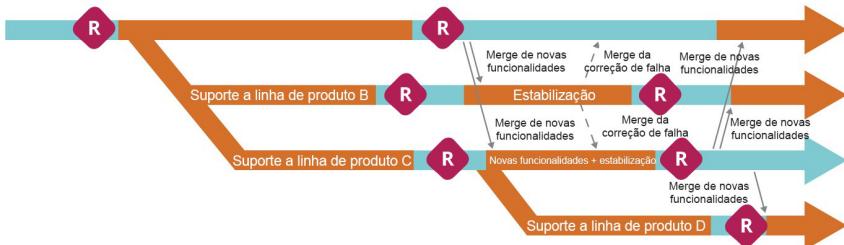


Figura 6.1: Branching versus desenvolvimento baseado em trunk

É muito fácil tornar uma sequência de eventos em uma história como uma tentativa de explicar o resultado — um viés cognitivo que Nassim Taleb descreve como *falácia da narrativa*. Isto é, indiscutivelmente, como metodologias nascem. O que nos chocou quando nós estudamos o caso *FutureSmart* foram as similaridades entre os métodos de gestão de programa do time de engenharia do *FutureSmart* com a abordagem que a Toyota usa para gerir inovação descrito no livro de Mike Rother (2010), chamado *Toyota Kata: Gerenciando Pessoas para Melhoria, Adaptabilidade e Resultados Excepcionais*.

## 6.2 DIMINUA OS CUSTOS ATRAVÉS DA INOVAÇÃO CONTÍNUA DE PROCESSOS USANDO MELHORIA KATA

A Melhoria Kata, como descrito por Mike Rother, é um

framework genérico e um conjunto de rotinas práticas para alcançar objetivos cujo caminho é incerto. Ele requer que procedamos em passos iterativos e incrementais, usando ciclos muito rápidos de experimentação.

Seguir a Melhoria Kata também melhora as capacidades e habilidades das pessoas fazendo o trabalho, uma vez que as pessoas resolvem os seus próprios problemas por meio de um processo de experimentação contínua, formando assim uma parte integral de qualquer organização de aprendizagem. Por fim, também reduz os custos através da identificação e eliminação de desperdícios em nossos processos.

A Melhoria Kata precisa ser primeiro adotado pela gestão da empresa, porque ele é uma filosofia de gestão que foca no desenvolvimento das capacidades daqueles que gerem, ao mesmo tempo em que permite que as organizações aproximem-se de seus objetivos em condições incertas. No final, todos na empresa devem estar praticando A Melhoria Kata habitualmente para alcançar objetivos e superar desafios. Isto é o que cria uma cultura de melhoria contínua, experimentação e inovação.

Para entender melhor como isto funciona, vamos primeiro examinar o conceito de *kata*. Um kata é “uma rotina que você pratica deliberadamente, assim o seu padrão se torna um hábito” (ROTHER, 2014). Pense em como praticar escalas para desenvolver memória muscular e destreza digital quando se aprende a tocar piano, ou praticar padrões básicos de movimento quando se aprende uma arte marcial (de onde o termo é derivado), ou um esporte. Nós queremos fazer da melhoria contínua um hábito, pois quando encaramos um ambiente onde o caminho até o objetivo é incerto, nós temos uma rotina instintiva e inconsciente para guiar nosso comportamento.

Na Toyota, uma das principais tarefas dos gerentes é ensinar o

padrão da Melhoria Kata para os times e facilitar a sua execução (incluindo treinar aprendizes) como parte do seu trabalho no dia a dia. Isto equipa os times com um método para resolver os seus próprios problemas. A beleza desta abordagem é que, se o objetivo ou o ambiente da empresa mudam, nós não precisamos mudar a maneira como trabalhamos. Se todos estão praticando a Melhoria Kata, a empresa vai automaticamente adaptar-se às novas condições.

A Melhoria Kata tem quatro estágios que nós repetimos em um ciclo, como mostrado na figura:

## OS QUATRO PASSOS DO MÓDELO KATA DE MELHORIA

Um padrão sistemático e científico de trabalho



Figura 6.2: A Melhoria Kata, cortesia de Mike Rother

### Entenda a direção

Nós começamos por entender a direção. A direção é derivada da visão estabelecida pela liderança da organização. Uma boa visão é aquela que é inspiradora e, potencialmente, inalcançável na prática. Por exemplo, a visão de longo prazo para as operações produtivas da Toyota é “Fluxo de uma peça ao menor custo possível”. No livro *Leading Lean Software Development*, Mary e Tom Poppendieck

(2009, quadro 4) descrevem que Paul O'Neill estabeleceu que o objetivo para a Alcoa seria “Segurança perfeita para todas as pessoas que tem qualquer relação com a Alcoa”, assim que assumiu como CEO em 1987.

As pessoas precisam entender que elas devem sempre estar trabalhando em direção à visão e que nunca estarão completos com melhorias. Nós vamos encontrar problemas à medida que nos movemos em direção à visão. A dica é tratá-los como obstáculos a serem removidos por meio de experimentação em vez de oposições à experimentação e à mudança.

Baseados na nossa visão e seguindo o Comando de Missão, nós devemos entender a direção em que estamos trabalhando, no nível organizacional e no nível de fluxo de valor. O objetivo poderia ser representado na forma de mapa de fluxo de valor de estado futuro (veja mais no *capítulo 7* sobre mapeamento de fluxos de valor). Ele deve resultar em um resultado mensurável para os nossos clientes, e nós devemos planejar alcançá-lo entre seis meses a três anos.

## **Planejando: compreenda a atual condição e estabeleça uma condição-alvo**

Depois que nós entendemos a direção nos níveis organizacionais e de fluxo de valor, nós incrementalmente e iterativamente nos movemos para o nível de processos. Rother recomenda estabelecer condições-alvo em um horizonte entre uma semana a três meses, com uma preferência por horizontes mais curtos para iniciantes. Para times que já estão usando métodos iterativos e incrementais para fazer desenvolvimento de produtos, faz sentido usar os mesmos marcos de iteração (ou *sprint*) também para as iterações de desenvolvimento de produto e Melhoria Kata. Times usando métodos baseados em fluxo como Kanban (que vemos no *capítulo 7*) e entrega contínua (descrito no *capítulo 8*) podem criar iterações

de Melhorias Kata no nível de programa.

Assim como todos os métodos iterativos de desenvolvimento de produtos, as iterações das Melhorias Kata envolvem uma parte de planejamento e uma parte de execução. Aqui, o planejamento envolve compreender a condição atual no nível de processos e estabelecer uma condição-alvo que nós visamos alcançar ao final da próxima iteração.

Analisando a condição atual, “é feita para obter os fatos e dados que você precisa para então descrever a próxima condição-alvo apropriada. O que você está fazendo é tentar encontrar o padrão atual de operação, assim você pode estabelecer um padrão de operação desejado (uma condição-alvo)” (ROTHER, 2014). A condição-alvo “descreve em detalhes mensuráveis como você quer que um processo funcione... [É uma] descrição e especificação do padrão de operação que você quer que um processo ou sistema tenha em uma data futura” (ROTHER, 2014).

O time comprehende a condição atual e estabelece uma condição-alvo juntos. Porém, na fase de planejamento, o time não planeja como mover-se à condição-alvo. Na Melhoria Kata, as pessoas que realizam o trabalho empenham-se em alcançar uma condição-alvo realizando uma série de experimentos, não seguindo um plano.

A condição-alvo identifica o processo sendo atacado, estabelece uma data em que nós visamos alcançar a condição determinada, e especifica detalhes mensuráveis do processo que nós queremos que exista. Exemplos de condições-alvo incluem limites do trabalho em andamento (*WIP*), a implementação de um Kanban ou um processo de integração contínua, o número de boas compilações que queremos alcançar por dia, e assim por diante.

## Alcançando a condição-alvo

Já que estamos nos engajando em um processo de inovação em condições de incerteza, nós não podemos saber com antecipação como nós alcançaremos a condição-alvo. Cabe às pessoas realizando o trabalho executar uma série de experimentos usando o Ciclo Deming (planejar, executar, verificar, agir), como descrito no *capítulo 3*. Os principais erros que as pessoas fazem ao seguir o Ciclo Deming são executá-lo de maneira muito infrequente e tomar muito tempo para completar um ciclo. Com a Melhoria Kata, todos devem realizar experimentos no dia a dia.

Todos os dias as pessoas do time devem responder as cinco questões a seguir (ROTHER, 2014):

1. Qual é condição-alvo?
2. Qual é a real condição agora?
3. Quais são os obstáculos que você pensa estarem impedindo-nos de alcançar a condição-alvo? Quais deles você está atacando agora?
4. Qual o seu próximo passo? (Comece o ciclo planejar-executar-verificar-agir.) O que você espera?
5. Quando nós podemos ver o que aprendemos com o passo que tomamos?

À medida que continuamente repetimos o ciclo, nós refletimos no último passo que tomamos para introduzir melhorias. O que nós esperávamos? O que realmente aconteceu? O que nós aprendemos? Nós devemos trabalhar no mesmo obstáculo por vários dias.

Esta abordagem experimental já é central em como engenheiros e designers trabalham. Designers que criam e testam protótipos para reduzir o tempo que um usuário precisa para completar uma tarefa estão engajados exatamente neste processo. Para desenvolvedores de software que usam desenvolvimento orientado a testes (TDD), toda linha de código de produção escrita é essencialmente parte de um

experimento para fazer um teste unitário passar. Isto é, de fato, um passo no caminho de aumentar o valor entregue pelo programa — que pode ser especificado na forma de condição-alvo, como nós descrevemos no *capítulo 9*.

A Melhoria Kata é uma simples generalização desta abordagem para melhoria, combinada com a aplicação em múltiplos níveis da organização, como nós discutiremos quando apresentamos a estratégia de implantação no *capítulo 15*.

## Como a Melhoria Kata é diferente de outras metodologias

Você pode pensar na Melhoria Kata como uma *metametodologia*, já que ela não se aplica a nenhum domínio em particular, nem diz a você o que fazer. Não é um guia; em vez disso, assim como o método Kanban, ensina os times a como *evoluir* o seu próprio guia. Neste sentido, diferencia-se de outras plataformas e metodologias ágeis.

Com a Melhoria Kata, não há a necessidade de fazer processos existentes adaptarem-se ao especificados na metodologia; espera-se que processos e práticas que você já usa evoluam ao longo do tempo. *Esta* é a essência de ágil: times não se tornam ágeis ao seguir uma metodologia. Em vez disso, times verdadeiramente ágeis são os times que estão constantemente trabalhando para evoluir seus processos para lidar com obstáculos pontuais que eles enfrentam a qualquer momento.

## **APRENDIZADO DE CICLO SIMPLES E APRENDIZADO DE CICLO DUPLO**

Mudar a maneira como pensamos e nos comportamos em relação à falha é crucial para um aprendizado efetivo. Isto é o que diferencia o *aprendizado de ciclo simples* e o *aprendizado de ciclo duplo* (veja a figura adiante). Estes termos foram cunhados pelo teórico de negócios Chris Argyris, que os sumariza como: “Quando um erro detectado e corrigido permite que a organização continue com as políticas presentes e alcance os objetivos presentes, então este processo de erro-e-correção é um aprendizado de ciclo simples. Aprendizado de ciclo simples é como um termostato que percebe quando está muito quente ou muito frio, e liga ou desliga o aquecedor. O termostato pode realizar esta tarefa, porque pode receber informação (a temperatura da sala) e tomar a ação corretiva. Aprendizado de ciclo duplo ocorre quando um erro é detectado e a sua correção envolve a modificação de normas, políticas e objetivos básicos da organização” (ARGYRIS; SCHÖN, 1978, p. 2–3).

Argyris defende que a principal barreira para o aprendizado de ciclo duplo é a defensividade quando confrontados com evidência de que nós precisamos de mudança na nossa maneira de pensar, o que pode acontecer tanto nos níveis individuais e organizacionais. Nós discutimos como superar a ansiedade e defensividade no *capítulo 11*.



Figura 6.3: Aprendizado de ciclo simples e aprendizado de ciclo duplo

Quando você pratica a Melhoria Kata, a melhoria de processos se torna trabalho planejado, similar a construir incrementos em produtos. A chave é que nós não planejamos *como* nós vamos alcançar a condição-alvo, nem criamos épicos, funcionalidades, histórias ou tarefas. Em vez disso, os times resolvem através de experimentação ao longo de uma iteração.

### **IMPLANTANDO A MELHORIA KATA**

O trabalho de Rother (2010) na Melhoria Kata foi um resultado direto da sua pesquisa sobre como as pessoas se tornavam gerentes na Toyota. Não há treinamento formal, nem há alguma instrução explícita. Contudo, para tornar-se um gerente na Toyota, a pessoa deve ter primeiro trabalhado no chão de fábrica, e assim participado da Melhoria Kata. Através deste processo, gerentes recebem treinamento implícito em como tornar-se um gerente na Toyota.

Isto apresenta um problema para as pessoas que querem aprender a gerenciar desta maneira ou adotar os padrões da Melhoria Kata. Isto também é um problema para a Toyota — que está buscando alcançar escala de maneira mais rápida do que o possível através do que efetivamente é um modelo de aprendiz para gerentes.

Consequentemente, Rother apresenta o Coaching Kata em complemento à Melhoria Kata. Ele é parte da implantação da Melhoria Kata, mas é também uma maneira de formar pessoas capazes de trabalhar com Melhorias Kata, inclusive gerentes.

Rother disponibilizou um guia para implantar a Melhoria Kata, *The Improvement Kata Handbook*, disponível gratuitamente no seu website, em <http://bit.ly/11iBzlY>.

## **6.3 COMO O TIME DA HP LASERJET IMPLEMENTOU A MELHORIA KATA**

A direção estabelecida pela liderança da HP LaserJet foi melhorar a produtividade dos desenvolvedores por um fator de 10,

para assim colocar o *firmware* fora do caminho crítico do desenvolvimento de produtos e reduzir custos (GRUVER, 2012, p. 144). Eles tinham três objetivos de alto nível:

1. Criar uma plataforma única para suportar todos os equipamentos;
2. Melhorar a qualidade e reduzir a quantidade de estabilização necessária antes de cada liberação;
3. Reduzir a quantidade de tempo usada em planejamento.

Eles não sabiam os detalhes do caminho até estes objetivos e não tentaram defini-los. A decisão chave foi trabalhar em iterações, e estabelecer condições-alvo para o fim de cada iteração de quatro semanas. A condição-alvo para a iteração 30 (aproximadamente 2.5 anos dentro do desenvolvimento da plataforma FutureSmart) são mostrados na figura adiante.

O primeiro ponto a observar é que as condições-alvo (ou “critério de saída” como é conhecido no FutureSmart) são todas condições mensuráveis. De fato, elas cumprem todos os elementos de objetivos SMART: eles são específicos, mensuráveis, alcançáveis, relevantes, e com limites de tempo (o limite por virtude do processo iterativo). Além disso, muitas das condições-alvo não estavam focadas em funcionalidades a serem desenvolvidas, mas sim em atributos do sistema, como qualidade, e em atividades projetadas para validar estes atributos, como testes automatizados. Finalmente, os objetivos para todo o programa distribuído com 400 pessoas para um único mês era capturado em um formato conciso para caber em uma única página de papel — similar ao método A3 usado pelo Sistema de Produção Toyota.

Como as condições-alvo são escolhidas? Elas são “objetivos agressivos que o time sente serem possíveis e importantes para alcançar em 4 semanas... Nós tipicamente nos dirigimos fortemente

para estes objetivos desafiadores, mas normalmente acabamos atingindo por volta de 80% do que nós pensávamos que podíamos alcançar no início do mês” (GRUVER, 2012, p. 40). Frequentemente, condições-alvo seriam alteradas ou mesmo removidas se o time descobre que tentar alcançá-las resulta em consequências não desejadas: “É surpreendente o que você aprende em um mês e tem de ajustar baseado no descobrimento durante o desenvolvimento” (GRUVER, 2012, p. 40).

<b>Rank</b>	<b>Tema</b>	<b>Critério de Saída</b>
		Objetivos alcançados / Objetivos não alcançados
0	Limites de Qualidade	Caso aberto de P1 < 1 semana Falhas no teste de resposta 24 horas de L2
1	Liberação trimestral	A) Pedidos finais de mudança P1 corrigidos B) Taxas de erros de confiabilidade no critério de liberação
2	Estabilidade da nova plataforma e cobertura de testes	A) 100% dos testes de aceitação do cliente passando B) 98% de todos os testes do pilar L2 passando C) Testes do pilar L4 criados D) Cobertura de testes de L4 para todos os requerimentos do produto Turn On E) 100% de execução de testes L4 para novos produtos
3	Dependências e funcionalidades chaves do produto Turn On	A) Imprimir por uma hora em velocidade e terminar com o grampeador B) Fazer cópias por uma hora em velocidade C) Habilitar o modo economizador de energia D) Execução noturna dos testes de fabricação E) Suporte a Biblioteca Comum de Testes para o display de 4-linhas do painel de controle
4	Construir a próxima geração de produtos	A) Compilação do sistema de ponta a ponta para o novo processador B) Análise de alto nível do desempenho do novo processador
5	Plano de integração de frota	Alinhar o conteúdo e cronograma das "fatias" do processo de testes ágil ponta a ponta com o laboratório de sistemas

Figura 6.4: Condições-alvo para a iteração 30 (GRUVER; YOUNG; FULGHUM, 2013)

### O QUE ACONTECE QUANDO NÓS NÃO ALCANÇAMOS A NOSSA CONDIÇÃO-ALVO?

Em culturas organizacionais burocráticas ou patológicas, não alcançar 100% das condições-alvo especificadas é tipicamente

considerado um fracasso. Em uma cultura geradora, entretanto, nós *esperamos* não sermos capazes de alcançar todas as condições-alvo. O propósito de estabelecer condições-alvo agressivas é para revelar obstáculos, pois assim podemos superá-los através de mais melhoria do trabalho.

Toda iteração deve terminar com uma retrospectiva (descrita no *capítulo 11*) na qual nós investigamos como podemos melhorar. Os resultados formam parte dos dados de entrada para as condições-alvo da próxima iteração. Por exemplo, se nós falhamos em alcançar a condição-alvo para o número de bens construídos pelo sistema por dia, nós podemos descobrir que o problema está no tempo para aprovisionar os ambientes de teste. Nós podemos então estabelecer uma condição-alvo de reduzir este tempo na próxima iteração.

Esta abordagem é um assunto comum em todo o pensamento Lean. O subtítulo do livro *Liderando Desenvolvimento Lean de Software* de Mary e Tom Poppendieck é: “Resultados não são o ponto”. Esta é uma afirmação provocativa que vai ao coração da mentalidade Lean. Se nós alcançamos o resultado ignorando o processo, nós não aprendemos como melhorar o processo. Se nós não melhoramos o processo, nós não podemos repetidamente alcançar melhores resultados. Organizações que colocam processos não modificáveis que todos são obrigados a seguir, mas que são ignorados em uma situação de crise, falha em ambos os lados.

Esta abordagem adaptativa e iterativa não é nova. De fato tem uma grande parte em comum com o que Tom Gilb propôs no seu livro de 1988, chamado *Princípios de Gestão de Engenharia de Software* (1988, p. 91):

*“Nós devemos estabelecer objetivos mensuráveis para cada próximo pequeno passo de entrega. Mesmo estes são suscetíveis a constantes modificações à medida que nós aprendemos sobre a realidade. Simplesmente não é possível estabelecer uma série de objetivos funcionais, de qualidade e de recursos, e estar certo de alcançá-los todos como planejado. Nós devemos estar preparados para comprometer e equilibrar. Nós devemos então projetar (engenharia) a solução técnica imediata, construí-la, testá-la, entregá-la e ter feedback. Este feedback deve ser usado para modificar o projeto imediato (se necessário), modificar as principais ideias arquiteturais (se necessário), e modificar tanto os objetivos de curto prazo como os de longo prazo (se necessário)”.*

## **PLANEJANDO PARA O DESENVOLVIMENTO ITERATIVO**

Em grandes programas, demonstrar melhora dentro de uma iteração requer ingenuidade e disciplina. É comum sentir que nós não podemos mostrar progresso significante em 2 a 4 semanas. Sempre tente encontrar alguma coisa pequena para atacar para alcançar um pouco de melhora em vez de tentar fazer alguma coisa que você crê que terá mais impacto, mas que tomará mais tempo.

Esta não é uma ideia nova, claro. Ótimas equipes têm trabalhado desta forma por décadas. Um caso conhecido é o projeto do Apple Macintosh, em que um time de aproximadamente 100 pessoas — localizadas em um único edifício — projetou o hardware, o sistema operacional e as aplicações para o que viria a ser um produto importantíssimo da Apple.

Os times integravam frequentemente o hardware, sistema operacional e software para mostrar progresso. O projetista de hardware, Burrell Smith, empregou chips lógicos programáveis (PALs), pois assim ele podia criar protótipos com diferentes abordagens para desenhos de hardware rapidamente no processo de desenvolver o sistema, adianto o ponto em que se tornava fixo — um grande exemplo do uso de opcionalidade para adiar a tomada de decisões finais (KOTTKE, 1981).

Depois de dois anos de desenvolvimento, a nova plataforma de sistema embarcado FutureSmart foi lançada. Como resultado, a HP *evoluiu* um conjunto de processos e ferramentas que substancialmente reduziu o custo de atividades que não agregavam valor ao processo de entrega enquanto significativamente aumentou

a produtividade. O time foi capaz de alcançar "liberações previsíveis, pontuais e regulares, assim novos produtos poderiam ser lançados na hora" (GRUVER, 2012, p. 89). O software embarcado deixou de ser o caminho crítico para novos lançamentos de produtos pela primeira vez em 20 anos. Isto, por sua vez, permitiu a eles criar confiança com o departamento de marketing de produto.

Como resultado do novo relacionamento entre marketing de produto e a divisão de software embarcado, a equipe do FutureSmart foi capaz de reduzir consideravelmente o tempo usado em planejamento. Em vez de "comprometer-se com uma lista de funcionalidades com 12 meses de antecedência que nunca poderiam ser entregues devido a todas as mudanças de planos durante o período" (GRUVER, 2012, p. 67), eles revisavam cada iniciativa planejada uma vez a cada 6 meses, e faziam uma estimativa durante 10 minutos do número de meses de engenharia que seria necessário para cada uma, sendo divididas por equipe. Uma análise mais detalhada seria feita uma vez que o trabalho fosse programado para uma interação ou pequeno marco. Um exemplo de produto de um destes exercícios é mostrado na figura a seguir.

Estimativas de alto nível em meses de trabalho de engenharia

Prioridade	Iniciativa	Componente 1 (25-30)	Componente 2 (20-25)	Componente 3 (30-40)	Componente 4 (30-40)	Componente 5 (20-30)	Componente 6 (20-30)	Componente 7 (20-30)	Componente 8 (15-25)	Componente 10 (40-50)	Componente 11 (20-30)	Componente 12 (20-30)	Outras equipes	<b>TOTAL</b>
1	Iniciativa A			21			5	3	1					<b>30</b>
2	Iniciativa B	3							4				17	<b>24</b>
3	Iniciativa C		5							2	1	1		<b>9</b>
4	Iniciativa D							10		2	2	2		<b>16</b>
5	Iniciativa E					20						3	5	<b>28</b>
6	Iniciativa F	23							5	6			2	<b>36</b>
7	Iniciativa G									2				<b>2</b>
8	Iniciativa H										5			<b>5</b>
9	Iniciativa I											3		<b>3</b>
10	Iniciativa J		20	27			17		39	17	21	9		<b>150</b>
11	Iniciativa K			3	30		3		3	14			12	<b>65</b>
12	Iniciativa L									2				<b>2</b>
13	Iniciativa M	3						10		6	6	6		<b>31</b>
		29	25	51	30	20	25	23	12	74	26	38	59	<b>401</b>

Figura 6.5: Estimativas aproximadas das próximas iniciativas (GRUVER; YOUNG; FULGHUM, 2013)

Isto é significativamente diferente de como o trabalho é planejado e estimado em grandes projetos que frequentemente criam planos funcionais detalhados e épicos de arquitetura que devem ser quebrados em peças menores e menores, analisadas em detalhe, estimadas, e colocadas em uma lista de itens priorizados *antes* de serem aceitas dentro do desenvolvimento.

Em última análise, o teste mais importante do processo de planejamento é se somos capazes de manter os compromissos que assumimos com os nossos stakeholders, incluindo os usuários finais. Como vimos, um processo de planejamento mais leve resultou no desenvolvimento do software embarcado sair do caminho crítico e,

ao mesmo tempo, reduzir os custos de desenvolvimento e demanda-falha. Já que esperávamos que a demanda-falha aumentasse uma vez que aumentamos o rendimento do processo, isso é duplamente impressionante.

Três anos após suas medidas iniciais, um segundo exercício de contabilidade-por-atividade ofereceu uma fotografia dos resultados que a equipe FutureSmart conseguiu com a sua abordagem, mostrada na *Tabela 6-2*.

**Tabela 6-2 — Atividades do time HP LaserJet Firmware em 2011**

% dos custos	Atividade	Anteriormente
2%	Integração contínua	10%
5%	Planejamento ágil	20%
15%	Um único branch	25%
10%	Supporte ao produto	25%
5%	Testes manuais	15%
23%	Criando e mantendo suítes de testes automáticos	0%
~40%	Inovação	~5%

No geral, a divisão de software embarcado da HP LaserJet mudou a economia do processo de entrega de software por meio da adopção de entrega contínua, abrangente automação de testes, uma abordagem iterativa e adaptativa para a gestão do programa, e um processo de planejamento mais ágil.

## BENEFÍCIOS ECONÔMICOS DA TRANSFORMAÇÃO AGILE DA HP FUTURESMART

- Os custos globais de desenvolvimento foram reduzidos em ~40%.
- Programas em desenvolvimento aumentaram em ~140%.
- Os custos de desenvolvimento por programa diminuiu 78%.
- Recursos gerando inovação aumentou oito vezes.

O ponto mais importante a lembrar deste estudo de caso é que as enormes reduções de custos e melhorias de produtividade só foram possíveis com base em um *investimento* grande e contínuo, feito pela equipe em automação de testes e integração contínua. Ainda hoje, muitas pessoas pensam que Lean é uma atividade liderada pela gerência e que se trata de simplesmente *cortar custos*. Na realidade, ele requer *investimento* para remover desperdício e reduzir demanda-falha — é uma atividade liderada pelos trabalhadores que, em última instância, pode continuamente reduzir os custos e melhorar a qualidade e produtividade.

## 6.4 GERENCIANDO A DEMANDA

Até agora, temos discutido a forma de melhorar o rendimento e a qualidade do processo de entrega. No entanto, é muito comum que este tipo de trabalho de melhoria fique preterido por outras demandas de negócios, tais como o desenvolvimento de novas funcionalidades. Isso é irônico dado que todo o propósito do trabalho de melhoria é aumentar a taxa em que nós podemos entregar tão bem quanto a qualidade do que é entregue. É muitas

vezes difícil fazer o resultado do trabalho de melhoria tangível — por isso é importante torná-lo visível através de contabilidade por atividade, incluindo medir o tempo do ciclo e o tempo gasto em *failure demand*, como retrabalho.

A solução é usar o mesmo mecanismo para gerir as novas demandas e a melhoria do trabalho. Um dos benefícios de usar a abordagem Melhoria Kata é que ele cria alinhamento com os resultados que deseja alcançar durante a próxima iteração em todo o programa. Na Melhoria Kata original, as condições-alvo estão preocupadas com a melhoria de processos, mas podemos usá-las para gerenciar a demanda também.

Há duas maneiras de fazer isso. Em organizações com uma cultura geradora (ver *capítulo 1*), podemos simplesmente especificar os objetivos de negócio desejados como as condições-alvo. Deixe que as equipes venham com ideias para funcionalidades, e realize experimentos para medir se elas terão o impacto desejado. Nós descrevemos como usar o mapeamento de impacto e desenvolvimento orientado a hipótese para conseguir isto no *capítulo 9*. No entanto, empresas mais tradicionais normalmente terão um acúmulo de trabalho priorizado em nível de programa dividido por suas linhas de negócios ou por donos de produto.

Podemos tomar algumas abordagens diferentes para integrar uma lista de atividades de nível de programa com a Melhoria Kata. Uma possibilidade é que as equipes que trabalham no âmbito do programa implantem o método Kanban, conforme descrito no *Capítulo 7*. Isso inclui a definição de limites para o trabalho em andamento (WIP) que são mantidos e geridos por estas equipes. Trabalho novo só será aceito quando o trabalho existente é concluído (em que "concluído" significa que é, pelo menos, integrado, totalmente testado com todos os testes automáticos concluídos, e pronto para ser liberado).

### **GERENCIANDO TRABALHO TRANSVERSAL**

A implementação de algumas funcionalidades dentro de um programa envolverá várias equipes trabalhando juntas. Para conseguir isso, a divisão HP FutureSmart criava uma equipe por funcionalidade que era pequena, temporária e "virtual", cuja função era coordenar o trabalho através das equipes relevantes.

O programa HP FutureSmart, que estava usando Scrum em algumas equipes, tomou a abordagem de especificar uma velocidade-alvo a nível do programa. O trabalho que acrescentava velocidade-alvo era aceito para cada iteração, aproximando-se de um limite de trabalho em andamento. A fim de implementar esta abordagem, todo o trabalho foi analisado e estimado em alto nível antes de ser aceito. A análise e a estimativa foram mantidas ao mínimo necessário para serem capazes de cumprir consistentemente as condições-alvo gerais a nível de programa, como mostrado na última figura.

### **NÃO USE A VELOCIDADE DO TIME FORA DOS TIMES**

É importante notar que estabelecer uma velocidade-alvo a nível do programa *não* exige que tentemos medir ou gerenciar a velocidade no nível da equipe, ou que as equipes devem usar Scrum. Velocidade em nível de programa estabelece a capacidade esperada de todas as equipes com base em estimativas de alto nível, como mostrado na última figura. Se uma equipe utilizando Scrum aceita um trabalho com base nestas especificações de funcionalidades de alto nível, ela então

pode criar histórias de baixo nível com as quais pode trabalhar.

A medida de velocidade em nível de time Scrum não é tão significativa fora do contexto de uma equipe em particular. Gerentes nunca devem tentar comparar velocidades de equipes diferentes ou tentar agregar estimativas entre equipes. Infelizmente, temos visto a velocidade de uma equipe sendo usado como uma medida para comparar a produtividade entre as equipes, uma tarefa para a qual não foi concebida, nem é adequada. Tal abordagem pode levar equipes a "trapacear" a métrica, e até mesmo parar de colaborar de maneira efetiva uns com os outros. Em qualquer caso, não importa quantas histórias nós concluímos se não alcançamos os resultados de negócios que nos propomos atingir sob a forma de condições-alvo em nível de programa.

Neste e no próximo capítulo, nós descrevemos uma maneira muito mais efetiva de medir progresso e gerenciar a produtividade — e uma que não requer que todas as equipes usem Scrum, ou "padronizar" estimativas ou velocidade. Usamos contabilidade por atividade e mapeamento de fluxo de valor (descrito no *capítulo 7*) para medir a produtividade, e nós usamos o mapeamento de fluxo de valor combinado com a Melhoria Kata para aumentá-la — fundamentalmente, ao nível do fluxo de valor em vez do nível de equipes individuais. Nós medimos e gerenciamos o progresso pela utilização de condições-alvo a nível programa, e se é preciso aumentar a visibilidade, podemos reduzir a duração de iterações.

## 6.5 CRIANDO A EMPRESA ÁGIL

Muitas organizações consideram tentar e adotar métodos ágeis

para melhorar a produtividade de suas equipes. No entanto, os métodos ágeis foram originalmente concebidos em torno de equipes pequenas, multifuncionais, e muitas organizações têm se esforçado para usar esses métodos em grande escala. Alguns frameworks para escalar ágil focam na criação de equipes pequenas e, em seguida, na adição de estruturas para coordenar o seu trabalho a nível do programa e portfólio.

Gary Gruver, diretor de engenharia da FutureSmart, contrasta esta abordagem de "tentar ativar as eficiências de pequenas equipes ágeis em uma grande empresa" com a abordagem da equipe FutureSmart de "tentar tornar ágil uma grande empresa usando os princípios ágeis básicos" (GRUVER, 2012). Na abordagem FutureSmart, enquanto os times andavam dentro de caminhos rigorosos em termos de práticas de engenharia (que serão discutidos em mais detalhes no *capítulo 8*), havia relativamente pouca atenção ao fato de que eles tinham, por exemplo, implementado Scrum no nível da equipe. Em vez disso, as equipes têm relativa autonomia para escolher e evoluir os seus próprios processos, desde que sejam capazes de satisfazer as condições-alvo em nível de programa para cada iteração.

Isso exigiu que a gestão de engenharia tivesse a liberdade para definir seus próprios objetivos em nível de programa. Ou seja, eles não tinham de obter a aprovação de orçamento para pagar pelo trabalho de melhoria de processos, como a automação de teste, ou construir o conjunto de ferramentas para integração contínua. Na verdade, o negócio não foi sequer consultado sobre este trabalho. Toda a demanda de negócios também foi gerida no nível do programa. Notavelmente, os pedidos de marketing de produto sempre passaram pelo processo em nível de programa, sem alimentar diretamente o trabalho para as equipes.

Outra consideração importante é a forma como as empresas

tratam métricas. Em uma cultura de controle, métricas e metas são muitas vezes criadas centralmente e nunca atualizadas em resposta às mudanças no comportamento que elas produzem. Organizações geradoras não gerenciam por métricas e metas. Em vez disso, a gestão do FutureSmart "usa as métricas para entender onde é preciso ter conversas sobre o que não está sendo finalizado" (GRUVER, 2012, p. 38). Esta é parte da estratégia da "gestão andando por aí" lançada pelo fundadores da HP Bill Hewlett e Dave Packard.

Talvez melhor caracterizado por "gestão andando por aí e fazendo perguntas". No Sistema Toyota de Produção, isto é chamado de caminhadas *gemba*.

Uma vez que descobrimos um problema, perguntamos o que podemos fazer para ajudar a equipe ou a pessoa que tem dificuldade. Nós descobrimos uma oportunidade para melhorar. Se as pessoas são punidas por não cumprir as metas ou métricas, um dos efeitos é que elas começam a manipular trabalho e informações para parecer que elas estejam cumprindo as metas. Como mostra a experiência da FutureSmart, ter boas métricas em tempo real é uma abordagem melhor do que depender de Scrums, ou Scrums de Scrums ou reuniões de acompanhamento do Escritório de Gerenciamento de Projetos, para descobrir o que está acontecendo.

## 6.6 CONCLUSÃO

A Melhoria Kata fornece uma maneira de alinhar as equipes e, de maneira geral, as organizações, tomando objetivos e dividindo-os em resultados (condições-alvo) pequenos e incrementais que nos fazem chegar mais perto do nosso objetivo. A Melhoria Kata não é

apenas uma metametodologia de melhoria contínua no nível de empresa e programa; é uma maneira de empurrar a responsabilidade para alcançar esses resultados para as bordas da organização, seguindo o Princípio da Missão. Como mostraremos no *capítulo 9*, ele também pode ser usado para executar grandes programas de trabalho.

As características chave da Melhoria Kata são a sua iteratividade e a capacidade de conduzir uma abordagem experimental para alcançar as condições-alvo desejadas, o que o torna adequado para trabalhar em condições de incerteza. A Melhoria Kata é também uma forma eficaz de desenvolver as capacidades das pessoas em toda a empresa para que elas possam se auto-organizar em resposta às novas condições.

O estudo de caso FutureSmart mostra como uma equipe grande e distribuída aplicou o metamétodo Melhoria Kata para aumentar a produtividade em oito vezes, melhorando a qualidade e reduzindo substancialmente os custos. Os processos e ferramentas que a equipe usou para alcançar essa transformação mudaram e evoluíram substancialmente ao longo do curso do projeto. Esta é uma característica de uma organização verdadeiramente ágil.

A implementação de um processo de melhoria contínua de nível empresarial é um pré-requisito para qualquer esforço de transformação em curso (como a adoção de uma abordagem ágil para entrega de software) em grande escala. A verdadeira melhoria contínua nunca termina porque, como a nossa organização e ambiente evoluem, nós descobrimos que o que funciona para nós hoje não será eficaz quando as condições mudem. Organizações de alto desempenho estão evoluindo constantemente para se adaptar ao seu ambiente, e fazem isso de uma maneira orgânica, não através de comando e controle.

## **Questões para o leitor**

- Você sabe quanto tempo o seu departamento de engenharia gasta em atividades sem nenhum valor agregado e na manutenção da demanda-falha *versus* servindo a demanda-valiosa? E quais são as principais fontes de desperdício?
- As equipes de engenharia devem obter permissão para investir em trabalho que reduz o desperdício e atividades sem agregar em toda a cadeia de valor, tais como construção, teste e automação de implementação e refatoração? Tais pedidos são negados por motivos como "não há orçamento" ou "não temos tempo"?
- Será que todos dentro da organização conhecem os resultados de curto e longo prazo que eles estão tentando alcançar? Quem decide estes resultados? Como eles estão estabelecidos, comunicados, revisados e atualizados?
- As equipes da sua organização regularmente refletem sobre os processos que elas usam e encontram maneiras de experimentar para melhorá-los? Que ciclos de feedback estão estabelecidos para descobrir quais ideias funcionam e quais ideias não? Quanto tempo leva para se obter este feedback?

## CAPÍTULO 7

# IDENTIFIQUE VALOR E AMPLIE O FLUXO

*“Não há nada tão inútil quanto fazer com eficiência aquilo que não deve ser feito de maneira nenhuma.” — Peter Drucker*

*“A medida de execução no desenvolvimento de produtos é a nossa capacidade de alinhar constantemente os nossos planos para o que seja, a cada momento, a melhor opção econômica.”*  
— Donald Reinertsen e Stefan Thomke

A maioria das empresas está afundando em um mar de excesso de trabalho, muito do qual fornece pouco valor para os clientes. Além de melhorar os produtos existentes e fornecer novos produtos, cada empresa tem várias iniciativas e projetos estratégicos em andamento a qualquer momento, e cada dia de trabalho não planejado chega para nos distrair de alcançar os nossos objetivos.

Uma resposta comum para este problema é tentar aumentar a utilização (trabalhar mais), melhorar a eficiência (trabalhar mais rápido), e reduzir os custos utilizando processos de gestão ultrapassados e contraproducentes. O pensamento Lean fornece uma alternativa comprovada que "pode ser resumido em cinco princípios: especificar com precisão o valor de cada produto

específico, identificar o fluxo de valor para cada produto, fazer o valor fluir sem interrupções, fazer com que o cliente puxe o valor do fornecedor, e buscar a perfeição" (WOMACK; JONES, 2010).

No capítulo anterior, mostramos como implementar uma estratégia de melhoria contínua em nível de programa para melhorar a produtividade e qualidade, e reduzir custos. Neste capítulo, mostraremos como os cinco princípios Lean foram adotados pela Maersk para reduzir o tempo de ciclo de novas funcionalidades em mais de 50%, aumentando simultaneamente a qualidade e retorno do investimento.

## 7.1 O ESTUDO DE CASO MAERSK

Em *Black Swan Farming using Cost of Delay* (<http://costofdelay.com>), Joshua J. Arnold e Özlem Yüce (2013) discutem como eles abordaram a redução de tempo de ciclo na Maersk Lines, maior companhia de logística marítima do mundo. O departamento de TI da Maersk possuía um orçamento anual de mais de US\$ 150 milhões, com grande parte do seu desenvolvimento realizada por fornecedores de *outsourcing* distribuídos globalmente. Eles enfrentavam uma grande demanda e lento *time-to-market*: em 2010, o tempo de atravessamento médio para uma funcionalidade era de 150 dias, sendo que 24% delas levando mais de um ano para serem entregues (desde a concepção até software em produção).

No momento da análise, em outubro de 2010, mais de 2/3 dos 4.674 requisitos identificados como estando em desenvolvimento estavam no "início difuso", à espera de serem analisados em detalhe e financiados. Em um caso, "uma funcionalidade que levou apenas 82 horas para ser desenvolvida e testada levou um total de 46 semanas para ser entregue de ponta a ponta. O tempo de atravessamento consumiu 38 semanas deste período",

principalmente no início difuso (figura a seguir).

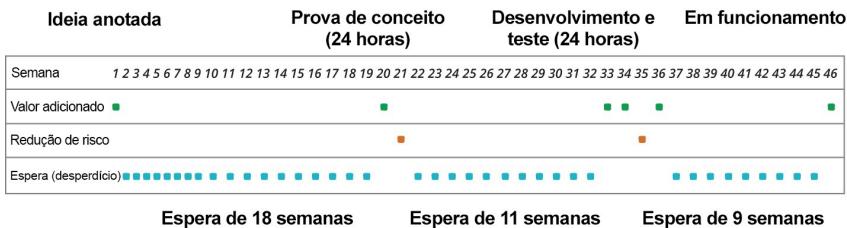


Figura 7.1: Mapa de fluxo de valor de uma única feature da Maersk — cortesia de Joshua J. Arnold and Özlem Yüce (2013)

Com base nos resultados desejados de "mais valor, fluxo mais rápido e melhor qualidade", Arnold e Yüce (2013) escolheram oito metas para todas as equipes:

1. Entrar em priorização mais rápido;
  2. Melhorar a priorização usando custo de atraso;
  3. Puxar requisitos da lista de prioridade dinâmica;
  4. Reduzir o tamanho dos requisitos;
  5. Chegar rapidamente ao desenvolvimento de código;
  6. Gerenciar ativamente o trabalho em andamento;
  7. Captar feedback mais rapidamente;
  8. Fluxo fluido e sustentável.

Anteriormente, funcionalidades sempre haviam sido agrupadas em projetos, resultando em muitas funcionalidades de menor valor sendo entregues juntamente com algumas poucas de alto valor. O método HiPPO (*Highest Paid Person's Opinion* — opinião da pessoa mais bem paga) foi usado para decidir quais funcionalidades eram de alto valor, e um grande esforço foi gasto tentando encontrar as

"ideias certas" e analisando-as em detalhe para criar projetos, obter a aprovação e justificar o investimento.

Arnold e Yüce implementaram um novo processo de gestão de requisitos. Eles criaram uma lista de requisitos — inicialmente no nível do projeto, e depois nos níveis de programa e de portfólio — chamada lista de prioridades dinâmicas. Quando novas funcionalidades eram propostas, eles passariam rapidamente por uma triagem, causando a repriorização da lista. Quando havia disponibilidade de capacidade de desenvolvimento, a funcionalidade de maior prioridade seria "puxada" da lista.

Funcionalidades eram priorizadas usando o método de Custo de Atraso (descrito em detalhes mais adiante neste capítulo), que estima o valor de uma funcionalidade em dólares, calculando quanto dinheiro perdemos por não ter a funcionalidade disponível quando precisamos dela. Usando essa abordagem, podemos determinar o impacto do tempo sobre o valor, e tomar decisões de priorização usando base econômica.

Por exemplo, o custo do atraso para a funcionalidade mostrada na primeira figura foi cerca de US\$ 210.000 por semana, o que significa que o atraso incorrido por haver espera em filas durante 38 semanas custar US\$ 8 mil. Investir um esforço extra para calcular o valor em dólares é essencial para revelar suposições, chegar a um entendimento comum, e afastar-se do modelo onde se confia na pessoa mais sênior da sala de fazer a priorização.

O número real usado para priorizar funcionalidades é conhecido como Custo de Atraso por Duração (ou CD3). É calculado pelo custo do atraso de uma funcionalidade dividido pela quantidade de tempo estimado para desenvolver e entregar essa funcionalidade. Isso leva em conta o fato de que temos pessoas e recursos limitados disponíveis para completar o trabalho, e que, se uma determinada funcionalidade leva um longo tempo para ser

desenvolvida, vai "empurrar para fora" as demais. Logicamente, se temos duas funcionalidades com o mesmo custo de atraso, mas uma vai demorar o dobro do tempo que a outra para se desenvolver, devemos desenvolver aquela de mais curta duração primeiramente. Um dos impactos de contabilizar a duração é que ela encoraja as pessoas a quebrar o trabalho em partes menores e mais valiosas, que por sua vez aumenta a pontuação CD3.

Implementar a lista de prioridade dinâmica e utilizar o CD3 para priorizar o trabalho ajudou a equipe a atingir vários outros objetivos em sua lista, como a priorização inicial mais rápida, reduzir o tamanho dos requisitos, escrever código mais rapidamente, e criar um fluxo mais suave. Em julho de 2011, o tempo de ciclo médio tinha sido reduzido em cerca de 50% nos dois serviços pilotados. Um dos serviços-piloto foi um sistema centralizado de contabilidade SAP.

Arnold e Yüce apresentam dois fatores que causaram a redução do tempo de ciclo: aumento da urgência gerado pelos exercícios de cálculo do Custo de Atraso, e a diminuição do tamanho do lote causada por pessoas quebrando o trabalho em pedaços menores para aumentar o CD3. Além disso, a satisfação dos clientes aumentou de forma significativa nos projetos-piloto.

Talvez o mais interessante tenha sido calcular o custo de atraso, pois esclareceu qual trabalho era mais importante. Nos dois sistemas analisados, a distribuição de custos de atraso seguiu uma curva exponencial. Os números de custo de atraso por semana para as funcionalidades do serviço-piloto, mostrados na figura a seguir, deixam muito claro quais três requisitos devem ser priorizados em detrimento de outros. Estes requisitos não foram identificados como sendo da mais alta prioridade antes do uso do custo de atraso.

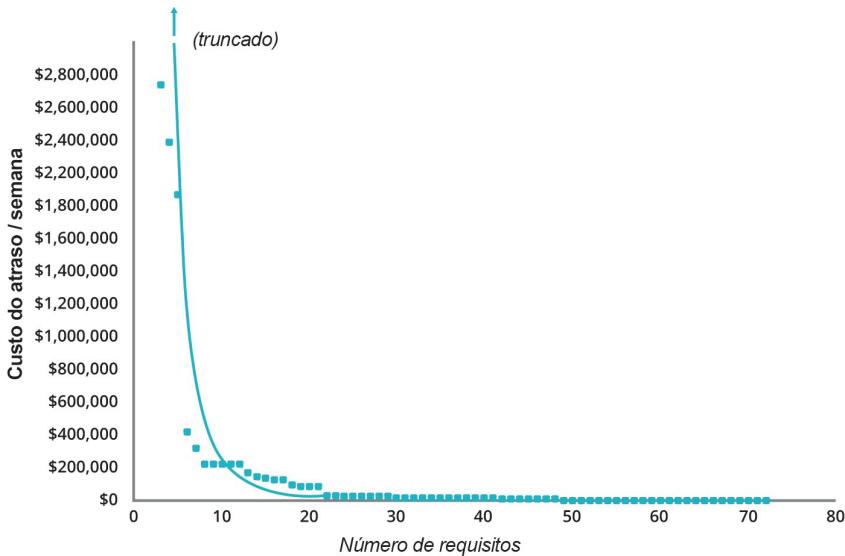


Figura 7.2: CD3 por feature — cortesia de Joshua J. Arnold and Özlem Yüce (2013)

O estudo de caso da Maersk demonstra a importância da utilização de uma abordagem baseada em fluxo para o desenvolvimento de produtos, em vez de grandes lotes de trabalho entregues em projetos. Também demonstra a importância de usar o custo de atraso, e não intuição ou HiPPO, para medir a prioridade relativa do trabalho a ser feito.

## 7.2 AMPLIAR O FLUXO

Como discutimos no *capítulo 6*, queremos melhorar o desempenho do processo de entrega antes de abordar a melhoria do alinhamento. No entanto, se queremos ver melhorias substanciais em termos de desempenho, temos de começar por escolher os lugares certos para concentrar os nossos esforços. É comum ver grandes organizações desperdiçando muito esforço para fazer alterações em processos ou comportamentos que são muito visíveis ou fáceis de mudar, mas não são motivadores principais para os

problemas em geral. Precisamos começar qualquer esforço de melhoria por meio da compreensão sobre onde surgem os problemas, e garantir que eles são entendidos em todos os níveis da organização. Só então teremos contexto certo para determinar o que fazer a seguir.

## Mapeie seu fluxo de valor no desenvolvimento de produto

A melhor maneira de entender onde os problemas começam é por meio de uma atividade chamada *mapeamento de fluxo de valor*. Cada organização tem muitos fluxos de valor, sendo definidos como o fluxo de trabalho a partir de um pedido do cliente até o cumprimento dessa solicitação. Cada fluxo de valor vai atravessar várias funções dentro de uma organização, como mostrado na figura a seguir.

Mapeamento de fluxo de valor foi descrita pela primeira vez por Rother e Shook (2009), e é o tema de um livro excelente de Karen Martin and Mike Osterling (2014).

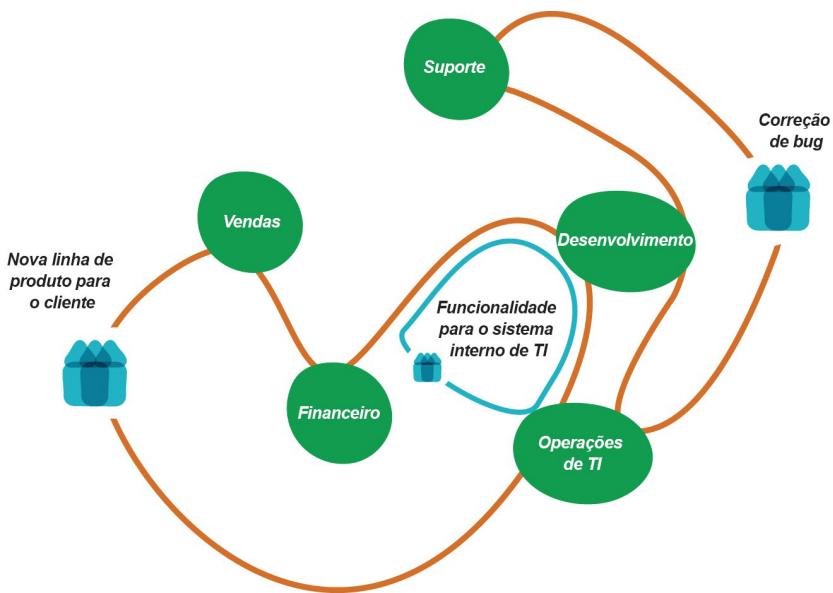


Figura 7.3: Fluxos de valor passando por departamentos

No contexto de explorar ideias validadas para o software, os fluxos de valor que nos interessam estão relacionados ao desenvolvimento de produtos, de tomar um pedido ou ideia de um cliente para uma funcionalidade ou correção de bug até entregá-lo aos usuários. Cada produto ou serviço terá o seu próprio fluxo de valor.

Para começar, selecione o produto ou serviço que deseja estudar e mapear o fluxo de valor existente para refletir a condição atual. Para evitar o erro comum de tentar melhorar por meio da otimização local, é essencial criar um fluxo de valor de estado futuro que representa como queremos que o fluxo de valor aconteça em algum momento futuro — normalmente um a três anos. Isto representa nossa condição-alvo. Os fluxos de valor atual e futuro podem então serem usados como base para a melhoria do trabalho, aplicando a Melhoria Kata no escopo de toda a cadeia de valor, conforme mostrado na figura:

## OS QUATRO PASSOS DO KATA DE MELHORIA



Figura 7.4: Mapeamento do fluxo de valor no contexto da Melhoria Kata

Para executar um exercício de mapeamento de fluxo de valor, devemos reunir pessoas de todas as partes da organização envolvidas no fluxo correspondente. No caso de design e entrega de produto, podemos incluir os produtos da unidade de negócios, marketing de produto, design, finanças, desenvolvimento, controle de qualidade e operações. Mais importante, a equipe de mapeamento de fluxo de valor deve incluir aqueles que são capazes de autorizar o tipo de mudança necessária para atingir o fluxo de valor de estado futuro. Muitas vezes, reunir todas as partes interessadas a se comprometer a gastar 1 a 3 dias juntos em uma única sala, ao mesmo tempo, é a parte mais difícil do processo. Busque a menor equipe possível que cumpra esses critérios — certamente não mais de 10 pessoas.

Realizar o mapeamento de fluxo de valor envolve a definição, em uma grande superfície (figura a seguir), dos vários blocos do processo de entrega do produto. Como você fatia e corta o fluxo de

valor em blocos de processo (também conhecido como *ciclos de fluxo de valor*) é um pouco de arte. Queremos detalhes suficientes para ser útil, mas não tanto que se torne desnecessariamente complexo e nos percamos discutindo minúcias. Martin e Osterling sugerem buscar entre 5 e 15 blocos de processo (MARTIN; OSTERLING, 2014, p. 63). Para cada bloco de processo dentro da cadeia de valor, registramos a atividade e o nome da equipe ou função que o executa.

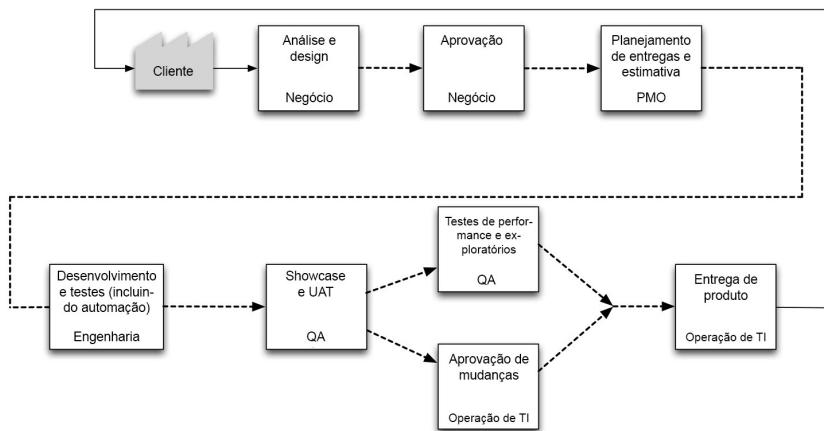


Figura 7.5: Esboço de um mapa de fluxo de valor mostrando blocos de processo

Uma vez que temos um diagrama de blocos, temos de reunir os dados necessários para compreender o trabalho dentro do fluxo de valor. Queremos saber o número de pessoas envolvidas em cada processo e quaisquer barreiras significativas ao fluxo. Observamos também a quantidade de trabalho dentro de cada bloco processo, bem como as filas entre os blocos. Finalmente, registramos três métricas-chave: tempo de atravessamento, tempo de processo e percentual completo e preciso, como mostrado na *Tabela 7-1*.

**Tabela 7-1 — Mapeamento de fluxo de valor**

Métrica	O que mede
Tempo de atravessamento (LT)	O tempo entre o momento em que um processo aceita um item de trabalho até momento em que o entrega para o processo seguinte
Tempo de processo (PT)	O tempo que seria necessário para completar um único item de trabalho, se a pessoa que executa dispusesse de todas as informações e os recursos necessários para completá-lo, podendo trabalhar de forma ininterrupta
Percentual completo e preciso (%C/A)	A proporção de vezes que um processo recebe um item de um processo anterior podendo utilizá-lo sem necessidade de retrabalho

Ao mapear um fluxo de valor, sempre registramos o estado dos processos como eles são no dia em que se realiza o exercício. É extremamente tentador registrar números que representam um estado ideal ou melhor em vez do estado comum, mas olhar para os números atuais ajuda a manter as pessoas alinhadas. Sempre que possível, a equipe deve realmente ir para os lugares onde o trabalho é feito e perguntar quais os números reais às pessoas que estão de fato trabalhando. Isso ajuda a equipe a experimentar os diferentes ambientes onde os trabalhos se realizam em toda a cadeia de valor.

O resultado de um exercício simples de mapeamento de fluxo de valor para um único requisito que passa por um fluxo de desenvolvimento do produto relativamente simples, mostrado na figura a seguir. Se for útil, poderíamos entrar em mais detalhes sobre cada uma das etapas do processo e descrever o que ocorre quando um processo rejeita uma entrada como incompleta ou incorreta. Isto é particularmente importante quando a relação entre tempo de atravessamento e o tempo de processo é grande, ou quando o processo seguinte tem um fraco %C/A.

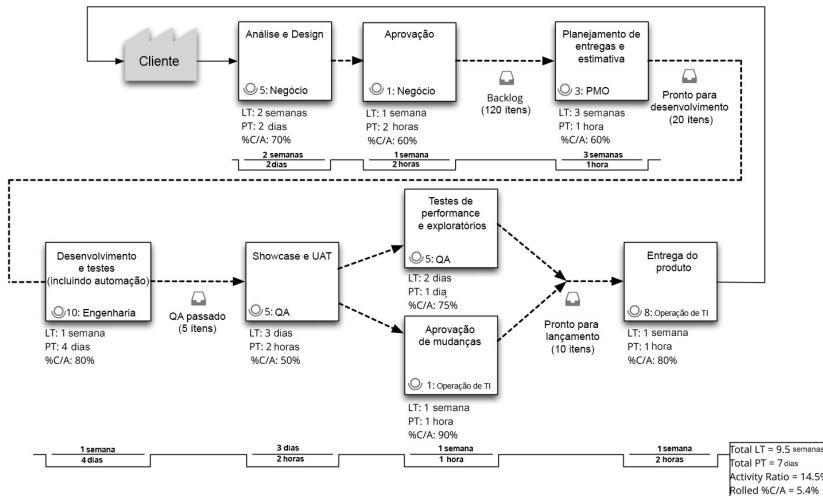


Figura 7.6: Exemplo de um mapa de fluxo de valor de uma feature

Executar este exercício pela primeira vez em uma organização é sempre esclarecedor. As pessoas geralmente se surpreendem — e muitas vezes se chocam — sobre como os processos em que eles não participam realmente funcionam e são impactados por seu trabalho.

Vimos discussões acontecerem! Em última análise, ao criar uma ideia melhor de como o trabalho se move através da organização, o mapeamento de fluxo de valor aumenta o alinhamento, a empatia e o entendimento compartilhado entre as partes interessadas.

Talvez a métrica mais valiosa quando fizer este exercício seja o %C/A. É muito comum descobrir que muito tempo é desperdiçado em demandas falhas, como retrabalho: desenvolvedores descobrem falhas no design, testadores pegam builds que não podem ser executados ou implementados, clientes pedem por mudanças quando veem features mostradas e defeitos críticos, ou problemas de performances são descobertos em produção ou reportados por usuários. Os facilitadores desses exercícios devem se armar com questões para descobrir e capturar o retrabalho, tais como:

- Em quais pontos descobrimos problemas no design?
- O que acontece nesse caso?
- Quem está envolvido nesse passo?
- Como funciona a transferência de tecnologia? Em que ponto descobrimos se a feature realmente entrega o valor esperado para os clientes?
- Onde os problemas de arquitetura (como performance e segurança) são descobertos?
- Qual é o efeito no tempo de atravessamento e qualidade?

Estas questões deveriam ser capturadas no mapa de fluxo de valor. Sua probabilidade deveria ser registrada no formulário de %C/A nos processos que as descobriram e (se possível) atribuídas à parte do fluxo de valor onde foram realmente causadas.

O total de desperdício em um fluxo de valor de uma empresa é normalmente bem decepcionante. Enquanto todo mundo tem uma compreensão intuitiva que os fluxos de valor são ineficazes, ver todo o fluxo de valor, da ideia ao resultado mensurável para o cliente, frequentemente revela uma quantidade impressionante de lixo. Esse lixo se manifesta no percentual de tempo que não agrega valor, em como o trabalho está frequentemente ocioso em filas e, crucialmente, nos números de %C/A que nos mostram onde falhamos em construir qualidade durante o processo de formação do fluxo.

Finalmente, o mapeamento de fluxo de valor revela a loucura das otimizações locais. Em quase todos os casos que vimos, fazer um bloco de processo mais eficaz terá um efeito mínimo no fluxo de valor como um todo. Já que o tempo de retrabalho e tempo de

atravessamento são alguns dos maiores contribuintes para o tempo total de entrega, adotar processos “ágiles” dentro de uma única função (como desenvolvimento) geralmente tem pouco impacto no fluxo de valor global e, consequentemente, nos resultados de nossos clientes.

Na maioria dos casos, precisamos repensar toda nossa abordagem de entrega de valor ao transformar todo o fluxo de valor, começando ao definir os resultados mensuráveis de cliente e organizacionais que desejamos alcançar por meio de nosso redesign. Para mitigar a disruptão desse tipo de mudança, normalmente limitamos nossos esforços para um único produto ou conjunto de capacidades em um produto — algo que mais beneficiaria os clientes e a organização como um todo.

Então, criamos um mapa de fluxo de valor para o futuro, que descreve como queremos que o fluxo de valor funcione no futuro. O objetivo dessa atividade de design é melhorar a performance. Martin e Osterling (2014, p. 101) definem a performance ótima como “entregar valor ao cliente de maneira tal que a organização não tenha despesas desnecessárias; o trabalho flui sem atrasos; a organização está totalmente em conformidade com todas as leis locais, estaduais e federais; a organização preenche todos os requisitos definidos pelo cliente; e os funcionários estão seguros e são tratados com respeito. Em outras palavras, o trabalho deveria ser projetado para eliminar atrasos, melhorar a qualidade e reduzir custos, esforços e frustrações desnecessários”.

Não há, claro, uma “resposta certa” para criar um mapa de fluxo de valor para o futuro, mas uma boa regra é focar em reduzir significantemente o tempo de travessamento e melhorar o %C/A (indicando que fizemos um trabalho melhor em construir qualidade). É importante que os participantes deste exercício sejam ousados e considerem uma mudança radical (*kaikaku*). Alcançar

um estado futuro certamente demandará que algumas pessoas aprendam novas habilidades e mudem o trabalho que estão fazendo, e alguns papéis (mas não as pessoas que os desempenham) se tornarão obsoletos. Por essa razão, como discutiremos no *capítulo 11*, é essencial fornecer suporte para o aprendizado de novas habilidades e comportamentos, e comunicar amplamente e frequentemente que ninguém será punido por executar um trabalho de melhoria — se não você provavelmente terá resistência.

Neste estágio, não tente adivinhar como o estado futuro será alcançado: foque nas condições-alvo para atingi-lo. Uma vez que os mapas do fluxo de valor atual e futuro estão prontos, podemos usar a Melhoria Kata para seguirmos para o estado futuro. No *capítulo 6*, descrevemos o uso da Melhoria Kata para guiar a melhoria contínua em nível de programa. As condições-alvo para o mapa de fluxo de valor para o futuro deveriam ser adicionadas aos ciclos da Melhoria Kata no nível de programa. Contudo, onde fluxos de valor vão além dos times envolvidos nos programas de trabalho — talvez dentro de operações de TI e unidades de negócios — precisamos também estabelecer ciclos da Melhoria Kata em nível de fluxo de valor, com pessoas que estabeleçam e acompanhem indicadores-chave de performance e monitorem o progresso.

### ATENÇÃO

As organizações que usam o paradigma estágio-portão (descrito na figura no começo da *Parte III*) acharão os princípios descritos nos capítulos seguintes muito difíceis de serem implementados sem alterar sua estrutura organizacional fundamentalmente. A Melhoria Kata descrita no *capítulo 6* pode (e deveria) ser implementado em todo lugar, já que faz quase nenhuma pressuposição sobre estrutura organizacional. Acabamos de falar sobre como mapear e melhorar o fluxo de

trabalho em sua organização, o que permitirá que você comece a mudar incrementalmente a forma de sua organização e os papéis do seu pessoal.

O *capítulo 8* discute práticas de engenharia que promovem uma entrega mais rápida, de qualidade mais alta com um custo menor. Se sua organização terceiriza a engenharia de software, seus parceiros terceirizados precisarão implementar essas práticas, e isso provavelmente demandará mudanças em seu relacionamento, incluindo potenciais mudanças contratuais. O *capítulo 9* descreve uma abordagem experimental de desenvolvimento de produto, em que designers, engenheiros, testadores, especialistas de infraestrutura e pessoal de produto trabalhem de maneira colaborativa em iterações bem curtas. Isso é extremamente difícil de fazer quando qualquer uma dessas funções é terceirizada; é bem mais fácil quando todos os times são internos, mas ainda se exige que todos trabalhem de maneira coordenada.

Todos podem e deveriam começar a jornada que descrevemos na *Parte III*. Esteja avisado: implementar o programa inteiro será disruptivo para a maioria das organizações e exigirá anos de investimento e experimentação. Não tente implementar todo o programa em sua organização rapidamente — use o mapeamento do fluxo de valor e divida os mapas de fluxo de valor para o futuro em blocos para fazer isso iterativamente e incrementalmente, fluxo de valor por fluxo de valor.

## **Limite o trabalho em processo**

Se nosso objetivo é aumentar o fluxo de trabalho de alto valor por meio de fluxo de valor de desenvolvimento de produto, o mapeamento do fluxo de valor representa um primeiro passo

essencial. Contudo, devemos dar passos adiante para gerenciar o fluxo de trabalho por meio do sistema, para diminuir os tempos de travessamento e aumentar a previsibilidade.

No contexto de desenvolvimento de produto, o Método Kanban fornece princípios e práticas para apoiar esse objetivo, como descrito em *Kanban: Successful Evolutionary Change for your Technology Business*, de David J. Anderson (2010). Primeiro, devemos visualizar o fluxo de trabalho por meio do fluxo de valor: pegamos o mapa de fluxo de valor atual e o traduzimos para um quadro físico ou virtual com colunas representando blocos e filas entre os blocos. Então fazemos um cartão para cada pedaço de trabalho que está passando pelo fluxo de valor no momento, como mostrado na figura a seguir. Esses cartões são movidos pelo quadro conforme o trabalho progride pelo fluxo de valor.

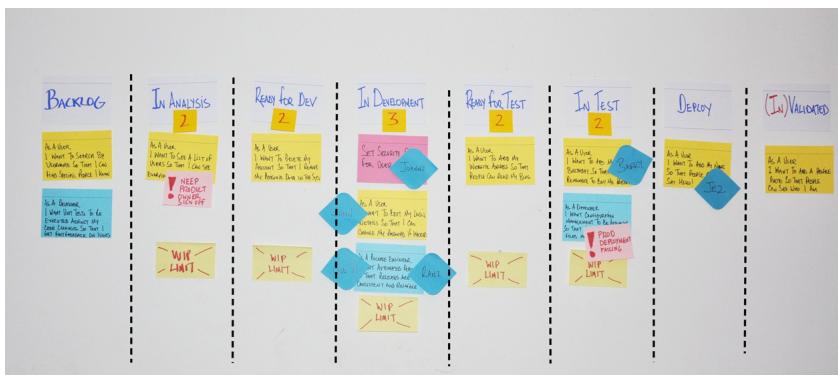


Figura 7.7: Exemplo de um quadro Kanban

Podemos visualizar a dinâmica do fluxo de valor ao criar um diagrama de fluxo cumulativo que mostra a quantidade de trabalho em cada fila e bloco de processo com o passar do tempo. Um exemplo do diagrama de fluxo cumulativo é mostrado na figura adiante. Ela mostra claramente a relação entre o trabalho em processo (WIP) e o tempo de travessamento: conforme reduzimos

o WIP, o tempo de atravessamento cai.

Essas duas quantidades estão, na verdade, casualmente relacionadas; na área matemática chamada de Teoria das Filas, isso é conhecido como Lei de Little.

No estudo de caso Maersk, discutimos duas maneiras de reduzir o tamanho dos pacotes de trabalho que se movem pelo fluxo de valor de desenvolvimento de produto: reduzir o tamanho dos requisitos e transformar projetos em requisitos que podem ser priorizados independentemente. Limitar o WIP é outra maneira poderosa de reduzir o tamanho do trabalho. Já que reduzir o tamanho dos pacotes de trabalho é o fator mais importante em sistematicamente aumentar o fluxo e reduzir a inconstância, e tem efeitos secundários importantes como melhoria da qualidade e aumento de confiança entre os stakeholders, devemos buscar essas práticas incansavelmente e medir nosso progresso.

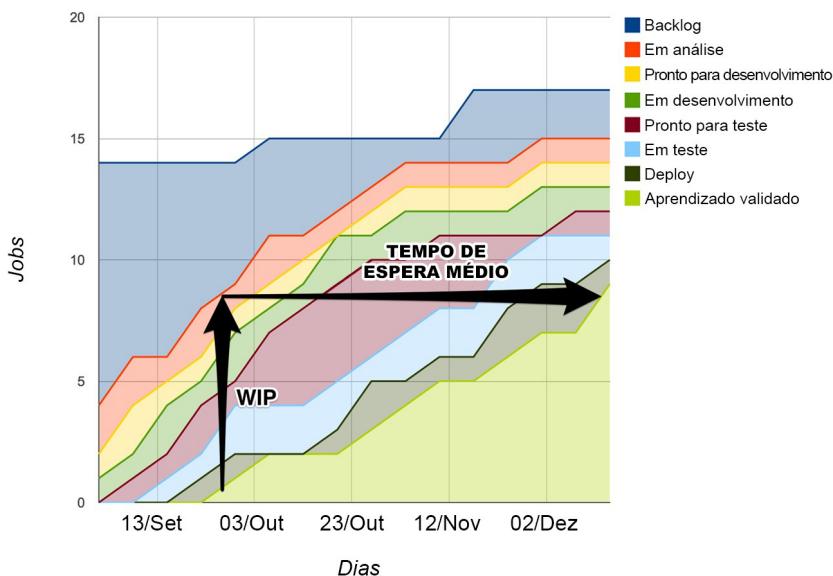


Figura 7.8: Um diagrama de fluxo cumulativo

O Método Kanban oferece uma maneira abrangente de gerenciar o fluxo de trabalho por meio do fluxo de valor de desenvolvimento de produto, usando as seguintes práticas:

- *Visualizar o fluxo de trabalho* criando um quadro que mostre o atual trabalho em processo dentro do fluxo de valor em tempo real.
- *Limitar o trabalho em processo* ao definir limites de WIP para cada bloco e fila de processo dentro de um fluxo de valor, e atualizá-los para trocar tempo de travessamento por utilização (quão ocupadas as pessoas estão).
- *Definir classes de serviço* para diferentes tipos de trabalho e os processos por meio dos quais serão

gerenciados, para garantir que trabalho urgente ou que dependa de tempo seja priorizado apropriadamente.

- *Criar um sistema de demanda*, concordando em como o trabalho será aceito em cada bloco de processo quando a capacidade se tornar disponível — talvez definindo uma reunião regularmente na qual os stakeholders decidem qual trabalho deveria ser priorizado baseado na capacidade disponível.
- *Fazer “reviews operacionais” regulares* para os stakeholders de cada bloco de processo analisarem sua performance e atualizarem os limites de WIP, classes de serviço e método pelo qual o trabalho é aceito.

#### LIMITES DE WIP DEVEM SER DOLOROSOS

Parte do propósito dos limites de WIP é revelar oportunidades para melhoria. Impor limites de WIP focará nossa atenção no trabalho que está bloqueado ou difícil de ser terminado, já que nossa incapacidade para terminá-lo nos impede de começar um trabalho novo. A essa altura, é tentador relaxar os limites de WIP para garantir que “algo está sendo feito”. É essencial evitar esse tipo de tentação e, em vez disso, identificar as fontes do problema.

O Método Kanban segue quatro princípios de melhoria contínua projetados para minimizar a resistência à mudança:

- Comece com o que você faz agora;
- Concorde em buscar mudança incremental, evolucionária;

- Inicialmente, respeite os atuais papéis, responsabilidades e cargos;
- Encoraje atos de liderança em todos os níveis.

O Método Kanban é uma ferramenta poderosa para melhorar a performance e aumentar a qualidade e confiança dentro do time em um ambiente onde não há concordância para melhoria contínua no nível gerencial sênior. Em tal situação, recomendamos fortemente que os times implementem o Método Kanban como primeiro passo para melhorar. Uma vez que você revelou que houve uma melhoria mensurável, você ainda precisa buscar, em nível de empresa, a melhoria contínua, já que, em um contexto típico de empresa, a adoção do Kanban em nível de time provavelmente levará apenas a melhorias incrementais.

## **GERENCIANDO O TRABALHO EM PROCESSO EM NÍVEL DE EMPRESA**

O objetivo principal de limitar o WIP é terminar o trabalho com um nível suficientemente alto de qualidade, assim como aumentar a taxa de transferência. Reduzir tempos de atravessamento dessa maneira requer que haja folga (*slack*) no sistema para gerenciar o WIP efetivamente. A folga é também essencial para fornecer tempo para o trabalho de melhoria de processo. Já que as teorias de gestão do taylorismo enfatizam a maximização da utilização dos empregados, isso requer uma mudança significativa no pensamento para muitas organizações.

Em empresas, um indicador de muito WIP é o número de pessoas designadas para mais de um projeto. Essa prática perniciosa inevitavelmente leva a tempos de atravessamento mais longos e baixa qualidade devido à mudança constante de contexto. Em vez de designar pessoas para múltiplos projetos, tenha um time centralizado que possa fornecer suporte especializado extra para os times sob demanda, mas não designe essas pessoas para quaisquer times e monitore cuidadosamente seu uso para mantê-lo bem abaixo de 100%.

Pawel Brodzinski (2014) escreveu um bom artigo sobre os limites de WIP para gerenciamento de portfólio.

## **7.3 CUSTO DE ATRASO: UMA ESTRUTURA DE TRABALHO PARA DESCENTRALIZAR DECISÕES ECONÔMICAS**

Um dos maiores problemas em desenvolvimento de produto é

entregar features valiosas tardivamente, de maneira que elas não mais forneçam vantagem competitiva. Como mostra o estudo de caso Maersk, um grande contribuinte para esse problema é empilhar features em projetos e entregar os resultados para os clientes depois de meses de espera. O mapeamento do fluxo de valor frequentemente revela — como fez em Maersk — que muito do tempo gasto esperando foi na fase de análise, com features paradas no backlog de produto esperando para serem analisadas, estimadas, aprovadas e priorizadas para o desenvolvimento.

Como vimos no *capítulo 3*, a informação fornecida por essas atividades é de valor bem baixo. Estruturas como o Scrum recomendam priorizar o backlog por valor de negócio, mas oferecem pouco direcionamento de como calculá-lo. Priorizar por valor de negócio também é falho em deixar a urgência do trabalho explícita. Contudo, uma estrutura de trabalho poderosa para tomar decisões racionais de priorização baseadas em economia existem na forma de Custo de Atraso (REINERTSEN, 2009). O time de Maersk reduziu tempos de ciclo para features de alto valor por meio de uma combinação de separar as features dos projetos e usar o Custo de Atraso como um método leve para identificar e priorizar o trabalho com o custo de oportunidade mais alto.

Para usar o Custo de Atraso, começamos decidindo sobre a métrica que estamos tentando otimizar por meio do nosso fluxo de valor. Para organizações comprometidas com desenvolvimento de produto, é normalmente o lucro do ciclo de vida. Entretanto, uma companhia de logística pode usar uma métrica como custo por tonelada por milha (quanto custa para mover uma tonelada por uma milha). Quando tivermos uma decisão, olhamos para todas as partes do trabalho afetadas por essa decisão e calculamos como maximizar nossa Única Métrica que Importa (ver *capítulo 4*), dadas as várias opções que temos. Para isso, temos de resolver, para cada pedaço de trabalho, o que acontece com nossa métrica-chave

quando atrasamos esse trabalho (daí o “custo de atraso”).

Vamos começar com um simples e ingênuo exemplo para delinear os mecanismos. Digamos que temos dois pedaços de trabalho que poderíamos começar assim que chegarmos, na segunda-feira de manhã. Temos certeza (absoluta) de que ambos são de alta prioridade. O que devemos fazer?

Começamos calculando quanto custará se não fizermos o trabalho. A Tarefa A é atualizar uma peça central de software para uma nova versão que suporta dados de cartão de crédito encriptados para atender um prazo de compliance, que é daqui duas semanas. Seremos multados em \$50.000 por dia de trabalho que não estivermos em compliance. O custo de não fazer esse trabalho é zero até o ponto em que a penalidade entra em ação e o custo de atraso para o trabalho é de \$250.000 por semana após o prazo. A Tarefa A levará duas semanas.

A Tarefa B é completar uma feature-chave requisitada por clientes em potencial, que já anunciamos que estará pronta daqui uma semana. Esperamos fechar \$100.000 de negócios por semana quando lançarmos essa nova feature. Além disso, um de nossos concorrentes está logo atrás de nós, e acreditamos que eles lançarão uma nova versão de seu software com essa feature daqui um mês. A Tarefa B leva uma semana para ser feita.

A matemática é simples e nossas opções estão mostradas na figura a seguir. Se fizermos a Tarefa A, então atrasaremos a Tarefa B em duas semanas, o que nos custará \$200.000. Se fizermos a Tarefa B primeiro, atrasaremos a Tarefa A em uma semana, o que nos custará \$250.000. Dessa forma, devemos executar a Tarefa A primeiro.

**Tarefa A:** 2 semanas, CdA \$250k/semana

**Tarefa B:** 1 semana, CdA \$100k/semana



Figura 7.9: Como é que vamos priorizar as tarefas A e B com Custo de Atraso?

Também podemos calcular o que acontece se tentarmos fazer as duas tarefas simultaneamente. Supondo que designemos metade de nossa capacidade para cada um, levará duas semanas para completar a Tarefa B e três semanas para completar a Tarefa A. Isso leva a um custo de atraso total de \$350.000, mostrando que devemos executar a Tarefa A antes da Tarefa B.

#### USE CD3 PARA ENCORAJAR BLOCOS DE TRABALHO MENORES

Aplicando o método CD3 como descrevemos no estudo de caso de Maersk, o CD3 da Tarefa A é 125.000 enquanto que o CD3 da Tarefa B é 100.000, o que nos diz que a Tarefa A tem prioridade mais alta. Suponha que temos uma alternativa à Tarefa B: a Tarefa C. A Tarefa C nos fornecerá o mesmo valor de 80% dos clientes que queriam a feature entregue pela Tarefa B (\$80.000 por semana), mas estimamos que completar a C levará metade do tempo da B (meia semana). O CD3 da Tarefa C será de 160.000, deixando-a como mais prioritária que a Tarefa A. Usar o CD3 consistentemente tem um importante efeito colateral — nos encoraja a dividir o trabalho em pedaços menores e mais valiosos.

Há muitas consequências ao se usar o Custo de Atraso. Ao calculá-lo para cada feature, não dependemos mais apenas de um *product owner* para estimar o valor de negócio para nossas histórias no backlog, o que é uma maneira ruim de priorizar, já que essa pessoa deve constantemente recalcular para levar em consideração a urgência de tempo do valor de negócio. Em vez disso, dado que temos capacidade limitada, pensamos em priorização como escolher o que atrasar.

Quando a capacidade de desenvolvimento se torna disponível, o

time simplesmente pega o item com o maior custo de atraso naquele momento. Isso é uma vantagem-chave de usar o Custo de Atraso: seguindo o Princípio de Missão, ele permite que todo mundo na organização tome decisões econômicas transparentes racionais sem a necessidade de mecanismos de comando-e-controle, tais como reviews onerosas, aprovações e priorização pela pessoa mais sênior na sala. Separamos várias questões que podem ser abordadas independentemente no nível correto da organização:

1. *Qual é a métrica econômica que estamos tentando otimizar?* (Lembre-se da Única Métrica que Importa, do *capítulo 4*). Comunicar essa métrica é responsabilidade da liderança.
2. *Qual o impacto nessa métrica de se atrasar cada pedaço de trabalho?* Calcular esse custo de atraso é a meta-chave de análise.
3. *Como devemos programar e priorizar o trabalho?* Isso pode ser determinado de maneira autônoma, dando aos times as informações das questões 1 e 2.

Além disso, o Custo de Atraso nos fornece um argumento econômico para limitar o trabalho em processo. Como vimos no exemplo anterior, o Custo de Atraso de tentar fazer A e B simultaneamente foi maior do que completar cada tarefa sequencialmente.

Claro que o exemplo é bem simples. Primeiramente, supomos que o Custo de Atraso permaneceria constante com o tempo. Esse é raramente o caso na vida real. Por exemplo, o Custo de Atraso para a Tarefa A aumenta para \$50.000 por dia depois do último dia em que podemos começar o trabalho de atualização para completá-lo a tempo de cumprir o prazo externo. Mas, para a Tarefa B, a quantidade de negócios que podemos fechar é provavelmente dependente do tempo, dado que nossa concorrência terá uma

feature similar em breve.

A dependência do tempo do custo de atraso é capturada em um perfil de urgência. Perfis de urgência que podem ser usados na vida real para a Tarefa A e a Tarefa B são mostrados na figura adiante. O custo de atraso, no eixo y, representa a quantidade de dinheiro que custa atrasar trabalho por unidade de tempo. Para calcular o custo de atraso total, medimos a área sombreada embaixo do gráfico. Apesar de, na teoria, existirem muitos perfis de urgência possíveis, em geral quase todas as tarefas em uma dada organização se encaixaram em um punhado de perfis-padrão. Estes podem ser moldados dentro de um sistema Kanban, usando-se classes de serviço.

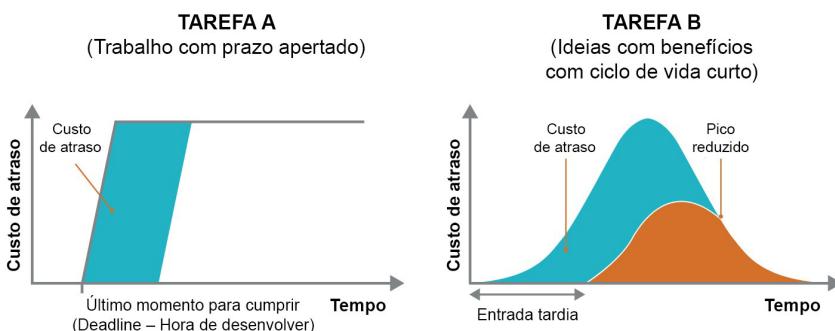


Figura 7.10: Perfis de urgência para Tarefa A e Tarefa B

O segundo problema que enfrentamos é que, em muitos casos, é extremamente difícil ter uma cifra precisa para o custo de atraso. Para se chegar a um número, normalmente fazemos várias suposições e incluímos múltiplos fatores. Por exemplo, não completar a Tarefa A a tempo pode levar a perda de confiança do consumidor na nossa capacidade de mandar os dados em segurança, o que pode impactar nas vendas futuras. É importante deixar essas suposições explícitas, visíveis e registradas junto com o trabalho sendo discutido, para que possamos validá-las.

A coisa mais importante para se ter em mente é que estamos focando em exatidão, e não em precisão, em nossas estimativas. Se a incerteza em torno do custo de atraso é muito alta, isso significar que estamos no domínio da “investigação” e devemos testar nossas suposições, talvez usando uma das técnicas como modelar uma métrica-chave com simulações Monte Carlo ou testar hipótese de negócio com MVPs, vistas na *Parte II*.

Um dos maiores benefícios do Custo de Atraso é que, em vez de discutir sobre respostas sobre as questões que colocamos, podemos argumentar sobre as suposições que fazemos em nossos modelos e focar em validá-las. O Custo de Atraso torna possível uma mudança cultural importante — de brigas políticas sobre qual trabalho é mais importante a expor e validar nossas suposições e seus efeitos nas variáveis econômicas. Isso requer um certo nível de maturidade organizacional. Como em todas as mudanças de processo, recomendamos começar com um produto que as pessoas realmente querem testar usando o Custo de Atraso, e fornecendo o suporte necessário para fazer experiências com ele.

Aplicar o Custo de Atraso na empresa nos permite descentralizar a tomada de decisão e deixar as decisões sob controle econômico. Para fazer isso, temos de mudar a maneira que pensamos a gestão do trabalho, particularmente o papel do grupo responsável por priorizar as decisões em nível de portfólio e programa (também conhecido como equipe de gestão de projetos — *Project Management Office* ou PMO). No modelo do Custo de Atraso, a responsabilidade principal desse grupo muda de tomar as decisões para criar e atualizar a estrutura para se tomar as decisões: trabalhar com projetos individuais e financeiros para criar perfis de urgência padrão e templates, reunindo e analisando dados, implantando o custo de atraso na organização e criando loops de feedback para continuamente melhorar a qualidade do processo de tomada de decisão.

As decisões reais podem então ser tomadas pelos times envolvidos no trabalho, baseadas nas informações sobre o Custo de Atraso atualizadas quando há um novo trabalho, ou quando as suposições baseadas em cálculos existentes mudam. Essa é uma grande mudança que necessitará de apoio da diretoria, um piloto e melhorias baseadas em aprendizado antes de ficar pronta para ser adotada pela organização como um todo.

Por fim, há coisas que não você não deve fazer com o Custo de Atraso. Evite simplesmente adicionar o Custo de Atraso em métodos de priorização existentes. Seu propósito é melhorar a qualidade da demanda, permitindo que identifiquemos e evitemos trabalho de baixo valor ao passo que diminuímos o tempo de atravessamento para trabalhos de alto valor. Se tornarmos o Custo de Atraso em um processo “peso-pesado” que fica junto com processos existentes, não atingiremos essa meta. O Custo de Atraso fornece a maior recompensa quando as filas são grandes — em outras palavras, quando há um longo tempo de atravessamento. É essencialmente uma contramedida que se torna valiosa quando há muito trabalho no sistema. Se você não tem grandes filas e muito trabalho, usar o Custo de Atraso provavelmente não lhe fornecerá muito valor global.

No contexto de produtos e serviços de software, implementar práticas de engenharia enxutas é essencial para reduzir tempos de atravessamento globais e reunir feedback rápido do cliente o mais cedo possível no ciclo de vida do produto. A solução definitiva para o déficit de taxa de transferência — e, dessa maneira, de produtividade — é reduzir os tempos de atravessamento globais reduzindo os pacotes de trabalho, gerenciando o trabalho em processo e reduzindo o custo de entregar valor aos clientes. Em termos de mensuração de valor, não há um substituto para enviar e receber feedback de clientes reais.

## 7.4 CONCLUSÃO

Em organizações de alta performance, liderança e gestão têm um foco nítido no valor que a organização está criando para seus clientes. Trabalhar cientificamente em direção a metas desafiadoras, o que leva a identificar e remover ou evitar atividades que não agregam valor é a essência do pensamento Lean, e isso requer uma mudança de mentalidade significativa para a maioria das organizações.

O mapeamento do fluxo de valor é uma ferramenta poderosa para visualizarmos nosso trabalho, para que saibamos nossas condições atuais e possamos criar um entendimento comum de onde estamos e para onde queremos ir. Um exercício eficaz de mapeamento do fluxo de valor pode ser realmente revelador ao permitir que times vejam, pela primeira vez, o fluxo de trabalho pela organização em resposta à demanda do consumidor, e sua verdadeira contribuição dentro desse fluxo. O resultado de um exercício de mapeamento do fluxo de valor é então usado para definir condições-alvo para a Melhoria Kata, descrito no *capítulo 6*. Nós então podemos usar o Método Kanban para gerenciar o fluxo de trabalho através do fluxo de valor, definir e atualizar os limites do WIP e classes de serviço, e criar um sistema de demanda para incrementar o fluxo.

Melhorar o fluxo de trabalho pela organização é um lado da moeda. O outro é se certificar de que estamos trabalhando as coisas certas. O Custo de Atraso fornece uma maneira de medir o valor do tempo, permitindo que os times tomem decisões transparentes de priorização. Ao quantificar o valor do trabalho que estamos fazendo, podemos evitar fazer trabalho de baixo valor. Se limitamos o trabalho em processo pelo fluxo de valor e apenas trabalhamos em tarefas de alto valor, podemos rapidamente reduzir o tempo para comercializar o trabalho, o que é tem maior valor para nossos

clientes.

## Questões para os leitores

- Como o trabalho é priorizado em seu time ou organização? Você usa um modelo econômico ou os HiPPOs decidem?
- Como você tentaria implementar um modelo econômico em seu backlog atual? Por que não ver os itens dentro da sua fila atual de iteração e estimar o Custo de Atraso para cada uma?
- Qual o seu atual tempo de atravessamento para alterações? Como você poderia reduzir o tamanho dos pacotes para reduzir o tempo de atravessamento?
- Como você descobre o impacto econômico de seu trabalho depois de entregue? O time foca apenas em métricas de entrada e saída, como velocidade? Como você poderia trabalhar com os departamentos financeiro, de gerenciamento de produto e outros para entender os resultados de negócios e impactos de receita de seu trabalho?
- Você agrupa features em projetos como parte de seus processos de planejamento e financiamento? Como você pode desembaraçar projetos e ir para um modelo em que você, incrementalmente, financia e entrega apenas o trabalho de alto valor (falaremos sobre gerenciamento financeiro no *capítulo 13*)? Como você poderia coordenar o trabalho por meio de fluxos de valor?

## CAPÍTULO 8

# ADOTE PRÁTICAS DE ENGENHARIA LEAN

*“Não dependa da inspeção em massa para alcançar qualidade. Melhore o processo e desenvolva a qualidade do produto em primeiro lugar.” — Edwards Deming*

Uma capacidade de inovação eficaz depende da capacidade de frequentemente testar ideias com usuários reais. Crucialmente, a taxa pela qual podemos aprender, atualizar nosso produto ou protótipo baseado em feedback é uma poderosa vantagem competitiva. Esta é a proposta de valor nas práticas de engenharia Lean que descreveremos neste capítulo. Andy Hertzfeld (<http://bit.ly/1v70zdR>), um dos engenheiros que trabalharam no Apple Macintosh original, observa que “em vez de discutir sobre novas ideias de software, precisamos na verdade testá-las, escrevendo rápidos protótipos, mantendo as ideias que mais funcionaram e descartando as outras. Sempre temos algo acontecendo que representou nossa melhor ideia à época”.

Em muitas organizações, implementar um software em um ambiente de produção integrada é um processo que ainda leva dias ou mesmo semanas. Mas as organizações que tratam o software como uma vantagem competitiva, em vez de um mal necessário,

investem substancialmente em reduzir esse tempo de espera. Para se ter noção do que é possível em grande escala, em maio de 2011, a Amazon conseguiu um tempo médio entre deployments para sistemas de produção de 11,6 segundos, com 1079 desses deployments em uma única hora, agregados em milhares de serviços que integram a plataforma da Amazon. Alguns desses deployments afetaram mais de 10.000 hosts — de acordo com a palestra de Jon Jenkins, na Velocity 2011, *Velocity Culture (the unmet challenge in Ops)*. A Amazon, claro, está sujeita a normas, como a Sarbanes-Oxley e PCI-DSS.

A principal razão de a Amazon ter investido nessa capacidade é tornar extremamente fácil e com baixo risco para os times projetarem e executarem experimentos. Em muitos casos, executar um experimento não requer um processo de solicitação de mudança burocrático. Isso dá aos times de entrega multifuncionais da Amazon a habilidade de testar ideias ousadas — com segurança de que, se algo der errado, o experimento pode ser encerrado com uma pequena porcentagem de usuários impactados por um pequeno período de tempo.

Apesar do nome, a entrega contínua não se trata de implantar em produção muitas vezes ao dia. O objetivo da entrega contínua é deixar seguro e econômico o trabalho em pequenas doses. Isso, por sua vez, leva a tempos de espera menores, mais qualidade e custos menores. É por essas razões que o time HP Futuresmart rearquitetou seu *firmware* do zero para minimizar o tempo de espera entre o check-in do código e ter o software validado, pronto para o release.

Definitivamente, a entrega contínua resulta em deployments seguros, no estilo “aperte o botão” em vez de provações longas e dolorosas que devem ser feitas depois do horário comercial.

Este capítulo é direcionado para leitores que desejam entender

os princípios e práticas por trás da entrega contínua. Para aqueles que buscam apenas um panorama, apresentamos um sumário executivo de práticas de engenharia Lean na próxima seção.

## 8.1 OS FUNDAMENTOS DA ENTREGA CONTÍNUA

Entrega contínua é a habilidade de colocar mudanças — experimentos, features, alterações de configuração, correção de bugs — em produção ou nas mãos de usuários em segurança e rapidamente, de maneira sustentável. Vamos examinar cada um desses requisitos.

- **Segurança**

Para garantir que os deployments sejam seguros, construímos um *deployment pipeline*, que sujeita cada mudança proposta a uma bateria de testes automatizados de diferentes tipos, seguidos por validações manuais, como teste exploratório e de usabilidade. Em seguida, habilitamos os deployments semiautomáticos (em que um humano tem de decidir fazer o push) de builds validados para os ambientes *downstream*, como, por exemplo, ambientes de teste e de produção, liberação para manufatura, ou para uma loja de aplicativos (dependendo do tipo de software). O objetivo principal do deployment pipeline é detectar e rejeitar mudanças que são arriscadas, contenham regressões ou nos deixem aquém do desempenho aceitável. Como um subproduto da implementação de um deployment pipeline, temos uma trilha de auditoria de onde cada mudança foi introduzida, quais testes estão sendo executados, por quais ambientes passou,

quem implementou, e assim por diante. Essa informação é inestimável como evidência de compliance.

- **Rapidez**

Devemos constantemente monitorar e reduzir o tempo de espera para apresentar as mudanças aos usuários. Mary e Tom Poppendieck (2006, p. 59) questionam: “Quanto tempo leva para sua organização implementar uma mudança que envolve apenas uma única linha de código?”. Reduzimos o tempo de espera ao trabalharmos para simplificar e automatizar o processo de desenvolvimento, teste e release. Devemos ser capazes de criar ambientes de teste sob demanda, e executar testes automatizados de forma muito rápida. Se necessário, os testes devem executar em paralelo, em várias máquinas, para garantir que seus resultados sejam gerados muito rápido. Usando este processo, é possível conseguir um alto nível de confiança de que nosso software está bom para release. Normalmente, isso envolve arquitetura (ou rearquitetura) para garantir a capacidade adequada de testes, com seus ambientes e paralelismo. Um importante efeito colateral deste trabalho é que o time de produto alcança um feedback muito rápido sobre a qualidade de seu trabalho, e problemas são encontrados logo após serem introduzidos em vez de serem encontrados somente mais tarde, nas fases de integração e testes, quando sua correção é muito mais cara.

- **Sustentabilidade**

O objetivo disso tudo é tornar economicamente viável trabalhar em pequenos lotes. Grandes lotes de trabalho

são entregues com pouca frequência. As releases de grandes lotes são dolorosas e mais caras, logo, a solução é reduzir sua frequência. O mantra da entrega contínua é: “Se machuca, faça com mais frequência, e tente eliminar a dor”. Se integração, teste e deployment são dolorosos, devemos tentar fazê-los toda vez que alguém commitar algo no controle de versão. Isso revela o desperdício e ineficiência em nosso processo de entrega para que possamos lidar com ele por meio de melhoria contínua. Contudo, para tornar econômico trabalhar em pequenos lotes, precisamos investir em testes extensivos e automação de implementação e em uma arquitetura que suporte tudo isso.

Há duas regras de ouro da entrega contínua que devem ser seguidas por todos:

1. O time não pode dizer que já “terminou” qualquer trabalho até que seu código esteja no *trunk* (tronco) no controle de versão e pronto para release (para serviços hospedados, o nível é ainda mais alto — “terminar” significa implementar para produção). Em *The Lean Startup*, Eric Ries (2011) argumenta que, para novas features que não são simples requisitos de usuários, o time deve também executar experimentos com usuários reais para determinar se a feature consegue o resultado desejado.
2. O time deve ter como prioridade manter o sistema funcionando antes de fazer um novo trabalho. Isso significa que, a qualquer momento, podemos pegar o sistema atual no controle de versão e gerar um entregável por meio de um processo automatizado. Se isso não for possível, precisamos parar o trabalho e corrigir este problema. Este é o conceito de *jidoka*, no Sistema de Produção Toyota, aplicado à entrega de

software.

Deveríamos enfatizar que seguir esses passos consistentemente será difícil e requer disciplina — mesmo para times pequenos e experientes.

#### **APLICANDO SUA DEFINIÇÃO DE “PRONTO”**

Os gerentes do HP FutureSmart tinham uma regra simples para aplicar essas regras de ouro. Quando alguém queria demonstrar uma nova feature (que era o requisito para declará-la “pronta”), precisava perguntar se o código havia sido integrado ao trunk e se a nova funcionalidade seria demonstrada em um ambiente similar ao de produção, com a execução de testes automatizados. A demonstração só pode proceder se a resposta foi “sim” para ambas perguntas.

No *capítulo 6*, discutimos os enormes aumentos em qualidade e produtividade e reduções no custo que o time HP FutureSmart foi capaz de conseguir. Essas melhorias foram possíveis quando o time colocou os princípios da entrega contínua no coração da sua estratégia de entrega. O time eliminou as fases de integração e testes do processo de desenvolvimento de seu software ao incorporar integração e testes em seu trabalho diário. Foi também possível alterar as prioridades rapidamente em resposta às necessidades mutantes do marketing do produto e dos usuários:

*“Sabemos de nossa qualidade dentro de 24 horas de qualquer conserto no sistema (...) e podemos testar amplamente mesmo para pequenos ajustes de última hora, para garantir que o conserto de um bug não cause falhas inesperadas. Ou podemos arcar com o acréscimo de novas features bem depois de declararmos ‘funcionalidade*

*completa" - ou, em casos extremos, mesmo depois de declararmos um candidato a release".* (GRUVER, 2012, p.60).

Vamos olhar para os padrões de engenharia que possibilitaram ao time HP FutureSmart conseguir um aumento de produtividade oito vezes maior.

## 8.2 INTEGRAÇÃO CONTÍNUA E AUTOMAÇÃO DE TESTE

Em muitos times de desenvolvimento, é comum que desenvolvedores trabalhem em suas ramificações do controle de versão. Em times pequenos, experientes, em uma mesma localização, isso pode funcionar. Contudo, o resultado inevitável de escalar esse processo é o “inferno da integração”, em que times e pessoas passam dias ou semanas integrando e estabilizando suas ramificações para liberar o código. A solução é todos os desenvolvedores trabalharem em um único *trunk* e integrarem seu trabalho a ele pelo menos uma vez por dia. Para conseguir fazer isso, os desenvolvedores precisam aprender a quebrar grandes pedaços de trabalho e transformá-los em passos pequenos, incrementais, que mantenham o *trunk* em bom estado e passível de entrega.

Validamos que o *trunk* está em bom estado fazendo o build da aplicação ou serviço toda vez que ocorre uma mudança no controle de versão. Também executamos testes de unidade na última versão do código e damos ao time um feedback minutos depois, caso o build ou teste tenha falhado. O time deve então corrigir o problema ou, se o problema não puder ser corrigido em poucos minutos, reverter a mudança. Dessa forma, asseguramos que nosso software está sempre em bom estado durante o processo de desenvolvimento.

A integração contínua é a prática de se trabalhar em pequenos lotes e usando testes automatizados para detectar e rejeitar

alterações que apresentem uma regressão. É, em nossa opinião, a prática técnica mais importante das práticas ágeis, e forma a base da entrega contínua, para a qual, além disso, exigimos que cada alteração mantenha o código pronto para release no trunk. Contudo, isso pode ser difícil de ser adotado para times que não estão acostumados.

Em nossa experiência, as pessoas tendem a cair em dois grupos: aqueles que não podem entender como é possível (particularmente, em escala) e aqueles que não acreditam que pessoas possam trabalhar de outra maneira. Asseguramos a você que é possível, tanto em larga escala quanto em pequena, seja qual for o seu domínio.

Vamos falar sobre o problema de escala com dois exemplos. Primeiro, o estudo de caso do HP FutureSmart demonstra a integração contínua sendo efetiva com um time de 400 pessoas trabalhando em um sistema embarcado. Segundo, notaremos que quase os mais de 10 mil desenvolvedores do Google, distribuídos em mais de 40 escritórios, trabalham em um único trunks. Todo mundo trabalhando neste trunk desenvolve e faz o seu release, e todos os builds são criados na fonte. De 20 a 60 alterações no código são submetidas a cada minuto, e 50% do código-base muda todo mês (LIMONCELLI, 2014). Os engenheiros do Google construíram um poderoso sistema de integração contínua que, em 2012, estava executando mais de 4 mil builds e 10 milhões suítes de teste (aproximadamente 60 milhões de testes) todo dia (PENIX, 2013).

A integração contínua não é apenas possível em times grandes distribuídos, mas é o único processo conhecido para se escalar efetivamente sem as fases dolorosas e imprevisíveis de integração, estabilização e de “amadurecimento” associadas com outras abordagens, tais como trens de release ou *branches* (ramos) de feature. A entrega contínua é projetada para eliminar essas

atividades.

### PRINCÍPIOS DA AUTOMAÇÃO DE TESTE

Como pode ser visto nos exemplos do Google e do HP FutureSmart, a integração contínua baseia-se em automação de testes completa. A automação de teste é ainda controversa em algumas organizações, mas é impossível atingir tempos de espera curtos e releases de alta qualidade sem ela. A automação de teste é um tópico importante e complexo, sobre o qual muitos livros já foram escritos — recomendamos Freeman e Price (2009) e Crispin e Gregory (2009) —, mas aqui estão alguns dos pontos mais importantes:

- A automação de teste não é sobre reduzir o número de testadores — mesmo que ela mude o papel e as habilidades necessárias para ser um testador. Testadores devem estar focados em teste exploratório e trabalhar com os desenvolvedores para criar e melhorar suítes de testes automatizados, e não devem somente trabalhar com teste manual de regressão.
- É impossível evoluir suítes de teste automatizado de alta qualidade a não ser que os testadores colaborem com os desenvolvedores pessoalmente (independentemente do time ou estrutura de gerência). Criar suítes sustentáveis de testes automatizados requer um forte conhecimento de desenvolvimento de software. Também é necessário que o software seja projetado tendo a automação de teste em mente, o que é impossível quando os desenvolvedores não estão envolvidos

nos testes.

- A automação de testes pode se transformar em um pesadelo de manutenção se as suítes de testes automatizados não são efetivamente mantidas. Um pequeno número de testes que rodam rapidamente e detectam bugs com confiança é melhor do que um número grande de testes que não são confiáveis e com os quais os desenvolvedores não se importam.
- A automação de testes deve ser projetada tendo em mente a paralelização. Executar testes em paralelo possibilita aos desenvolvedores obterem feedback rápido e previne más práticas, como dependências entre os testes.
- Testes automatizados complementam outros tipos de testes, como testes exploratórios, de usabilidade e de segurança. O objetivo do teste automatizado é validar a funcionalidade central e detectar regressões para que não desperdicemos tempo tentando testar manualmente (ou implementar) versões do software que contenham sérios problemas.
- Testes automatizados confiáveis requerem uma configuração completa e gerenciamento de infraestrutura. Deve ser possível criar um ambiente de teste virtual similar ao de produção sob demanda, seja dentro do ambiente de integração contínua ou na estação de trabalho do desenvolvedor.
- Apenas gaste tempo e esforço com automação de

teste para produtos ou features uma vez que eles foram validados. Automação de testes para experimentos é perda de tempo.

A principal objeção à integração contínua vem de desenvolvedores e seus gerentes. Dividir qualquer nova feature ou esforço de rearquitetura em pequenos passos é mais difícil do que completá-los isoladamente em uma branch, e leva mais tempo se você não está acostumado à disciplina de trabalhar em pequenos lotes. Isso significa que pode levar mais tempo, no começo, para terminar as tarefas de desenvolvimento. Isso pode, por sua vez, diminuir a velocidade do desenvolvimento e criar a impressão de que a eficiência do time diminuiu — aumentando a pressão arterial dos gerentes de desenvolvimento.

Contudo, não deveríamos ser otimistas com a taxa com que declararmos as coisas “prontas” isoladamente em uma branch. Deveríamos ser otimistas com o tempo de espera total — o tempo que levamos para entregar um software valioso para usuários. Ser otimista com o tempo de “dev complete” é precisamente o que causa o “inferno da integração”. Um final de processo doloroso e imprevisível de integração e teste, por sua vez, perpetua os longos ciclos de release, que são um grande fato em atrasos de projeto, software de baixa qualidade, custos totais mais altos e usuários insatisfeitos.

### VOCÊ REALMENTE ESTÁ FAZENDO INTEGRAÇÃO CONTÍNUA?

A integração contínua (em inglês, *continuous integration*, ou CI) é difícil e, em nossa experiência, a maioria dos times que dizem que a estão praticando, na verdade, não estão. Alcançar a CI não é simplesmente um caso de instalar e executar uma

ferramenta de CI; é uma maneira de pensar. Um dos nossos artigos favoritos sobre CI discute como fazê-la sem qualquer ferramenta de CI — usando apenas uma estação de trabalho, uma galinha de borracha e um sino (SHORE, 2006). Claro, você precisará mais do que isso em um time grande de desenvolvimento, mas, proporcionalmente, os princípios são os mesmos.

Para descobrir se você realmente está fazendo CI, faça as seguintes perguntas para seu time:

- Todos os desenvolvedores do time estão commitando no trunk (não apenas fazendo a fusão dela com suas branches ou cópias de trabalho) pelo menos uma vez por dia? Em outras palavras, eles estão desenvolvendo baseados no trunk e trabalhando em pequenos lotes?
- Toda mudança no trunk inicia um processo de build, incluindo a execução de um conjunto de testes automatizados para detectar regressões?
- Quando o processo de build e teste falha, o time corrige o build dentro de poucos minutos, seja consertando a falha ou revertendo a alteração que causou a falha nele?

Se a resposta para a maioria dessas perguntas for “não”, você não está praticando a integração contínua. Em particular, reverter alterações ruins é uma técnica insuficientemente praticada. No Google, por exemplo, qualquer um tem poder para reverter uma alteração ruim no controle de versão, mesmo se ela foi feita por alguém em um time diferente: eles priorizam deixar o sistema em bom estado em vez de gerar novas features.

Claro que, se você está no meio do trabalho em uma aplicação grande e usando várias branches, não é fácil movê-la para integração contínua. Nessa situação, o objetivo deveria ser empurrar os times em direção ao trabalho no trunk, começando com as branches mais voláteis. Em uma organização grande, levou um ano para sairmos de 100 ramificações para cerca de 10-15.

### 8.3 O PIPELINE DE ENTREGA

Lembre-se da segunda regra de ouro da entrega contínua: devemos dar prioridade a manter o sistema trabalhando em vez de começar um novo trabalho. A integração contínua é um importante passo em direção a esse objetivo — mas, normalmente, não nos sentiríamos confortáveis revelando para os usuários um software que passou apenas por testes unitários.

O pipeline de entrega deve avaliar cada alteração feita no sistema, detectar e rejeitar alterações que carreguem riscos altos ou impactem negativamente na qualidade, e deve fornecer ao time um feedback a tempo sobre suas alterações para que eles façam uma triagem dos problemas rapidamente e com custo baixo. O pipeline leva cada check-in para o controle de versão, cria pacotes para a versão que são implementáveis em qualquer ambiente e desempenha uma série de testes para aquela versão, para detectar defeitos conhecidos e para verificar que a feature funciona.

Se o pacote passa por esses testes, devemos nos sentir confiantes para implementar o build particular do software. Se qualquer estágio do pipeline falhar, aquela versão do software não pode ir para a frente, e os engenheiros devem imediatamente fazer uma triagem para encontrar a fonte do problema e corrigi-lo.

Mesmo o pipeline mais simples, como o mostrado na figura a seguir (um pipeline mais complexo é mostrado na figura mais adiante), possibilita que os membros do time desempenhem deployments semiautomáticos de builds que passaram de CI para testes exploratórios com ambientes similares de produção, ou ambientes de teste de aceitação do usuário. Deveria ser possível fornecer ambientes de teste e implementar qualquer build de CI para eles usando um processo totalmente automatizado. Este mesmo processo deveria ser usado para promoção para produção.

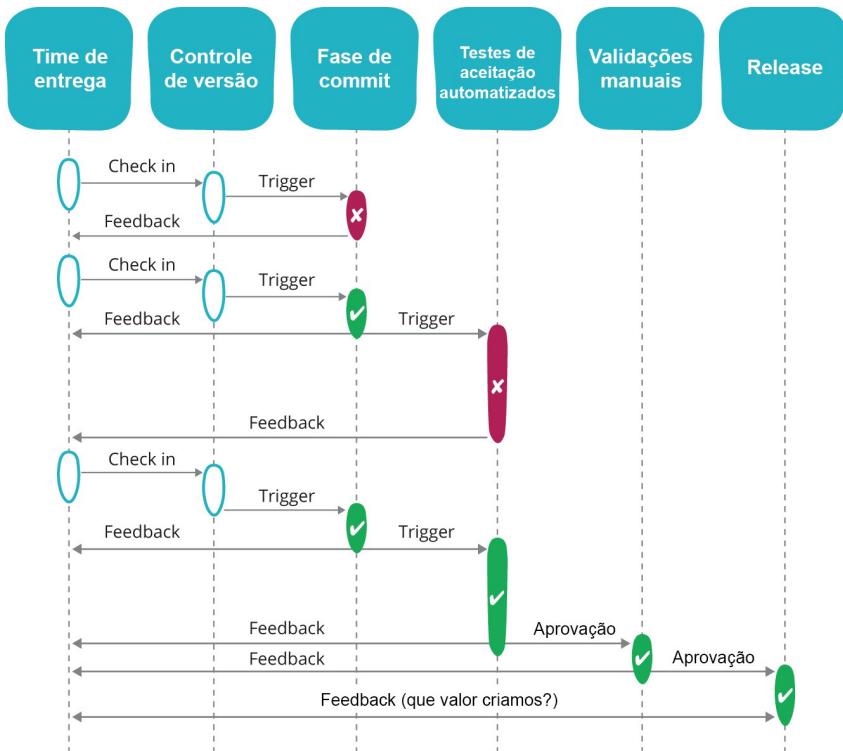


Figura 8.1: Alterações se movendo por um simples pipeline de entrega

O pipeline conecta todos os passos necessários para ir do check-in para a produção (ou distribuição em uma loja de aplicativo). Ele também conecta todas as pessoas envolvidas com a entrega do

software — desenvolvedores, testadores, engenheiros de release e operações —, o que o torna uma importante ferramenta de comunicação.

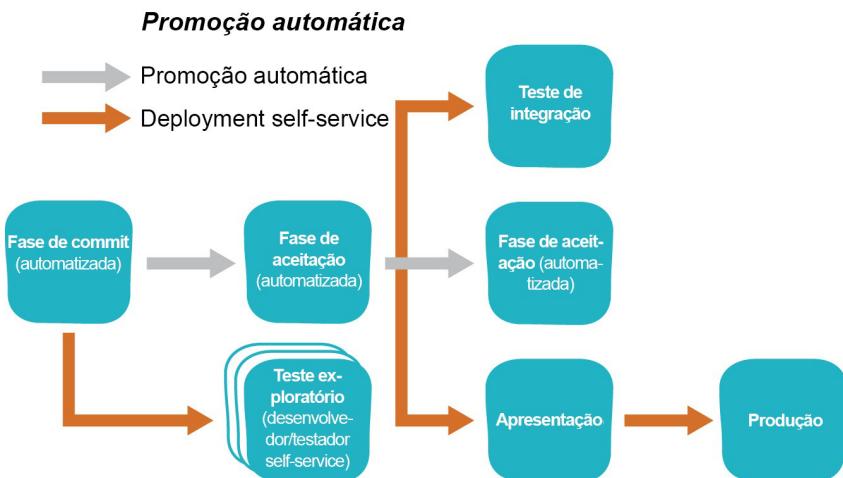


Figura 8.2: Um pipeline mais complexo

### O PIPELINE DE ENTREGA DO FUTURESMART

O pipeline do time FutureSmart permite que um time distribuído de 400 pessoas integre de 100-150 alterações — cerca de 75-100 mil linhas de código — no trunk em sua base de código de 10 milhões de linhas todo dia. A cada dia, o pipeline produz de 10-14 bons builds do firmware do Nível 1. Todas as alterações — incluindo desenvolvimento de features e alterações em larga escala — são feitas no trunk. Os desenvolvedores agregam seu código ao trunk várias vezes por semana.

Todas as mudanças em qualquer sistema — ou nos sistemas em

que roda — deveriam ser feitas por meio de controle de versão e então promovidas via pipeline. Isso inclui não apenas a fonte e o teste do código, mas também migrações e implementação do banco de dados e scripts de provisão, assim como alterações nas configurações de servidor, rede e infraestrutura.

O pipeline, dessa forma, se torna o registro de quais testes estão rodando contra um dado build e quais foram os resultados, quais builds foram implementados em quais ambientes e quando, quem aprovou a promoção de um build em particular e quando, qual a exata configuração de cada ambiente — de fato, todo o ciclo de vida do código e das alterações na infraestrutura conforme eles se movem por diferentes ambientes.

Isso, por sua vez, significa que a implementação de um pipeline tem outros muitos usos importantes, além de rejeitar alterações de alto risco ou problemáticas no sistema:

- Você pode reunir informações importantes no seu processo de entrega, como estatísticas do ciclo de tempo das alterações (a média, o desvio-padrão), e descobrir os gargalos em seu processo.
- Ela fornece uma riqueza de informações para propósitos de auditoria e compliance. Auditores amam o pipeline de entrega, porque ele permite que eles rastreiem cada detalhe de quais comandos exatamente estavam rodando em quais boxes, quais foram os resultados, quem os aprovou e quando, e assim por diante.
- Ele pode formar a base de um processo de gerenciamento de mudança leve, mas abrangente. Por exemplo, a telefônica australiana altamente regulada National Broadband usou um pipeline de entrega para

enviar tickets de gerenciamento de alterações automaticamente quando alterações fossem feitas na infraestrutura de produção, e para atualizar seu banco de dados de gerenciamento de configuração automaticamente quando abastecia novos sistemas e fazia promoção para produção (CUNNINGHAM; MYERS, 2012).

- Ele possibilita que os membros do time façam deployments do build de sua escolha para o ambiente de sua escolha. As ferramentas para implementar pipelines normalmente permitem que essas homologações sejam feitas por ambiente e para que fluxos de trabalho em torno da promoção de build sejam cumpridos.

#### **ENTREGA CONTÍNUA E CONTROLE DE ALTERAÇÃO**

Muitas empresas têm tradicionalmente usado conselhos consultivos de alterações ou sistemas de controle de alterações similares, como forma de reduzir o risco de alterações nos ambientes de produção. Contudo, o 2014 State of Devops Report (FORSGREN; KIM; KERSTEN; HUMBLE, 2014), que pesquisou mais de 9 mil indivíduos em muitas indústrias, descobriu que processos de homologação externos aos times de desenvolvimento fazem muito pouco para melhorar a estabilidade dos serviços (mensurados em termos de tempo para restaurar o serviço e porcentagem de alterações que falharam), ao mesmo tempo em que atua como um significante entrave na taxa de transferência (mensurada em termos de tempo de espera para alterações e frequência de alterações).

A pesquisa comparou processos externos de homologação de

alterações com mecanismos de revisão em par, como programação em par ou o uso de solicitações de recebimento. Uma análise estatística revelou que, quando times de engenharia se responsabilizavam pela qualidade de seu código por meio da revisão em par, os tempos de espera e a frequência de release melhoravam consideravelmente, com um impacto insignificante na estabilidade do sistema. Dados adicionais do relatório, que apoia o uso de técnicas discutidas neste capítulo, são apresentados no *capítulo 14*.

Os dados sugerem que é hora de reconsiderar o valor fornecido por processos pesos-pesados de controle de alteração. A revisão em par das alterações do código combinada com uma deployment pipeline fornece uma substituição poderosa, segura, auditável e de alta performance para homologação externa de alterações. O estudo de caso da National Broadband Network (citado anteriormente) mostra um método para implementar um processo leve de controle de alteração que é compatível com estruturas como ITIL em um ambiente regulado. Para saber mais sobre compliance e gerenciamento de risco, veja o *capítulo 12*.

Implementar a entrega contínua requer pensar cuidadosamente sobre arquitetura de sistemas e processos, e fazer uma certa quantidade de planejamento anteriormente. Quaisquer atividades manuais que são repetidas deveriam ser consideradas potenciais desperdícios e, dessa forma, candidatas para simplificação e automação. Isso inclui:

- **Build** — Deveria ser possível criar pacotes da fonte, implementável em qualquer ambiente, em um único passo usando um script que é armazenado no controle

de versão e pode ser rodado por qualquer desenvolvedor.

- **Provisionamento** — Qualquer um deveria ser capaz de fazer um ambiente de teste a sua escolha (incluindo configuração de rede, configuração de host, quaisquer aplicativos e software) de maneira totalmente automatizada. Este processo deveria também usar informações e scripts que são mantidos no controle de versão. Alterações na configuração do ambiente deveriam sempre ser feitas por meio do controle de versão, e de maneira barata e sem dor para eliminar boxes existentes e reprovisão da fonte.
- **Deploy** — Qualquer um deveria ser capaz de implementar pacotes de aplicativos em qualquer ambiente que tenha acesso, usando um processo totalmente automatizado que use script mantido no controle de versão.
- **Teste** — Deveria ser possível para qualquer desenvolvedor rodar a suíte de teste completamente automatizada em sua máquina, assim como qualquer conjunto de testes selecionados. As suítes de teste deveriam ser abrangentes e rápidas, e conter testes unitários e de aceitação de nível.

Requisitamos, como um princípio para a automação, um excelente gerenciamento de configuração. Em particular, tudo requisitado para reproduzir nosso sistema de produção e o build, teste e implementação de nossos serviços devem ser no controle de versão. Isso significa não apenas o código-fonte, mas o build, teste e scripts de deployment, infraestrutura e configuração de ambiente, esquemas de base de dados e scripts de migração, assim como a

documentação.

## 8.4 DISSOCIAZ DEPLOYMENT E RELEASE

O princípio mais importante para fazer releases de baixo risco é este: dissociar o deployment e o release. Para entender este princípio, primeiro devemos definir esses termos.

Deployment é a instalação de uma dada versão de um pedaço de software em um determinado ambiente. A decisão de fazer um deployment — incluindo para produção — deve ser meramente técnica. Release é o processo de fazer uma feature, ou um conjunto de features, disponível para clientes. O release deve ser uma decisão puramente de negócios.

Frequentemente, esses dois termos são tratados como sinônimos — ou seja, usamos deployment como nosso mecanismo primário para fazer releases. Isso tem uma consequência muito negativa: ligar uma decisão técnica de fazer deploy à decisão de negócios de fazer o release. Essa é uma importante razão pela qual políticas organizacionais são injetadas no processo de deploy, em detrimento de todos.

Existe um número de técnicas para implementar um software em um ambiente de produção com segurança sem deixar sua funcionalidade disponível para os usuários — para que possamos validar se nosso sistema se comporta corretamente. A mais simples — e uma das mais poderosas — é o deployment azul-verde (algumas vezes chamado de deployment preto-vermelho). Esse padrão requer dois ambientes de produção separados, com os nomes de azul e verde. A qualquer momento, apenas um deles ficará ativo; na figura adiante, é o verde.

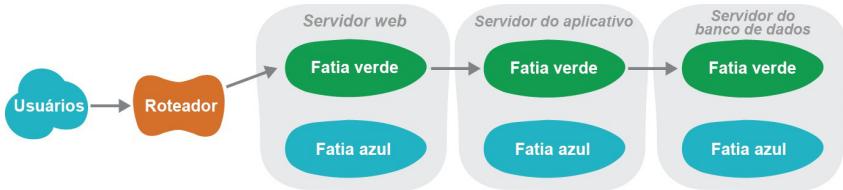


Figura 8.3: Deployment azul-verde, cortesia de Martin Fowler (2009a)

Quando queremos fazer o release de uma nova versão de nosso serviço, implementamos os pacotes com novas features para o ambiente que não está atualmente ativo (azul, neste exemplo) e os testamos à vontade. O processo de release então simplesmente altera o roteador para apontar para o ambiente azul; para reverter, direcionamos o roteador de volta para o ambiente verde. Uma variação mais sofisticada gradualmente constrói o tráfego para o ambiente azul com o tempo.

Crucialmente, para companhias com processos de deployment árduos que não podem fazer o release em horários de pico, os deployments azul-verde permitem que esses processos possam ser feitos em segurança durante o horário comercial, dias antes de um release planejado, se necessário. O processo muito mais simples (e reversão, se necessário) pode ser então feito em horários fora do pico, de maneira remota, por um grupo de pessoas bem menor.

Algumas organizações usam seus *datacenters* principais e de back-up para seus ambientes azul e verde, verificando dessa forma que podem fazer um processo de recuperação de desastre toda vez que fizerem o deployment. Contudo, os ambientes azul e verde não têm de estar fisicamente separados. Eles podem ser ambientes virtuais ou lógicos rodando na mesma infraestrutura física (já que o ambiente não vivo normalmente consome poucos recursos).

O deployment e o release podem também ser dissociados em nível de feature ou componente, em vez de em nível de sistema,

usando uma técnica conhecida como “release obscuro”. Em sua palestra sobre o processo de release do Facebook, o gerente de release Chuck Rossi diz que todas as principais features que serão lançadas nos próximos seis meses já estão em produção — você apenas não pode vê-las ainda. Os desenvolvedores protegem as novas features com “feature flags”, para que os administradores possam dinamicamente garantir o acesso a grupos de usuários particulares, por feature.

Dessa maneira, as features podem ficar disponíveis primeiramente para a equipe do Facebook, depois para um pequeno grupo de usuários como parte de um teste A/B (ver *capítulo 9*). Features validadas podem então ser escaladas totalmente para a base de usuários — e desligadas se forem sobre carregadas ou se um defeito for encontrado. Alternância de features também pode ser usada para deixar diferentes conjuntos de features disponíveis para diferentes grupos de usuários de uma única plataforma.

#### **RELEASE OBSCURO PARA APlicativos MOBILE**

Em vez de lançar novos aplicativos mobile diretamente na loja de aplicativos, crie uma marca diferente para implementar e valide-a antes de lançá-la com sua marca oficial.

## **8.5 CONCLUSÃO**

A entrega contínua representa uma alternativa a processos de grandes lotes de desenvolvimento e release. Ela foi adotada por grandes organizações de engenharia em diferentes domínios, incluindo indústrias altamente reguladas, como as de serviços financeiros. Apesar de suas origens em serviços web, este paradigma

de engenharia foi aplicado com sucesso em pacote de software, firmware e desenvolvimento mobile. Ele torna possível que organizações respondam rapidamente a mudanças nas necessidades do consumidor e aumenta a qualidade do software, ao mesmo tempo em que reduz tanto o risco no release quanto o custo do desenvolvimento do software.

A cultura também desempenha um importante papel em possibilitar a entrega contínua. Uma cultura em que interações entre times de desenvolvimento, operações e segurança da informação são geralmente uma relação de ganha-ganha é altamente relacionada à alta performance, por ser uma cultura que está no fim “gerador” da tipologia de Westrum (*capítulo 1*).

Conforme as organizações trabalham para implementar a entrega contínua, elas terão de mudar a maneira como abordam o gerenciamento de controle de versão, desenvolvimento de software, arquitetura, teste e base de dados. A figura seguinte é a síntese de nosso estudo de diferentes organizações.

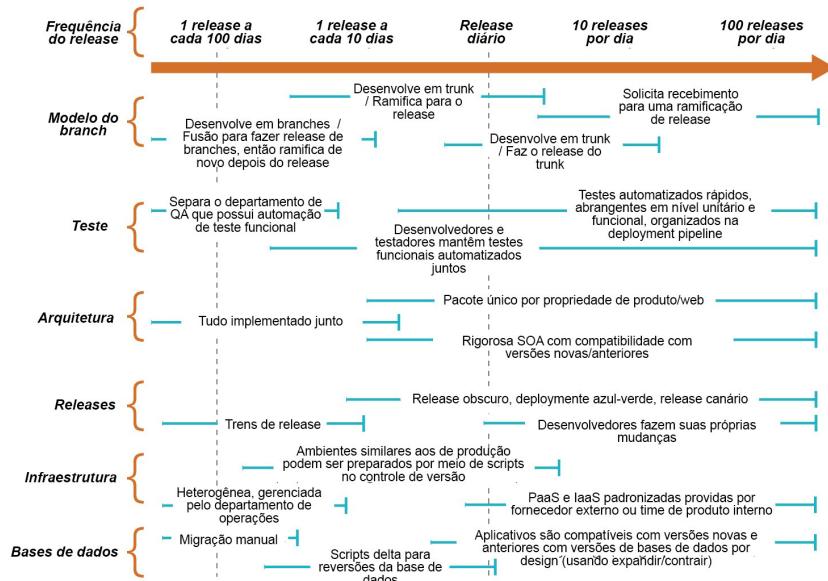


Figura 8.4: As forças G do deployment, adaptado de Paul Hammant (2013)

Claro que todas essas áreas estão inter-relacionadas. Por exemplo, construir uma suíte de testes sustentável, completa e automatizada requer uma arquitetura que permita o software ser implementado na estação de trabalho local do desenvolvedor, o que, por sua vez, requer que os ambientes similares aos de produção possam ser montados por scripts de controle de versão. Trabalhar no que se pode atacar por primeiro, no caso de sistemas existentes, pode ser complexo. Discutiremos a mudança evolutiva da arquitetura no *capítulo 10*.

Recomendamos fortemente que você comece implementando um gerenciamento de configuração completo, integração contínua e desenvolvimento baseado em trunk. É também importante criar uma cultura de automação de teste com os desenvolvedores, o que, por sua vez, requer que os ambientes de teste possam ser providenciados sob demanda. Em nossa experiência, tentativas de dedicar aos problemas no release ou em operações, discutidos no

*capítulo 14*, não produzem melhoria significante sem integração contínua, automação de teste e provisão de ambiente automatizado.

## Questões para os leitores

- Qual é sua definição para “pronto” para uma feature ser aceita? Deve — no mínimo — estar integrada ao trunk e demonstrada em um ambiente similar com o de produção ao se executar testes automatizados?
- Você está praticando a integração contínua como definimos neste livro? Como você faria para introduzi-la?
- As relações entre desenvolvedores, testadores e o pessoal de operações de TI são colaborativas ou adversárias? O que você pode fazer para melhorá-las?
- Os deployments de produção são dolorosos, eventos “big bang” que envolvem interrupções planejadas fora do horário de trabalho? Como você poderia mudá-los para executar mais do trabalho dentro das horas normais de trabalho?

## CAPÍTULO 9

# REALIZE UMA ABORDAGEM EXPERIMENTAL PARA DESENVOLVIMENTO DE PRODUTO

*“A dificuldade em definir qualidade é traduzir futuras necessidades do usuário em características mensuráveis, para que um produto possa ser projetado e seu resultado seja satisfatório pelo preço que o usuário pagará.” — Walter Shewhart*

Até agora, passamos toda a *Parte III* mostrando como melhorar a velocidade com a qual podemos entregar valor para os clientes. Neste capítulo, vamos mudar o foco para discutir alinhamento — como usar a capacidade de entrega para assegurar que estamos construindo as coisas certas para clientes, usuários e para nossa organização.

No *capítulo 7*, mostramos como usar o Custo de Atraso para priorizar o trabalho. Em uma organização em que TI é

essencialmente um provedor de serviço, isso é uma maneira efetiva de evitar trabalho em tarefas de baixo valor que consomem tempo e recursos preciosos. Contudo, em organizações de alta performance, projetos e requisitos não são apenas jogados para o departamento de TI construir. Em vez disso, engenheiros, designers, testadores, equipe de operações e gerentes de produto trabalham juntos para criar resultados de alto valor para clientes, usuários e para a organização como um todo. Além disso, essas decisões — feitas localmente pelo time — levam em consideração as metas estratégicas mais abrangentes da organização.

No *capítulo 6*, descrevemos a Melhoria Kata, uma abordagem interativa para melhoria de processo na qual estabelecemos condições-alvo para a próxima iteração, e então deixamos os times decidirem qual trabalho fazer para alcançar essas condições-alvo. A inovação principal que apresentamos neste capítulo é usar o mesmo processo para gerenciar desenvolvimento de produto. Em vez de aparecer com requisitos ou casos de uso e colocá-los em um backlog para que os times os construam por ordem de prioridade, nós descrevemos, em termos mensuráveis, os resultados de negócio que queremos alcançar na próxima iteração. Então fica a cargo dos times descobrirem ideias para features que vão atingir esses resultados de negócios, testá-los e construir aquelas que alcancem os resultados desejados. Dessa maneira, usamos a habilidade e talento de toda a organização para atingirmos as metas de negócio com mínimo desperdício e a todo vapor.

Como abordagem para executar desenvolvimento de software ágil em escala, isso é diferente da maioria das estruturas de trabalho. Não há backlog em nível de programa; em vez disso, os times criam e gerenciam seus próprios backlogs e são responsáveis por colaborar para atingir as metas de negócio. Essas metas são definidas em termos de condições-alvo em nível de programa, e atualizadas regularmente como parte do processo de Melhoria Kata (veja no

*capítulo 6*). Dessa forma, a responsabilidade pelo atingimento das metas de negócios é empurrada para os times, e estes focam nos resultados de negócios em vez de mensurações como o número de histórias completas (velocidade do time), linhas de código escritas ou horas trabalhadas. De fato, o objetivo é minimizar produção e, ao mesmo tempo, maximizar resultados: quanto menos linhas de códigos que escrevemos e horas que trabalhamos para alcançar nossas metas de negócio desejadas, melhor. Sistemas enormes e complexos demais e equipe exausta são sintomas de foco em produção em vez de resultados.

Uma coisa que não fazemos neste capítulo (ou neste livro) é prescrever quais processos os times devem usar para gerenciar seu trabalho. Os times podem — e devem — ser livres para escolher os métodos e processos que melhor funcionam para eles. De fato, no programa HP FutureSmart, times diferentes usaram, com sucesso, diferentes metodologias e não houve uma tentativa de se impor um “processo-padrão” ou metodologia para os times. O importante é que os times foram capazes de trabalharem juntos efetivamente para atingir as metas do negócio.

Dessa forma, não apresentamos métodos ágeis padrão, como XP e Scrum, ou alternativos, como o Kanban. Há vários livros excelentes que tratam desses métodos detalhadamente, como *Kanban: Successful Evolutionary Change for Your Technology Business*, de David Anderson (2010), *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, de Kenneth S. Rubin (2012), e *The Scrum Field Guide: Practical Advice for Your First Year*, de Mitch Lacey (2012). Em vez disso, discutimos como times podem colaborar para definir abordagens que atinjam as condições-alvo, e então projetar experimentos para testar suas suposições.

As técnicas descritas neste capítulo exigem um alto nível de confiança entre diferentes partes da organização envolvidas no fluxo

de valor do desenvolvimento do produto, assim como entre líderes, gerentes e aqueles que se reportam a eles. Elas também exigem times de alto desempenho e tempos de espera curtos. Assim, a não ser que estes fundamentos estejam prontos (descritos em capítulos anteriores), implementar essas técnicas não irá produzir o valor de que são capazes.

## 9.1 CRIANDO HIPÓTESES PARA A PRÓXIMA ITERAÇÃO

O resultado do processo de planejamento da iteração Melhoria Kata (descrito no *capítulo 6*) é uma lista de condições-alvo mensuráveis que desejamos atingir durante a próxima iteração, descrevendo a intenção do que estamos tentando alcançar e seguindo o Princípio de Missão (ver *capítulo 1*). Neste capítulo, descreveremos como usar o mesmo processo para guiar o desenvolvimento do produto.

Conseguimos isso criando condições-alvo baseadas no cliente e em resultados organizacionais como parte de nosso processo de planejamento de iteração, além de condições-alvo de melhoria de processo. Isso nos permite usar, em nível de programa, melhoria contínua para desenvolvimento de produto também, adotando uma abordagem orientada para a meta para engenharia de requisitos.

Nossas condições-alvo de desenvolvimento do produto descrevem metas de cliente ou negócio que queremos alcançar, as quais são guiadas pela nossa estratégia de produtos. Exemplo incluem aumento da receita por usuário, focar em um novo segmento de mercado, resolver um problema de uma persona em particular, aumentando a performance de nosso sistema, ou redução do custo de transação. Contudo, não propomos soluções para atingir essas metas, ou escrever histórias ou features (especialmente,

as “épicas”) em nível de programa. Em vez disso, cabe aos times dentro do programa decidirem como eles atingirão essas metas. Isso é crítico para se conseguir alta performance em escala, por duas razões:

- As soluções iniciais que propomos raramente são as melhores. Soluções melhores são descobertas criando, testando e refinando múltiplas opções para descobrir quais resolvem o problema da melhor maneira.
- Organizações só podem se mover rapidamente em escala quando as pessoas trabalhando nas soluções possuem um profundo entendimento tanto das necessidades do usuário quanto da estratégia do negócio, e expõem suas próprias ideias.

Um backlog em nível de programa não é uma maneira efetiva de conduzir esses comportamentos — apenas reflete a quase irresistível tendência humana de especular “os meios para se fazer algo, em vez do resultado que queremos” (GILB, 1988, p. 23).

## CHEGANDO ÀS CONDIÇÕES-ALVO

Engenharia de requisitos orientada por meta tem sido usada por décadas — veja Yu, Giorgini, Maiden e Mylopoulos (2010), Lapouchnian (2005) e Gilb (2005) para engenharia de requisitos orientada para metas —, mas a maioria das pessoas ainda está acostumada a definir o trabalho em termos de features e benefícios, em vez de resultados de negócios mensuráveis e clientes. A abordagem features-e-benefícios trabalha com nossa tendência natural de chegar a soluções, e temos de pensar mais para especificar os atributos de uma solução apenas aceitável.

Se você tem features e benefícios e quer chegar às condições-alvo, uma simples abordagem é se perguntar por que clientes dão importância a um benefício em particular. Você pode precisar se perguntar “por que” várias vezes para chegar a algo que se pareça com uma condição-alvo real — este é um velho truque usado por Taiichi Ohno, chamado “os cinco porquês”. É também essencial se assegurar que as condições-alvo tenham critérios de aceitação mensuráveis, como mostrado na figura adiante.

Gojko Adzic apresenta uma técnica chamada mapeamento de impacto para desmembrar metas de negócios de alto nível para o nível de programa, de hipóteses testáveis. Adzic descreve um mapa de impacto como “uma visualização do escopo e suposições subjacentes, criado colaborativamente por um grupo multifuncional de stakeholders. É um mapa mental aumentado durante uma discussão facilitada pelas respostas às seguintes questões: Por quê? Quem? Como? O quê?” (ADZIC, 2012, l. 146). Um exemplo do

mapa de impacto é mostrado na figura:

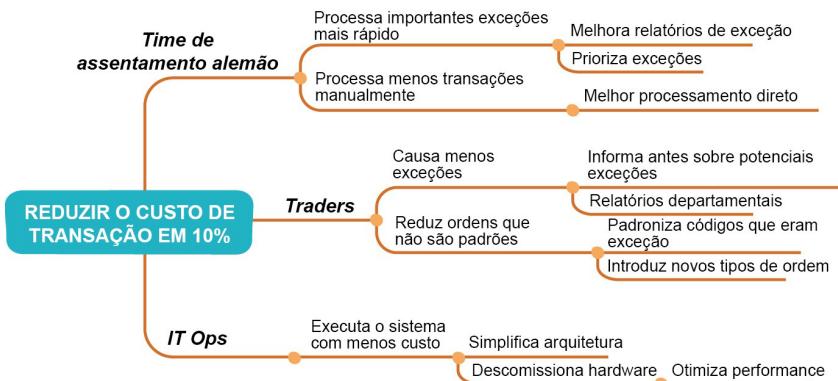


Figura 9.1: Exemplo de um mapa de impacto, cortesia de Gojko Adzic (2012)

Começamos um mapa de impacto com uma condição-alvo em nível de programa. Ao definir um objetivo, incluindo a sua intenção (por que nos importamos com ele, de uma perspectiva de negócios), nos asseguramos de que todos que estão trabalhando para atingir a meta entendem o propósito do que estão fazendo, seguindo o Princípio da Missão. Também fornecemos critérios de aceitação claros para que possamos determinar em que momento alcançamos a condição-alvo.

O primeiro nível de um mapa de impacto enumera todos os stakeholders com interesse naquela condição-alvo. Isso inclui não apenas os usuários finais que serão afetados pelo trabalho, mas também as pessoas dentro da organização envolvidas ou que serão impactadas, ou que podem influenciar no progresso do trabalho — tanto positivamente quanto negativamente.

O segundo nível do mapa de impacto descreve maneiras possíveis pelas quais os stakeholders podem ajudar — ou impedir — a atingir o objetivo. Essas mudanças de comportamento são impactos que pretendemos criar.

Até aqui, não devemos mencionar sobre as possíveis soluções que nos moverão em direção à nossa condição-alvo. É apenas no terceiro nível do mapa de impacto que propomos opções para alcançar a condição-alvo. Primeiramente, devemos propor soluções que não envolvam escrever código — como atividades de marketing ou simplificação de processos de negócio. O desenvolvimento de software deve sempre ser um último recurso, por causa do custo e complexidade de construir e manter um software.

As possíveis soluções propostas no mapa de impacto não são o produto-chave. Pensar em possíveis soluções simplesmente nos ajuda a refinar nosso pensamento em relação à meta e aos stakeholders. As soluções em que pensamos nesta fase serão dificilmente as melhores — esperamos, em vez disso, que as pessoas trabalhando para entregar os resultados apareçam com melhores opções e as avaliem para determinar quais atingirão nossa condição-alvo. O mapa de impacto pode ser considerado um conjunto de suposições — por exemplo, na figura anterior, assumimos que códigos de exceção padronizados reduzirão pedidos não padronizados, o que reduzirá o custo de processamento de transações não padronizadas.

Para essa ferramenta funcionar efetivamente, é importante ter as pessoas certas envolvidas no exercício do mapeamento de impacto. Pode ser um time pequeno, multifuncional, incluindo stakeholders, equipe técnica, designers, analistas de qualidade, operações de TI e suporte. Se o exercício é conduzido puramente pelos stakeholders de negócio, eles perderão a oportunidade de examinar as suposições por trás das condições-alvo e de receber ideias de designers e engenheiros que têm mais intimidade com o problema. Um dos objetivos mais importantes do mapeamento de impacto é criar um entendimento comum entre os stakeholders.

Uma vez que priorizamos uma lista de condições-alvo e mapas

de impacto criados colaborativamente pelo pessoal técnico e de negócios, cabe aos times determinar o caminho mais curto possível para a condição-alvo.

Essa ferramenta se difere de maneiras importantes de muitas abordagens-padrão de pensar sobre requisitos. Aqui estão algumas das importantes diferenças e motivações por trás delas:

- **Não há listas de features em nível de programa.**

As features são simplesmente mecanismos para se atingir a meta. Parafraseando Adzic, se atingir uma condição-alvo com um conjunto de features completamente diferente do que previmos não contar como sucesso, nós escolhemos a condição-alvo errada. Especificar condições-alvo em vez de features nos permite responder rapidamente a mudanças em nosso ambiente e a informações que colhemos de stakeholders enquanto trabalhamos para alcançar a condição-alvo. Isso evita o “churn de feature” durante a iteração. Mais importante, isso é a maneira mais efetiva de fazer uso do talento daqueles que trabalham para nós; isso os motiva ao dar-lhes a oportunidade de buscar maestria, autonomia e propósito.

- **Não há estimativa detalhada.**

Objetivamos uma lista de condições-alvo que seja uma meta elástica — em outras palavras, se todas as nossas suposições são boas e todas as nossas apostas tiverem êxito, descobriremos que é possível atingir as condições. Contudo, isso raramente acontece, o que significa que podemos não alcançar algumas das condições-alvo menos prioritárias. Se regularmente estivermos alcançando muito menos, precisamos

reequilibrar nossas condições-alvo em prol das metas de melhoria do processo. Manter as iterações curtas — de 2 a 4 semanas, inicialmente — faz com que possamos ajustar as condições-alvo em resposta ao que descobrimos durante a iteração. Isso permite que rapidamente detectemos se estamos no caminho errado e tentemos uma abordagem diferente antes de investir muito nas coisas erradas.

- **Não existem “épicos arquiteturais”.**

As pessoas fazendo o trabalho devem ter completa liberdade para fazer quaisquer trabalhos de melhoria que quiserem (incluindo mudanças de arquitetura, automação e refactoring) para melhor alcançar as condições-alvo. Se queremos excluir metas particulares que necessitarão de trabalho de arquitetura, como compliance ou performance melhorada, nós as especificamos nas condições-alvo.

## 9.2 PESQUISA DE USUÁRIO

O mapa de impacto nos fornece uma quantia de possíveis soluções e um conjunto de suposições para cada solução. Nossa tarefa é encontrar o caminho mais curto para a condição-alvo. Selecionamos uma que pareça mais curta e validamos a solução — junto com suas suposições — para ver se é realmente capaz de entregar o valor esperado (como vimos, features geralmente falham em entregar o valor esperado). Há muitas maneiras de validar nossas suposições.

Primeiro, criamos uma hipótese baseada em nossa suposição. No livro *Lean UX*, Josh Seiden e Jeff Gothelf (2013, p. 23) sugerem o modelo mostrado a seguir para usar como ponto de partida para

capturar hipóteses.

**ACREDITAMOS QUE**

[construir essa feature]

[para essas pessoas]

**resultará [neste resultado].**

**Saberemos que tivemos sucesso quando tivermos**

[este sinal do mercado].

Neste formato, descrevemos os parâmetros do experimento que faremos para testar o valor da feature proposta. O resultado descreve a condição-alvo que queremos atingir.

Tal como acontece com o formato de história de usuário, resumimos o trabalho (por exemplo, a feature que queremos construir ou a mudança no processo de negócios que queremos fazer) em poucas palavras para que lembremos da conversa que tivemos sobre ela com o time. Também especificamos a persona, cujo comportamento mediremos quando rodarmos o experimento. Finalmente, especificamos o sinal que vamos mensurar no experimento. Em experimentos online controlados, que serão discutidos na próxima seção, isso é conhecido como critério de avaliação global para o experimento.

Uma vez que temos uma hipótese, podemos começar a projetar o experimento. Essa é uma atividade multifuncional que requer colaboração entre especialistas em design, desenvolvimento, teste, techops e análise, apoiada por experts no assunto, se for o caso. Nossa objetivo é minimizar a quantidade de trabalho que teremos

para reunir uma quantia suficiente de dados para validar ou não as suposições de nossa hipótese. Há muitos tipos de pesquisa de usuário que podemos fazer para testar nossa hipótese, como mostrado na figura seguinte. Para saber mais sobre diferentes tipos de pesquisa de usuário, leia *UX for Lean Startups*, de Laura Klein (2013).



Figura 9.2: Diferentes tipos de pesquisa de usuário, cortesia de Janice Fraser (2011)

O resultado-chave para um experimento é informação: queremos reduzir a incerteza quanto ao fato do trabalho proposto atingir a condição-alvo. Há muitas maneiras de rodarmos o experimento para reunir informação. Tenha em mente que experimentos frequentemente têm um resultado negativo ou inconclusivo, especialmente em condições de incerteza; isso significa que frequentemente temos de ajustar, refinar e evoluir

nossas hipóteses ou criar um novo experimento para testá-las.

A chave para a abordagem experimental para desenvolvimento de produto é que nós não dominamos um novo trabalho de desenvolvimento sem antes criar uma hipótese para que possamos determinar se nosso trabalho entregará o valor esperado. Em muitos aspectos, esta abordagem é apenas uma extensão do desenvolvimento guiado por teste. Chris Matts teve uma ideia similar, a qual chama de injeção de feature.

### 9.3 EXPERIMENTOS CONTROLADOS ONLINE

No caso de serviços baseados na internet, podemos usar um método poderoso chamado de experimento controlado online, ou teste A/B, para testar uma hipótese. Um teste A/B é um experimento controlado, randomizado, para descobrir quais das duas possíveis versões de uma página produz melhor resultado. Quando executamos um teste A/B, preparamos duas versões de uma página: um controle (normalmente, a versão existente dela) e um novo tratamento que queremos testar. Quando um usuário visita nosso site pela primeira vez, o sistema decide a quais experimentos aquele usuário está sujeito, e para cada experimento ele escolhe aleatoriamente se ele acessará o controle (A) ou o tratamento (B). Nós medimos a interação do usuário com o sistema tanto quanto possível para detectar quaisquer diferenças de comportamento entre o controle e o tratamento.

**A MAIORIA DAS BOAS IDEIAS NA VERDADE ENTREGA VALOR ZERO OU NEGATIVO**

Talvez o resultado mais revelador do teste A/B é quantas ideias aparentemente boas não melhoram o valor, e como é completamente impossível distinguir os erros

antecipadamente. Como discutido no *capítulo 2*, dados coletados de testes A/B por Ronny Kohavi, que dirigiu o grupo de Data Mining e Personalização da Amazon antes de se juntar à Microsoft como gerente geral de sua Plataforma Experimental, revelam que 60% a 90% das ideias não melhoraram a métrica que se destinavam a melhorar.

Assim, se não estamos fazendo experimentos para testar o valor de novas ideias antes de desenvolvê-las completamente, as chances são que cerca de 2/3 do trabalho que estamos fazendo é de valor zero ou negativo para nossos clientes — e certamente de valor negativo para nossa organização, já que este trabalho apresenta três formas de custo. Além do custo de desenvolver as features, há um custo de oportunidade associado com mais trabalho de valor que poderíamos ter feito, e o custo da nova complexidade que eles adicionam aos nossos sistemas (que se manifestam como custo de manutenção do código, um entrave na taxa em qual podemos desenvolver uma nova funcionalidade e, frequentemente, estabilidade e desempenho operacionais reduzidos).

Apesar dessas terríveis probabilidades, muitas organizações acharam difícil acatar experimentos em execução para mensurar o valor de novas features ou produtos. Alguns designers e editores sentem que desafia sua expertise. Executivos se preocupam que isso possa ameaçar seu emprego como tomadores de decisão e que eles possam perder o controle sobre essas decisões.

Kohavi, que cunhou o termo *HIPPO*, diz que este trabalho é “dizer aos clientes que seu bebê é feio”, e carrega consigo hipopótamos de borracha para dar a estas pessoas, para deixar o clima mais leve e lembrá-los de que a maioria das “boas” ideias não é boa e que é impossível dizer quais serão

problemáticas sem termos dados.

Ao executar o experimento com um grupo de usuários grande o suficiente, queremos reunir dados para demonstrar uma diferença estatisticamente significante entre A e B para a métrica de negócio que queremos, conhecida como critério de avaliação global, ou OEC (*Overall Evaluation Criterion*, em inglês) — compare com a Única Métrica que Importa, do *capítulo 4*. Kohavi (2013) sugere otimizar e mensurar o valor de vida do cliente em vez de receita a curto prazo. Para um site como o Bing, ele recomenda usar uma boa soma de fatores, como tempo no site por mês e frequência de visita por usuário, com o objetivo de melhorar a experiência global do cliente e fazê-lo retornar.

Diferente de data mining (mineração de dados), que pode apenas descobrir correlações, o teste A/B tem o poder de mostrar uma relação casual entre uma mudança em uma webpage e uma mudança correspondente na métrica com a qual nos preocupamos. Empresas como a Amazon e a Microsoft normalmente fazem centenas de experimentos em produção a qualquer momento e testam toda nova feature usando este método antes de implantá-la. Cada visitante do Bing, o serviço de busca da Microsoft, estará participando de cerca de 15 experimentos por vez (KOHAVI, 2013).

## USANDO TESTES A/B PARA CALCULAR O CUSTO DE ATRASO PARA MELHORIAS DE PERFORMANCE

Na Microsoft, o time de Ronny Kohavi queria calcular o impacto de melhorar a performance das buscas no Bing. Eles fizeram isso rodando um teste A/B no qual introduziram uma demora artificial no servidor para usuários que estavam vendo a versão “B”. Eles foram capazes de calcular uma quantia em dólar para o impacto na receita de melhorias de performance, descobrindo que “um engenheiro que melhora a performance do serviço em 10 milissegundos mais do que paga pelos seus custos anuais”. Esse cálculo pode ser usado para determinar o custo de atraso por melhorias de performance.

Quando criamos um experimento para usar como parte do teste A/B, queremos ter muito menos trabalho do que teríamos com a total implementação da feature sob consideração. Podemos calcular a quantidade máxima que deveríamos gastar em um experimento ao determinar o valor esperado da informação que ganharemos ao fazê-lo, como discutido no *capítulo 3* (apesar de que, normalmente, nós gastaremos muito menos).

No contexto de um site, aqui estão algumas maneiras de reduzir o custo de um experimento:

- *Use a regra do 80/20 e não se preocupe com casos excepcionais:* construa os 20% da funcionalidade que entregarão 80% do benefício esperado.
- *Não construa em escala:* experimentos em um site ativo são normalmente apenas vistos por uma pequena porcentagem de usuários.

- *Não se preocupe com compatibilidade entre navegadores:* com um simples código de filtragem, você pode se assegurar que apenas usuários com o navegador correto podem ver o experimento.
- *Não se preocupe com cobertura de teste significativa:* você pode adicionar cobertura de teste mais tarde quando a feature estiver validada. Um bom monitoramento é muito mais importante quando trabalhamos com experimentação.

## 9.4 UM EXEMPLO DE TESTE A/B

O Etsy é um site onde pessoas podem vender bens de artesanato. O site usa teste A/B para validar todas as ideias de novos produtos. Por exemplo, um produtor percebeu que a busca por um item específico na loja de alguém não dava nenhum resultado e queria descobrir se uma feature que mostra itens similares de lojas de outras pessoas aumentaria a receita. Ele criou um arquivo de configuração para determinar qual a porcentagem de usuários que veria o experimento.

Os usuários que acessavam a página em que o experimento estava rodando seriam aleatoriamente alocados ou para um grupo-controle (A) ou para o grupo que veria o experimento (B), baseado na carga no arquivo de configuração. Experimentos arriscados serão vistos apenas por uma pequena porcentagem de usuários. Uma vez que um usuário é alocado para uma das versões, ele vai para lá em todas as visitas para que o site tenha uma aparência consistente para ele.

### **EXPERIMENTO SEGURO PARA FALHAR**

O teste A/B permite que times definam as restrições, limitações ou limites para criar um experimento seguro para falhar. O time pode definir o limite de controle de uma métrica-chave antes do teste para que possam revertê-lo ou abortá-lo se esse limite é alcançado (por exemplo, conversão cai além de um número estabelecido). Determinar, compartilhar e concordar sobre esses limites com todos os stakeholders antes de conduzir o experimento estabelecerá até onde o time pode experimentar com segurança.

O comportamento subsequente do usuário é então rastreado e mensurado — por exemplo, podemos ver quantos chegam à página de pagamento. O Etsy tem uma ferramenta, mostrada na figura a seguir, que mede a diferença de comportamento para vários endpoints e indica quando foi atingida a significância estatística em um intervalo de confiança de 95%. Por exemplo, para “site – page count”, “+0,26%” indica que o experimento produz uma melhoria estatisticamente significante de 0,26% sobre a versão controle. Experimentos normalmente têm de rodar por alguns dias para produzir dados estatisticamente significantes.

Gerar uma mudança de mais de alguns pontos percentuais em uma métrica de negócio é raro e pode normalmente ser atribuído à Lei de Twyman: “Se uma estatística parece interessante ou inusitada, ela provavelmente está errada”.

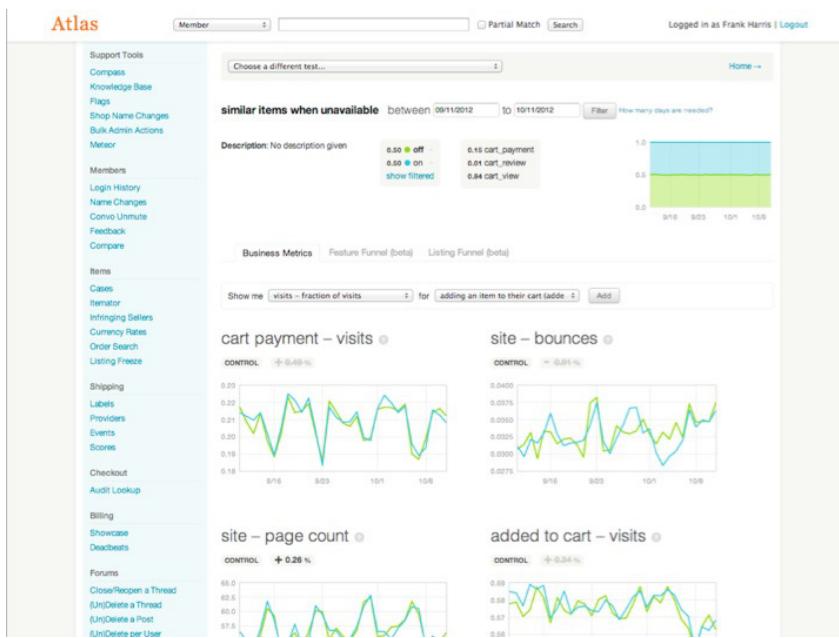


Figura 9.3: Mensurando mudanças no comportamento do usuário usando teste A/B

Se a hipótese é validada, mais trabalho pode ser feito para se construir a feature e escalá-la, até que finalmente a feature se torne disponível para todos os usuários do site. Tornar visível para 100% dos usuários é equivalente a fazer o release da feature publicamente — uma importante ilustração da diferença entre deployment e release que discutimos no *capítulo 8*. O Etsy sempre tem um número de experimentos rodando em produção a qualquer momento. Em um painel, você pode ver quais experimentos estão planejados, quais estão em execução e quais foram completados, o que permite que as pessoas mergulhem nas métricas correntes para cada experimento, como mostrado na figura:

Etsy Launch Calendar				
<a href="#">Home</a>   <a href="#">By Team</a>   <a href="#">Add an item</a>   <a href="#">Last Updated</a>   <a href="#">More</a>				
Have an upcoming launch? <a href="#">Add an item</a>				
<a href="#">Active experiments</a> <a href="#">Upcoming launches</a>		<a href="#">Recent 100% launches</a>		
Date	Name	Team	Notes	
Nov 7	<a href="#">Gift Ideas browse pages</a>  	Buyer Experience	This is a gift guide browse destination. Subsections will focus on recipient (for him, for her, for kids, etc.) and price (under \$25, under \$100, etc.). It will work just like all other browse pages. There will be NO HAND ...	
Nov 7	<a href="#">Etsy for iPhone (v2.1.1)</a>	Mobile	<a href="#">Example</a> — We submitted the app on Friday. We will be pushing it out when it's approved by Apple; our hope is that it's approved by Wednesday. There will be no coordination with PR or blog post. We may send ...	
Nov 2	<a href="#">Winter Holidays browse pages</a>  	Buyer Experience	<a href="#">Example</a> — These are browse pages for the Winter Holidays and will feature subsections for holiday decor, cards, etc. They'll be similar to our holiday merch hub from last year, but much deeper in terms of browsing opportunities. Those in UK ...	
Nov 1	<a href="#">Updated treatment of homepage browse links</a>  	Buyer Experience	<a href="#">Example</a> — Over a two week period we observed 4%-5% increases in browse landing page and subsection page views. There were also slight increases in add to cart and listings viewed events. Visits with a search and search events were down ...	
Oct 24	<a href="#">Next day availability of DC funds</a>  	Payments	We plan to allow established sellers to be able to deposit their funds prior the next day after a sale. Non established sellers will still need to ship items to have available funds.	
Oct 23	<a href="#">Reduce one-time hold from 10 days to 5 days</a>	Payments	Whenever a new seller signs up for direct checkout, a 10 day hold is placed on deposits. This also occurs anytime a bank account is updated. We have decided to reduce this standard hold period to 5 days. The main ...	
Oct 23	<a href="#">Etsy for iPhone (v2.1)</a>  	Mobile	<a href="#">Example</a> — Update: We have been approved by Apple and will be launching Tuesday, 10/23 at 8am ET. ... Our target submit date to Apple is Wednesday 10/10. Depending on Apple's turnaround time, we expect the app to be ...	
Oct 22	<a href="#">Recipient Query Rewriting</a>	Search & Destroy	<a href="#">Example</a> — This didn't move metrics positively or negatively. However we decided to keep it because this is the first step towards using recipient in search, and encouraging users to properly associate their listing w/ a recipient. We will reevaluate how ...	
Oct 19	<a href="#">Parcel Insurance for Shipping Labels</a>  	Seller Team	<a href="#">Example 1</a> , <a href="#">Example 2</a> — Rampup started 10/9. Scheduled to finish 10/19.	
Oct 18	<a href="#">Search Ads respecting filters</a>	Search & Destroy	This experiment didn't hurt inventory: <a href="https://splunk.etsycorp.com/en-US/app/search/filterTimeline?sid=1350940765.163366&amp;vs=h8m3sk4b">https://splunk.etsycorp.com/en-US/app/search/filterTimeline?sid=1350940765.163366&amp;vs=h8m3sk4b</a> Also it looks like CTR might have improved.	

Figura 9.4: Experimentos rodando atualmente no Etsy

## ALTERNATIVAS PARA O TESTE A/B

Apesar de gastarmos muito tempo falando sobre teste A/B neste capítulo, ele é apenas um de muitas técnicas experimentais para colher dados. Designers de experiência do usuário têm uma gama de ferramentas para obter feedback de usuários, desde protótipos *lo-fi* a métodos de pesquisa etnográfica como questionário contextual, mostrado na figura *Diferentes tipos de pesquisa de usuário, cortesia de Janice Fraser (2011)*. O livro *Lean UX: Applying Lean Principles to Improve User Experience* discute muitas dessas ferramentas e como aplicá-las no contexto do desenvolvimento guiado por hipótese (GOTHELF; SEIDEN, 2013).

## 9.5 PRÉ-REQUISITOS PARA UMA ABORDAGEM EXPERIMENTAL EM DESENVOLVIMENTO DE PRODUTO

Convencer as pessoas a coletar — e prestar atenção a — dados reais vindos de experimentações, como teste A/B, é bastante difícil. Mas uma abordagem experimental e científica que crie valor para o cliente tem implicações na maneira como trabalhamos, assim como na maneira como pensamos sobre valor.

Como aponta Dan McKinley (2012), do Etsy, a experimentação não pode ser aparafusada a um processo-cascata de desenvolvimento de produto. Se chegarmos ao final de várias semanas (ou meses) de trabalho e tentarmos um experimento, há uma boa chance de descobrirmos que fizemos uma enorme quantidade de trabalho com zero efeito ou que deixa tudo pior. A essa altura, teremos de jogar tudo fora porque já não há uma maneira precisa de identificar o efeito de cada mudança específica que foi feita.

Essa é uma decisão extremamente dolorosa para ser tomada e, na prática, muitos times sucumbem à falácia dos custos irrecuperáveis ao dar peso indevido ao investimento feito até a data quando tomarem a decisão. Eles ignoram os dados e implementam o produto como está porque arquivar o trabalho é considerado uma falha completa, visto que um deploy bem-sucedido de qualquer coisa para produção é percebido como sucesso — desde que dentro do tempo e do orçamento.

Se vamos adotar uma abordagem experimental minuciosa, precisamos mudar o que consideramos ser resultado de nosso trabalho: não apenas ideias validadas, mas a informação que conseguimos durante a execução dos experimentos. Também precisamos mudar a maneira de pensar sobre novas ideias;

particularmente, é essencial trabalhar com pequenos lotes e testar toda suposição por trás da ideia que estamos validando. Isso, por sua vez, requer que implementemos a entrega contínua, como descrito no *capítulo 8*.

Trabalhar em pequenos lotes cria um fluxo — um elemento-chave do Pensamento Lean. Mas pequenos lotes são difíceis de se conseguir, tanto por razões filosóficas como por razões técnicas. Algumas pessoas acham difícil usar uma abordagem incremental para criar produtos. Uma objeção comum à abordagem experimental é que ela leva a decisões localmente ideais, mas globalmente abaixo do ideal, e que isso compromete a integridade do produto como um todo, matando uma linda visão holística por milhares de testes A/B.

Enquanto é certamente possível terminarmos com um produto feio e altamente complexo quando os times falham ao executar uma abordagem holística na experiência do usuário, isso não é um resultado inevitável dos testes A/B. A experimentação não quer substituir ter uma visão sobre seu produto. Em vez disso, ela permite que você evolua sua estratégia e visão rapidamente, em resposta a dados reais de clientes usando seu produto. Testes A/B não serão efetivos na ausência de uma visão e uma estratégia. Gerentes de produto, designers e engenheiros precisam colaborar e aplicar as lições de design thinking para obter uma visão de longo prazo das necessidades do usuário e estabelecer uma direção para o produto.

## MAS O QUE É DESIGN THINKING?

Tim Brown, CEO e presidente da IDEO e um dos principais personagens em design thinking, diz: “Como um estilo de pensamento, design thinking é geralmente considerado a capacidade de combinar empatia pelo contexto de um problema, criatividade na geração de insights e soluções e racionalidade para analisar e encaixar soluções no contexto”. Discutimos design thinking e Lean UX no *capítulo 4*.

Há dois obstáculos a mais para se usar uma abordagem experimental em desenvolvimento de produto. Primeiro, projetar experimentos é complicado: temos de evitar que eles interfiram uns nos outros, aplicar alertas para detectar anomalias e projetá-los para produzirem resultados válidos. Ao mesmo tempo, queremos minimizar a quantidade de trabalho que devemos ter para reunir dados significantes estatisticamente.

Por último, usar uma abordagem científica para clientes e desenvolvimento de produtos requer uma colaboração intensiva entre equipes de produto, design e técnica por todo o ciclo de vida de todo produto. Esta é uma grande mudança cultural para muitas empresas, em que a equipe técnica geralmente não contribui com o processo global de design.

Esses obstáculos são a razão pela qual nós fortemente desencorajamos as pessoas a adotarem as ferramentas discutidas neste capítulo, sem antes colocar em prática os fundamentos descritos nos capítulos anteriores da *Parte III*.

## INOVAÇÃO REQUER UMA CULTURA DE EXPERIMENTAÇÃO

Greg Linden, que desenvolveu o primeiro mecanismo de recomendações da Amazon, montou uma hipótese que dizia que mostrar recomendações personalizadas na hora do check-out poderia convencer as pessoas a fazer compras por impulso — de maneira similar às prateleiras nas filas do caixa em supermercados, mas compilado de maneira personalizada para cada cliente por um algoritmo. Contudo, um vice-presidente sênior que viu o demo de Greg estava convencido de que isso distrairia as pessoas no check-out. Greg foi proibido de levar adiante o trabalho na feature (LINDEN, 2006)

Linden desobedeceu ao vice-presidente e colocou um teste A/B em produção. O teste demonstrou um aumento na receita tão claro quando as pessoas receberam recomendações personalizadas no check-out que a feature foi desenvolvida e lançada com urgência.

É concebível que um engenheiro em sua empresa coloque um teste A/B forçadamente em produção mesmo com a censura de um executivo sênior? Se os dados do experimento provarem que o executivo estava errado, qual a probabilidade de a feature ser escolhida em vez de enterrada? Como escreve Linden (2006), “a criatividade deve fluir de todos os lados. Seja você o estagiário ou o diretor de tecnologia, qualquer boa ideia deve ser capaz de passar por um teste objetivo, de preferência um que exponha a ideia a clientes reais. Todos devem ser capazes de experimentar, aprender e iterar. Posição, obediência e conformidade não devem impedir. Para a inovação florescer, a mensuração deve governar”.

Uma cultura baseada em mensuração e experimentação não é uma antítese para ideias loucas, pensamento divergente e raciocínio abdutivo. Pelo contrário, ela dá às pessoas a permissão para ir atrás de suas ideias loucas — facilitando a

reunião de dados reais para embasar o louco bom e rejeitar o louco ruim. Sem a capacidade de executar experimentos baratos e seguros para falhar, tais ideias são normalmente esmagadas por um HiPPO e pela mediocridade da decisão feita por comitê.

Um dos desafios mais comuns encontrados no desenvolvimento de software é o foco dos times, gerente de produto e organizações ao gerenciar custo no lugar de valor. Isso normalmente se manifesta em esforço indevido em atividades que não adicionam valor, tais como análise inicial detalhada, estimativa, gerenciamento de escopo e preparação de backlog. Esses sintomas são o resultado de se focar em maximizar a utilização (manter nosso pessoal ocupado) e produção (mensurar seu trabalho), em vez de se focar em resultados, minimizando a produção requerida para consegui-los e reduzindo os tempos de espera para obter feedbacks rápidos em nossas decisões.

## 9.6 CONCLUSÃO

A maioria das ideias — até mesmo as aparentemente boas — entregam zero ou valor negativos para os usuários. Ao focar nos resultados que queremos atingir, em vez de focar em soluções e features, podemos separar o que estamos tentando fazer das maneiras possíveis de se fazer. Então, segundo o Princípio da Missão, os times podem desempenhar pesquisa de usuário (incluindo experimentos online de baixo risco, seguros para falhar) para determinar o que realmente fornecerá valor para os clientes — e para a nossa organização.

Ao combinar mapa de impacto e pesquisa de usuário com a estrutura de trabalho Melhoria Kata, apresentada no *capítulo 6*,

podemos escalar entrega de software ágil e combiná-lo com design thinking e uma abordagem experimental para desenvolvimento de produto. Isso nos permite rapidamente descobrir, desenvolver e entregar soluções de alto valor e alta qualidade para usuários em escala, aproveitando a habilidade e talento de todos na organização.

## Questões para os leitores

- O que acontece em sua organização quando uma quantidade substancial de esforço foi investida em uma ideia que, no fim das contas, fornece pouco valor a usuários ou à organização, ou até mesmo piora as coisas?
- Os resultados esperados para as features nas quais você está trabalhando foram quantificados? Você tem uma maneira para medir os resultados reais?
- Que tipo de pesquisa do usuário você fez nos protótipos antes de liberá-los mais amplamente? Como você pode receber o feedback mais rapidamente e de maneira mais barata?
- Quando foi a última vez que você observou pessoalmente seu produto ser usado?
- Você consegue pensar em uma maneira barata de testar o valor do próximo trabalho em seu backlog?

## CAPÍTULO 10

# IMPLEMENTE O COMANDO DA MISSÃO

*“Quanto mais alinhado você está, mais autonomia você pode conceder. Um viabiliza o outro.”* — Reed Hastings

*“Os melhores gerentes descobrem como produzir melhores resultados setando o contexto apropriado, em vez de tentar controlar as pessoas.”* — Reed Hastings

Em sua apresentação realizada em 2009 sobre a cultura do Netflix, Liberdade e Responsabilidade, o CEO Reed Hastings (2009) descreve uma dinâmica comum a várias organizações em crescimento. À medida que as organizações se tornam maiores, elas se tornam mais complexas em termos dos sistemas que estão em evolução e em execução, o ambiente empresarial em que operam, e sua capacidade de "fazer as coisas".

Eventualmente, a empresa torna-se demasiadamente complexa para ser executada informalmente, e processos formais são postos em prática para evitar que ela afunde no caos. Processos fornecem um certo grau de previsibilidade, mas eles nos atrasam e fazem pouco para evitar resultados ruins, oriundos de eventos que não podem ser gerenciados através do processo (por exemplo, trabalho

que vai de acordo com o plano, mas que não agrega valor ao cliente).

Gestão através de controle de processo é aceitável em certos contextos dentro dos processos de fabricação (o tipo de sistemas para os quais Six Sigma faz sentido), mas não no desenvolvimento de produtos — em que seu resultado é a otimização para eficiência e previsibilidade à custa da inovação e capacidade de adaptação a cenários de mudanças. Geoff Nicholson, o criador dos Post-It, afirma que a adoção pela 3M do Seis Sigma a mando do CEO James McNerney (ex-GE e agora da Boeing) "matou a inovação" (HUANG, 2013). Processos prescritivos, baseados em regras, também atuam como um freio na melhoria contínua, a menos que as pessoas que operam o processo tenham permissão para modificá-lo. Finalmente, uma dependência excessiva ao processo tende a expulsar as pessoas que mexem, assumem riscos, e executam experimentos "*safe-to-fail*". Esses tipos de pessoas tendem a se sentir sufocados em um ambiente de processo pesado, mas eles são os drivers essenciais de uma cultura de inovação.

Da mesma forma, à medida que as organizações crescem, os sistemas que elas constroem e operaram aumentam de complexidade. Para levar novas funcionalidades rapidamente ao mercado, muitas vezes trocamos qualidade por uma maior velocidade. Esta é uma decisão sensata e racional. Mas em algum momento, a complexidade dos nossos sistemas torna-se um fator limitante na nossa capacidade de entregar novos trabalhos, e chegamos a uma rua sem saída. Muitas empresas têm milhares de serviços em produção, incluindo sistemas de missão crítica rodando em plataformas legadas. Estes sistemas são muitas vezes interligados de forma que torna muito difícil mudar qualquer parte do sistema sem também mudar os outros, o que se transforma em um significativo arrasto na sua capacidade de inovação em grande escala.

Estas preocupações organizacionais e de arquitetura são muitas vezes as maiores barreiras para a execução da estratégia para avançar rapidamente em escala com base nos princípios de Comando da Missão. Começaremos apresentando uma execução virtuosa de uma estratégia para gerenciar a complexidade organizacional e sistêmica na era web: a Amazon. Em seguida, apresentaremos princípios organizacionais, de arquitetura e de liderança que permitem às organizações crescerem com sucesso.

## 10.1 A ABORDAGEM DA AMAZON PARA O CRESCIMENTO

Em 2001, a Amazon tinha um problema: a enorme e monolítica "grande bola de lama", que rodava o seu website, um sistema chamado Obidos, não conseguia escalar. O fator limitante eram as bases de dados. O CEO Jeff Bezos transformou este problema em uma oportunidade. Ele queria que a Amazon se tornasse uma plataforma que outras empresas poderiam aproveitar, com o objetivo final de atender melhor as necessidades dos clientes. Com isto em mente, ele enviou um memorando ao corpo técnico direcionando-os a criar uma arquitetura orientada a serviços, o que Steve Yegge resume assim:

O lendário "discurso da plataforma" de Steve Yegge é leitura obrigatória para líderes técnicos. Veja em <https://plus.google.com/+RipRowan/posts/eVeouesvaVX>.

1. Todas as equipes vão expor seus dados e funcionalidades através de interfaces de serviço.
2. Equipes precisam se comunicar entre si por meio dessas interfaces.

3. Não será permitida nenhuma outra forma de comunicação entre processos: nenhuma vinculação direta, nenhum acesso direto à base de dados de uma outra equipe, nenhum modelo de memória compartilhada, nenhum tipo de *backdoor*. A única comunicação permitida é por meio de chamadas de interface de serviço pela rede.
4. Não importa qual tecnologia seja usada. HTTP, Corba, Pubsub, protocolos próprios — não importa. Bezos não se importa.
5. Todas as interfaces de serviços, sem exceção, precisam ser desenhadas desde o início para serem externalizáveis. Isso quer dizer que a equipe precisa planejar e projetar para ser capaz de expor a interface para desenvolvedores no mundo exterior. Sem exceções.
6. Qualquer um que não fizer isso será demitido.

Bezos contratou Rick Dalzell — graduado na West Point Academy e ex-Ranger do Exército — para aplicar essas regras. Bezos determinou outra mudança importante, juntamente com estas regras: cada serviço seria de propriedade de uma equipe multifuncional, que construiria e executaria o serviço ao longo do seu ciclo de vida. Como Werner Vogels, CTO da Amazon, diz: "você construiu, você cuida."

O artigo de Werner Vogel sobre a ida da Amazon para uma arquitetura SOA também é leitura necessária. Veja <http://queue.acm.org/detail.cfm?id=1142065>.

Isto, em conjunto com a regra de que todas as interfaces de serviço devem ser projetadas para serem externalizáveis, apresenta algumas consequências importantes. Como ressalta Vogels, esta

forma de organização de equipes "coloca os desenvolvedores em contato com a operação de seu software no dia a dia. Também os põe em contato diariamente com o cliente. Este ciclo de feedback do cliente é essencial para melhorar a qualidade do serviço".

Assim, cada equipe é efetivamente envolvida em desenvolvimento de produtos, mesmo as pessoas que trabalham com os componentes de infraestrutura que compõem a Amazon Web Services, tais como EC2. É difícil enfatizar suficientemente a importância dessa transição de paradigma de financiamento e entrega, baseado em projeto, para um paradigma baseado no desenvolvimento de produtos.

Um dos maiores problemas que as organizações sofrem ao crescer é manter uma comunicação eficaz entre as pessoas e entre as equipes. Uma vez que você move as pessoas para um andar diferente, um prédio diferente, ou para um fuso horário diferente, a via de comunicação torna-se drasticamente limitada e torna-se muito difícil manter compartilhado o entendimento, a confiança e uma colaboração eficaz. Para controlar este problema, a Amazon estipulou que todas as equipes devem estar em conformidade com a regra de "duas pizzas": a equipe deve ser pequena o suficiente para que duas pizzas possam alimentar a todos — geralmente, cerca de 5 a 10 pessoas.

A limitação de tamanho tem quatro efeitos importantes:

1. Assegura que a equipe tenha uma compreensão clara e compartilhada do sistema que eles estão trabalhando. À medida que as equipes ficam maiores, a quantidade de comunicação necessária para que todos saibam o que está acontecendo escala de uma forma combinatória.
2. Limita a taxa de crescimento do produto ou serviço sendo trabalhado. Ao se limitar o tamanho da equipe, limita-se a velocidade na qual o sistema pode evoluir. Isso também ajuda

- a garantir que a equipe mantenha uma compreensão compartilhada do sistema.
3. Talvez ainda mais importante, descentraliza o poder e cria autonomia, segundo o Princípio de Missão. Cada equipe de duas pizzas (2PT) é a mais autônoma possível. O líder da equipe, trabalhando com a equipe executiva, decidirá sobre as principais métricas de negócios que a equipe será responsável, conhecida como função de aptidão, que torna os critérios globais de avaliação para experimentos da equipe. A equipe então é capaz de agir autonomamente para maximizar essa métrica usando as técnicas que descrevemos no *capítulo 8*.
  4. Liderar uma 2PT é uma maneira para que os funcionários ganhem alguma experiência de liderança em um ambiente onde a falha não tenha consequências catastróficas — o que "ajuda a empresa a atrair e reter talentos empreendedores" (CRAWFORD, 2013).

Um elemento essencial da estratégia da Amazon foi a relação entre a estrutura organizacional de uma 2PT e a abordagem arquitetural de uma arquitetura orientada a serviços.

#### **UMA BREVE INTRODUÇÃO A ARQUITETURAS ORIENTADAS A SERVIÇOS**

Um princípio fundamental de uma arquitetura orientada a serviços (SOA) está na decomposição de sistemas em componentes ou serviços. Cada componente ou serviço fornece uma interface (também conhecida como APIs, *Application Programming Interfaces*, em inglês) para que outros componentes possam se comunicar com ele. Outras partes do sistema — e as equipes que os criam — não precisam conhecer os detalhes de como os componentes ou serviços consumidos são construídos. Em vez disso, eles precisam conhecer apenas a interface. Isto significa também que não

existe necessidade de haver comunicação entre as equipes que usam um serviço ou componente e a equipe que o constrói e mantém. De fato, se a API é suficientemente bem concebida e documentada, nenhuma comunicação é necessária.

Qualquer sistema pode ser decomposto de vários modos. Entender como decompor um sistema é uma arte e, à medida que o sistema evolui, a decomposição ideal provavelmente mudará. Existem duas regras que arquitetos seguem ao decompor sistemas. Primeiro, garantir que a adição de um novo recurso mude apenas um serviço ou um componente. Isso reduz a rotatividade da interface (PARNAS, 1972). Em segundo lugar, evitar a comunicação "exagerada" ou de granularidade fina entre serviços. Serviços "tagarelas" escalam mal e são mais difíceis de personificar para fins de teste.

Todos os sistemas bem desenhados são divididos em componentes. O que diferencia uma arquitetura orientada a serviços é que seus componentes podem ser implantados em produção de forma independente um do outro. Não há mais lançamentos "big bang" de todos os componentes do sistema ao mesmo tempo: cada serviço tem seu próprio cronograma de lançamento independente. Esta abordagem arquitetural é essencial para a entrega contínua (*continuous delivery*) de sistemas de larga escala. A regra mais importante a ser seguida é a seguinte: a equipe que gerencia um serviço tem de garantir que seus consumidores não quebrarão quando uma nova versão for lançada.

Para evitar a sobrecarga de comunicação que pode matar a produtividade à medida que ampliamos o desenvolvimento de software, a Amazon aproveitou uma das mais importantes leis de

desenvolvimento de software, a Lei de Conway: “As organizações refletem os sistemas de design e são limitadas a produzir desenhos que são cópias das estruturas de comunicação dessas organizações”.

Uma maneira de aplicar a lei de Conway consiste em delimitar as fronteiras de uma API às fronteiras da equipe. Desta forma, podemos distribuir equipes por todo o mundo. Contanto que cada serviço seja desenvolvido e gerido por uma equipe única, localizada no mesmo lugar, autônoma e cross-funcional; com isso, uma farta comunicação entre as equipes deixa de ser necessário.

Organizações muitas vezes tentam lutar contra a Lei de Conway. Um exemplo comum é dividir equipes por função, por exemplo, colocando os engenheiros e testadores em locais diferentes (ou, pior ainda, através da terceirização de testadores). Outro exemplo é quando o front-end de um produto é desenvolvido por uma equipe, a lógica de negócios por uma outra equipe, e o banco de dados por uma terceira equipe. Dado que qualquer nova funcionalidade requer mudanças em todos os três pontos, é necessária uma grande quantidade de comunicação entre elas, comunicação esta que é severamente impactada caso as equipes estejam em locais separados. Dividir equipes por função ou camada arquitetural tipicamente leva a uma grande quantidade de retrabalho, desentendimentos sobre especificações, handoffs pobres, e pessoas ociosas aguardando alguém finalizar o trabalho.

A abordagem da Amazon certamente não é a única maneira de se criar velocidade em escala, mas ilustra a importante ligação entre estruturas de comunicação, liderança e arquitetura de sistemas.

## 10.2 CRIE VELOCIDADE EM ESCALA ATRAVÉS DO COMANDO DA MISSÃO

Conforme as organizações crescem, processos informais e canais

de comunicação tornam-se cada vez mais ineficazes na obtenção dos resultados ao nível do sistema que desejamos. Na verdade, é fácil para as pessoas perderem de vista os resultados ao nível do sistema em face do rápido crescimento. Conforme as organizações crescem, elas se movem para o domínio complexo. Em particular, duas características de sistemas adaptativos complexos começam a se tornar importantes. Em primeiro lugar, não há nenhuma perspectiva privilegiada, a partir do qual o sistema como um todo pode ser entendido — nem mesmo a partir do escritório do CEO. Em segundo lugar, ninguém pode esperar compreender mais do que uma pequena parte do todo, dependendo da informação e do contexto disponível para eles.

Portanto, se não formos cuidadosos na forma como crescemos nossa organização, acabaremos com um sistema onde as pessoas otimizam para o que é visível para eles e para o feedback que recebem, que é mais ou menos determinado com quem as pessoas interagem no seu dia a dia. Dessa forma, cada departamento ou divisão otimiza para seu próprio benefício, e não porque as pessoas são estúpidas ou más, mas porque elas simplesmente não têm visibilidade suficiente sobre os efeitos de suas ações sobre a organização mais ampla. Um diagrama simplificado de uma empresa tradicionalmente estruturada é mostrado na figura a seguir.

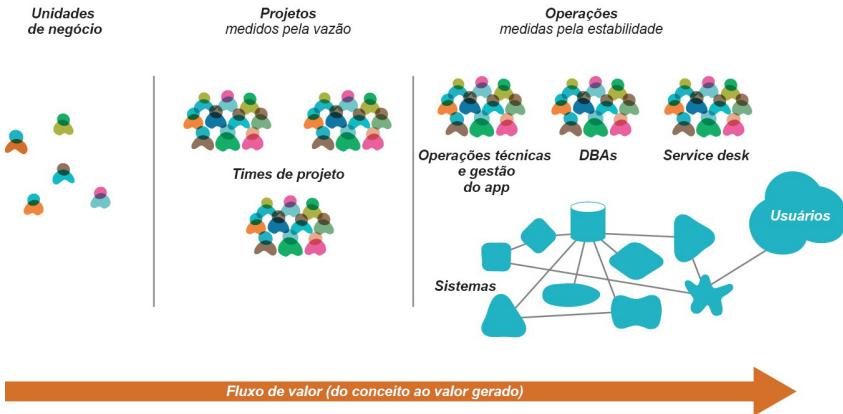


Figura 10.1: Um exemplo de uma corporação tradicional

A chave para se mover rapidamente em escala é criar muitas equipes pequenas, autônomas e descentralizadas, com base no modelo de Comando da Missão. Nas organizações verdadeiramente descentralizadas, seguimos o princípio da subsidiariedade, em que decisões devem ser tomadas pelas pessoas diretamente afetadas por essas decisões. Níveis mais altos de burocracia só devem executar tarefas que não podem ser realizados de forma eficaz a nível local — ou seja, a autoridade de níveis mais altos de burocracia deve ser subsidiária a dos níveis locais. Se tomarmos essa ideia até à sua conclusão lógica, vamos acabar com o que é conhecido como holocracia (<http://holacracy.org/constitution>).

A empresa brasileira Semco é um exemplo de empresa que segue um modelo radicalmente descentralizado (SEMLER, 1995).

Várias grandes organizações bem-sucedidas têm seguido esse princípio por muitos anos, por exemplo, o Gore Company,

Southwest Airlines, e o banco sueco Handelsbanken, todos os quais têm consistentemente demonstrado um desempenho melhor que a média em seus mercados.

Nosso ponto de partida é definir a base organizacional da unidade — uma equipe de até 10 pessoas (seguindo regra de duas pizzas, da Amazon). Depois que se atinge mais de 10 pessoas, a coordenação e a dinâmica do grupo ficam mais difíceis de gerir, e torna-se difícil tomar decisões de consenso e alcançar uma compreensão compartilhada do contexto para todos na equipe.

Em um contexto empresarial, as equipes geralmente colaboram para alcançar as metas de nível de programa, e produtos e serviços de maior dimensão exigirão múltiplas equipes, talvez incluindo pessoas dedicadas a marketing e suporte. Como diz Reed Hastings, o nosso objectivo é criar equipes que são altamente alinhadas, mas fracamente acopladas. Nós garantimos que equipes estão alinhadas usando a Melhoria Kata, tal como descrito nos capítulos 6 e 9 — isto é, tendo iterações no nível de programa, com condições-alvo definidas e tendo equipes colaborando na decisão de como alcançá-los.

Estas são algumas estratégias que empresas têm aplicado com sucesso para criar autonomia para as equipes individuais:

- *Dê as equipes as ferramentas e autoridade para conduzir as alterações até a produção.*

Em empresas como a Amazon, Netflix e Etsy, em muitos casos as equipes não precisam criar bilhetes e ter mudanças revisadas por um conselho consultivo para conseguir implantá-los em produção. De fato, no Etsy esta autoridade é transferida não apenas para as equipes, mas para os engenheiros individuais. É esperado que os engenheiros se consultem mutuamente

antes de submeter alterações e certos tipos de alterações de alto risco. Mas, em geral, espera-se que os engenheiros executem testes automatizados e consultem outros membros de sua equipe para determinar o risco de cada alteração, e confia-se que agem de forma adequada com base nessas informações. O ITIL suporta este conceito na forma alterações padrão. Todas as alterações que são lançadas no escuro (e que, portanto, formam a base de testes A/B) devem ser consideradas alterações padrão. Em contrapartida, é essencial que as equipes sejam responsáveis por apoiar suas alterações. Para mais informações, veja o *capítulo 14*.

- *Certifique-se de que as equipes tenham as pessoas que necessitam para projetar, executar e evoluir os experimentos.*

Cada equipe deve ter a autoridade e as competências necessárias para chegar a uma hipótese, planejar um experimento, colocar um teste A/B em produção, e coletar os dados resultantes. Uma vez que as equipes são pequenas, isso geralmente significa que elas são multifuncionais com uma gama de pessoas: alguns generalistas com uma ou duas especialidades profundas — às vezes conhecidos como "as pessoas em forma de T" (GUEST, 1991) —, juntamente com um pessoal especializado, como um administrador de banco de dados, um especialista em UX e um especialista em domínio. Isso não impede a existência de equipes centralizadas de especialistas que dão apoio sob demanda às equipes de produto.

- *Certifique-se de que as equipes tenham autoridade para*

*escolher suas próprias ferramentas.*

Impor um conjunto de ferramentas à uma equipe é um exemplo de otimização feita mais para as necessidades das áreas de Compras e de Finanças do que para pessoas que realizam o trabalho. As equipes precisam ser livres para escolher suas próprias ferramentas. Uma exceção diz respeito à pilha de tecnologia usada para rodar os serviços em produção. Idealmente, a equipe usará uma plataforma como serviço (*platform as a service* — PaaS) ou serviço de infraestrutura (*infrastructure as a service* — IaaS) fornecido pela equipe interna de TI ou um provedor externo, permitindo que as equipes tenham autonomia para realizar implantações para testes e (quando aplicável) sob demanda até os ambientes de produção através de uma API (e não por meio de um sistema de tickets ou por e-mail). Se tal sistema não existe, ou é inadequado, a equipe deve ser autorizada a escolher a sua própria pilha tecnológica, mas deve estar preparada para atender a quaisquer restrições regulamentares aplicáveis e aguentar os custos de dar suporte ao sistema em produção. Nós cobrimos este tema espinhoso em mais detalhes no *capítulo 14*.

- *Certifique-se de que as equipes não necessitem de aprovação de financiamento para realizar experiências.*

As técnicas descritas neste livro barateiam a realização de experiências, de forma que o financiamento não deve ser uma barreira para testar novas ideias. As equipes não devem precisar de uma aprovação para gastar dinheiro, até um certo limite (por exemplo, um limite mensal ou por transação).

- *Certifique-se de que os líderes se concentrem na implementação do Comando da Missão.*

Em uma organização em crescimento, os líderes devem trabalhar continuamente para simplificar processos e a complexidade dos negócios, para aumentar a eficácia, autonomia e capacidades das menores unidades organizacionais, e fomentar novos líderes dentro dessas unidades.

Um exemplo de como isso pode ser é mostrado na figura adiante. No caso de produtos instalados pelo usuário, aplicações móveis e sistemas embarcados, PaaS/IaaS é usado para fins de teste, mas o processo de lançamento acontece sob demanda em vez de continuamente. Note que esta estrutura não exige uma mudança a quem as pessoas se reportam. As pessoas ainda podem se reportar às linhas funcionais tradicionais (por exemplo, testadores se reportando para um diretor de testes), mesmo se no dia a dia elas trabalhem em equipes multifuncionais. As empresas muitas vezes desperdiçam uma grande quantidade de tempo em reorganizações desnecessárias e disruptivas — quando seria melhor simplesmente colocar na mesma sala as pessoas que trabalham no mesmo produto ou serviço (ou, para os produtos de maiores dimensões, no mesmo andar).

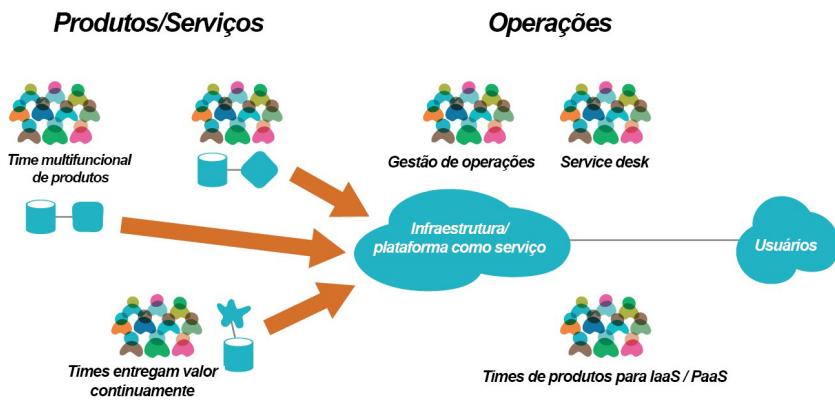


Figura 10.2: Equipes de produto trabalhando juntas, com uma camada de serviços para realizar deployments

#### **CERTIFIQUE-SE DE QUE PRÊMIOS E RECOMPENSAS ESTEJAM ALINHADOS COM UM COMPORTAMENTO DESEJADO**

Embora não seja necessário que estruturas de *report* refletem a organização da equipe, má gestão pode facilmente destruir a colaboração, recompensando pessoas por um comportamento otimizado para a sua função em detrimento dos resultados dos clientes ou a objetivos organizacionais mais amplos. Exemplos disso incluem premiar desenvolvedores por funcionalidades que estão "prontas", mas não estão "prontas para produção", ou premiar testadores para o total de bugs que eles encontram.

Em geral, recompensar pessoas pela sua produção em vez de seus resultados de nível de sistema induz à disfunção e, em qualquer um dos casos, no contexto de um trabalho de conhecimento, bônus ou recompensas monetárias têm demonstrado reduzir o desempenho. Cobrimos o tema de incentivos e da cultura em mais detalhes nos capítulos 1 e 11.

Criar equipes pequenas e autônomas faz com que seja econômico elas trabalharem em pequenos lotes. Quando feito corretamente, essa combinação traz vários benefícios importantes:

- *Aprender mais rápido, melhorar serviço ao cliente, gastar menos tempo em trabalho que não agrega valor.*

Autonomia — combinada a uma arquitetura corporativa que a apoie — reduz as dependências entre as equipes de modo que podem liberar alterações mais rápido. Este é um componente fundamental para permitir que as equipes criem protótipos de novos produtos e funcionalidades para obter feedback dos clientes, executar testes A/B e melhorar o serviço ao cliente, respondendo rapidamente às solicitações dos usuários por melhorias e correções de bugs. Se pudermos aprender rapidamente o que os usuários realmente usam, podemos parar de perder tempo criando coisas que não agregam valor. A métrica mais importante é: o quão rápido podemos aprender?

- *Melhorar a compreensão das necessidades dos usuários.*

Nas organizações nas quais o trabalho se move por meio de silos funcionais, o ciclo de feedback dos usuários aos designers e engenheiros que criam o produto é muitas vezes lento e tem baixa fidelidade. Quando todos na equipe podem construir pequenos experimentos, subi-los em produção e analisar as métricas, toda a equipe entra em contato com os usuários em regime diário.

- *Motivar toda a equipe.*

Quando podemos projetar um experimento ou

disponibilizar uma correção de bug ou melhoria para os usuários e ver os resultados quase que imediatamente, vivemos uma experiência incrivelmente poderosa — uma prova de nossa autonomia, domínio e propósito. Não conhecemos ninguém que já tenha trabalhado dessa maneira que queria voltar à velha maneira de fazer as coisas.

- *Fica mais fácil de calcular lucros e perdas.*

Equipes multifuncionais voltadas para o cliente que possuem um serviço ao longo do seu ciclo de vida tornam muito mais fácil o cálculo de perdas e ganhos para o serviço. O custo do serviço é apenas o custo dos recursos consumidos pela equipe, além de seus salários. Isso nos permite usar números monetários simples para identificar as equipes que geram as maiores margens para a empresa. Note que isto é independente da ideia de estorno interno, que, se implementada dogmaticamente, muitas vezes exige altos níveis de complexidade dos negócios para determinar os custos com uma precisão desnecessária.

Uma coisa é a adotar os princípios da Missão de Comando em uma startup em expansão, mas é outra coisa completamente diferente em uma empresa com uma abordagem mais tradicional e centralizada para gestão e tomada de decisão. O Comando da Missão altera drasticamente a maneira como pensamos sobre gestão, em especial, a gestão de risco, custo e outros resultados ao nível do sistema. Muitas organizações adotam uma abordagem *one-size-fits-all* para gestão de riscos e custos, com processos centralizados de gestão de versão de software (pelo departamento de TI) e orçamento (pelo departamento de finanças).

No Comando da Missão, as equipes têm a autoridade e a responsabilidade de gerenciar custos e os riscos de forma adequada em seu próprio contexto particular. Muda-se papel das finanças, o escritório de gerenciamento de projeto, arquitetos corporativos, as equipes de GRC, e outros grupos centralizados: eles especificam as metas alvo, ajudam a tornar o estado atual transparente e prestam apoio e ferramentas sempre que solicitados, mas não ditam como os custos, processos e riscos são geridos. Discutimos abordagens enxutas para a governança e para finanças na *Parte 4*.

### 10.3 EVOLUINDO SUA ARQUITETURA ATRAVÉS DO PADRÃO DE ESTRANGULAMENTO.

Equipes autônomas trarão pouca diferença para os resultados do cliente se a arquitetura corporativa impede que as equipes façam experimentos e responda rapidamente às necessidades do cliente. Para viabilizar tanto a entrega contínua quanto a descentralização, as equipes precisam ser capazes de liberar mudanças rapidamente e com segurança.

Infelizmente, a realidade é que, em muitas empresas, existem milhares de sistemas rígidos, e é muito difícil fazer alterações em qualquer um deles sem a navegação por uma teia de dependências. Muitas vezes, uma das dependências é um sistema de registro mantido por uma equipe que libera atualizações a cada poucos meses ao custo de um heroísmo significativo.

## **ARQUITETAR PARA ENTREGA CONTÍNUA E ORIENTADO A SERVIÇOS**

Arquitetar para entrega contínua e arquitetura orientada a serviços (SOA) significa evoluir os sistemas que são testáveis até transformá-los em implantáveis. Sistemas testáveis são aqueles para os quais podemos ganhar rapidamente um elevado nível de confiança na certeza do sistema, sem depender de extensos testes manuais em caros ambientes integrados. Sistemas implantáveis são aqueles que são projetados para serem implantados de forma rápida, segura e de forma independente para testes e (no caso de sistemas de web) em ambientes de produção. Estes requisitos "multifuncionais" são tão importantes quanto desempenho, segurança, escalabilidade e confiabilidade, mas são muitas vezes ignorados ou relegados a um status inferior.

Uma resposta comum ao se descobrir preso em uma grande bola de lama é financiar um grande projeto de substituição de sistemas. Tais projetos tipicamente levam meses ou anos antes de entregar qualquer valor para os usuários, e a transição do antigo para o novo sistema é frequentemente realizada no estilo "big bang". Estes projetos também possuem um alto risco de atraso, de ficarem acima do orçamento e de serem cancelados. A rearquitetura de sistemas não deve ser feita como um grande programa de trabalho financiado pelo orçamento da capital. Ela deve ser uma atividade contínua que acontece como parte do processo de desenvolvimento do produto.

A Amazon não substituiu sua monolítica arquitetura Obidos através de um programa de substituição estilo "big bang". Em vez disso, eles mudaram de forma incremental para uma arquitetura

orientada a serviços, enquanto continuavam a oferecer novas funcionalidades, utilizando uma padrão conhecido como "estrangulamento da aplicação". Conforme descrito por Martin Fowler, o padrão envolve a substituição gradual de um sistema através da implementação de novas funcionalidades em uma nova aplicação que está fracamente acoplada ao sistema existente, portando funcionalidade da aplicação original só onde era necessário (FOWLER, 2004; POLS; STEVENSON, 2004). Com o tempo, a antiga aplicação é "estrangulada" — exatamente como uma árvore envolta por um figo tropical estrangulador (figura a seguir).

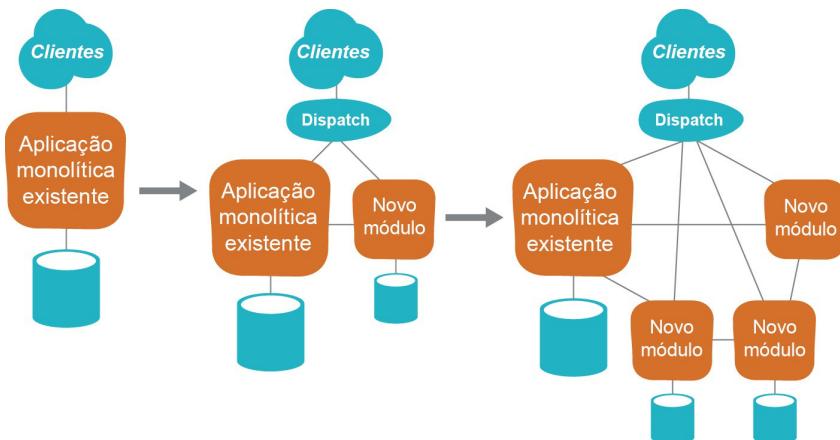


Figura 10.3: A evolução dos estranguladores

Aplicações estranguladoras devem usar os métodos descritos anteriormente neste livro. Existem algumas regras importantes a seguir ao implementar o padrão de estrangulamento:

- *Comece entregando novas funcionalidades, pelo menos em um primeiro momento.*

Sempre encontre maneiras de satisfazer uma necessidade que não é atendida pelo software existente, e priorize recursos usando o custo de atraso dividido

pela duração (CD3), como descrito no *capítulo 7*, para garantir que você entregue a maior quantidade de valor no menor tempo possível.

- *Não tente portar funcionalidade existente a menos que seja para apoiar uma mudança de processos de negócio.*

O maior erro que as pessoas cometem é portar recursos existentes da forma como estão. Isso geralmente significa reproduzir a complexidade criada para servir os processos de negócios como eles eram anos atrás, o que é extremamente dispendioso. Sempre que lhe for solicitado adicionar uma funcionalidade que representa uma mudança para um processo de negócio, vá e observe o processo desde o início e procure maneiras de simplificá-lo antes de implementar o código que vai apoiá-lo. Você descobrirá que muito da complexidade acidental em processos de negócio na verdade vem de ser forçado a usar o velho software que você está substituindo!

- *Entregue algo rápido.*

Faça o lançamento inicial do seu novo aplicativo pequeno o suficiente para que você o tenha implantado e gerando valor em algumas semanas ou alguns meses. Ao construir o primeiro módulo, é difícil — mas essencial — resistir à tentação de adicionar novas funcionalidades. A medida do sucesso para o primeiro lançamento é o quão rápido você pode fazê-lo, não quanta funcionalidade ele tem. Tipicamente, isto é alcançado através da utilização da abordagem de "fatia vertical" em que construímos pequenos incrementos de funcionalidade de ponta a ponta por toda a pilha de tecnologia.

- Projete para a testabilidade e a capacidade de implantação.

Funcionalidade na nova aplicação deve sempre ser construída utilizando boas práticas de desenvolvimento de software: desenvolvimento orientado a testes, integração contínua e baixo acoplamento. Aplicações estranguladoras são uma oportunidade para testar tais práticas, portanto certifique-se de que a equipe trabalhando nisso esteja entusiasmada com esses métodos e tenha experiência suficiente para ter uma boa chance de sucesso.

- Faça com que a arquitetura do novo software seja executado em um PaaS.

Trabalhe com Operações para conduzir a concepção do software em conjunto com a plataforma como um serviço, como descrevemos no *capítulo 14*. Se a equipe de operações não está preparada para fazer isso, trabalhe com eles para garantir que o sistema não aumente a complexidade do ambiente operacional existente.

Existe, naturalmente, um *trade-off* para a migração de uma forma incremental. Geralmente, leva mais tempo para fazer a substituição de forma incremental em comparação com um rearquitetura hipotética "big bang" entregando a mesma funcionalidade. No entanto, uma vez que uma aplicação estranguladora entrega valor ao cliente desde o início, evolui em resposta às necessidades dos clientes, e pode avançar ao seu próprio ritmo, é quase sempre preferível.

A arquitetura corporativa é geralmente conduzida por caros planos *top-down* para "racionalizar" a arquitetura, se deslocam de

sistemas legados para uma plataforma moderna, e eliminam a duplicação para criar uma única fonte de verdade. Frequentemente, o estado final é representado por um diagrama belo que se encaixa em uma única (e grande) folha de papel. No entanto, o estado final é raramente alcançado, porque o ecossistema que a arquitetura serve sempre muda rápido demais e é ainda mais raro para os benefícios prometidos se concretizarem. Normalmente, o que acontece é que as novas estruturas são adicionadas, mas os sistemas que deveriam ser substituídos nunca realmente são desligados, levando a uma complexidade cada vez maior que torna cada vez mais difícil mudar as coisas no futuro.

## CRIE ALINHAMENTO DE ARQUITETURA ATRAVÉS DE CONDIÇÕES-ALVO ESPECIAIS, E NÃO POR PADRONIZAÇÃO OU ÉPICOS ARQUITETURAIS

Nossa experiência é que a padronização em um determinado conjunto de ferramentas ou tecnologia de pilha não é nem necessária nem suficiente para atingir as metas de arquitetura corporativa, tais como permitir que as equipes respondam rapidamente a mudanças de requisitos, criar sistemas de alto desempenho em escala, ou reduzir o risco de invasão ou roubo de dados. Assim como nós conduzimos o produto e a inovação de processos através da Melhoria Kata, também podemos conduzir o alinhamento arquitetural através dele.

Metas de arquitetura — por exemplo, desempenho, disponibilidade e segurança desejados — devem ser abordadas por especificar iterativamente as condições-alvo ao nível do programa. Seguindo o Princípio da Missão, defina uma visão clara das metas de sua arquitetura corporativa sem especificar como elas devem ser alcançadas, e crie um contexto no qual as equipes possam determinar a forma de alcançá-las por meio da experimentação e colaboração. Cobrimos alternativas para a normalização e questões relacionadas com mais detalhes no *capítulo 14*.

Somos muito melhores ao aceitar que sempre estaremos em um estado de mudança, e trabalhando devagar e de forma incremental para reduzir a complexidade através do padrão de aplicação estrangulador. Encontre uma maneira de mensurar a área de superfície dos sistemas destinados a serem descontinuados, e a torne visível para que as equipes possam trabalhar para reduzi-la e, finalmente, eliminar tais sistemas — enquanto eles continuam a

entregar valor aos clientes. Aceitar que evoluir uma arquitetura corporativa — e reduzir complexidade desnecessária — é um processo contínuo e sem fim.

## 10.4 CONCLUSÃO

Mover-se rapidamente em escala requer a execução do Comando da Missão. Uma abordagem comumente usada é criar pequenas equipes que são altamente alinhadas, mas fracamente acopladas. No entanto, dada o forte acoplamento entre a arquitetura de sistemas e os fluxos de comunicação observados por Melvin Conway e codificado na lei que leva seu nome, também precisamos evoluir para uma arquitetura de sistemas que suporte este tipo de organização descentralizada.

Mover-se de um modelo centralizado mais tradicional para o tipo de estrutura descrita neste capítulo é difícil. Devemos proceder lenta e gradualmente. Isso requer mudanças nos processos centralizados correntes — em particular, orçamentação, compras, gestão de riscos, governança e gestão de lançamentos. Estes são discutidos na última parte deste livro, a *Parte 4*.

Ainda que a mudança seja difícil e leve tempo, não devemos ser dissuadidos. A chave é encontrar maneiras de fazer pequenas mudanças incrementais que proporcionam melhores resultados aos clientes — e então, continuar avançando. Assim como nós aplicamos o padrão estrangulador à arquitetura empresarial, também podemos aplicá-lo aos processos e à nossa cultura organizacional — este é o tema do capítulo final neste livro, no *capítulo 15*.

### Questões para os leitores

- Suas equipes podem fazer experiências e alcançar

resultados com os clientes de forma independente, ou elas dependem de outras equipes a fim de fazer qualquer coisa?

- Você pode implantar partes do seu sistema de forma independente entre si, ou você precisa liberar tudo de uma vez?
- Qual é a menor unidade de trabalho possível que você poderia fazer para viabilizar ou a experimentação ou uma implantação independente para uma única equipe ou componente/serviço?
- Como as pessoas do seu time são recompensadas? Isso os encoraja ou desencoraja de colaborar com outras pessoas no seu time ou em outros times?

# Transformar

*“Todo mundo quer mudar o mundo, mas ninguém quer mudar a si mesmo.” — Leo Tolstoy*

Se você chegou até aqui, você já deve ter uma boa noção de como aplicar conceitos e princípios Lean para fazer grandes produtos de software, e sobre a importância da estratégia e cultura para permitir a descoberta e exploração de novos negócios. Mas para tirar o máximo proveito de nossos esforços, princípios e conceitos Lean precisam ser escalados ao longo de toda a organização. Só quando isso acontece é que perceberemos todo o valor do trabalho que investimos, investigando novas ideias e explorando aquelas que entregam valor aos clientes.

Nós prontamente compreendemos que estes conceitos funcionam bem para atender as necessidades de ambientes que mudam rapidamente e com concorrência feroz. Contudo, é difícil ampliar os conceitos Lean para melhoria de processo, aplicações COTS e a evolução e apoio de sistemas internos, especialmente sistemas de registro. Relações entre fornecedores e vendedores apresentam um obstáculo a mais. A natureza de nossa relação com fornecedores de soluções proprietárias, especializadas ou customizadas frequentemente inibe a colaboração, o feedback rápido ou pequena mudança incremental. Precisamos procurar fornecedores que estão dispostos a nos tratar como esperamos tratar nossos próprios clientes. Devemos encorajar os fornecedores a nos ouvir, entender o que precisamos e experimentar. Eles devem estar dispostos a embarcar na jornada de melhoria conosco.

Para adicionar mais complexidade a este problema, muitas de nossas abordagens tradicionais para governança, risco e compliance (GRC), gerenciamento financeiro, aquisição, gerenciamento vendedor/fornecedor e recursos humanos (recrutamento, promoção, compensação) criam desperdício e gargalos adicionais. Estes só podem ser eliminados quando toda a organização adota os conceitos Lean e todo mundo trabalha junto na mesma direção.

Tornar uma empresa Lean não é um show de apenas uma pessoa ou departamento. Não funcionará por meio de uma força-tarefa tática especial. Não podemos obrigar que, a partir de agora, todos trabalharão desta maneira e esperar que eles se ajustem ao nosso plano de implementação. A verdadeira transformação enxuta é o resultado de líderes corajosos e comprometidos que encorajam e disponibilizam o pensamento enxuto para que se propague ao longo de toda a organização — e não apenas em produtos voltados ao cliente. Os que estão no topo devem ser modelo para todos. Eles precisam deixar de lado egos, ouvir e respeitar opiniões contrárias e construir uma relação de confiança em todos os níveis da organização. Isso é essencial para novos líderes emergirem e para práticas e conceitos Lean se entremarem na cultura da organização.

As pessoas devem se sentir empoderadas para tomar decisões que envolvam riscos e para tentar novas ideias, ao mesmo tempo em que reconhecem suas responsabilidades com os clientes e mantêm-se alinhadas com a estratégia global da organização. Como líderes, precisamos definir limitações e contexto para todos, mas também garantir que eles não são indevidamente restritivos. Quando todos estão unidos na busca de um propósito comum, temos empatia com nossos clientes e colocamos suas necessidades na frente, a maioria das pessoas pode descobrir quais riscos são aceitáveis e quais não são.

O conflito surge quando nossos valores defendidos não

correspondem à nossa prática. É aqui que modelar o comportamento que queremos ver em todos é mais importante. Não há fórmulas, instruções ou rituais que funcionarão para todos. Cada um de nós precisa de um tempo por dia, talvez até várias vezes por dia, para refletir sobre nossas próprias ações e decidir se elas correspondem aos valores que declaramos e trabalham para nos levar na direção certa.

Uma mentalidade enxuta não pode prosperar em uma organização com um estilo de gerenciamento centralizado, do tipo comando-e-controle. Mesmo assim, ainda precisamos manter a visibilidade e a transparência no que todo mundo está fazendo. Não é fácil para grandes organizações encontrar esse equilíbrio, e devemos admitir que ajustes constantes serão necessários. Muitas pessoas dentro da organização vão perceber essa mudança cultural como ameaçadora e vão rejeitá-la. Comando e controle é fácil; eu sigo as regras e, se não funcionar, não é minha culpa. Contudo, se você me pedir para tomar decisões e me responsabilizar por elas, alguém pode me responsabilizar pelos erros que provavelmente vou cometer. Me dê comando e controle!

Se somos bem-sucedidos ao criar uma empresa enxuta, teremos falhas e revezes, em todos os níveis e com todas as pessoas. Se não, isso significa que não criamos uma cultura de alta confiança e alta performance, e continuamos a julgar nossa performance por métricas de vaidade, e não por resultados reais. Nunca teremos a cultura de uma organização de aprendizado se não nos é permitido errar e piorar em algo antes de melhorarmos.

Nesta parte, concentraremos nossa discussão em como buscar trabalho de transformação sem-fim para a empresa. Abordaremos algumas das áreas mais comuns nas quais vimos uma discrepância entre conceitos enxutos e os princípios, práticas e processos de gerenciamento e liderança predominantes. Essas lacunas são

reveladas quando enfrentamos obstáculos que nos impedem de fazer melhor ao entregar valor para os clientes, ou quando nos são roubadas a satisfação e a realização em nosso trabalho diário. Nossa esperança é que os leitores se inspirarão para encontrar maneiras de superar esses obstáculos e compartilhar seus sucessos — e falhas — com outros.

# CULTIVE UMA CULTURA DE INOVAÇÃO

*“A capacidade de nossa empresa em ser competitiva e sobreviver não depende tanto das soluções em si, mas na capacidade das pessoas na organização de compreender um situação e desenvolver soluções.” — Mike Rother*

*“Nós agora aceitamos o fato de que o aprendizado é um processo vitalício de manter-se a par da mudança. A tarefa mais urgente é ensinar às pessoas como aprender.” — Peter Drucker*

*“Sabe, eu sou totalmente a favor do progresso. É à mudança que eu tenho ressalva.” — Mark Twain*

A cultura é o fator mais crítico para a capacidade de uma organização adaptar-se ao ambiente em transformação. Entretanto, por ser intangível, é difícil analisar e ainda mais modificar a cultura. Toda organização tem sua cultura ímpar, há "tantas culturas bem sucedidas quanto há empresas bem-sucedidas" (The Economist, 410, no. 8869, p. 72). No *capítulo 1*, apresentamos as características de uma cultura generativa, de alto de desempenho. Neste capítulo, discutiremos como entender a cultura de sua organização e o que é possível fazer para transformá-la.

Em todas as organizações, a cultura está em constante transformação. Novos funcionários e líderes chegam, pessoas saem, estratégias e produtos evoluem e morrem, e o mercado muda constantemente. A pergunta mais importante: é possível, de forma consciente, fazer evoluir a cultura organizacional para responder a estas mudanças no ambiente?

Para entender como influenciar a cultura organizacional, precisamos compreender seus fundamentos. Apresentamos um modelo de cultura organizacional e discutimos como medi-lo. Seguimos com estratégias para dar o pontapé inicial da transformação organizacional, com o objetivo de fazer com que estas estratégias sejam autossustentáveis. Por fim, examinamos a relação entre indivíduos e organizações, e discutimos como contratar e reter pessoas "boas".

## 11.1 MODELE E MENSURE SUA CULTURA

*"CEOs podem falar e falar sobre cultura, mas os funcionários sabem quem são os babacas." — Jack Welch*

No *The Corporate Culture Survival Guide*, Schein define cultura como "um padrão de pressupostos tácitos compartilhados que foi aprendido por um grupo ao resolver seus problemas de adaptação externa e integração interna, que funcionou bem o suficiente para ser considerado válido e, portanto, ser ensinado a novos membros como a forma correta de perceber, pensar e sentir em relação a estes problemas" (SCHEIN, 2009). O termo "tácitos" da definição é importante — e é o que torna a cultura tão intangível.

Shanley Kane, autor de *Your Startup Is Broken*, oferece outra

perspectiva ao comentar que “nossa verdadeira cultura é feita *primariamente* pelas coisas que ninguém diz. [...] Cultura diz respeito à dinâmica de poder, prioridades e crenças não ditas, mitologias, conflitos, normas sociais, criação de "panelinhas" e distribuição de recursos e controle dentro de empresas” (KANE, 2014).

Apesar de intangível, a cultura é mensurável, e há uma vasta bibliografia dedicada a exatamente esta tarefa. É claro que toda metodologia é baseada em um modelo subjacente e que todos os modelos são, em alguma medida, limitados. Ainda assim, tais medições são importantes como modo de tornar visível a cultura e encorajar as pessoas a dar atenção a ela. Aqui estão alguns exemplos de projetos executados para medir cultura:

- Karen E. Watkins e Victoria J. Marsick desenvolveu as *Dimensions of the Learning Organization Questionnaire* (DLOQ), pesquisa fartamente estudada na literatura acadêmica. Você pode responder gratuitamente o questionário em: <http://www.partnersforlearning.com/instructions.html>.
- A pesquisa Q12, do Gallup, é baseada no que eles acreditam ser "as 12 únicas perguntas que importam" para medir a satisfação de funcionários. Você encontra as perguntas e outras informações em: <http://q12.gallup.com/>.
- No *capítulo 1*, discutimos como o *2014 State of DevOps Report* mede tanto a satisfação no trabalho quanto a cultura (usando o modelo de Westrum) e o impacto das duas coisas no desempenho organizacional. A análise demonstrou que o modelo de Westrum previu tanto a satisfação no trabalho quanto o desempenho organizacional no contexto do trabalho do conhecimento (VELASQUEZ, 2014).

### **PRATICIDADES EM RODAR PESQUISAS DE CULTURA**

Seja usando um serviço ou criando sua própria pesquisa, tenha cuidado com a quantidade de informação a ser coletada. Para obter respostas sinceras, não peça às pessoas informações que possam servir para identificá-las. Apresente somente os resultados agregados. Pode ser útil coletar certas informações demográficas para analisar, por exemplo, se os resultados variam conforme o gênero ou o papel. Faça isso apenas se o número de respondentes for grande o suficiente para garantir o anonimato.

Tenha atenção aos modos como a informação pode ter um efeito nocivo aos respondentes. Em uma grande empresa, a reação dos gestores aos resultados ruins de uma pesquisa no departamento foi demandar seus próprios relatórios, retratando-os de modo mais favorável, na oportunidade seguinte.

Desvincule pesquisas de cultura das revisões de salário e desempenho. Faça com que os resultados agregados estejam disponíveis para todos os funcionários e certifique-se de que os executivos agendem reuniões para discutir os resultados e planejar os passos posteriores. Rode pesquisas a cada ano ou a cada semestre para estabelecer um marco para medição e comparação ao longo do tempo.

Avaliar a cultura organizacional e tornar os problemas visíveis é o primeiro passo. Em seguida, é preciso investigar por que uma cultura é como é. Para este estudo, o modelo de Schein é útil. Ele separa a cultura em três camadas: artefatos, valores e pressupostos subjacentes (figura a seguir).

**Artefatos:** aspectos visíveis e observáveis da cultura como estrutura organizacional e processos, e como as pessoas se vestem e se comportam. Difícil de decifrar.



**Valores expostos:** estratégias, objetivos, filosofias, como aquelas frases com os valores da empresa expostas em websites e estampadas em seu lobby.



**Suposições subjacentes:** crenças, percepções, pensamentos, sentimentos (tudo que governa valores e comportamento).

Figura 11.1: Camadas de cultura organizacional

Discrepâncias entre os valores e comportamentos observados em uma organização são comuns. Comportamentos observados são indicadores melhores dos reais valores. Quem é recompensado por quais comportamentos? Quem é contratado, promovido ou demitido? Para compreender a natureza e fonte dos valores reais é preciso aprofundar-se ao nível dos pressupostos subjacentes. Este nível é difícil de acessar, mas é o mais importante de se compreender.

Schein apresenta uma tipologia exaustiva de pressupostos tácitos, dos quais os mais importantes são as crenças de líderes e gestores sobre os funcionários. Em seu clássico sobre gestão *The*

*Human Side of Enterprise*, Douglas McGregor descreve dois conjuntos opostos de crenças que ele observou em gestores, ele chama esses conjuntos de Teoria X e Teoria Y. Gestores que adotam os pressupostos da Teoria X acreditam que as pessoas são preguiçosas e sem ambição, e que elas valorizam mais a estabilidade no emprego do que a responsabilidade. Técnicas de motivação (cenoura-e-vara) são as mais efetivas para lidar com funcionários.

Uma sofisticada e divertida de uma organização com Teoria X, baseada no trabalho de Ricky Gervais, está disponível no seguinte link: <http://bit.ly/1v71WJq>.

Em contraste, gestores Teoria Y acreditam que funcionários poderiam e associariam seus próprios objetivos aos da organização, delegariam mais, operariam mais como professores e treinadores e auxiliariam os funcionários a criar incentivos e controles que eles mesmos monitorariam (SCHEIN, 2009, p. 64).

Como vimos no *capítulo 1*, motivadores extrínsecos, como bônus, são efetivos no mundo taylorista da rotina e trabalho mecânico. Entretanto, em contextos de trabalho baseado em conhecimento, eles chegam a reduzir o desempenho. Pessoas dedicadas a trabalhos não rotineiros são motivadas por fatores intrínsecos, sintetizados por Daniel Pink como "1. Autonomia — o desejo de direcionar nossas próprias vidas; 2. Maestria — a urgência de tornar-se cada vez melhor em algo que importa; 3. Propósito — a vontade de fazermos o que fazemos a serviço de algo maior do que nos mesmos".

Pink também faz referência a diversos estudos que demonstram de modo definitivo que a motivação extrínseca diminui o desempenho do trabalho baseado em conhecimento. Veja mais em: <http://www.danpink.com/drive-the-summaries>.

O que é particularmente problemático é que ambos os tipos de gestor acreditam que seu estilo é o melhor, pois geram respostas comportamentais que se alinham ao próprio estilo. Pessoas cuja estratégia de gestão corresponde à Teoria X acabam por ter funcionários passivos, resistentes à mudança, pouco dispostos a aceitar responsabilidades e que “**almejam benefícios econômicos**” (MCGREGOR, 1995, p. 42.). Trata-se de uma reação racional à não satisfação de principais necessidades pelo trabalho. O trabalho passa a ser algo a ser suportado em troca do salário.

Em uma organização cujos líderes compartilham dos pressupostos da Teoria Y, o trabalho destes é “criar condições para que os membros da organização atinjam seus próprios objetivos melhor quando dirigem seus esforços em direção ao sucesso da empresa” (MCGREGOR, 1995, p. 49), gerando valores para os clientes e a organização enquanto desenvolvem as próprias habilidades. Enquanto líderes e gestores com atitudes da Teoria X não trabalharem para adotar a mentalidade da Teoria Y, e demonstram isso de forma consistente através de suas ações, não serão capazes de provocar mudanças perceptíveis nos comportamentos das pessoas. A história do NUMMI, descrita no *capítulo 1*, é um bom exemplo de mudança de mentalidade e comportamento.

É difícil mudar a cultura através de um projeto. Como diz Schein, “A cultura é muito estável e difícil de transformar, pois

representa o aprendizado acumulado de um grupo — o modo de pensar, sentir e perceber o mundo que fizeram o grupo ser bem-sucedido." (SCHEIN, 2009, p. 27–28).

## 11.2 TRANSFORME SUA CULTURA

Na revolucionária obra *Pedagogia do Oprimido*, publicada em 1970, Paulo Freire descreve aquele que ainda é hoje modelo predominante de ensino. Neste modelo, estudantes são vistos como "contas bancárias" zeradas a serem preenchidas pelos professores — não como participantes com algo a dizer sobre o que e o como aprendem. Este modelo não é projetado para habilitar estudantes a aprender — particularmente aprender a pensar por si mesmos. Ao contrário, são projetados para controlar o processo de aprendizado, o acesso dos estudantes à informação e sua capacidade de analisá-la criticamente. Desta forma, o sistema educacional perpetua as estruturas sociais e as hierarquias de poder existentes.

De modo semelhante, a maior parte das empresas parece tratar seus funcionários como contas bancárias recheadas cujas habilidades e conhecimento podem sacar a serviço dos objetivos da empresa. É isto que está implícito quando nos referimos a funcionários como "recursos" e nos perguntamos como aumentar a taxa de utilização e produtividade, sem nos atentarmos ao desenvolvimento pessoal. Este tipo de comportamento é indicativo de um ambiente em que os funcionários existem antes de tudo como provedores de esforço, não como participantes ativos na criação de valor. Em contraste, organizações de alto desempenho são efetivas tanto em desenvolver quanto em aproveitar as habilidades únicas de suas pessoas.

O conceito chave aqui é a ideia de que trabalhadores são recursos fungíveis — isto é, essencialmente intercambiáveis. Sempre que se ouvem referências às pessoas como "recursos", é isto que está implícito.

Organizações que agem como se funcionários fossem como contas bancárias tendem a abordar a mudança de modo transacional. Esta abordagem tão comum e falha inclui financiar um programa de transformação que se espera que "conserte" a organização para que atenda ao propósito. A transformação organizacional é tratada como um produto — vendido por consultores, pago pela liderança e consumido, conforme orientação, pelo restante da organização.

Estes programas de transformação comumente concentram-se em reorganização de equipes e estruturas de subordinação, envio de funcionários para cursos de curta duração e implantação de ferramentas e metodologias em toda a organização. Tais estratégias costumam não funcionar, pois não são efetivas em modificar os padrões de comportamento das pessoas. Como aponta Mike Rother, em Toyota Kata, "o que é definidor não é a forma da organização, mas o modo como as pessoas agem e reagem" (ROTHER, 2010, p. 236). Isto é determinado antes de tudo pelas ações da liderança e dos gestores. Para tomar alguns exemplos: as pessoas têm autonomia para agir e recebem a confiança para assumir riscos? Falhas são punidas ou levam à investigação e a melhorias em nossos sistemas? A comunicação transversal é recompensada e desencorajada?

Iniciamos este livro com a discussão do caso NUMMI, em que uma organização quebrada foi restaurada sob um novo paradigma

de liderança e gestão. Mesmo tendo recontratado as mesmas pessoas, a NUMMI alcançou níveis extraordinários de qualidade e produtividade, e reduziu custos. Em artigo publicado no periódico *MIT Sloan Management Review*, John Shook (2010), primeiro funcionário estadunidense da Cidade da Toyota, reflete sobre a conquista da transformação cultural:

*“O que a experiência na NUMMI me ensinou de muito poderoso é que o modo de transformar a cultura é começar não por mudar como as pessoas pensam, mas sim como elas se comportam — aquilo que fazem. Aqueles que estão tentando transformar a cultura de nossas organizações precisam definir aquilo que queremos fazer, os modos como queremos que nós mesmos e os demais se comportem, oferecer treinamento e então fazer o que for necessário para reforçar tais comportamentos. A cultura mudará como resultado [...]. O que transformou a cultura na NUMMI não foi uma noção abstrata de “envolvimento dos funcionários” ou uma “organização de aprendizagem”, nem mesmo uma “cultura”. O que transformou a cultura foi dar aos funcionários os meios para que pudessem trabalhar de forma bem-sucedida. Foi comunicar a eles, de forma clara, quais eram suas tarefas e prover os treinamentos e ferramentas que os permitiram desempenhar tais tarefas com sucesso”.*

Shook nos dá sua própria interpretação do modelo de Schein, mostrando o contraste entre a abordagem mais comum à mudança cultural e a abordagem adotada na NUMII, na figura adiante.

A NUMMI teve uma vantagem para alcançar a transformação cultural. Toda a força de trabalho era recém-contratada — com muitos trabalhadores recém-demitidos da Freemont Assembly. É difícil conquistar uma mudança sistêmica e sustentada sem uma crise. Em seu livro *The Corporate Culture Survival Guide*, Schein se pergunta se a crise é uma condição necessária para transformações bem-sucedidas. Sua resposta é seguinte: "Porque humanos evitam a

imprevisibilidade, e logo criam culturas, o argumento fundamental para o aprendizado adulto é que, de fato, nós precisamos de algum estímulo novo que perturbe o equilíbrio. O melhor modo de pensar em tal estímulo é como uma desconfirmação: algo de inesperado é percebido ou sentido e isto perturba algumas de nossas crenças e pré-concepções [...] a desconfirmação cria a ansiedade da sobrevivência — algo ruim acontecerá se não mudarmos — ou culpa — compreendemos que não estamos alcançando nossos próprios ideais ou objetivos" (SCHEIN, 2009, p. 106).

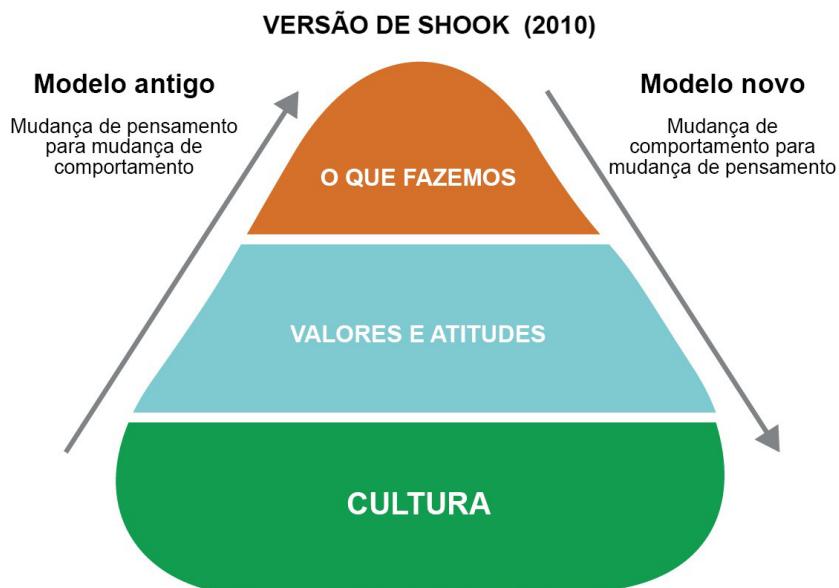


Figura 11.2: Abordagem antiga e nova à mudança cultural (SHOOK, 2010) — publicado na MIT Sloan Management Review/Massachusetts Institute of Technology, todos os direitos reservados, distribuído por Tribune Content Agency, LLC

A desconfirmação pode vir naturalmente de diversas origens com potencial de ameaçar nossa sobrevivência: econômica, política, tecnológica, jurídica, moral ou o simples entendimento de que não estamos alcançando nosso propósito. Uma causa comum de desconfirmação não planejada é a ação de líderes em contradição com os valores por eles propagados. É possível também gerar

desconfirmação de modo controlado através de iniciativas conjuntas, atividades planejadas da liderança ou criando uma crise artificial.

Quando as pessoas aceitam a necessidade de mudança, confrontam-se com o medo de não conseguir aprender as novas habilidades e comportamentos requeridos, ou então de perder status ou alguma parte de sua identidade — a este fenômeno Schein dá nome de *ansiedade do aprendizado*.

Schein postula que para uma mudança bem sucedida, a ansiedade da sobrevivência deve ser maior do que a ansiedade do aprendizado, e para chegar a isto, a "ansiedade do aprendizado deve ser reduzida, em vez de aumentar a ansiedade do aprendizado" (SCHEIN, 2009, p. 114). Muitos líderes e gestores cometem o erro de tentar realizar a mudança através do aumento da ansiedade da sobrevivência. Isso cria um ambiente de medo que, por sua vez, resultam em muita energia gasta em desviar da culpa, evitar responsabilidade e em jogadas políticas.

A ferramenta sistemática mais poderosa para reduzir a ansiedade da sobrevivência que encontramos é a Melhoria Kata, descrito no *capítulo 6*. Um ambiente em que erros são aceitos como oportunidades de aprendizado para construção de sistemas e processos que reduzam o impacto de erros futuros é essencial para criar uma cultura de alto desempenho.

## **Faça com que seja seguro falhar**

A atitude de sua organização diante de falhas — seja um esforço de transformação ou simplesmente uma decisão — é fator crítico para a criação de uma organização adaptativa, resiliente. O acadêmico Russell L. Ackoff, estudioso da teoria das organizações, observou que "é nossa forma de lidar com erros que leva a uma estabilidade que, por sua vez, evita mudanças significativas". Se as

pessoas ouvem que cometer erros é ruim e se são punidas por cometê-los, a consequência inevitável é que passarão a evitar decisões arriscadas (<http://youtu.be/MzS5V5-0VsA?t=6m>).

Em um sistema complexo e adaptativo, como uma empresa, ninguém tem informação perfeita. Cada decisão acarreta consequências não desejadas, cujas causas são praticamente impossíveis de detectar *a priori*, ainda que fiquem muito claras quando observadas em retrospecto. Sempre que pareça que uma pessoa foi responsável por determinado resultado, devemos ser honestos e nos perguntar: "se eu estivesse na mesma situação, é plausível que eu viesse a tomar as mesmas decisões?". Normalmente, a resposta é "sim".

Em lugar de punir erros, devemos: assegurar que as pessoas tenham acesso à informação necessária para tomar decisões efetivas; buscar formas de limitar o impacto negativo das decisões; e sermos disciplinados em aprender com os erros. Por exemplo, como gestores e líderes da sua organização reagem a falhas? Eles promovem a culpabilização, a justiça ou a investigação?

Uma prática comum em organizações como culturas de alto desempenho é uma cerimônia pós-morte sem culpa após cada incidente ou acidente. O objetivo da cerimônia pós-morte é aprimorar o sistema para que as pessoas estejam mais bem informadas e equipadas em situações futuras similares, assim diminuindo o impacto negativo.

No início de toda cerimônia de pós-morte, cada participante deve ler em voz alta as palavras da Diretiva Primária das Retrospectivas: "Independentemente do que descubramos, compreendemos e acreditamos verdadeiramente que todos fizeram o melhor trabalho possível dado o conhecimento que tinham no momento, suas competências e habilidade, os recursos disponíveis e a situação em curso" (KERTH, 2001). Uma cerimônia de pós-morte

deve gerar:

Estes dois pontos foram levantados por John Allspaw (2013). Se você tiver interesse em saber como a Knight Capital perdeu 460 milhões de dólares em 30 minutos, vale a pena ler a publicação integralmente.

- Uma descrição e explicação sobre o incidente ocorrido da perspectiva das pessoas envolvidas e das afetadas, incluindo uma linha do tempo de eventos e a lista dos fatores incidentes.
- Artefatos (recomendações, remediações, listas de verificação, atualização de guias etc.) para melhor prevenção, detecção e reação a eventos futuros similares, melhorando a forma de lidar com eles.

Cerimônias de pós-morte não devem buscar a identificação de uma causa raiz. A ideia de que um evento único pode ser identificado como causa de uma falha é um erro de compreensão sobre a natureza de sistemas adaptativos complexos. Conforme indicam os especialistas em segurança Sidney Dekker, Erik Hollnagel, David Woods e Richard Cook (2008, p. 6):

*“Nosso entendimento sobre como acidentes acontecem passou um desenvolvimento dramático no último século. Inicialmente, acidentes eram vistos como a conclusão de uma sequência de eventos (que incluíam “erros humanos” como causa ou fator de contribuição). Isto agora tem sido progressivamente substituído por uma visão sistêmica, segundo a qual acidentes emergem da complexidade das atividades das pessoas em um contexto organizacional e técnico. Estas atividades normalmente são dedicadas a prevenir acidentes, mas tocam também*

*outros objetivos (vazão, produção, eficiência, controle de custos), o que significa que conflitos entre objetivos podem surgir, sempre sob a pressão de recursos escassos (por exemplo, tempo, dinheiro, conhecimento). Acidentes emergem de uma conjunção de condições e acontecimentos que estão, em geral, associadas à busca pelo sucesso, mas cuja combinação específica torna-se, ao contrário, um gatilho para falha”.*

Cada falha é resultado de múltiplas coisas que dão errado — frequentemente de modo invisível (Dekker menciona sistemas adaptativos complexos "flutuando em direção à falha") — há um guia rápido para falhas em sistemas complexos disponível em <http://bit.ly/1F7O3Mg>. Cada cerimônia de pós-morte deve ter como resultado diversas ideias de melhoria incremental. Devemos também agendar o acompanhamento para avaliar se estas melhorias foram efetivas, idealmente através de um exercício de simulação de falha similar, conforme descrito no *capítulo 14*.

### 11.3 NÃO HÁ ESCASSEZ DE TALENTOS

O título desta seção é emprestado de uma apresentação de Andrew Shafer. Veja em [https://www.youtube.com/watch?v=P\\_sWGI7MzhU](https://www.youtube.com/watch?v=P_sWGI7MzhU).

Na indústria de tecnologia, é comum ouvir falar sobre "escassez de talentos" e a dificuldade de encontrar "pessoas boas". Nesta seção, desconstruiremos os pressupostos que embasam este tipo de afirmação. Analisaremos o que queremos dizer com "pessoas boas" olhando o caso particular de uma função — engenheiro de software — e então ampliar para o caso geral.

É crença comum que, entre o melhor e o pior engenheiro, a diferença é de ordem de grandeza (SACKMAN; ERIKSON; GRANT, 1968; MCCONNELL, 2011). Na verdade, o número 10x é, para ser gentil, "parcamente amparado em evidências empíricas" (BOSSAVIT, 2013). Entretanto, o que se depreende de um mergulho aprofundado é que o debate é, em realidade, sobre a validade ou utilidade de medidas de produtividade individual no contexto de uma organização.

A produtividade individual é normalmente medida pelo tempo que uma tarefa padrão leva para ser realizada em condições controladas. Esta abordagem é embasada por uma perspectiva taylorista do trabalho, em que gestores definem tarefas a serem realizadas e trabalhadores tentam completá-las o mais rapidamente possível. Assim, métricas antiquadas, tais como linhas de código por dia e número de horas trabalhadas, são usadas para medir a produtividade individual de engenheiros de software.

Os defeitos destas métricas tornam-se óbvios se considerarmos os resultados ideais: o menor número possível de linhas de código para a resolução de um problema e a criação de processos comuns e interações com o cliente simplificados para redução da complexidade em sistemas de TI. As pessoas mais produtivas são aquelas que encontram maneiras inteligentes de evitar escrever código.

Em muitas organizações, é fútil preocupar-se indevidamente com variações entre indivíduos. Se há algo a aprender do caso da NUMMI, descrito no *capítulo 1*, é que cultura organizacional e liderança tornam irrelevantes as diferenças entre indivíduos. Como diz o jornalista Malcolm Gladwell, "O mito do talento presume que as pessoas fazem as organizações serem inteligente. Na maior parte dos casos, observa-se o contrário. [...] Nossas vidas são claramente beneficiadas pelo brilhantismo individual. Grupos não escrevem

romances e não foi um comitê que criou a teoria da relatividade. Empresas, porém, funcionam diferente. Elas não simplesmente criam, mas executam, competem e coordenam esforços de muitas pessoas. As empresas mais bem-sucedidas nestas tarefas são aquelas em que o sistema é a estrela". Como observou W. Edwards Deming, "Um sistema ruim sempre derrota uma pessoa boa".

Nossa taxa de compreensão e solução de problemas complexos — a competência chave para a qual ainda precisamos mais de pessoas do que de máquinas — é determinada tanto pelo ambiente quanto por nossas próprias competências e habilidade. Não se podem culpar as pessoas por falhas de aprendizado e solução de problemas quando suas oportunidades estão limitadas por diversos motivos: silos organizacionais que isolam os trabalhadores uns dos outros e dos clientes, por tempos de ciclo muito longos.

Dado que a cultura de uma organização tem um efeito tão dominante no desempenho dos indivíduos, devemos sequer nos preocupar sobre as específicas habilidades e atitudes dos indivíduos? Em vez de ter uma visão "conta bancária" que incide sobre as capacidades existentes das pessoas, é mais importante considerar sua capacidade de adquirir novas competências — particularmente no campo da tecnologia, onde o conhecimento e as habilidades úteis mudam rapidamente.

Carol Dweck, professora de psicologia na Universidade de Stanford, passou anos pesquisando a psicologia da aprendizagem, desenvolvimento e motivação. Sua pesquisa revela que há uma maneira de julgar o quanto bom as pessoas serão em aprender novas habilidades. Dweck descobriu que nossa capacidade de aprender é determinada por nossas crenças acerca da questão: a capacidade é inata, ou pode ser aprendida? Podemos observar, com base no comportamento das pessoas, onde caem em um contínuo entre dois extremos (MOREHEAD, 2012):

*“Em uma mentalidade fixa, os alunos acreditam que suas habilidades básicas, sua inteligência e seus talentos são apenas características fixas. Eles têm uma certa quantidade e isso é o que é, e depois seu objetivo torna-se parecer inteligente o tempo todo e nunca parecer estúpido. Em uma mentalidade de crescimento, os alunos compreendem que seus talentos e habilidades podem ser desenvolvidas através de esforço, bons ensinamentos e persistência. Eles não necessariamente pensam que todo mundo é igual ou que qualquer um pode ser Einstein, mas eles acreditam que todos podem ficar mais espertos se trabalharem para isso”.*

Dweck mostrou através de uma série de experimentos que a nossa mentalidade determina como decidimos nossos objetivos, como reagimos ao fracasso, quais são as nossas crenças sobre esforço e estratégias, e qual é a nossa atitude para com o sucesso dos outros (figura seguinte). Nossa mentalidade é particularmente importante em termos de nossa atitude perante o fracasso. Pessoas com uma mentalidade fixa temem falhar que eles acreditam tornar suas limitações inatas visíveis para os outros, enquanto aqueles com uma mentalidade de crescimento são menos avessos ao risco por ver o fracasso como uma oportunidade para aprender e desenvolver novas habilidades.

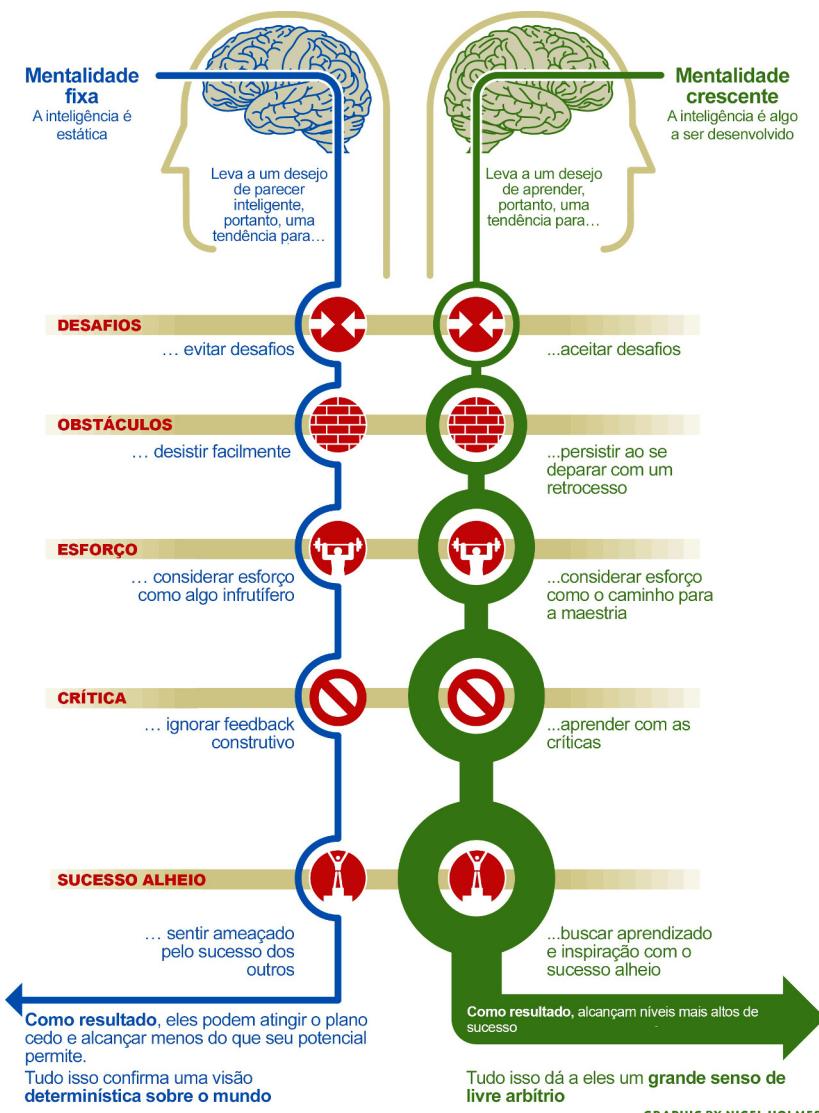


Figura 11.3: As duas mentalidades de Dweck, cortesia de Nigel Holmes

A boa notícia é que podemos mudar nossas crenças, como mostrado por uma das experiências mais interessantes de Dweck. O trabalho de Dweck mostra que, se recompensar as pessoas pelo esforço que elas colocam na resolução dos problemas que eles

achem desafiador, os move para uma mentalidade de crescimento. Se, ao contrário, elogiamos e recompensamos as pessoas pela sua capacidade de empregar suas habilidades existentes, criamos uma mentalidade fixa. Isto tem implicações importantes tanto para os gestores de pessoas e para os departamentos de RH, particularmente no contexto de avaliações de desempenho.

Gladwell (2002) tem um artigo que vale a pena ser lido na íntegra. A pesquisa de Dweck também tem implicações importantes para a forma como criamos nossos filhos, especialmente filhas, que são muitas vezes elogiadas por serem "boas" ou "bonitas", criando uma mentalidade fixa. Esta é apenas uma maneira em que enviesamentos implícitos reforçam-se mutuamente. Veja mais em <http://bit.ly/1zkRLOK>.

Você pode ter certeza de que o comportamento e as atitudes das pessoas em sua organização — da cultura organizacional — afetam a mentalidade dos indivíduos dentro dela e, portanto, sua capacidade de aprender. Dessa forma, a cultura organizacional determina não apenas a produtividade e o desempenho das pessoas que trabalham nela, mas também a sua capacidade de adquirir novas competências, a sua atitude perante o fracasso e a novos desafios, e seus objetivos. Dando às pessoas metas mais difíceis que as obrigue a aprender novas habilidades, oferecendo-lhes apoio, treinamento e tempo de folga para reduzir a ansiedade do aprendizado, e criar uma cultura em que a colaboração é recompensada, e o fracasso leva à reflexão e melhoria em vez da culpa — tudo isso trabalha para instigar uma mentalidade de crescimento nos funcionários e deve ser um objetivo-chave da mudança organizacional. Você pode ter certeza de que o comportamento e as atitudes das pessoas em sua organização — da

cultura organizacional — afetam a mentalidade dos indivíduos dentro dela e, portanto, sua capacidade de aprender. Dessa forma, a cultura organizacional determina não apenas a produtividade e o desempenho das pessoas que trabalham nela, mas também a sua capacidade de adquirir novas competências, a sua atitude perante o fracasso e a novos desafios, e seus objetivos. Dando às pessoas metas mais difíceis que as obrigue a aprender novas habilidades, oferecendo-lhes apoio, treinamento e tempo de folga para reduzir a ansiedade do aprendizado, e criar uma cultura em que a colaboração é recompensada, e o fracasso leva à reflexão e melhoria em vez da culpa — tudo isso trabalha para instigar uma mentalidade de crescimento nos funcionários e deve ser um objetivo-chave da mudança organizacional.

O trabalho de Dweck nos diz que existem de fato "jogadores A" e "jogadores B." Jogadores A são simplesmente pessoas com uma mentalidade de crescimento que, ao adentrarem uma equipe, tentarão descobrir como fazer a equipe ter sucesso, trabalhando para adquirir as habilidades necessárias no processo. Em contraste, as pessoas com uma mentalidade fixa, os verdadeiros Jogadores B, são a maior barreira para a mudança organizacional e melhoria contínua. Estes são o tipo de pessoas que resistem à experimentação dizendo que as abordagens dos outros "não tem como funcionar aqui." Eles também são susceptíveis de contratar pessoas que julgam como piores do que eles, de modo a evitar desafios para o seu estado e identidade.

Enquanto essas pessoas são capazes de mudar a sua mentalidade, eles também podem envenenar tentativas de mudar a cultura, atrapalhando o alto desempenho equipes. Para reduzir a ansiedade do aprendizado durante os esforços de mudança, deve ser amplamente divulgado que o apoio e os recursos estarão disponíveis para ajudar as pessoas a adquirir novas habilidades, que ninguém vai perder o emprego caso estejam dispostos a aprender, e que

aqueles que desejarem sair receberão um generoso pacote de indenização.

Por último, a responsabilidade mais importante dos líderes de uma organização é para com sua cultura, demonstrada pela forma como eles tratam os outros. Por exemplo, Dweck argumenta que enquanto Steve Jobs possuía uma mentalidade de crescimento quando se tratava de suas próprias habilidades, ele tinha uma atitude de mentalidade fixa em relação aos outros: "Ele queria que eles fossem perfeitos e eles viviam com medo de chegar até ele e conseguir sua desaprovação, em vez de sua aprovação" (MOREHEAD, 2012).

### COMO O GOOGLE RECRUTA

O trabalho de Dweck exige que repensem o recrutamento. Nós não devemos contratar pessoas apenas com base das habilidades que elas já possuem. Isto é particularmente míope na indústria de software, em que tecnologia e, dessa forma, as habilidades necessárias, mudam tão rapidamente. Também não deveríamos usar quebra-cabeças ou resultados de testes, que Laszlo Bock, vice-presidente sênior de operações de pessoas no Google, descreve como "inútil como critério de contratação. [...] Eles não preveem coisa alguma" (BRYANT, 2013). O Google tem feito um grande esforço investigativo sobre o que torna um processo de recrutamento eficaz no contexto da tecnologia. Os três principais critérios são (FRIEDMAN, 2014):

- Capacidade de aprendizagem, incluindo a capacidade de "processar na hora" e "juntar pedaços díspares de informação".
- Liderança, "em particular a liderança emergente

em vez da liderança tradicional. Liderança tradicional é: você foi presidente do clube de xadrez? Você era vice-presidente de vendas? Quão rapidamente que você chegou lá? Não estamos nem aí. O que nos interessa é, quando confrontado com um problema e você é um membro de uma equipe, se você, no momento oportuno, intervém e lidera. E tão crítico quanto, você dá um passo atrás e para de liderar, você deixa que alguém lidere? Porque o que é fundamental para ser um líder eficaz neste ambiente é que você tem de estar disposto a ceder o poder".

- Mentalidade. "Gente brilhante e bem sucedida raramente experimenta o fracasso, e assim eles não aprendem como aprender com esse fracasso. [...] Em vez disso, eles cometem o fundamental erro de atribuição, que é se algo de bom acontecer, é porque eu sou um gênio. Se algo ruim acontece, é porque alguém, ou é um idiota, ou eu não obtive os recursos, ou o mercado mudou."

Bock prossegue e observa que as pessoas mais bem sucedidas do Google "terão uma posição feroz. Eles vão discutir como o inferno. Eles vão ser fanáticos sobre o seu ponto de vista. Mas se você disser, aqui está um fato novo, e eles vão dizer, Ah então, isso muda tudo; você está certo". Isso reflete o conselho de Paul Saffo, diretor do Palo Alto Institute for the Future, que diz que "para lidar com um futuro incerto e ainda avançar, as pessoas devem ter opiniões fortes, que são mantidas fragilmente" (SUTTON, 2006).

A estratégia de recrutamento do Google é libertadora, porque

ela enormemente expande o leque de candidatos qualificados. Em vez de procurar "moscas brancas" que tenham exatamente as habilidades e experiência necessárias para um trabalho, devemos olhar para as pessoas que podem rapidamente adquirir as competências necessárias e, em seguida, investir em um ambiente que lhes permita fazê-lo.

## Criando talentos

O problema da "escassez de talentos" é resolvido através da criação de um ambiente no qual as pessoas podem aprender no trabalho e contratar pessoas com uma mentalidade de crescimento. Investir no desenvolvimento dos colaboradores é uma das poucas oportunidades que as empresas têm de criar uma vantagem competitiva sobre startups (os outros são pesquisa e desenvolvimento, e a busca da optionalidade em Horizonte 3, conforme descrito no capítulo 2). Há muitas maneiras nos quais as empresas podem investir nas pessoas:

- *Ajudar os funcionários a criar e atualizar planos de desenvolvimento pessoal.*

Para ajudar os funcionários a assumir o controle do seu próprio desenvolvimento e garantir que os gerentes saibam como ajudá-los, é essencial que eles, seus gestores, e as pessoas que lhes dão um feedback compreendam seus objetivos de carreira. Criar e atualizar periodicamente um plano de desenvolvimento pessoal simples é a base do desenvolvimento do empregado.

- *Separar avaliações de desempenho das avaliações de compensação.*

O objetivo das avaliações de desempenho é proporcionar uma oportunidade para que os funcionários obtenham feedback sobre o seu progresso rumo a seus objetivos de desenvolvimento pessoal, atualizar suas metas e discuti-las com o seu gerente imediato. Acoplar avaliações de desempenho com avaliações de compensação, e particularmente a prática de "classificar" empregados, é baseado em ideias ultrapassadas de motivação extrínseca que incentivam os funcionários a competir entre si em vez de cooperar uns com os outros, e reduz o engajamento dos funcionários.

- *Facilitar o feedback periódico.*

Os funcionários devem compartilhar feedback informal regularmente para se ajudarem mutuamente a irem em direção a seus objetivos pessoais. Feedback positivo é oportuno, projetado para o benefício de quem recebe, e dado com permissão. Em um processo formal (como durante uma avaliação de desempenho, a repreensão oficial, ou entrevista de saída), ninguém deve ouvir um feedback que eles ainda não tenha recebido informalmente.

- *Dar aos funcionários o acesso a fundos de treinamento.*

Os funcionários aprendem através de diferentes canais e devem ter fácil acesso aos fundos que lhes permitam comprar livros, participar de conferências e de treinamento, ou se envolver em outras atividades que os ajudem a irem em direção a seus objetivos de desenvolvimento pessoal. As condições para a despesa devem ser tão liberais quanto possível, dentro das limitações da regulamentação tributária aplicável.

- *Dar aos funcionários tempo para perseguir seus próprios objetivos.*

Muitas organizações inovadoras reservam tempo para as pessoas trabalharem no que quiserem. A 3M tem permitido que os empregados gastem 15% do seu tempo em seus próprios projetos desde 1948. O bloco de notas Post-It é apenas uma das inovações criadas como resultado desta iniciativa ([http://solutions.3m.com/innovation/en\\_US/stories/time-to-think](http://solutions.3m.com/innovation/en_US/stories/time-to-think)). Em 2004, na carta de IPO, Sergey Brin e Larry Page do Google escreveram, "Nós incentivamos nossos funcionários, além de seus projetos regulares, a gastar 20% do seu tempo de trabalho no que eles acho que mais vai beneficiar o Google. Isso lhes dá poder para serem mais criativos e inovadores. Muitos dos nossos significativos avanços tem acontecido dessa maneira. Por exemplo, o AdSense para conteúdo e o Google News foram ambos prototipados nesse 'período de 20%'. A maioria dos projetos mais arriscados fracassam, muitas vezes nos ensinando algo. Outros são bem-sucedidos e se tornam negócios atrativos" (<http://investor.google.com/corporate/2004/ipo-founders-letter.html>).

Norman Bodek conta uma história sobre Taiichi Ohno fechando um armazém em uma subsidiária da Toyota: "Livre-se deste armazém e, em um ano, eu vou voltar e olhar! Quero ver esse armazém transformado em oficina mecânica e eu quero ver todos treinados como mecânicos" (BODEK, 2004, p. 29). Bodek relata que as ordens de Ohno foram seguidas, e em um ano o armazém tinha sido substituído por uma oficina mecânica e os trabalhadores treinados novamente. Em sintonia com a política corporativa japonesa do pós-Segunda Guerra Mundial de proporcionar às

pessoas emprego para toda vida, a Toyota espera treinar pessoas para fazer diferentes tipos de trabalho ao longo de suas carreiras.

Funcionários da Toyota compreendem que parte de seu trabalho é adquirir novas competências. Toyota fornece o treinamento e apoio necessário para isso, removendo uma grande parte da ansiedade de aprendizado que é o mais sério obstáculo à criação de uma organização de aprendizagem e mudança organizacional. Mais importante ainda, quando as pessoas são tratadas com respeito e têm a oportunidade de seguir a autonomia, domínio e propósito, elas tornam-se altamente motivadas para entregar valor. A satisfação dos funcionários no trabalho é o melhor indicador do desempenho organizacional.

## **Elimine o viés oculto**

Outro grande colaborador para a "escassez de talentos" em tecnologia é o grande número de pessoas qualificadas que decidem não entrar no campo ou sair prematuramente. Olhe para suas equipes de tecnologia e perceba que as mulheres, em particular, são muito pouco representadas, assim como as pessoas não brancas nos Estados Unidos e na União Europeia. Dado que "as diferenças sexuais biológicas em aptidão inerente para matemática e ciências são pequenas ou inexistentes" (CECI; WILLIAMS, 2010; MOSS-RACUSIN et al., 2012), e o mesmo vale para as diferenças entre as raças, qual é a causa desta sub-representação?

Uma série de estudos feitos sobre os processos de recrutamento com o objetivo de contratar no mérito mostra que o nosso preconceito implícito de gênero desempenha um papel importante na rejeição de mulheres devidamente qualificadas. Em um estudo realizado em 2012, pesquisadores pegaram 127 professores de biologia, química e física de todo os Estados Unidos e lhes deram material de inscrição de emprego para um estudante de graduação

em ciências se candidatando a um emprego como gerente de laboratório de ciências. Os materiais foram todos idênticos, mas foram aleatoriamente designados a um nome masculino ou feminino. Os participantes foram convidados a avaliar do aluno "sua competência e 'contratabilidade', bem como o valor do salário e da quantidade de orientação que eles ofereceriam ao aluno" (MOSS-RACUSIN et al., 2012). Os resultados estão reproduzidos na figura adiante.

Talvez o mais interessante, ambos os professores homens e mulheres demonstraram o mesmo viés, mostrando que não é intencional ou explícita, mas sim "moldada por preconceitos implícitos ou não intencionais, decorrente de exposição repetida a estereótipos culturais difundidas que retratam as mulheres como menos competentes". Outros estudos têm demonstrado os mesmos efeitos em diferentes domínios, bem como um efeito semelhante em relação à raça (BERTRAND, M.; MULLAINATHAN, 2004; NICOT, 2008; CÉDIEY; FORONI; GARNER, 2008).

Veja, por exemplo, Bertrand e Mullainathan (2004) e Nicot (2008) que faz referência a Cédiey, Foroni e Garner (2008) no viés implícito de raça, e <http://bit.ly/1v72MG7> que faz referência a Goldin e Rouse (2000) sobre o efeito das audições cegas no aumento do número de mulheres em orquestras.

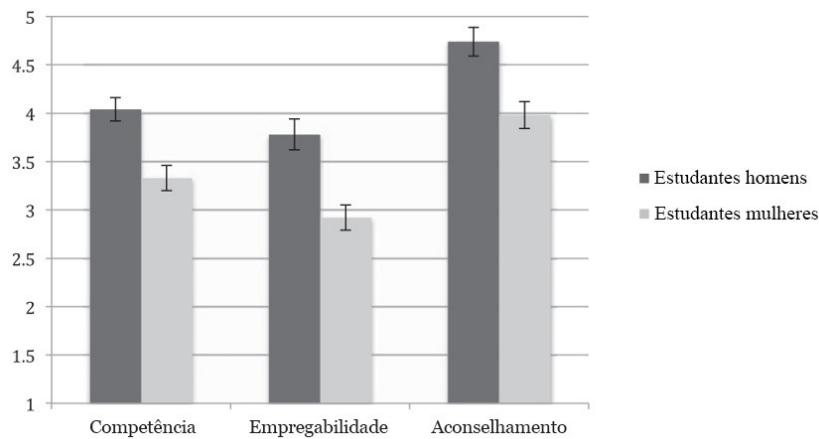


Figura 11.4: Os efeitos do preconceito implícito de gênero na contratação — Parte I

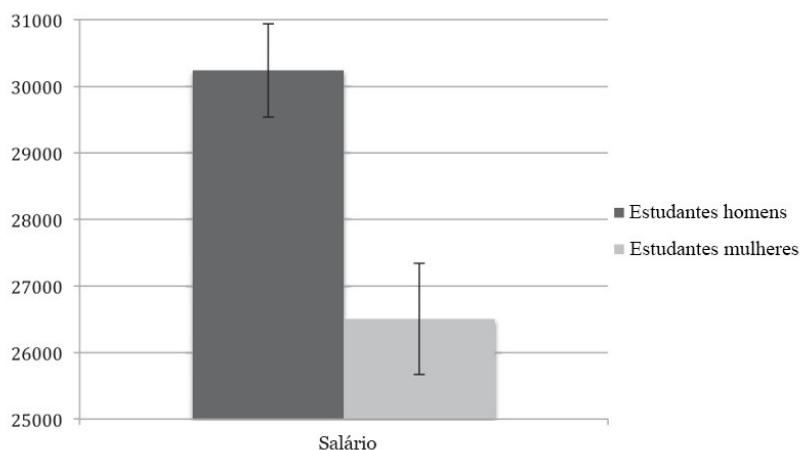


Figura 11.5: Os efeitos do preconceito implícito de gênero na contratação — Parte II

Estes preconceitos implícitos não estão limitados ao recrutamento ou sexo. O viés implícito e acesso desigual aos recursos atuam em todas as fases da nossa vida educacional e profissional, resultando em dominação masculina e branca nos campos da ciência, tecnologia, engenharia e matemática (STEM, na sigla original em inglês) — em geral, profissões altamente bem

pagas tendem a ser dominadas por homens.

A universidade da Califórnia, Berkeley, recentemente redesenhou seu curso introdutório de ciência da computação, levando à inscrição de 106 mulheres e 104 homens (BROWN, 2014).

Uma pesquisa representativa de 19.000 pessoas realizadas nos Estados Unidos pela Level Playing Field Institute entre 2001 e 2006 constatou que o custo anual para empresas americanas atribuível à rotatividade voluntária de gerentes e profissionais devida unicamente à injustiça foi de US\$64 bilhões. Os entrevistados citaram os seguintes comportamentos: grosseria, ter colegas de trabalho em um nível semelhante ou superior que são menos educados ou menos experientes, outros que tomam crédito pela seu trabalho, receber atribuições que são geralmente considerados abaixo do seu nível de emprego, sentindo-se excluído da equipe, e ser estereotipado (KLEIN, 2013, p. 7-8).

O que podemos fazer sobre isso? Aqui está uma seleção de estratégias que se mostraram úteis. Como leitura complementar, consulte o livro de Freada Kapor Klein (2007), *Giving Notice*.

Um sumário excelente sobre a razão pela qual as mulheres deixam a indústria de tecnologia e o que fazer sobre isso pode ser encontrado em <http://bit.ly/1toep4k>.

- *Garanta pagamento equitativo.*

É impossível fazer comparações exatas entre indivíduos, então, em vez disso, examine salários por função. Comparar o salário médio de homens brancos dentro de uma determinada função (como analista de UX ou engenheiro sênior) com o salário médio de pessoas de grupos sub-representados. Corrija quaisquer discrepâncias que você encontrar. O processo anual de revisão de remuneração no Netflix segue uma regra simples: o salário de cada funcionário é ajustado pelo "top de mercado", garantindo que sejam pagos "mais do que [qualquer outra empresa] provavelmente o faria; tanto quanto um substituto custaria; tanto quanto pagaríamos para mantê-los se eles tivessem uma oferta mais elevada de qualquer outro lugar" (HASTINGS, 2009). Se implementada de forma abrangente, essa prática tem o efeito de corrigir as desigualdades salariais para grupos historicamente desfavorecidos.

- *Crie condições-alvo para recrutamento e promoção.*

A Melhoria Kata pode e deve ser usada como parte de esforços para aumentar a diversidade. Condições-alvo de contratação e promoção de grupos sub-representados são um exemplo de uma utilização adequada desta ferramenta. Uma grande empresa queria aumentar o número de mulheres em posições de gerência sênior. Para evitar acusações de discriminação positiva, eles não criaram uma cota para os cargos, mas de fato impuseram uma condição-alvo para a proporção de mulheres na lista de candidatos (por exemplo, "50% dos candidatos para o cargo devem ser mulheres") (FOWLER, 2015). Uma abordagem semelhante pode ser usada para o recrutamento e mix de equipe.

- *Monitore o tempo de casa, a taxa de avanço e a satisfação no trabalho.*

Colete dados sobre o tempo de casa médio dos homens brancos em comparação com pessoas de grupos sub-representados. Veja quanto tempo leva para grupos diferentes receberem promoções. Descubra qual a proporção de cada grupo sub-representado que tem, pelo menos, uma outra pessoa se reportando a eles. Analise sua pesquisa de satisfação de trabalho para revelar diferenças entre dados demográficos. Maior rotatividade dos funcionários de grupos sub-representados, maior tempo para promoção, e menor satisfação no trabalho são indicadores claros de (no máximo) um viés implícito dentro de sua organização.

- *Regularmente reveja as políticas, interações e processos de RH.*

O viés implícito não apenas desempenha um papel no recrutamento — ele permeia o ambiente corporativo. Para dar apenas um exemplo, as mulheres são muito mais propensas a receber feedback crítico em avaliações de desempenho (a palavra "abrasivo" é usado quase que exclusivamente em feedback para mulheres). Padrões semelhantes são evidentes no feedback para outros grupos sub-representados (PYNCHON, 2014).

É essencial criar políticas claras, ter líderes que definam pública e regularmente expectativas sobre comportamento aceitável, e garantir que eles moldem o comportamento apropriado e sejam vistos tomando medidas em caso de comportamento inadequado. Contrate um especialista externo para avaliar interações, políticas e processos de RH, fazer

recomendações, e retornar regularmente para monitorar a implementação e avaliar os progressos.

## 11.4 CONCLUSÃO

Em uma organização de alto desempenho, os funcionários apreciam e têm orgulho no seu trabalho diário, e os líderes e gestores se dedicam a apoiar os funcionários em sua busca pelo propósito da organização. Nenhuma organização faz isso perfeitamente, e aqueles que o fazem melhor estão constantemente trabalhando para melhorar.

Para criar esse tipo de ambiente, temos de abordar o comportamento de todos na organização, começando com os executivos. Como John Kotter observa, "a maioria dos funcionários, talvez 75 por cento de toda a gestão, e praticamente todos os altos executivos, precisam acreditar que uma grande mudança é absolutamente essencial" (KOTTER, 2012, p. 51). A essência de uma mentalidade Lean é o entendimento de que este deve ser o caso não apenas em uma crise, mas o tempo todo. Mudança, melhoria e desenvolvimento são comuns em uma organização verdadeiramente magra.

Mudança de cultura é alcançada pela prática repetida, deliberada e consciente de todos na organização. Líderes e gestores devem facilitar isto, investindo no desenvolvimento dos funcionários e criando condições para apoiar as pessoas que trabalham em conjunto para melhorar continuamente os processos, o conhecimento e o valor entregue aos clientes. Finalmente, é essencial que os líderes moldem os comportamentos que eles esperam que o resto da organização adote. Líderes cujas ações contradizem suas palavras, particularmente quando o seu estado é ameaçado ou em momentos de estresse, perderão a confiança de suas equipes.

## Questões para os leitores

- Sua organização enviou uma pesquisa anônima (pelo menos anualmente) para medir a satisfação no trabalho e outros indicadores de cultura? Os resultados agregados são publicados para estimar o progresso rumo às metas para a satisfação no trabalho, diversidade e uma real mudança cultural? Os resultados são discutidos e postos em prática?
- O que acontece quando algo dá errado? Existe um processo sistemático de aprender com os acidentes, a fim de melhorar os sistemas, ou gerentes se concentram em atribuir a culpa?
- O que a sua organização faz para investir no crescimento em longo prazo dos funcionários?
- A sua empresa vê mudança de cultura como contínua ou baseada em eventos? Que práticas você poderia começar para se mover na direção de um modelo contínuo?
- A sua organização contrata aqueles que tenham experiência e habilidades específicas, ou pessoas com a capacidade e atitude para aprender as habilidades relevantes para ajudar a sua equipe ter sucesso?
- A sua organização investiu em reduzir e eliminar os efeitos de enviesamentos implícitos sistemáticos? Como você está avaliando seu progresso?

## CAPÍTULO 12

# ADOTE O PENSAMENTO LEAN PARA GOVERNANÇA, RISCO E CONFORMIDADE

*"Tudo está sujeito à interpretação. A interpretação que prevalece em algum ponto do tempo é uma função de poder e não de verdade." — Friedrich Nietzsche*

*"A confiança não é simplesmente uma questão de verdade ou mesmo de constância. É também uma questão de amizade e de boa vontade. Confiamos naqueles que têm nosso melhor interesse no coração e desconfiamos daqueles que parecem surdos em relação a nossas preocupações." — Gary Hammel*

Frequentemente ouvimos que os princípios da Startup Enxuta, bem como as técnicas e práticas que sugerimos neste livro, nunca funcionariam em grandes empresas por causa da governança. "Isso não atenderá aos requisitos regulatórios", "Aquilo não se encaixa em nosso processo de gestão de mudanças", "Nossa equipe não tem acesso aos servidores de produção". Estes são apenas alguns poucos exemplos de muitas razões que as pessoas têm dado para repudiar a

possibilidade de mudar a forma que trabalham.

Quando ouvimos tais objeções, reconhecemos que as pessoas não estão realmente falando de governança; elas estão se referindo a processos que foram adotados para gerenciar riscos e conformidade e que confundiram com governança. Assim como quaisquer outros processos em uma organização, aqueles estabelecidos para gerenciar a Governança, os Riscos e Conformidade (GRC) devem ser alvos da melhoria contínua para garantir que contribuam para o valor geral.

Processos da GRC típicos incluem o controle de acesso; a entrega de soluções (gerência de projeto); a gestão de mudanças e as atividades relacionadas à redução de riscos com o uso da TI.

Existem muitas organizações grandes que foram capazes de aplicar práticas de engenharia Lean e desenvolveram uma cultura de experimentação como descrito anteriormente. Estas empresas estão sujeitas ao mesmo nível de exigências regulatórias e revisões que as outras. Por isso sabemos que pode ser feito.

Neste capítulo, buscaremos guiar você através do labirinto que é a GRC, especialmente como ela se relaciona com a gestão de conceitos e práticas necessárias para se tornar uma empresa Lean. Essa área é às vezes pouco entendida por quem não fez da GRC seu foco de carreira, então, apresentaremos algum contexto para lhe ajudar a chegar a um entendimento comum com equipes da GRC. Assim, será mais fácil discutir como processos da GRC e controles podem ser melhorados e permitir às equipes de produtos desbravarem continuamente e melhorarem seu trabalho. Ofereceremos alguns exemplos de como conceitos e princípios Lean podem ser aplicados para melhorar os processos da GRC,

resultando em uma melhor governança e em riscos reduzidos, enquanto ainda se atende à conformidade.

Por todo este capítulo, usaremos o termo "equipes da GRC." Por mais clareza, nossa discussão e nossos exemplos darão enfoque a equipes que influenciam fortemente em como a tecnologia pode ser utilizada nas organizações. As mais comuns são o PMO, a arquitetura técnica, a segurança da informação; a de riscos e conformidade, além das equipes de auditoria interna.

## 12.1 ENTENDENDO GOVERNANÇA, RISCOS E CONFORMIDADE

Na introdução da *Parte I*, declaramos que a principal responsabilidade dos líderes é conduzir a organização como um todo em direção a seus objetivos, ajustando o curso conforme necessário. Isto é governança. Infelizmente, nas organizações o termo governança é frequentemente mal utilizado e confundido com teorias de gestão, modelos e processos projetados para atender às necessidades de uma era que já se foi.

*Governança* é sobre manter nossa organização no caminho. É a principal responsabilidade do conselho administrativo, mas se aplica a todas as pessoas e outras entidades que trabalham para a organização, e requer que os seguintes conceitos e princípios sejam aplicados em todos os níveis:

- **Responsabilidade** — Cada indivíduo é responsável pelas atividades, tarefas e decisões que tomar em seu trabalho diário, e responsável também pelos impactos que essas decisões terão sobre a capacidade geral de entregar valor às partes interessadas.
- **Autoridade e responsabilidade** — Existe um

entendimento sobre quem tem o poder e a responsabilidade de influenciar os comportamentos em uma organização e sobre como isso funciona.

- **Visibilidade** — Todos, a qualquer momento, podem visualizar os resultados atingidos pela organização, com base em dados reais e atuais. Isto, por sua vez, pode ser mapeado aos objetivos e metas estratégicas da organização.
- **Empoderamento** — A autoridade para agir na melhoria da entrega de valor às partes interessadas é concedido ao nível correto (as pessoas que terão de lidar com os resultados daquela decisão).

*Risco* é uma exposição à possibilidade da ocorrência de algo desagradável. Todos gerenciamos riscos diariamente, seja no trabalho, em casa ou quando nos divertimos. Como é impossível eliminar todos os riscos, a questão a ser respondida na gestão de riscos é: "Quais riscos você está disposto a conviver com?". Na medida em que você toma atitudes para mitigar riscos em uma área, inevitavelmente são introduzidos mais riscos em outra área. Um exemplo clássico disso é restringir o acesso da equipe de desenvolvimento ao hardware, forçando-a a se apoiar em uma equipe de infraestrutura centralizada e separada para criar acessos e ambientes para testes e experimentos. Isso pode ser efetivo para o objetivo da equipe de suporte de reduzir o risco de instabilidade nos sistemas, mas aumenta o risco de atraso na entrega na medida em que as equipes devem enviar chamados para outras equipes e esperar até que sejam atendidos.

*Conformidade* é uma obediência às leis, regulamentos da indústria, contratos com vinculação legal e até normas culturais. A intenção da conformidade obrigatória normalmente é proteger os

interesses das partes interessadas em relação à privacidade da informação, segurança física e aos investimentos financeiros. Quando vinculados pela lei, por regulamento ou contrato, a conformidade não é opcional. Se escolhermos não obedecer, aumentamos nossos riscos de multas, paralisações operacionais ou danos à nossa reputação. Em casos extremos, a pena de prisão pode ser o resultado da consciente e sistemática desvirtuação da conformidade de uma organização.

## **GERÊNCIA NÃO É GOVERNANÇA**

Como estabelecido no COBIT 5 Framework (ISACA, 2012), o COBIT formalmente significa *Control Objectives for Information and Related Technology* (objetivos de controle para a informação e tecnologia relacionada). Ele procura oferecer uma visão de negócio de ponta a ponta da governança da TI da organização. Os auditores, assim como as equipes de risco e conformidade, utilizam o framework e as ferramentas relacionadas para criar e avaliar a governança sobre a utilização de tecnologia para a entrega de valor. Para mais informações, veja <http://www.isaca.org/cobit/pages>.

O COBIT 5 claramente explica a diferença entre governança e gerência.

A governança garante que as necessidades, condições e opções das partes interessadas sejam avaliadas para determinar objetivos equilibrados e acordados a serem atingidos; determina a direção por meio da priorização e da tomada de decisão; e monitora o desempenho e a conformidade em relação à direção e aos objetivos estabelecidos.

A gerência planeja, constrói, executa e monitora atividades em alinhamento com a direção estabelecida pelo corpo de governança para atingir os objetivos empresariais.

Por exemplo, a governança envolve criar a visão e os objetivos para a implantação de mudanças tecnológicas a uma taxa que dê condições ao negócio ter sucesso. Ela define o que deveria ser medido para determinar se estamos indo na direção correta para atingir nossos objetivos. A gerência determina como a organização

atingirá aquela visão. No caso das mudanças tecnológicas, que incluem a estruturação de equipes de entrega, suas restrições e qual o nível de decisão que elas estão empoderadas para exercitar.

Será um processo único, geral, que serve para todas as ocasiões, *top-down*, ou será que as equipes terão autonomia e empoderamento para tomar decisões sem ter de esperar por aprovações de alto nível? Uma boa gestão da GRC mantém o equilíbrio entre controle suficiente para evitar que coisas ruins aconteçam e, ao mesmo tempo, permitir que a criatividade e a experimentação possam melhorar continuamente o valor entregue para as partes interessadas.

## **Adote uma abordagem evolutiva em relação à gestão de riscos**

Um conflito que vivenciamos frequentemente quando implementamos estruturas e processos da GRC para conformidade é pensar neles como algo escrito em pedra em vez de algo que poderia ser alterado, modificado e melhorado. Para permitir a boa governança, mudanças nos processos da GRC devem acontecer o tempo todo em resposta a mudanças nas necessidades da organização e no ambiente do mercado em que ela existe.

Quando bem conduzidos, os processos de gestão da GRC melhoram a entrega de valor por meio da gestão efetiva de riscos. A intenção é melhorar a comunicação, a visibilidade e o entendimento de quem está fazendo o que, quando, como e por quê, assim como os resultados do trabalho que está pronto. Isto está fortemente alinhado com o que as equipes de entrega de produto estão tentando atingir. A questão então torna-se: por que os processos da GRC são vistos como impedimentos quando procuramos maneiras de melhorar nossa produtividade e o valor que entregamos aos cliente e à nossa organização?

Infelizmente, muitos dos processos de gestão nas empresas são projetados e implementados no paradigma do comando e controle. Eles são altamente centralizados e vistos como a extensão de equipes especializadas na GRC que não têm de prestar contas pelos resultados dos processos que exigem. Os processos e os controles que essas equipes decretam são frequentemente derivados de frameworks populares sem nenhuma consideração ao contexto em que serão aplicados, e sem considerar o impacto deles em toda a cadeia de valor do trabalho que afetam.

Os processos e controles frequentemente falham em manter o passo com as mudanças tecnológicas e as capacidades que permitiriam que se atingissem os resultados desejados usando meios mais leves e responsivos. Isto força as equipes de entrega a completarem atividades que não adicionam valor, criam gargalos e aumentam o risco geral de falhar para entregar em tempo hábil.

## 12.2 APLIQUE PRINCÍPIOS LEAN AOS PROCESSOS DA GRC

Como tudo que tratamos neste livro, a jornada na aplicação dos princípios Lean para processos da GRC — e os resultados decorrentes — parecerão diferentes em cada organização, dependendo da natureza do negócio e de onde operamos. Não existe um livro de receitas que sirva a todas as circunstâncias (como frameworks respeitáveis, a exemplo do ITIL REF3 e do COBIT, explicam). Contudo, os princípios e conceitos Lean podem ser aplicados a quaisquer processos de gestão da GRC: visualizar a cadeia de valor, aumentar o feedback, ampliar o aprendizado, empoderar equipes, reduzir desperdícios e atrasos, limitar o trabalho em andamento, fazer mudanças incrementais e melhorar continuamente para atingir melhores resultados.

O ITIL (*Information Technology Infrastructure Library*) — veja <http://www.itil-officialsite.com> — é um framework que vem sendo refinado há 20 anos e oferece conjuntos de práticas recomendadas para gerenciar TI com base na experiência dos setores público e privado. É largamente utilizado na TI tanto pelos gestores quanto pelos demais trabalhadores.

Uma tensão natural existe entre as equipes da GRC — incumbidas em recomendar e aconselhar sobre como reduzir riscos e atender à conformidade com as leis e regulamentos aplicáveis — e o resto da organização que apenas quer entregar e o quanto antes melhor. Entretanto, a tensão pode ser boa, pois gera criatividade, mas que só é positiva se todas as partes envolvidas souberem e lutarem para atingir objetivos comuns e forem, no final das contas, medidas pelos mesmos padrões. Quando a tensão é negativa, o resultado obtido é menor em colaboração, visibilidade e conformidade na medida em que os indivíduos e as equipes desenvolvem maneiras secretas para driblar os processos da GRC. Isso leva a decisões baseadas em informações inadequadas, o que enfraquece a governança geral.

As metas e os objetivos das equipes da GRC normalmente resultam em mais trabalho para todas as outras equipes. Um pouco disso é bom, pois a atenção aos riscos, ameaças e aos controles antes de se começar o trabalho pode evitar muita dor de cabeça durante os últimos passos até produção. Ser capaz de provar que temos medidas adequadas de controle postas em prática é importante durante auditorias e ajuda a nos manter em conformidade. O desafio é encontrar o equilíbrio correto entre o controle que permita às equipes avançarem rapidamente e ainda mantenha os riscos relacionados à conformidade em um nível aceitável.

## **Defina o valor dos processos da GRC na perspectiva do consumidor**

Para extrair valor dos processos da GRC, tais como controle de acesso, gestão de mudanças técnicas e ciclo de vida da entrega de soluções, devemos sempre começar com um entendimento comum dos objetivos, valores e resultados esperados dos processos da organização. Precisamos de uma visão comum de como nosso trabalho diário contribui para estes no nível organizacional, independente de com qual equipe nos associamos.

Isso quer dizer que nossas equipes da GRC precisam tomar a responsabilidade pelos resultados (bons ou ruins) da conformidade e das atividades de gestão de risco e seus impactos na habilidade das equipes em entregar valor em tempo hábil. Também, as equipes de entrega de produto deveriam entender a linguagem, a intenção e o propósito dos processos e controles estabelecidos para a conformidade e governança. Apenas então essas equipes, que normalmente são vistas trabalhando ortogonalmente, serão capazes de "parar de lutar estupidamente e fazer mais maravilhas" (ROBBINS, 2013).

Portanto, as equipes da GRC devem se sentir como membros de uma equipe de entrega de produto, aprender sobre as capacidades da tecnologia e das técnicas utilizadas na engenharia Lean, e ajudar as equipes a alavancá-las para oferecer evidências da conformidade sem criar desperdícios e gargalos. Ao mesmo tempo, a equipe de entrega como um todo precisa começar a prestar atenção à linguagem e aos frameworks usados pelas equipes da GRC para entenderem exatamente o que as equipes da GRC buscam obter.

Vimos muito desperdício e muitas tensões destrutivas entre as equipes da GRC e as de entrega, pois muitos processos e práticas de gestão da GRC estão desconectados de como as equipes trabalham.

Tipicamente, as equipes da GRC dão enfoque em realizar e medir a conformidade (por exemplo, perguntando "Todos seguiram a atividade conforme descrita em nosso framework?"), não em melhorar os resultados ("Estamos fazendo o que nos permitirá ser conformes e continuar a entregar valor em tempo hábil?").

### EVITE A ABORDAGEM "NÃO SERIA TERRÍVEL SE" PARA A GESTÃO DE RISCOS

No livro *How to Measure Anything*, Douglas Hubbard relata que Peter Tippet do Cybertrust, discutindo "o que ele entende ser o modo predominante de pensar sobre [segurança da TI], chama de abordagem 'não seria terrível se...'. Nesse framework, os especialistas em segurança imaginam a ocorrência de um evento catastrófico. Independente de sua probabilidade, deve ser evitado a qualquer custo. Tippet observa: 'acontece que todas as áreas têm um 'não seria terrível se...', então todas as coisas precisam ser feitas. Não existe nenhum senso de priorização'" (HUBBARD, 2010, pág. 188).

Quando estivermos priorizando o trabalho em nosso portfólio, não pode existir um passe livre para que o trabalho voltado a mitigar "coisas ruins" avance para o começo da fila. Em vez disso, quantifique os riscos considerando seus impactos e probabilidades utilizando o mapeamento de impactos (veja o capítulo 9), e então faça uso do Custo de Atraso (veja o capítulo 7) para equilibrar o trabalho de mitigação contra outras prioridades. Dessa forma, podemos gerenciar os riscos com segurança e a conformidade utilizando um framework econômico em vez do medo, da incerteza e da dúvida.

As equipes da GRC são medidas pelo "Estamos em

conformidade?", já as equipes de produto são medidas por "Com que velocidade conseguimos entregar valor pelo uso da tecnologia?". Ambas as perguntas estão erradas porque elas medem o desempenho de uma equipe a partir de uma perspectiva funcional isolada e não como o valor líquido para a organização. É fácil estar em conformidade com as leis quando as equipes da GRC podem exigir processos e forçar que todos os itens de um checklist sejam realizados. Contudo, quando as métricas das equipes não estão alinhadas no nível organizacional, podemos estar conformes e, ainda assim, tomar decisões incrivelmente ruins sobre a entrega de valor para as partes interessadas. Isso é realmente irônico, pois a maioria das leis e regulamentos foram estabelecidos com o intuito de proteger e melhorar o valor para as partes interessadas.

#### **ABORDAGENS BASEADAS EM REGRAS LEVAM AO TEATRO DA GESTÃO DE RISCOS**

Quando as equipes da GRC não usam uma abordagem com base em princípios e prescrevem, em vez disso, regras que as equipes devem seguir cegamente, o resultado comum é o teatro da gestão de riscos: uma apresentação cara projetada para dar a aparência de gerenciar os riscos, mas que na realidade aumenta as chances de consequências negativas não intencionais.

Em uma grande empresa europeia que trabalhei, o processo de aprovação de mudanças fazia com que os desenvolvedores prenchessem uma planilha com sete abas, que era enviada por e-mail para um gerente de mudanças em outro país que então decidia se aprovaria ou não. A mudança não poderia ir em frente sem essa aprovação, nem tampouco se o formulário não estivesse completamente preenchido. O gerente de mudanças não entendia verdadeiramente o conteúdo da planilha; antes de aprovar, ele dependia de conversas com os desenvolvedores

para determinar quais eram os riscos e se as atividades planejadas de mitigação eram adequadas. Os desenvolvedores sabiam disso e faziam o menor esforço possível no preenchimento da planilha, muitas vezes apenas mudando a data e o título de uma versão enviada anteriormente e reenviando-a como uma nova requisição. O gerente de mudanças sabia que os desenvolvedores faziam isso, mas não fazia diferença para ele, desde que o processo documentado estivesse sendo seguido à risca. Não havia nenhuma adição de valor em termos de gestão de riscos, apesar de tornar desnecessariamente doloroso para a equipe conseguir colocar suas mudanças no ar. Contudo, a conformidade estava sendo atendida por meio da "evidência" documentada na requisição de mudança. O valor real estava acontecendo nas conversas e em completar atividades de mitigação antes da aplicação da mudança.

Quando as equipes de produto se recusam a participar do teatro de gestão de riscos, a resposta padrão é que há exigências por parte de algum framework popular como o ITIL ou o COBIT, ou por alguma lei ou regulamento. Contudo, com algumas poucas exceções, nem os frameworks nem as leis prescrevem alguns processos em particular. Por exemplo, muita gente acredita que a segregação de funções é exigida por uma lei dos EUA chamada de Sarbanes-Oxley, seção 404, de forma que as organizações estabelecem controles elaborados sobre acesso a sistemas e ambientes de TI para atender a interpretação delas do que essa lei quer dizer. O fato é que em nenhum lugar nessa lei — nem nas regras de SEC que foram criadas pela lei — a segregação de funções é mencionada.

A segregação de funções é um conceito que busca evitar erros e atividades maliciosas exigindo que pelo menos duas pessoas completem uma transação fim a fim. Outro jeito de abordar

isso é assegurar que nenhuma pessoa possa completar uma transação sem que esta tenha sido detectada ou controlada por, pelo menos, uma outra pessoa.

Se você acredita que é esperado de você seguir um processo que comprometa sua habilidade de fazer um bom trabalho, vale a pena conversar com as pessoas que criaram o processo para discutir suas intenções. Volte ao Princípio da Missão no *capítulo 1* e use-o como uma oportunidade de colaboração, e para construir relacionamentos e desenvolver um entendimento comum. Você pode se surpreender ao descobrir que é possível ter uma conversa produtiva sobre como atingir as metas delas de um jeito diferente, ou mesmo ver se seu trabalho realmente está no escopo para a lei ou regulamento em questão.

Se lhe disserem que um processo em particular é "requerido" por algum regulamento, educadamente pergunte onde você pode encontrar mais informações sobre esta exigência. Em muitos casos, regras e processos da GRC onerosos que são aplicados são simplesmente interpretações de alguém sobre o que é necessário, não algo mandatório pelo regulamento em questão.

## 12.3 MAPEIE A CADEIA DE VALOR, CRIE O FLUXO E ESTABELEÇA UM SISTEMA PUXADO

Com um entendimento comum dos processos da GRC e dos objetivos da equipe de entrega de produto e seus métodos, a colaboração para atingir os objetivos organizacionais pode realmente começar. Como discutido no *capítulo 7*, o mapeamento da cadeia de valor é uma poderosa ferramenta que pode ser usada

para nos dar uma visão do estado atual e identificar áreas de melhoria. No contexto dos processos GRC, é importante colocá-las sobre estas atividades da equipe de entrega e entender como elas influenciam a habilidade da equipe de finalizar seu trabalho.

A maior parte dos processos da GRC é projetada isoladamente para aplicar controles como aprovações necessárias, acesso limitado, segregação de funções, monitoramento e revisão de atividades. Estas servem para oferecer visibilidade e transparência sobre quem faz o que, quando e com qual autoridade. Acima de tudo, os frameworks usados mais comumente pelas equipes da GRC para criar os processos enfatizam melhorar a eficiência geral e a eficácia da organização. Infelizmente, muitos dos processos e controles fazem justamente o oposto quando considerados em uma cadeia de valor maior vista de ponta a ponta.

## **Os controles errados interrompem o fluxo**

Os controles podem ser preventivos por natureza pela aplicação de uma barreira. De forma alternativa, eles podem agir como detetives — monitorando e revisando eventos depois de suas ocorrências, e levantando as respostas apropriadas para a descoberta de exceções potenciais tais como erros, omissões ou ações maliciosas.

Muitos de nós cometem o erro de pensar que controles preventivos são mais efetivos: podemos criar barreiras ou tirar a possibilidade das pessoas de fazerem alguma coisa, mas não funciona. A realidade é que as pessoas precisam entregar as coisas. Se você tentar pará-las, muitas vão se tornar criativas e descobrir formas de driblar quaisquer barreiras criadas. A reação então é travar ainda mais, o que encoraja ainda mais soluções criativas por baixo dos panos, cultivando uma cultura subversiva de comportamento arriscado. Um bom exemplo são equipes que

compartilham um usuário com privilégios para acessar ambientes diferentes. Seria muito melhor conceder acesso a cada integrante da equipe com seus próprios IDs, e então monitorar o uso desses privilégios.

Um resultado ainda mais trágico de usar controles preventivos em demasia é quando as equipes simplesmente deixam de se importar e passam a agir de uma forma autômata, abandonando quaisquer esforços de melhorar as coisas.

Controles preventivos, quando executados no nível inadequado, frequentemente levam a custos altos e desnecessários, forçando as equipes a:

- Esperar outra equipe para completar tarefas braçais que poderiam ser facilmente automatizadas e executadas quando necessário;
- Obter aprovações de pessoas ocupadas que não têm um bom entendimento dos riscos envolvidos na decisão, tornando-se, portanto, gargalos;
- Criar grandes volumes de documentação de acurácia questionável que se torna obsoleta logo depois de concluída;
- Empurrar grandes lotes de trabalho para equipes e comitês especiais pedindo aprovação e processamento, e então esperar pelas respostas.

Se os controles preventivos não forem executados de forma adequada e consistente, eles não serão mais efetivos. Eles devem ser monitorados continuamente para garantir que estão sendo aplicados de forma correta e que ainda são relevantes. Sem o monitoramento e as ações corretivas decorrentes, os controles preventivos são menos efetivos que controles de detetive bem

executados, tais como monitoramento constante, teste e revisão frequentes e aplicados logo de início, além de métricas altamente visíveis dos resultados.

Apesar do senso de segurança que o uso de controles preventivos pode trazer, eles são extremamente valiosos quando aplicados no nível adequado, e são a melhor solução em algumas circunstâncias. Contudo, eles nunca deveriam ser aplicados unilateralmente, apenas em conjunto com outros controles e no nível correto de granularidade. Devemos sempre considerar o efeito deles na capacidade de entrega das equipes.

Portanto, quando desempenhamos o mapeamento de valor dos processos de governança sobre os processos de entrega, precisamos olhar com cuidado para todos os controles e fazer duas perguntas:

- O propósito do controle está sendo atendido?
- O controle está contribuindo de verdade para a efetividade e eficiência geral da organização?

Precisamos olhar com cuidado para o nível de autoridade dado para nossas equipes. O objetivo é trazer as decisões de aprovação para o nível correto e dar às equipes o máximo de autoridade que as possibilitem fazer seu trabalho. Isto envolve definir limites e garantir que as equipes saibam como e quando escalar decisões que estão fora de sua autoridade. Também precisamos garantir que a documentação seja mantida a um nível razoável e, quando feita, garantir que seja acessível, fácil de se entender e atualizada conforme necessário, de preferência automaticamente.

"Confie, mas verifique" (provérbio russo popularizado por Ronald Reagan) é um conceito que vem ganhando aceitação em círculos da GRC. Em vez de impedir as equipes de acessar os ambientes e o hardware para evitar que façam algo errado,

confiamos que as pessoas farão o correto, por isso damos acesso e controle sobre os sistemas e o hardware que elas precisam usar diariamente. Então verificamos se a equipe não está abusando da autoridade por meio de um monitoramento bom e da revisão frequente dos processos, com o intuito de garantir que os limites estabelecidos estejam sendo observados e que visibilidade e transparência completas estejam incluídas no trabalho da equipe.

#### **REDUZINDO OS CICLOS DE FEEDBACK NAS ATIVIDADES DE CONFORMIDADE**

Atender às demandas de conformidade para Segurança da Informação tem sido uma pedra no sapato de muitas equipes de entrega. No espírito da metodologia Big Bang de entrega de projetos, a equipe de segurança é envolvida no último momento — dias antes do lançamento — para executar uma revisão final de código para procurar por vulnerabilidades de segurança e da conformidade requerida.

A comunidade de Segurança da Informação agora percebe que essa abordagem não funciona. Na maioria dos produtos, simplesmente existe complexidade e volume demais para completar uma revisão mínima. Quando as vulnerabilidades ou outras falhas na conformidade são descobertas dessa forma, geralmente é tarde demais para fazer muita coisa a respeito. Acaba sendo mais arriscado corrigir as vulnerabilidades em um sistema frágil ou esperar pelas mudanças do que deixar que as vulnerabilidades cheguem a produção com a promessa de arrumá-las depois.

Para atender a questões de conformidade e redução de riscos de segurança, muitas organizações agora incluem especialistas em segurança da informação como membros das equipes

multifuncionais de produto. O papel deles é ajudar a equipe a identificar quais são as possíveis ameaças de segurança e qual nível de controle será necessário para reduzi-las a um nível aceitável. Eles são consultados desde o início e são engajados em todos os aspectos da entrega do produto:

- Contribuindo com o design em relação a privacidade e segurança;
- Desenvolvendo testes automatizados de segurança que podem ser incluídos no pipeline de deploy;
- Pareamento com as pessoas de desenvolvimento e de testes para ajudá-las a entender como prevenir a inclusão de vulnerabilidades comuns na base de código;
- Automatizando o processo de testes de patches de segurança nos sistemas.

Eles também criam seus próprios ambientes para realizar revisões de código e testes de segurança mandatórios de forma que eles não impeçam a equipe de executar outro trabalho enquanto este é feito.

Como membros da equipe, os especialistas ajudam a encurtar os ciclos de feedback relacionados à segurança, ajudam a reduzir os riscos gerais de segurança na solução, melhoraram a colaboração e o conhecimento de problemas de segurança da informação em outras equipes, e elas mesmas aprendem mais sobre o contexto do código e as práticas de desenvolvimento. Todos saem ganhando.

Na medida em que todos nos tornamos melhores em criar fluxo

para as equipes mudando os processos de governança, as equipes da GRC também se beneficiam. Utilizando controles desenhados em colaboração com as equipes da GRC, as equipes de entrega podem embutir evidência de conformidade real no trabalho diário e nas ferramentas, além de afastar o teatro da gestão de risco. Como fazemos com a qualidade funcional e de desempenho, construímos evidência de conformidade em nosso trabalho diário de forma a não ter de apelar para grandes lotes de inspeção após a maior parte do trabalho estar pronta.

O efeito global para as equipes da GRC é que elas agora podem extrair informações relacionadas à conformidade das equipes de entrega a qualquer momento, sem interromper o fluxo geral da equipe, a não ser que algo desagradável ou não previsto pareça estar acontecendo. As auditorias anuais são menos dolorosas porque as equipes de entrega entendem a intenção dos controles que os auditores estão pedindo e podem gerar evidências do cumprimento dessas intenções por meio de seus processos.

Ao usar um framework econômico (a exemplo de Custo de Atraso, discutido no *capítulo 7*), podemos quantificar os trade-offs econômicos que fazemos quando implementamos controles para mitigar riscos. Isto nos permite priorizar trabalho da GRC em relação a outros tipos de trabalho que realizamos — portanto puxar trabalho adicional requerido para conformidade no momento correto para o negócio.

#### **ESTUDO DE CASO: IMPLEMENTAÇÃO PCI-DSS NA ETSY**

A Etsy é um mercado on-line para itens feitos à mão e itens vintage com mais de um bilhão de dólares em vendas brutas em 2013. Na cultura de alta confiança da Etsy, os desenvolvedores normalmente colocam suas próprias

mudanças no ar — realmente, como parte da integração de novos engenheiros, os desenvolvedores utilizam o sistema de implantação automatizado para atualizar seus perfis ao vivo no site logo nos primeiros dias. Os engenheiros também podem trabalhar em — e ter acesso a — todas as partes do sistema.

Contudo, devido à Etsy processar transações de cartão de crédito, ela está sujeita ao PCI-DSS, um padrão da indústria que é bastante prescritivo em como gerenciar sistemas para armazenamento ou transmissão de informações do titular do cartão (esses sistemas são conhecidos como o ambiente de informações do titular, ou *cardholder data environment*, CDE). Por exemplo, o CDE deve ser fisicamente separado e deve também haver a segregação de funções por pessoas que trabalham nos sistemas dentro do CDE.

A segregação de funções normalmente é interpretada no sentido de (entre outras coisas) dizer que os desenvolvedores não deveriam ter acesso à base de dados de produção e que eles não deveriam ser capazes de colocar suas próprias modificações em produção. Ambos requisitos conflitam com a forma que a Etsy tipicamente opera. Veja como eles abordaram a conformidade em relação ao padrão PCI-DSS.

- 1. Minimize as consequências de uma conformidade requerida. Entenda que não existe uma solução de conformidade que sirva para tudo e arquitete sistemas que separem as preocupações relacionadas a diferentes demandas de conformidade.**

A cultura principal na Etsy é otimizada para a velocidade de inovação. Contudo, o processamento de cartões de crédito é uma área na qual a segurança dos dados é de máxima importância. A Etsy reconhece que diferentes partes de seus

sistemas têm preocupações diferentes e devem ser tratadas de forma diferente.

A decisão arquitetural mais importante da Etsy foi desacoplar o ambiente CDE do resto do sistema, limitando o escopo dos regulamentos do PCI-DSS para uma área separada, prevenindo a "contaminação" de todos seus sistemas de produção. Os sistemas que formam o CDE são separados (e gerenciados de outra forma) do resto dos ambientes da Etsy nos níveis físico, de rede, de código-fonte e de infraestrutura lógica.

Aliás, o CDE é construído e operado por uma equipe multifuncional que é responsável exclusivamente pelo CDE. Novamente, isto limita o escopo dos regulamentos PCI-DSS apenas a esta equipe.

### **1. Estabeleça e limite o raio de impacto de frameworks e regulamentos.**

Sempre comece perguntando: "Qual é o menor conjunto de mudanças que precisamos fazer em nossa arquitetura e cultura ideais que ainda atenda à conformidade a que estamos sujeitos?". Então adote uma abordagem incremental e iterativa na implementação e validação dessas mudanças.

Por exemplo, apesar do PCI-DSS exigir a segregação de funções, isso não impede que uma equipe multifuncional CDE trabalhe junta no mesmo espaço. Quando integrantes da equipe CDE querem fazer alguma alteração em produção, eles criam um chamado que é aprovado pelo líder técnico; fora isso, o código é comitado e o processo de deploy é completamente automatizado, como acontece com o ambiente principal da Etsy. Não existem gargalos e atrasos, pois a segregação de funções é mantida local: uma mudança é aprovada por alguém diferente da pessoa que a fez.

## **1. Utilize controles compensatórios.**

É fundamental respeitar os resultados dos regulamentos que estamos tentando obter, apesar de reconhecer que existem diversas formas de atingir tais resultados. Por exemplo, o PCI-DSS permite que as organizações implementem "controles compensatórios" — uma alternativa projetada para criar o mesmo resultado — onde exista uma restrição legítima, seja técnica ou de negócio, que impeça a implementação de um controle em particular (WILLIAMS, 2010).

No caso do PCI-DSS, você deveria falar com seu auditor de segurança qualificado (QSA — Qualified Security Auditor) e com o banco adquirente para discutir possíveis alternativas aos controles que têm um impacto inaceitável, seja técnico ou de negócio. Por exemplo, o pipeline de deploy descrito no *capítulo 8* é usado pela Etsy oferece um conjunto poderoso de controles compensatórios que pode oferecer uma alternativa à segregação de funções nos outros sistemas deles.

A vantagem de se utilizar princípios Lean e a entrega contínua no desenvolvimento de produtos é que eles possibilitam uma abordagem bem granular e adaptativa para a gestão de riscos. Conforme vamos trabalhando em lotes pequenos e rastreando cada mudança em nossos sistemas do check-in até a implantação, poderemos quantificar o risco de cada mudança e gerenciá-la de forma apropriada.

A melhor maneira de atingir os objetivos de uma boa GRC é incluindo a conformidade e a gestão de riscos nas atividades diárias das equipes de produto, incluindo o design de sistemas e de UX e testes. Conforme as organizações se afastam do paradigma do comando e controle e as equipes da GRC adotam uma abordagem

colaborativa para a gestão de risco, começamos a valorizá-las como conselheiras confiáveis e especialistas em seu domínio de conhecimento. Para muitas equipes da GRC, isso requer uma mudança grande em seus papéis, responsabilidades e comportamentos na organização. Esse é o movimento de um papel de policiamento para um de membro contribuidor da equipe que é medido pelos mesmos resultados que a equipe de produto, não apenas por uma perspectiva de conformidade.

## 12.4 CONCLUSÃO

Uma boa governança necessita que todos deem ênfase na descoberta de formas para melhorar o valor e oferecer informação correta sobre a qual embasar nossas decisões. Começamos com liderança e direção do Conselho e Executivos, e confiamos na habilidade dos empregados de aceitarem suas responsabilidades em tomar boas decisões no trabalho. Uma cultura de abertura, confiança e transparência é necessária para uma boa governança.

As estruturas e os processos da GRC devem ser desenvolvidos colaborativamente tanto pelas equipes da GRC quanto pelas equipes de produto que trabalham diariamente para entregar valor para os consumidores. Ao identificar a intenção das leis e regulamentos que temos de cumprir, nossas equipes da GRC podem colaborar como equipes de produto para determinar abordagens locais que se encaixam melhor na melhoria da entrega de valor.

Começamos explorando, com as equipes da GRC, como podemos minimizar os efeitos negativos em se apoiar em controles restritivos por meio do uso criativo da arquitetura de sistemas, melhoria de processos, contenção do escopo, aplicação de controles compensatórios e alavancagem de novas tecnologias. Podemos então tirar proveito do nosso aprendizado para melhorar continuamente nossos processos e oferecer tanto uma governança

melhor quanto melhores resultados para todas as partes interessadas.

## Questões para os leitores

- Como suas equipes de produto veem seus processos da GRC atuais? Até que ponto sua organização está engajada no teatro da gestão de riscos?
- Quais ações os líderes tomam para desenvolver um entendimento compartilhado da linguagem e dos frameworks da GRC pela organização?
- Suas estruturas da GRC (políticas, organização e processos) evitam que as equipes de produto façam melhorias no processo, ou exigem que elas peçam aprovação para quaisquer mudanças no processo? Se assim for, como você poderia apoiar as equipes na melhoria dos processos sem abrir mão da conformidade?
- Como você poderia possibilitar que as equipes da GRC colaborem com suas equipes de produto como membros confiáveis durante o processo de criação de valor?

## CAPÍTULO 13

# EVOLUINDO A GESTÃO FINANCEIRA PARA GUIAR A INOVAÇÃO DE PRODUTO

*“Aderir a regras de orçamentação não deveria ser mais importante que a boa tomada de decisão.” — Emily Oster*

*“Neste exato momento, sua empresa tem processos de negócio do século XXI, baseados na internet, processos gerenciais da metade do século XX, tudo apoiado em princípios de gestão do século XIX.” — Gary Hamel*

### 13.1 INTRODUÇÃO

Em muitas empresas grandes, os processos de gestão financeira (*Financial Management Processes*, ou FMP em inglês) são desenhados no paradigma de projeto. Isto se apresenta como um empecilho na condução de uma abordagem baseada em produto na busca pela inovação. É relativamente fácil para equipes pequenas trabalharem e colaborarem umas com as outras. Entretanto, em uma escala organizacional, em algum momento atingimos um

ponto em que a evolução é impedida por processos de gestão financeira rígidos e centralizados que guiam as entregas e os processos de aquisição, limitando as opções para inovar em escala.

Vamos tratar alguns dos problemas criados por estes FMPs, dando especial ênfase ao processo orçamentário. Reforçamos que, para superar as questões que sua organização vivencia como resultado de processos gerenciais financeiros, você precisará da ajuda da equipe de finanças. Comece construindo boas relações com eles e trabalhe de forma colaborativa para melhorar os resultados para os clientes e para o negócio.

Reconhecer a interdependência de todos os processos gerenciais, os comportamentos prejudiciais que eles podem exigir e as barreiras que eles apresentam para a melhoria contínua e para a inovação é essencial para ter sucesso em se tornar Lean. É difícil deixar o velho pensamento de que o controle forte e centralizado gera eficiências valiosas. Não importa quanto bem esses controles tenham-nos servido em uma era de baixa complexidade e avanços técnicos mais lentos, hoje cria barreiras que evitam que nos adaptemos rapidamente a oportunidades emergentes. Neste contexto, os recursos e esforços necessários para coletar informação, comunicar e monitorar processos rígidos e centralizados superam qualquer ganho de eficiência. Assim como um processo orçamentário fortemente controlado e centralizado promove a competitividade mais do que o comportamento interno e colaborativo. Isso é contraprodutivo à inovação, que requer trabalho em equipe.

Muitas grandes organizações multinacionais têm se transformado ao abandonar a crença de que comando e controle é o melhor jeito de gerenciar seus processos financeiros. Para se aprofundar mais sobre o tópico, recomendamos a leitura de *Beyond Budgeting* (HOPE; FRASER, 2003) e *Implementing Beyond Budgeting* (BOGSNES, 2009), assim como o site *Beyond Budgeting*

*Round Table* (<http://www.bbrt.org> ).

## 13.2 DANÇAR NO RITMO DO TAMBOR FINANCEIRO REDUZ A VELOCIDADE DA INOVAÇÃO

Planejamento, planejamento orçamentário, previsibilidade e monitoramento são essenciais na definição do que é sucesso para nós, em particular o nosso compromisso com os acionistas.

Regulamentos e padrões relativamente novos ou revisados, tais como a Sarbanes-Oxley e a International Financial Reporting Standards, têm intensificado a percepção da necessidade de se controlar e centralizar tais processos. Entretanto, a intenção desses regulamentos é melhorar a transparência e a visibilidade em relatórios financeiros, bem como nossa habilidade em tomar decisões melhores. Controle e tomada de decisão centralizados através de orçamentos anuais podem facilmente criar resultados contrários.

Neste capítulo, consideramos as práticas gerenciais financeiras da organização em empresas que são tipicamente reconhecidas como inibidoras da inovação:

- Basear decisões de negócio em ciclos orçamentários centralizados anuais, considerando exceções somente em circunstâncias extremas. Isso junta previsibilidade, planejamento e monitoramento em um único processo centralizado, realizado uma vez por ano, gerando resultados subotimizados destas importantes atividades.
- Usar a capacidade de atingir objetivos orçamentários como um indicador chave da performance individual,

do time e da organização como um todo, que meramente diz a você como boas pessoas atuam no processo, e não os resultados que eles têm de alcançar no último ano.

- Basear decisões de negócio na estrutura financeira de capital *versus* gastos operacionais. Isso limita a habilidade de inovar ao não começar um produto mínimo viável (MVP) que cresce gradualmente ou pode ser descartado a qualquer momento. O modelo CapEx/OpEx de reportar custos é amplamente baseado em recursos físicos e é baseado em projeto. Isso não se adapta bem ao uso de informações para experimentos, aprendizados e melhorias contínuas do produto no tempo.

Combinadas, essas práticas nos forçam a cronometrar decisões de negócio e planos anuais de trabalho para otimização dos departamentos financeiros e ciclos de prestação de contas, que por sua vez restringem quando e como inovações ocorrem na organização. Eles não estão em sintonia com nossa habilidade e necessidade de entregar valor continuamente para os consumidores. Programas inchados e amplamente financiados, que entregam valor questionável, não aproveitam oportunidades inesperadas, porque não há financiamento disponível para explorar e testar as hipóteses. Tempo que poderia ser gasto em inovação é gasto, em contrapartida, no gerenciamento e prestação de contas sobre "o orçamento".

### 13.3 LIBERTANDO-NOS DO CICLO ANUAL DE ORÇAMENTO

Processos orçamentários centralizados tipicamente planejam,

fazem previsões, monitoram e reportam sobre a posição financeira e o desempenho geral de uma organização. Eles guiam tudo do relatório da meta de receita até o planejamento e alocação de recursos. Contudo, no contexto do desenvolvimento de produtos, o ciclo anual de orçamento pode facilmente:

- Reduzir a transparência sobre os custos reais da entrega de valor — os custos são alocados por centros de custo funcionais ou de qual cesta vem o dinheiro, sem uma visão de ponta a ponta.
- Remover decisões das pessoas fazendo o trabalho — a alta gestão estabelece e demanda metas detalhadas.
- Afastar esforço da criação de valor ao exigir processos exaustivos para aprovar, rastrear e justificar custos.
- Medir o desempenho pela habilidade de agradar o chefe ou produzir saídas — não pelo resultados reais para os clientes — recompensando aqueles que atingem as metas orçamentárias, independentes qual seja o custo geral e de longo prazo.

Contudo, muitas grandes empresas estabelecidas encontraram alternativas aos processos orçamentários centralizados para conseguir atingir os objetivos de uma boa gestão financeira. A figura a seguir chama a atenção para a importância de se separar os objetivos orçamentários e sugere algumas abordagens possíveis.



Figura 13.1: Abordagens para o atingimento de objetivos orçamentários, cortesia de Bjarte Bogsnes (2009)

## Pare de confundir boa gestão financeira com "o orçamento"

“Odeio o planejamento orçamentário anual com o fogo de mil sóis.” — Anônimo

Um orçamento deveria ser visto como a soma de fundos reservada, ou necessária, para um propósito: “Qual é o teto de quanto podemos gastar nessa atividade?”. Ele não define o que vamos fazer realmente — isso seria estratégia. Não se trata tampouco de um plano para como executar a estratégia, nem faz previsão ou mede o sucesso na entrega de valor para os clientes. Quando embutimos todas essas atividades no processo de orçamento, perdemos nosso foco.

Ter um orçamento é algo bom, especialmente quando

estabelecemos algum desafio financeiro para nós mesmos. Restrições financeiras podem ser um bom catalisador para a criatividade, a colaboração e a inovação. Particularmente no domínio da prospecção, podemos estimular a inovação se deliberadamente reduzirmos o financiamento a áreas localizadas ou produtos, e permitirmos que equipes decidam como podem fazer melhor uso do investimento disponível, conforme descrito na *Parte II* deste livro. Contudo, esta abordagem não funcionará se simplesmente reduzirmos o investimento e dissermos às equipes quais as metas delas e como atingi-las.

Seguindo o princípio da subsidiariedade descrito no *capítulo 10*, a responsabilidade para gerenciar os fundos alocados deveria ser levado até o menor nível apropriado — geralmente, as pessoas que realizam o trabalho. Ainda precisamos oferecer às equipes definições claras dos limites, mas as equipes precisam ser confiadas e precisam receber a chance de tomar decisões. Como descrito no livro *Implementing Beyond Budgeting*, quando a gigante petroquímica europeia Borealis utilizou essa abordagem, eles esperavam que os custos subissem. Ao invés disso, houve uma queda (BOGSNES, 2009, p. 90).

Apesar da Borealis estar bem posicionada e preparada para a mudança cultural para apoiar esse passo, o CFO Bjarne Bogsnes atribui a maior parte do resultado à melhor visibilidade no que guia os custos por meio do uso dos princípios da contabilidade baseada em atividades: aqueles responsáveis pelas atividades que geram custos reportam suas finanças e as equipes assumem responsabilidade por melhorar o gerenciamento dos custos.

Contabilidade baseada em atividades é uma abordagem de atividades de custo e monitoramento que envolve traçar recurso de consumo e de custo de resultados finais (EDWARDS, 2008).

A grande falácia do planejamento, evidente no processo orçamentário centralizado, é que se desenvolvermos um plano financeiro detalhado de antemão para o próximo ano, ele simplesmente se desenrolará conforme o plano desde que nos atenhamos a ele. Os esforços para desenvolver estes tipos de planos são um desperdício de tempo e de recursos, pois o desenvolvimento de produto diz respeito tanto a descoberta quanto a execução. Os custos mudarão, novas oportunidades surgirão e uma parte do trabalho planejado não gerará os resultados desejados. No mundo atual da globalização, do rápido crescimento tecnológico e do aumento da imprevisibilidade, é uma tolice pensar que planos precisos e acurados são atingíveis ou mesmo desejáveis.

Uma abordagem mais interessante é estabelecer objetivos de alto nível e de longo prazo, cuidadosamente gerenciar o futuro próximo mais previsível e constantemente ajustar nos planos de curto prazo para chegar mais perto de nossas metas. Podemos adotar essa abordagem implementando o lançamento da estratégia descrito no *capítulo 15*.

O lançamento da estratégia utiliza o meta-método Kata da Melhoria apresentado no *capítulo 6* e cascateia-o por toda a organização, seguindo o Princípio da Missão, descrito no *capítulo 1*. Bjarte Bogsnes apresenta uma abordagem semelhante chamada de "Da ambição à ação" (*Ambition to Action*) no *Implementing Beyond Budgeting*.

*Ambition to Action*, apresentada no capítulo 4 de Bogsnes (2009), é derivada de abordagem *Balanced Scorecard* de Kaplan e Norton.

#### **SUBSTITUA O ORÇAMENTO ANUAL POR PREVISÃO CONTÍNUA**

Previsão contínua é uma ferramenta que pode ser útil para ajudar a melhorar o planejamento financeiro e diminuir a dependência do orçamento. Conforme cada período acaba, outro é ligado ao final da previsão de forma que sempre se cobre um mesmo tamanho de tempo para o futuro. O final não oferece muito detalhe, mas inclui linhas de custos conhecidas com estimativas sobre como serão no período em questão. Na previsão contínua, a atenção é concentrada no futuro próximo, com base nas informações acuradas e atuais. Não gastamos tanto tempo buscando detalhes para o futuro que provavelmente mudará de alguma forma desconhecida.

Ao adotar esta abordagem, lembre-se de que as previsões não são feitas para se definir metas ou gerenciar recursos. A não ser que você utilize uma abordagem como lançamento de estratégia ou da ambição para a ação para estabelecer metas e gerenciar recursos e desempenho, você acabará com um orçamento contínuo em vez de uma previsão contínua, o que Bogsnes descreve como "um pouco mais dinâmico, mas quatro vezes mais trabalho" (em uma conversa particular).

Ao separar as atividades requeridas para realizar uma boa gestão financeira daquelas necessárias ao processo de orçamento anual,

melhoramos nossa capacidade de entender nossas atuais condições. Focamo-nos em desenvolver o fluxo nas decisões e ajustes necessários aos objetivos que definimos para nós mesmos. A mudança acontece de “Eu tenho o dinheiro para fazer o que me pediram?” para “Isso é realmente necessário?”

## **Desassociar decisões de financiamento das decisões do ciclo financeiro anual**

O uso do ciclo financeiro anual para definir alocação de recursos promove uma cultura que frustra nossa capacidade de experimentar e inovar. Perpetua o desperdício e as ideias que são pouco prováveis de entregar valor. Nós precisamos reconhecer que as inovações têm um custo contínuo que não pode ser definido ou totalmente planejado com um ano de antecedência. Nós precisamos de processos leves para financiar inovação e sermos disciplinados em parar de trabalhar em qualquer coisa que não gere os resultados desejados.

Quando um processo anual é o único canal para obter recursos amarrados a linhas específicas de itens ou novas iniciativas, é quase impossível mudar a direção em resposta a novas informações. Ao contrário, todo ano precisamos investir uma boa quantidade de esforços para apresentar o melhor plano de negócio para obter a maior quantidade de financiamento possível, em vez de ser honesto sobre o que realmente precisamos. De todas as submissões feitas durante este evento anual, somente aqueles com as histórias mais atraentes passam inalteradas. As demais são cortadas ou colocadas no backlog para serem consideradas no próximo ano, acumulando custo de atraso.

Em vez disso, algumas empresas estão fazendo o chamado *Dynamic Resource Allocation* (Alocação Dinâmica de Recursos) apresentado na figura seguinte. Esta abordagem cria revisões

frequentes para decisões de financiamento e cada decisão tem menos riscos associados. Todas as decisões são baseadas em evidências empíricas, se tornando mais fáceis de realizar. Quando feito corretamente, acesso aos fundos expandem para mais times, são mais frequentes, tem menos riscos e traz melhores resultados. Encorajamos mais inovações e reduzimos riscos financeiros associados com grandes iniciativas.

### ALOCAÇÃO DINÂMICA DE RECURSOS

Uma mentalidade diferente - a consciência sobre os custos desde o primeiro centavo

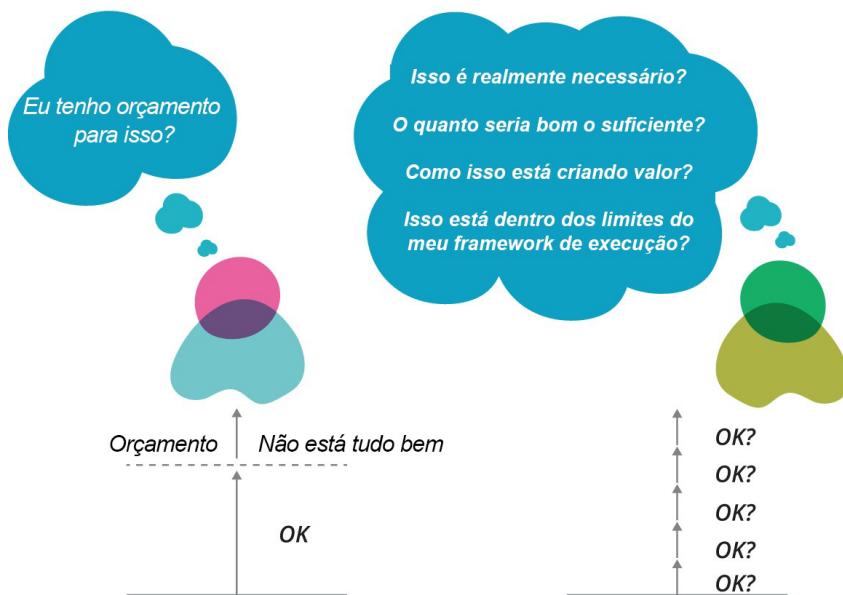


Figura 13.2: Alocação Dinâmica de Recursos, cortesia de Bogsnes (2009)

O modelo de desenvolvimento de produtos que discutimos neste livro funciona bem para alocação dinâmica de recursos. Quando temos uma nova ideia, precisamos começar com uma fase de desbravamento. O custo de explorar uma ideia pode ser medido em termos do custo operacional do time. Restrições são definidas: você pode ter um time pequeno por um período definido, e gastar

no máximo X. Uma vez que o time tenha evidências de que a ideia vai entregar valor, podemos prover mais recursos para encaminhar à fase de exploração.

Por todo o Horizonte 3, buscamos usar o Princípio da Opcionalidade para gerenciar nossos investimentos (veja o *capítulo 2* para mais informações sobre o 3 Horizontes e opcionalidade). Nossa objetivo é investir poucos recursos, em um número de possíveis oportunidades, esperando que muitas vão falhar, mas algumas vão se mostrar bem promissoras.

Equipes que conseguem sair do domínio "desbravamento" de forma bem-sucedida e escalam começam a praticar a melhoria contínua, como descrito no *capítulo 6*, constantemente removendo desperdícios no processo de entrega. É essencial evitar "recompensar" as equipes que conseguem melhorias de performance reduzindo seus custos operacionais, reduzindo ou dividindo a equipe. Isso desmotiva instantaneamente as equipes e destrói a mentalidade de inovação. Em vez disso, a equipe deve investir mais tempo em explorar novas ideias sem documentações onerosas, revisões e aprovações — contanto que mantenham suas performances altas e os custos dentro dos limites.

Usando um paradigma de produto em vez de um paradigma de projeto, começa a ficar mais fácil calcular ganhos e perdas em uma base por produto ou por serviço. Podemos calcular os custos de entrega e da realização de um produto ou serviço simplesmente através de custos operacionais da formação da equipe de sua condução. Isso torna mais fácil visualizar quando os custos associados a um produto ou serviço excedem os valores, ou quando não estamos obtendo a margem esperada. Quando queremos construir funcionalidades que estão relacionados a vários produtos, nós podemos usar o Custo do Atraso para basear uma decisão de investimento (veja *capítulo 7*).

Como a proposição de valor e desenvolvimento, bem como os custos de suporte de um produto, mudam pelo seu ciclo de vida, podemos mudar sua composição da equipe. Enfim, quando o produto começa a entregar um valor negativo, devemos retirá-lo assim que possível. Geralmente, um investimento é requerido para que produtos e serviços sejam retirados — e, novamente, Custo do Atraso pode ser usado para fazer a melhor decisão de investimento. Isso requer apoio dos executivos: sabemos de uma empresa da Fortune 500 que bonificou seus VPs com base no número de serviços retirados no ano, com o objetivo de reduzir a complexidade do sistema e promover inovação.

Iniciativas locais, menores e mais simples, que envolvam menos riscos, devem passar por menos revisão e um processo de aprovação mais leves que iniciativas de nível organizacional. Juntamente com isso, precisamos de um processo contínuo e critérios definidos para quando o investimento deve cessar. Revisão e fiscalização podem ser descentralizados criando equipes locais responsáveis por reportarem os resultados de suas decisões de investimento. Isso também pode escalar para o nível organizacional. Ainda queremos manter controles centralizados de alto nível sobre grandes iniciativas organizacionais, mas deve haver menos dessas em um dado momento no tempo. Veja a *Tabela 13-1* para mais exemplos de modelos de investimento.

**Tabela 13-1 — Exemplo de modelos de investimento**

Complexidade do relacionamento	Foco	Taxa de mudança necessária	Modelo de investimento
Simples, um a um	Voltado ao cliente	Rápido — vários times por dia, diariamente ou semanalmente	Pequena duração — 2 semanas. Pequenos times. Pequenos blocos de investimento. Uso de infraestrutura temporária

Complexidade do relacionamento	Foco	Taxa de mudança necessária	Modelo de investimento
Interdependência entre duas ou três equipes de produto	Valor médio — organização do valor de negócio entre equipes de produto	Moderado — 2 a 3 semanas	Pequena duração — 2 a 4 semanas. Equipes de produto misturadas e pequenas. Pequenos blocos de investimento. Inicialmente usa-se infraestrutura temporária.
Nível organizacional	Operações centrais — exemplos: ERPs, CRMs, Big Data, Reporting	Mais lenta — menos do que 4 equipes por ano	Durações longas — 3 a 6 meses. Comece com times pequenos e cresça com o tempo. Decisões de investimento contínuas sempre de 4 a 6 semanas. Blocos de investimento maiores para suportar as mudanças na infraestrutura principal.

Ao se livrar de um ciclo anual de orçamento altamente centralizado, não significa que nossas responsabilidades com uma boa gestão financeira estão diminuindo. Muitas empresas grandes globais, incluindo Handelsbanken, Maersk e Southwest Airlines, começaram uma jornada para evitar grandes orçamentos centralizados e gerenciar custos através de outros meios (LESS, 2012). Elas começaram descentralizando a responsabilidade financeira para operações, direcionando-as para unidades individuais de negócio:

- Gestão sênior não define as metas para todos os custos e receitas do próximo ano fiscal;
- Decisões de negócio críticas não são baseadas em orçamento;
- Equipes e indivíduos não são medidos por suas habilidades em ficar dentro do orçamento.

Todos ainda possuem metas e são cobrados por melhorar o valor que entregam. Entretanto, estas metas não são empurradas pela alta gestão, mas definidas pela própria equipe e alinhados com as metas e objetivos da organização.

## **Explore princípios da contabilidade baseada em atividades**

O consumo de recursos deveria ser diretamente atrelado às atividades que geram valor. Tradicionalmente, os custos são rastreados somente por centros de custos funcionais, como TI, e há pouca visibilidade no que guia estes custos. Departamentos de TI e equipes engajadas em desenvolvimento de produto frequentemente são vistos como centros de custo que podem ser gerenciados e controlados de forma independente do negócio. O pensamento tradicional é que conseguir o fornecimento de serviços mais baratos de TI reduzirá os custos e oferecerá resultados igualmente bons. Se fosse tão fácil assim, você provavelmente não estaria lendo este livro. A realidade é que nosso negócio é quem guia nossos custos de TI e de desenvolvimento de produto e não podemos gerenciá-los independentemente.

A contabilidade (custeio) baseada em atividades permite-nos destinar os custos totais dos serviços e atividades à atividade de negócio ou produto que guia esses custos. Ela nos oferece uma imagem melhor do verdadeiro valor financeiro sendo entregue pelo produto. Contudo, assim como muitos modelos e abordagens em relação ao negócio, a contabilidade baseada em atividades não é uma panaceia. Precisamos ser cuidadosos para não perseguir uma precisão desnecessária, criando modelos complicados e processos mais custosos do que o valor que buscamos oferecer. O objetivo é simplesmente obter as melhores informações para ajustar os planos e atividades para melhorar o valor — começando pequeno e parando quando tivermos evidência empírica suficiente na qual

basear nossas decisões.

#### TOMANDO MELHORES DECISÕES COM CUSTEIO BASEADO EM ATIVIDADES

Aqui está uma história de um trabalho em um antigo emprego que mostra como a contabilidade baseada em atividade traz clareza em como a tecnologia apoia o valor para os clientes.

Em uma tentativa de cortar custos, nosso comitê financeiro executivo havia exigido metas pouco realistas para o próximo ano fiscal. Eles tiveram dificuldades relacionadas aos itens das linhas do orçamento, como custos de servidores, e não entendiam a razão pela qual simplesmente não reduzíamos nossos níveis de pessoal e mantínhamos mais pessoas e sistemas ao mesmo tempo. Para dar uma visão que eles pudessem entender, usamos o princípio da contabilidade baseada em atividades de alocar custos a atividades de negócio (operações, gestão de receita, marketing, relacionamento com o consumidor, gestão da cadeia de fornecimento etc.) em vez de itens de linhas em nosso orçamento (pessoal de TI, software, hardware, IPS, servidores etc.). Nosso sistema de gestão financeira não estava configurado para oferecer tal visão e tínhamos uma data limite apertada, por isso usamos o que tínhamos à mão: planilhas, números atuais da operação, duas pessoas, dois dias e acesso total à gestão sênior da TI para conseguir informações e muita comida e bebida.

Não nos concentrarmos em ser 100% acurados ou precisos: entendemos que algo entre 90 a 95% de acurácia era bom o suficiente. Nossa objetivo era dar aos executivos o panorama geral de como os custos da TI estavam servindo a nossos clientes e ao crescimento da nossa organização.

Felizmente, já tínhamos um entendimento claro dos nossos serviços de TI e custos relacionados às atividades de negócio. Fomos capazes de ligar muitos custos diretamente a um produto ou serviço do negócio. Por exemplo, nosso centro de suporte ao cliente recebeu todos os custos associados com o software de reconhecimento interativo de voz. Outros custos, tais como serviços de e-mail, tiveram de ser divididos entre as unidades de negócio, então usamos o número de pessoas como uma medida para divisão dos custos proporcionalmente. Também alocamos custos específicos para nosso departamento — aqueles relacionados à gestão do serviço do departamento e nosso consumo de serviços comuns.

A saída desse esforço foi uma série de gráficos e tabelas que mostravam como as atividades de negócio dirigiam os custos de TI. Quando apresentados a essa informação no lugar dos tradicionais itens de linha do orçamento, os executivos ficaram mais confortáveis para tomar decisões sobre o que apoiar (ou não) em nossa submissão do orçamento. Ainda mais importante, todos tiveram um panorama melhor do real custo de propriedade dos produtos e serviços do negócio, e pudemos calcular melhor o custo de atraso para a retirada e a troca de sistemas.

### 13.4 EVITE USAR ORÇAMENTOS COMO BASE PARA MEDIR DESEMPENHO

Talvez o maior erro que podemos fazer com os orçamentos é usá-los como um indicador-chave de desempenho como base de recompensa e reconhecimento na capacidade de um indivíduo, uma equipe ou a organização como um todo em seguir um orçamento. Permanecer nos limites do orçamento apenas indica se gastamos ou

ganhamos tanto quanto dissemos que faríamos. Se dissermos às equipes que elas vão gastar mais ou menos do que elas precisam para realizar seus trabalhos, elas encontrarão uma forma de fazer acontecer, ou gastar um grande esforço justificando porque não conseguiram.

Contudo, isso evita que prestemos atenção às questões mais importantes: planejamos no nível correto, estabelecemos boas metas, tornamo-nos mais eficientes ou melhoramos a satisfação dos nossos consumidores? Nossos produtos estão melhorando ou morrendo? Estamos em uma situação financeira melhor do que antes?

Bônus e recompensas para bons resultados financeiros funcionam melhor quando compartilhados igualmente — não apenas com a alta gestão e executivos, mas com todos os empregados da organização. As equipes de trabalho vão em algum momento responder a organização com inércia e subterfúgio quando suas contribuições não forem reconhecidas e as recompensas forem baseadas em um processo percebido como injusto. Por outro lado, as pessoas tendem a seguir bons líderes, e tornam ótima a organização quando o reconhecimento e os incentivos são compartilhados igualmente com todos.

### **INCENTIVOS FINANCEIROS COM IMPACTO POSITIVO: O CASO DA WESTJET**

A WestJet tem sido uma das companhias aéreas mais bem-sucedidas financeiramente na América do Norte pelos últimos 15 anos. Os fundadores da WestJet sabiam que, se fosse para eles terem sucesso, era essencial criar uma cultura de responsabilidade e propriedade para todos os empregados. Para criar esta cultura, eles claramente declararam suas

estratégias e objetivos, filtraram e treinaram empregados para bom encaixe cultural e de valores, assim como estabelecerem incentivos financeiros que beneficiaram todos os empregados.

Duas vezes ao ano, uma parte da receita da companhia é distribuída aos empregados, de acordo com suas bases salariais. Todos os empregados são convidados a participar da festa dos *shareholders*, na qual os cheques são fisicamente entregues aos membros dos times por seus gerentes — cara a cara sempre que possível, logo os gerentes podem reconhecer pessoalmente todos os empregados por suas contribuições.

Além disso, WestJet permite aos funcionários comprar voluntariamente ações da WestJet, com até 20% do seu salário bruto base, e então a empresa contribui com a mesma quantidade em ações ao funcionário. Em 2012, mais de 85% dos funcionários participavam deste programa, se tornando também donos da WestJet.

Essa abordagem tem ajudado a WestJet se manter rentável em uma indústria difícil e altamente regulada por quase duas décadas. WestJet relatou lucro anual em 17 dos 18 anos de seu funcionamento.

Para mais informações, acesse:  
<http://www.westjet.com/pdf/greatWestJetJobs.pdf>,  
<https://www.westjet.com/pdf/global-reporting.pdf> e  
<http://bit.ly/1v73i6N>.

## 13.5 PARE DE BASEAR DECISÕES DE NEGÓCIO EM RECEITA VERSUS CUSTO OPERACIONAL

Preocupação sobre custos de capital (CapEx) *versus* operação

(OpEx) é importante para as organizações. Há vantagens de tarifas e impactos financeiros positivos de se reportar os gastos organizacionais desta forma, logo, muita atenção é gasta com isso. A premissa básica de capitalizar sistemas de software é vê-los como recursos que criam benefícios futuros para nossa organização. Isso pode ter impacto significativo em balancear planilhas e, em troca, no valor de mercado de uma organização.

Infelizmente, esta distinção é geralmente usada como fundação na qual decisões críticas de negócio são tomadas. Isso injeta outro elemento de complexidade no processo decisório e no investimento em inovação. Todos os custos associados com qualquer trabalho precisa ser categorizado em um ou dois buckets, e o processo tradicional para gerenciar os buckets assume que um trabalho em equipe é um ou outro, mas não pode ser ambos ao mesmo tempo.

O processo tradicional também contribui em obscurecer o verdadeiro custo de propriedade e escala custos operacionais. Um projeto será capitalizado totalmente, permitindo-nos divulgar o relatório daquele custo de um extenso período, de forma que ele tenha pouco impacto no curto prazo na receita. Entretanto, muitos dos itens que são capitalizados durante o projeto inicial tem um impacto negativo imediato no nosso OpEx, começando antes ou imediatamente depois que o projeto dissipar. Os custos operacionais de longo prazo necessários para suportar o aumento da complexidade dos sistemas criados pelos projetos não são calculados quando projetos capitalizados são aprovados (porque não vem do mesmo bucket). Suporte contínuo e obsolescência de produtos e serviços é um problema de OpEx. No final, equipes OpEX ficam presas na justificação de seus custos crescentes causados pelo inchaço e complexidade criada por decisões de CapEx.

Se somos sérios sobre inovação, não deveria importar de onde

vem o orçamento. Discussões abertas e francas, baseadas em evidências reais dos custos totais fim a fim dos produtos, são o que devemos usar como base de decisões de negócio. Financiar a alocação de um desenvolvimento de produto com CapEx ou OpEx deve ser feito por contadores depois de que a decisão de negócio é tomada.

Vamos olhar para o domínio "Desbravar" primeiro. Muitas ideias acabam por resultar em valor zero ou negativo, mas o CapEx se aplica aos recursos que entregam valores de longo prazo. Então faz sentido considerar todas as atividades de desbravamento como OpEx. Ainda precisamos de times para definir a quantidade de recursos que pretendem consumir no desbravamento, mas não deveriam precisar obter aprovação para financiamentos futuros toda vez que querem tentar alguma coisa nova dentro dos limites definidos.

Tratando agora do domínio "Explorar", equipes de produto e finanças precisam falar uns com os outros para determinar como alocar os investimentos. Queremos evitar a inserção de complexidade e desperdícios quando decidimos sobre o método de acompanhamento para determinar a alocação de recursos para CapEx ou OpEx. Deve ser fácil o suficiente para todos entenderem como funciona e proverem uma representação acertada e justa da proposta de valor de longo prazo. Ao final, custos alocados a CapEx devem ser tão simples quanto decidir a percentagem de recursos do time (ou tempo) gastos na criação de benfeitorias que serão capitalizadas no futuro. Seguem alguns tópicos para seguem discutidos com o departamento financeiro sobre gastos com CapEx:

- Como podemos criar um modelo flexível para alocar investimentos baseados em princípios de CapEx e OpEx, mas evitar usar regras e processos rígidos que requerem à todos competir por todos os investimentos

ao mesmo tempo?

- Que elementos, além de custos projetizados, devem afetar a quantidade de detalhes e rigor que aplicamos às decisões de investimento: complexidade, estimativas de tempo, tamanho de equipes, efeito líquido dos custos operacionais?
- Como podemos gerenciar iniciativas locais *versus* iniciativas de nível organizacional para reduzir atrasos e o tempo de resposta total para oportunidades locais?
- Como podemos estruturar as decisões de investimento para acomodar cronogramas curtos e melhorar a disponibilidade de mais equipes?
- Como baseamos decisões de investimentos posteriores em entregas de produtos funcionais demonstradas, em oposição à quantidade de atividades?

Nessas discussões, é importante levar em conta o tempo esperado de vida do produto. Tecnicamente, bens de software deveriam ser capitalizados apenas se o tempo esperado de vida do produto for igual ou superior ao período atual de depreciação para produtos de software — a maioria das empresas atualmente utiliza três anos. Contudo, parece pouco realista acreditar que todos esses sistemas tenham um período de vida que gere valor por três anos caso seja mantido sem alterações. Isto representa uma questão interessante. O que é menos arriscado e mais responsável:

- Categorizar o produto de software inteiramente como OpEx; ou
- Capitalizar os custos e resgatar um desconto no futuro se o produto for aposentado antes de ser totalmente depreciado?

Provavelmente não existe uma única resposta definitiva; será necessário considerar muitas variáveis que são diferentes para cada organização.

## 13.6 ALTERE SEUS PROCESSOS DE AQUISIÇÃO PARA GANHAR MAIOR CONTROLE SOBRE A ENTREGA DE VALOR

Em seu livro *Out of the Crisis*, Edwards Deming (2000) propôs 14 princípios de gestão necessários para que as empresas americanas pudessem melhorar a efetividade de seus negócios. O princípio de número 4 da lista declara: "Pare com a prática de contratar o negócio com base no preço. Em vez disso, minimize o custo total. Vá na direção de um único fornecedor para cada item em uma relação de longo prazo de lealdade e confiança".

Mais de 30 anos mais tarde, vemos muitas organizações que não capturaram o verdadeiro sentido deste princípio. Falhamos em quantificar o custo total e tratamos produtos e serviços como commodities fungíveis que podem ser facilmente produzidas por quaisquer fornecedores. Enquanto processos de aquisição podem resultar em relações de longo prazo, esses raramente são construídos sobre colaboração e confiança.

Na história da NUMMI, no *capítulo 1*, alguns dos problemas que a GM enfrentou ao adotar TPS têm relação ao fato de que fornecedores não estão acostumados com a ideia de trabalhar colaborativamente para melhorar a qualidade e a especificação de partes em resposta aos resultados em campo. Esta é uma analogia direta aos problemas que encontramos quando terceirizamos o desenvolvimento e, ao fim do contrato, descobrimos que o produto entregue não se encaixa no propósito.

O resultado líquido é que as políticas, os processos e práticas de

aquisição conspiram para nos impedir de melhorar o valor que oferecemos pela entrega de software.

O primeiro erro que cometemos é pensar que, com grandes quantidades de planejamento de antemão, podemos gerenciar o risco de chegar a algo que não entrega o valor esperado. Em muitas empresas grandes, o gerente terceirizado de serviços de entrega de software deve definir todas as saídas de antemão, na forma de uma requisição de proposta (RFP — *Request for proposal*). Isto é seguido por respostas detalhadas dos fornecedores sobre como eles entregariam as saídas esperadas e a que custo, frequentemente calculadas com base nos valores por hora e gastos somados a uma margem desejada. As decisões de contratação, então, são baseadas nas respostas recebidas e apresentações realizadas pelos grupos que entram na lista reduzida.

Este processo contratual doloroso e altamente detalhado tem vários efeitos colaterais negativos:

- **É uma forma ruim de gerenciar riscos do desenvolvimento de produto.** É com base na ideia equivocada de que podemos saber exatamente o que precisamos de antemão — em outras palavras, que durante a construção do sistema não faremos nenhuma descoberta significativa sobre o que os usuários acham valioso, nem encontraremos nenhuma complexidade inesperada significativa.
- **Favorece os incumbentes.** Fornecedores que já tem uma presença na organização tem acesso mais fácil às pessoas de forma que podem entender melhor seus orçamentos e metas a serem atingidas. Como o processo de aquisição para novos fornecedores é tão doloroso, é mais fácil renovar automaticamente os contratos estabelecidos mesmo com fornecedores

medíocres. Os custos percebidos e os riscos associados com a busca por um novo fornecedor frequentemente são considerados maiores do que renovar contratos existentes.

- **Favorece grandes fornecedores.** Grandes companhias empregam pessoas que se especializam em responder RFPs, o que pode ser feito com minúcias e caixas de seleção, mas que tem pouco ou nenhum efeito sobre os resultados.
- **Inibe a transparência.** O sucesso é normalmente recompensado em relação ao custo de entrega, com pouca preocupação com os custos relacionados de integração, de mudança nos processos de negócio ou nos custos operacionais do dia a dia. Raramente vemos se levar em conta o quanto os produtos oferecidos afetarão a entrega de valor de ponta a ponta. A entrega *offshore*, por exemplo, normalmente parece barata em termos de custo por unidade. Contudo, quando levamos em conta o aumento nos custos de comunicação e de viagens, assim como demoras nas respostas e retrabalho devido a diferenças de fuso horário, pode facilmente levar mais tempo que a entrega feita localmente — em custos gerais semelhantes.
- **É pouco acurado.** Devido à falácia do planejamento (veja o *capítulo 2*), tanto os fornecedores quanto os gerentes tendem a ser otimistas demais em suas estimativas sobre a quantidade de pessoas e de trabalho necessária, pois eles sabem que o preço do contrato tem mais peso na decisão. Os fornecedores sabem que podem ganhar mais dinheiro com requisição de

mudanças.

- **Ignoram-se os resultados.** O desempenho dos contratos é medido em relação à habilidade de oferecer serviços conforme contratado, no valor/hora contratado, pelo período contratado. Normalmente, não existe nenhuma menção aos resultados dos serviços entregues. Independente dos resultados, os fornecedores normalmente são recompensados com mais contratos para suporte, correção de problemas e melhora do serviço.

Caso você esteja com dificuldades de sair de um contrato de longo prazo, baseado em projeto, que especifica a entrega de uma solução ao final de um termo, mitigue os riscos oferecendo pagar o fornecedor de forma adiantada, com base na entrega de software incremental em funcionamento. Isso incentiva-os a lhe oferecer software que pode ser posto em produção e assim obter feedback dos clientes sobre o valor da solução. Dessa forma, você pode ajustar a direção do desenvolvimento do produto, com base nos resultados teste.

O segundo erro no processo de aquisição típico é que ele assume que todos os serviços são iguais em termos tanto de qualidade das pessoas trabalhando na entrega quanto na qualidade do software entregue. Depois de muita experiência dolorida, muitos de nós sabemos que não é o caso. Tem muitos fatores que determinam quão bem-sucedido será o resultado do engajamento com um fornecedor e o que menos importa é o custo. Quais são as taxas de ocorrência de erros e de correção deles? Qual o esforço associado com manutenção sobre a solução deles? Quantas linhas de código

compõem a solução (menos é melhor)? Com qual proximidade podemos trabalhar com eles? Podemos confiar neles? Apesar de podermos ganhar algum insight conversando com outros clientes do fornecedor, o teste *acid* é experimentação. Precisamos testar a relação e medir o resultado — o que requer modificar os processos que usamos para engajar fornecedores de serviço terceirizados.

### O GOVERNO DO REINO UNIDO MUDA SEU PROCESSO DE AQUISIÇÃO PARA ENCORAJAR A INOVAÇÃO

No começo de 2014, o governo do Reino Unido anunciou mudanças drásticas nas regras aplicadas à contratação e gestão dos contratos de serviço de TI (<http://bit.ly/1v73rXY>). Essas mudanças tinham como objetivo encorajar a competição no setor de serviço de TI e ajudar o governo a se tornar um cliente mais inteligente dos fornecedores de serviço. Eles acreditam que obterão melhores resultados em seus serviços de TI limitando contratos de TI grandes e aumentando a fonte de potenciais fornecedores.

Para facilitar que empresas de pequeno e médio porte possam fazer ofertas para contratos de TI do governo, quatro grandes mudanças foram anunciadas no processo de aquisição de serviços de TI do governo do Reino Unido:

- Nenhum contrato acima de £100m será assumido, exceto sob circunstâncias excepcionais.
- Empresas com um contrato para provisão de serviços não será permitido oferecer integração de sistemas na mesma parte do governo.
- Novos contratos de hospedagem terão duração máxima de dois anos.
- Não haverá renovação automática de contratos.

O governo do Reino Unido espera criar serviços mais eficientes e responsivos que atendam às demandas do público por meio de melhor acesso a soluções digitais inovadoras e com bom custo-benefício. Elas estão expandindo a variedade de fornecedores e criando mais oportunidades para avaliar e negociar os contratos que podem ser baseados em tecnologias desatualizadas e caras ou que apenas não estejam entregando valor.

O manifesto ágil diz que devemos preferir colaboração com o cliente mais do que negociação de contratos. Nós precisamos continuamente trabalhar com nossos fornecedores para criar resultados de alta qualidade. Os melhores relacionamentos e resultados são alcançados, quando não jogamos os requisitos por cima do muro, e então esperamos que um produto ou serviço apareça magicamente depois de alguns meses. Precisamos estar engajados, gerenciar contratos e relacionamentos, promover flexibilidade, e procurar oportunidades de experimentar diferentes fornecedores, para que possamos avaliar suas competências e suas capacidades de entregar valor.

## 13.7 CONCLUSÃO

Organizações que continuam estruturando decisões de financiamento em ciclos financeiros vão encarar sérios obstáculos para melhorar suas capacidades de inovação. Precisamos avançar para além do paradigma do orçamento centralizado e introduzir fluidez no processo de previsão financeira, planejamento e monitoramento. Isso é essencial se nós queremos ver mais do que benefícios incrementais de se aplicar princípios Lean.

Devemos desassociar decisões de investimento do ciclo de

orçamento anual, e parar de se preocupar se é capital ou despesa operacional. É assim que podemos tomar melhores decisões sobre o que e quando devemos investir para criar resultados que queremos. Nós ainda precisamos gerenciar nossos custos com cuidado, mas resultados melhores podem ser alcançados por meio de menores ciclos de planejamento, previsibilidade e monitoramento, juntamente com custos relacionados às atividades de negócio que os guiam. Restringir recursos e cronogramas para testar ideias nos ajuda facilmente a cortar investimentos das ideias que não vingam.

Finalmente, para apoiar inovação e experimentação, precisamos modificar nossos processos e regras de aquisição. Sem relacionamentos de longo prazo e de confiança, baseados em desejos mútuos de se tornar melhor em entregar valor, organizações grandes serão para sempre dificultadas por sistemas legados e produtos que estão sempre desatualizados antes mesmo de serem lançados.

## Questões para os leitores

- Seu time de produto pode experimentar abertamente novas ideias e tecnologias sem gastar grandes quantidades de tempo procurando aprovação e investimentos? Você pode facilmente obter investimentos para novas tecnologias a qualquer momento do ano, ou está restrito ao ciclo anual?
- Quais os seus critérios para um investimento de sucesso? É suficiente para os projetos estar no prazo e no orçamento, ou você intenciona medir os resultados dos consumidores e da organização?
- Quanto tempo é gasto gerenciando o orçamento das equipes no ano, incluindo relatórios de performance e justificando variações?

- Quão longe e quão frequente você espera planejar o detalhamento dos custos? Existe algum processo mais fácil para ajustes e relatórios contínuos dentro do plano?
- O processo de alocação de gastos *versus* gastos operacionais atrapalha as pessoas de realizar decisões de investimento responsáveis? Se sim, existe um jeito mais simples, de baixo risco, para experimentar?

## CAPÍTULO 14

# TORNE A TI EM UMA VANTAGEM COMPETITIVA

*“O padrão de centro de custo preenche o vácuo da nossa inabilidade de definir, modelar e medir o valor que a maior parte dos trabalhadores criam para a organização.” — Ken H. Judy*

Os departamentos de TI da empresa enfrentam forças poderosas e conflitantes. A primeira prioridade deles é manter em pé os existentes sistemas críticos em relação ao negócio, conforme tais sistemas envelhecem e aumentam em complexidade. Existe também uma pressão crescente de aumentar a velocidade em que novas funcionalidades e produtos podem ser entregues. Por fim, a TI tradicionalmente tem sido vista como um centro de custo de forma que existe uma pressão constante por eficiência (o que normalmente se manifesta em cortes de custo).

Estes objetivos conflitantes frequentemente levam a uma espiral negativa. Reduzir a complexidade e substituir sistemas legados exige investimento. Contudo, investimentos frequentemente vêm na forma de projetos multianuais que muitas vezes são abandonados

ou lançados ainda incompletos devido a custos crescentes e/ou mudanças de pessoal no nível executivo. Esta complexidade crescente, junto com a necessidade de ganhos mais profundos de eficiência, reduz a capacidade da TI de gerenciar o trabalho planejado de forma efetiva. O aumento na demanda por mudanças, quando em conjunto com um ambiente de TI instável, leva à proliferação de trabalho não planejado que reduz ainda mais a capacidade de TI.

Neste capítulo, vamos discutir algumas estratégias para aumentar a responsividade da TI para as necessidades de negócio em constante mudança, para melhorar a estabilidade dos serviços de TI e reduzir a complexidade dos nossos sistemas e infraestrutura de TI. Muitas destas estratégias vêm do movimento DevOps que tem como objetivo permitir que trabalhemos com segurança em escala em um ambiente de alta velocidade e altas consequências.

## 14.1 REPENSANDO A MENTALIDADE DA TI

A TI historicamente tem sido vista como um centro de custo e uma capacitadora de necessidades de negócio, não uma criadora de vantagem competitiva. Por muitos anos, a ortodoxia foi essa, como Nicholas Carr (2003) infamemente colocou, "A TI não tem importância" (veja mais em [http://www.nicholascarr.com/?page\\_id=99](http://www.nicholascarr.com/?page_id=99)). Mesmo entre os praticantes do Lean, a TI é algumas vezes vista como "apenas um departamento." Isto criou o que Marty Cagan (2008), autor de *Inspired: How to Create Products Customers Love*, chama de "mentalidade de TI", na qual a TI é simplesmente uma provedora de serviços para "o negócio" (figura a seguir).

*“Que nível de influência o seu parceiro provedor de software exerce no momento de decidir qual produto ou serviço o seu negócio deve entregar?”*

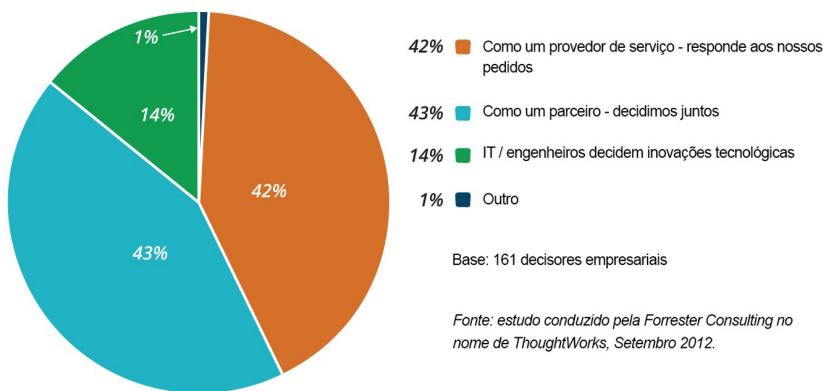


Figura 14.1: O que os líderes de negócios pensam sobre a relação negócio-TI

Este problema é exacerbado pelo típico modelo projetizado pelo qual os projetos de TI são financiados e gerenciados. O trabalho criado nos projetos de TI é tipicamente entregue (ou jogado) para operação de TI executar, assim as pessoas gerenciando os projetos têm pouco incentivo para pensar sobre as consequências de longo prazo de suas decisões de design — e grandes incentivos para lançar o máximo possível de funcionalidades em um tempo extremamente apertado em geral. Isto resulta em software difícil de operar, mudar, lançar, manter e monitorar, tudo isso adiciona complexidade aos ambientes operacionais, o que, por sua vez, torna mais difícil entregar os próximos projetos (BOTTCHER, 2010).

Como Charles Betz (2006, p. 300), autor de *Architecture and Patterns for IT Service Management, Resource Planning, and Governance*, diz:

*“Devido ao fato de ser a área melhor compreendida de atividade da TI, a fase de projetos é frequentemente otimizada em detrimento das outras áreas de processo, e portanto às custas da cadeia de valor como um todo. O desafio da gestão de projetos de TI é que os*

*objetivos mais amplos da cadeia de valor estão frequentemente 'fora de escopo' para um projeto em particular, e os projetos não são responsabilizados por suas contribuições para a entropia geral do sistema".*

Operações — um departamento dentro do departamento de TI e talvez visto como um verdadeiro centro de custos — vivencia as consequências destas decisões diariamente. Em particular, os sistemas integrados que eles devem manter em execução são incrivelmente complexos e intrincados, construídos durante anos e muitas vezes frágeis, então Operações evita mudá-los. Dado que a estabilidade é a prioridade número um, Operações desenvolveu a reputação de departamento que diz "não" — uma resposta completamente racional ao problema em mãos.

O departamento de Operações tem dois mecanismos primários usados para deter a correnteza: o processo de gestão de mudanças e a padronização. O processo de gestão de mudanças é utilizado para mitigar os riscos de mudanças nos ambientes de produção e também atender a requisitos regulatórios. Isto geralmente exige que cada mudança em produção seja revisada por uma equipe (conhecida como Comitê Consultivo de Mudança ou, mais popularmente em inglês, *Change Advisory Board*, nos termos da terminologia ITIL) antes de ser lançado. A padronização é usada para gerenciar a heterogeneidade dos ambientes de produção, reduzir os custos e prevenir falhas de segurança; também exige que todos os softwares utilizados em produção (e frequentemente em ambientes de desenvolvimento também) sejam aprovados para o uso.

O resultado destes processos é que a taxa de mudanças diminui enormemente nos ambientes de produção e as equipes não conseguem utilizar as ferramentas de preferência. Em certas circunstâncias, isto pode ser uma troca aceitável se estas limitações

puderem de fato melhorar a estabilidade dos ambientes de produção. Contudo, os dados mostram que não é o que acontece. Na verdade, muitas das presunções que fundamentam a área de operações dos departamentos de TI e as suas relações com outras partes da organização não são mais válidas.

No relatório *2014 State of DevOps*, mais de 9.000 pessoas mundo afora foram pesquisadas sobre o que cria organizações de alto desempenho, se a TI realmente é importante para o negócio e quais fatores impactam o desempenho dos departamentos de TI (FORSGREN; KIM; KERSTEN; HUMBLE, 2014). O primeiro grande resultado da pesquisa foi uma forma estatisticamente válida de medir o desempenho da TI. Organizações com TIs de alto desempenho são capazes tanto de conseguir alta vazão, medida em termos de tempo de aplicação das mudanças e frequência de deploy, quanto de alta estabilidade, medida como o tempo para restaurar o serviço após uma queda ou um evento que tenha causado degradação de qualidade do serviço. Organizações com TIs de alto desempenho também têm taxas 50% menores de falhas de mudança que as de desempenho médio e baixo.

Os dados mostram que as organizações com TI de alto desempenho são capazes de conseguir níveis mais altos tanto de vazão quanto de estabilidade. Além disso, as empresas com TI de alto desempenho têm probabilidade duas vezes maior de ultrapassar seus objetivos de rentabilidade, participação de mercado e de produtividade que aquelas com TI de baixo desempenho.

As práticas que mais se correlacionam com um TI de alto desempenho (tanto em vazão quanto em estabilidade) são:

- Manter as configurações do sistema, as configurações da aplicação e o código da aplicação no controle de versionamento.

- Sistemas de registro e monitoramento que produzem alertas de falha.
- Desenvolvedores quebrarem grandes funcionalidades em mudanças pequenas e incrementais que são integradas ao *trunk* diariamente (conforme discutido no *capítulo 8*).
- Desenvolvedores e Operações regularmente conseguem resultados bons para ambos quando interagem.

Existem dois outros fatores que indicam fortemente uma TI de alto desempenho. A primeira é uma cultura organizacional de alta confiança como descrito no *capítulo 1*. A segunda é um processo leve de aprovação de mudanças feito pelos pares. Muitas organizações possuem uma equipe independente para aprovar alterações que vão para produção. Contudo, os dados mostram que, enquanto tais processos diminuem significativamente a vazão, eles só têm um impacto negligenciável sobre a estabilidade. Mecanismos de aprovação de mudanças pelos pares (tal como programação em par ou revisão de código por outros desenvolvedores) são tão efetivos na criação de ambientes estáveis como Comitês Consultivos de Mudança — mas permitem uma vazão drasticamente superior.

Apesar de estes dados apoiarem as práticas existentes de empresas de alto desempenho, como a Amazon e o Google, eles diretamente contradizem a sabedoria dada de que a segregação de responsabilidades é uma forma efetiva de gerenciar riscos. Contudo, o trabalho de Westrum sobre cultura de segurança mostra que nenhum processo ou controle consegue compensar pelos efeitos de um ambiente no qual as pessoas não se preocupam com os resultados para o cliente e para a organização. Em vez de criar controles para compensar por culturas patológicas, a solução é criar uma cultura na qual as pessoas assumem responsabilidade pelas

consequências de suas ações — em especial os resultados para o cliente.

Existe uma receita simples, mas de grande alcance, para permitir este comportamento:

1. *Você constrói, você opera.* As equipes que constroem novos produtos e serviços devem assumir a responsabilidade pela operação e suporte destes serviços, pelo menos até que eles estejam estáveis e o peso da operação e suporte tenham se tornado previsíveis. Ao fazer isto, também garantimos que é fácil medir o custo de operar o serviço e o valor que ele entrega.
2. *Torne a TI central em uma organização de desenvolvimento de produto.* O ciclo de vida do produto e as estratégias descritas neste livro deveriam ser usados para entregar produtos e serviços internos assim como para o cliente externo.
3. *Invista em reduzir a complexidade dos sistemas existentes.* Utilize a capacidade recuperada no passo 1 para investir em trabalhos de melhoria do dia a dia com o objetivo de reduzir o custo e o risco de fazer mudanças em serviços existentes.

## 14.2 LIBERDADE E RESPONSABILIDADE

Para reduzir o peso sobre Operações, é essencial que as equipes passem a suportar os novos produtos, serviços e funcionalidades que constroem. Para isto, precisamos dar a elas tanto autonomia para lançar e operar novos produtos e funcionalidades quanto a responsabilidade de dar suporte a estes.

No Google, as equipes trabalhando em um novo produto devem passar por uma "revisão de prontidão para produção" antes que qualquer serviço possa ser colocado no ar. A equipe de produto é,

então, responsável por seus serviços quando inicialmente eles vão ao ar (de forma similar ao conceito do ITIL de Suporte para Período de Funcionamento Experimental).

Após alguns meses, quando o serviço se estabilizou, a equipe de produto pode pedir a Operações — chamada no Google de Engenheiros de Confiabilidade do Site — a assumir a operação do dia a dia do serviço, mas não antes de passar por uma "revisão de prontidão para passagem de bastão", para garantir que o sistema esteja pronto para tal passagem. Se o serviço encontra um problema sério após a passagem, a responsabilidade pelo suporte é transferida de volta à equipe de produto até que eles possam passar por outra revisão de prontidão (LIMONCELLI; CHALUP; HOGAN, 2014).

Para mais, veja a palestra *Thousands of DevOps Since 2004* de Tom Limoncelli, feita na SRE@Google de 2012, em <https://www.youtube.com/watch?v=iIuTnhdTzK0>.

Como discutido no capítulo 8, *risco e conformidade*, este modelo requer que as equipes de produto trabalhem com outras partes da organização responsáveis pela conformidade, segurança de informação e operações de TI durante todo o processo de desenvolvimento. Em particular, departamentos centralizados de TI são responsáveis por:

- Oferecer documentação clara e atualizada sobre quais processos e aprovações são necessárias para que serviços novos possam ser lançados e como as equipes podem acessá-los.
- Monitorar o tempo de atravessamento e outros SLAs para estes serviços, tais como aprovação de pacotes de

software, provisionamento de infraestrutura (a exemplo de ambientes de teste), e trabalhar para constantemente reduzi-los.

Para serviços no ar ainda com desenvolvimento ativo, os desenvolvedores compartilham igual responsabilidade com operações para (adaptado do post de John Allspaw, em <https://gist.github.com/jallspaw/2140086>):

- Responder a quedas e fazer plantões;
- Projetar e evoluir os sistemas de monitoramento e alerta, assim como as métricas das quais eles dependem;
- Configurar a aplicação;
- Projetar e revisar a arquitetura.

Os engenheiros que constroem novas funcionalidades deveriam poder colocar código em produção sem depender de ninguém, seguindo uma revisão por pares, exceto em caso de mudanças de alto risco. Contudo, eles devem estar disponíveis quando tais mudanças vão para o ar, de forma que possam dar suporte. Muitas mudanças de novo código (particularmente os de alto risco) deveriam ser lançadas "escuras" (como descrito no *capítulo 8*) e, ou desligadas em produção ou parte de um teste A/B.

Algumas pessoas descrevem este modelo como "no-ops", já que (caso seja bem-sucedido) nós drasticamente reduzimos a quantidade de trabalho de suporte reativo que o pessoal de operações deve realizar. Realmente equipes executando todos seus serviços em uma nuvem pública podem levar este modelo à sua conclusão lógica na qual as equipes de produto têm controle completo sobre — e responsabilidade pela — a construção, lançamento e execução dos serviços por todos os seus ciclos de vida (um modelo inaugurado em escala pelo Netflix). Isto levou a uma

boa dose de resistência das pessoas de operações que ficaram receosas de perderem seus empregos.

O rótulo "no-ops" é claramente provocativo, e entendemos como problemático; no modelo que descrevemos, a demanda pelas habilidades de operações na verdade é ainda maior, porque as equipes de entrega devem assumir a responsabilidade de construir, evoluir, operar e dar suporte aos produtos e serviços da organização. É verdade que as pessoas de operações tradicionais terão de passar por um período de intenso aprendizado e mudança cultural para serem bem-sucedidas neste modelo — mas isto é verdade para todos os papéis dentro de organizações adaptativas.

### NoOps

Este termo foi cunhado por Mike Gualtieri (2011). Respostas de John Allspaw da Etsy e Adrian Cockcroft da Netflix podem ser vistas em <https://gist.github.com/jallspaw/2140086>.

Deve-se reconhecer e aceitar que isto será assustador para muita gente. Suporte e treinamento devem ser oferecidos para ajudar aqueles que desejarem fazer a transição. É importante deixar claro que o modelo que descrevemos não tem como objetivo tornar as pessoas redundantes — mas todos precisam estar dispostos a aprender a mudar (veja o *capítulo 11*). Pacotes generosos para desligamentos deveriam ser oferecidos para aqueles que não estiverem interessados em aprender novas habilidades e assumir novos papéis na organização.

Remover o peso de criar e manter novos produtos e serviços libera as organizações de TI para que possam dar enfoque em operações, evoluir os serviços existentes e construir ferramentas e

plataformas para apoiarem as equipes de produto.

### 14.3 CRIANDO E EVOLUINDO PLATAFORMAS

O papel mais importante da TI central é apoiar o resto da organização, incluindo a gerência de bens, como computadores e licenças de software, a provisão de serviços de telefonia, gerência de usuários e infraestrutura. Isto é verdade tanto para organizações de alto quanto de baixo desempenho. A diferença está em como estes serviços são gerenciados e oferecidos.

Tradicionalmente, as empresas têm confiado em pacotes de fornecedores externos (como Oracle, IBM e Microsoft) para oferecer componentes de infraestrutura como bases de dado, armazenagem e poder computacional. Ninguém poderia perder a onda de ir para o paradigma da computação conhecido como "nuvem". Contudo, apesar de poucas empresas conseguirem evitar o movimento, muitas estão falhando em executá-lo corretamente.

Para serem bem-sucedidas, as organizações de TI devem escolher entre dois caminhos: ou terceiriza para fornecedores externos de infraestrutura ou plataforma como um serviço (IaaS ou PaaS), ou constrói e evolui a própria.

Apesar de que fazer esse movimento para fornecedores externos de nuvem tenha riscos diferentes se comparados com gerenciar a infraestrutura dentro de casa, muitas das razões comumente apontadas para se criar um "nuvem privada" não sobrevivem a uma análise um pouco mais minuciosa. Os líderes deveriam tratar objeções em relação a custo e segurança da informação com ceticismo: é razoável supor que a equipe da sua empresa responsável por segurança da informação fará um trabalho melhor que a Amazon, a Microsoft ou o Google, ou que sua organização será capaz de contratar hardware a um custo menor?

Dado que invasões em redes corporativas tornaram-se comuns (e às vezes até apoiada por estados), a ideia de que a informação de alguma forma está mais segura atrás dos firewalls corporativos é absurda. A única forma de proteger efetivamente a informação é por meio de criptografia forte associada com higiene rigorosa em relação ao gerenciamento de chaves e controles de acesso. Isto pode ser atingido tão efetivamente na nuvem quanto em uma rede corporativa. Muitas organizações estão construindo e executando alguns de seus datacenters para a Amazon (KONKEL, 2014). Muitos países agora estão atualizando suas regras para explicitamente permitir que dados sejam gravados em uma infraestrutura gerenciada externamente.

Existem duas grandes razões para ser cauteloso sobre nuvens públicas. O primeiro risco é ficar preso a um fornecedor, o que pode ser mitigado por meio de escolhas cuidadosas de arquitetura. O segundo é a questão da soberania. Qualquer empresa armazenando seus dados na nuvem "está sujeita às leis do país hospedando os servidores quanto às leis locais em relação a como aquela informação deveria ser protegida, levando a um potencial conflito de leis sobre a soberania dos dados. As implicações dessas sobreposições de obrigações legais dependem das leis específicas dos países envolvidos na relação e dos acordos entre os governos" (BARR, 2014).

Mesmo assim, existem razões convincentes para escolher fornecedores públicos de nuvem, razões como custos mais baixos e desenvolvimento mais rápido. Em particular, nuvens públicas permitem que as equipes de engenharia criem sua infraestrutura instantaneamente sob demanda. Isto reduz significativamente o tempo e o custo de desenvolver novos serviços e evoluir os existentes. No meio tempo, muitas empresas que alegam ter implementado "nuvens privadas" ainda exigem que os engenheiros abram chamados para pedir ambientes de teste e de produção, o que

leva dias ou semanas para serem provisionados.

Qualquer projeto de implementação de nuvem que não resulte nos engenheiros sendo capazes de criar ambiente ou fazer lançamentos instantaneamente sob demanda utilizando uma API deve ser considerado um fracasso. O único critério para o sucesso de uma rede privada deveria ser um aumento substancial do desempenho geral da TI com base em métricas como vazão e estabilidade apresentadas acima: tempo de aplicação de mudança, frequência de lançamento, tempo para restaurar um serviço e taxa de falhas de mudança. Isto, em contrapartida, resulta em maior qualidade e custos menores, assim como libera capital para investir no desenvolvimento de novos produtos e na melhor dos serviços e infraestrutura existentes.

A alternativa para o uso de um fornecedor externo é desenvolver internamente sua própria plataforma de entrega de serviço. Uma plataforma de entrega de serviço (SDP — *Service Delivery Platform*) lhe permite automatizar todas as atividades rotineiras associadas com a construção, teste e lançamento de serviços, incluindo o provisionamento e a gestão diária dos serviços de infraestrutura. Também é a fundação na qual os pipelines de implantação para a construção, testes e lançamento individual de serviços. *The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems* é um excelente guia para projetar e executar uma plataforma de entrega de serviço (LIMONCELLI; CHALUP; HOGAN, 2014).

Contudo, empresas que tiveram sucesso na criação de sua própria SDP (pelos critérios anteriores) tipicamente não o fez pelo caminho tradicional de TI de comprar, integrar e operar pacotes comerciais.

Isto reflete a forma como a Toyota aborda a questão do maquinário. Norman Bodek (2004, p. 37) relata que “A Toyota e os principais fornecedores, em vez de comprar máquinas que fariam ‘tudo que fosse possivelmente necessário no futuro’, eles preferem fabricar mais de 90% do seu maquinário para executar um trabalho específico necessário no momento”.

Em vez disso, elas utilizaram o paradigma de desenvolvimento de produto descrito neste livro para a criação e evolução de um SDP, preferindo usar componentes de código aberto como fundação. Esta abordagem exige uma mudança substancial de ferramental e realinhamento da TI para dar enfoque em explorar novas plataformas testando-as com um subconjunto de clientes internos (como discutido na *Parte II*), com o objetivo de entregar valor cedo e oferecer desempenho superior ao de fornecedores externos. Produtos validados deveriam ser evoluídos usando os princípios da *Parte III*, utilizando equipes de produto multifuncionais e medindo o sucesso delas pelas métricas de desempenho de TI anteriores.

## Preparando-se para desastres

As organizações que escolhem gerenciar sua própria SDP devem levar a continuidade de negócio muito seriamente. A Amazon, o Google e o Facebook injetam falhas em seus sistemas de produção de forma regular para testar seus processos de recuperação de desastres. Nestes exercícios, chamados de Dias de Jogos na Amazon e Testes de Recuperação de Desastres (DiRT — *Disaster Recovery Testing*) no Google, uma equipe dedicada é montada para planejar e executar um cenário de desastre.

Tipicamente isto inclui desligar fisicamente a energia elétrica de datacenters e desconectar as conexões de fibra com os escritórios ou datacenters. Isto tem consequências reais, mas é reversível em um evento de uma falha incontrolável. Espera-se que as pessoas executando os serviços afetados cumpram os acordos de nível de serviço (SLA) e as interrupções são planejadas cuidadosamente para não exceder os limites do que é necessário para executar o serviço. Principalmente, uma análise sem julgamento é feita após cada exercício (veja o *capítulo 11*) e as melhorias propostas são testadas algum tempo depois.

Kripa Krishnan, o gerente de programa do Google para os exercícios DiRT, comenta que "para eventos no estilo DiRT serem bem sucedidos, uma organização precisa primeiro precisar aceitar as falhas de sistemas e processos como meios de aprendizado. Coisas ruins acontecem e quando acontecem, o foco precisa ser em resolver o erro em vez de repreender um indivíduo ou equipe por uma falha de sistemas complexos... Nós projetamos testes que exigem a interação de engenheiros de diversos grupos que podem não trabalhar juntos diariamente. Desta forma, se um desastre de larga escala vier um dia a ocorrer, estas pessoas já terão estabelecido uma relação forte de trabalho" (ROBBINS; KRISHNAN; ALLSPAW; LIMONCELLI, 2012).

A Netflix leva essa ideia ao seu extremo lógico ao executar um conjunto de serviços conhecido como o Exército Símio, liderado pelo Chaos Monkey, um serviço que desliga servidores de produção em intervalos regulares para testar a resiliência do ambiente de produção. Como muitos sistemas da Netflix, o software por trás do Exército Símio é de código aberto e disponível no GitHub. As organizações que não têm estômago para realizar exercícios de injeção de falhas pelo menos anualmente não deveriam estar no jogo de desenvolver seus próprios serviços de infraestrutura — pelo menos não para sistemas críticos.

Finalmente, as organizações que desenvolvem seus próprios serviços de infraestrutura devem dar a opção a seus clientes internos a escolha de usá-los ou não. As empresas se apoiam na padronização dos serviços e bens oferecidos por operações da TI para gerenciar custos de suporte, por exemplo, mantendo uma lista de ferramentas e componentes de infraestrutura aprovados das quais as equipes podem escolher. Contudo, tendências como empregados levarem seus próprios dispositivos para o trabalho e equipes de desenvolvimento de produto usando componentes de código aberto não padrão (como bases de dado NoSQL) apresentam um desafio para este modelo.

Vimos casos nos quais componentes de código aberto eram necessários para conseguir níveis de desempenho, manutenibilidade e segurança exigidos pelos seus clientes, mas tiveram resistência de seus departamentos de operações de TI — resultando em uma grande quantidade de tempo e dinheiro desperdiçados tentando forçar os produtos a executar nos pacotes existentes.

A forma correta de tratar este problema é permitir que as equipes de produto utilizem as ferramentas e componentes que preferirem, mas exigir deles que assumam os riscos e custos de gerenciar e operar os produtos e serviços construídos por eles — para repetir a frase do CTO da Amazon, Werner Vogels, "Você constrói, você opera". Lembre-se da definição Lean sobre desempenho ótimo do *capítulo 7*: "Entregar valor para o cliente de uma forma na qual a organização não incorra em gastos desnecessários; o trabalho flui sem atrasos; a organização é 100% conforme com todas as leis locais, estaduais e federais; a organização atende todos os requisitos definidos pelos clientes; todos os empregados estão seguros e são tratados com respeito. Em outras palavras, o trabalho deveria ser projetado para eliminar atrasos, melhorar a qualidade e reduzir custos, esforços e frustração desnecessários" (MARTIN; OSTERLING, 2014, p. 101). Processos

que inibem um ótimo desempenho deveriam ser alvo de melhoria.

## 14.4 GERENCIANDO SISTEMAS EXISTENTES

Uma plataforma de entrega de serviços, seja criada internamente ou servida por um fornecedor, deve garantir a padronização e redução dos custos de execução de novos sistemas. Contudo, ela não ajudará a reduzir a complexidade dos sistemas existentes. Um grande número de sistemas existentes é um dos maiores fatores limitantes para a habilidade de departamentos de TI das empresas de se moverem rapidamente.

Nos departamentos de operações que devem manter centenas ou milhares de serviços existentes, colocar no ar mesmo uma funcionalidade nova aparentemente simples pode envolver múltiplos sistemas, e qualquer tipo de mudança em produção está repleta de riscos. Obter ambientes de teste integrados para tais mudanças é caro — mesmo uma parte do ambiente de produção não pode ser reproduzida sem um grande esforço (e normalmente é difícil dizer quanto precisa ser reproduzido, e em qual nível de detalhe, para propósito de execução dos testes). Junte isso com os silos funcionais, terceirização e equipes distribuídas lidando com múltiplas prioridades e nós rapidamente descobrimos o quão encravados estamos.

Nesta seção, apresentamos três estratégias para mitigar o problema. A estratégia de curto prazo é criar transparência sobre as prioridades e melhorar a comunicação entre as equipes trabalhando nestes sistemas. A solução de médio prazo é construir camadas de abstração sobre sistemas que são difíceis de mudar e criar dublês de teste para sistemas que tem de integrar com eles. A solução de longo prazo é incrementalmente rearquitetar os sistemas com a habilidade para mover rapidamente em escala como um objetivo arquitetural.

A solução de curto prazo — criar transparência sobre as prioridades e melhorar a comunicação — é importante ser extremamente efetivo. A TI deve servir a múltiplas partes interessadas com prioridades frequentemente conflitantes. Quem vence depende frequentemente sobre quem grita mais alto ou tem as melhores conexões políticas, não sobre um modelo econômico tal como Custo de Atraso (discutido em *capítulo 7*). É importante ter um entendimento compartilhado em todos os níveis na organização sobre quais são as prioridades atuais. Isto pode ser tão simples quanto um encontro semanal ou mensal com as partes chave interessadas, incluindo todos os clientes da TI para elaborar uma lista priorizada de uma página. A comunicação entre aqueles responsáveis por sistemas acoplados é essencial.

### **ACOPLAMENTO REQUER COMUNICAÇÃO FREQUENTE**

Uma grande empresa de viagens queria entregar continuamente novas funcionalidades a seu site. Contudo, o site precisava conversar com um sistema de reservas legado. Frequentemente novas funcionalidades eram atrasadas devido às dependências destas com o sistema de reserva, que era atualizado a cada seis meses. Isto estava acarretando custos enormes em custos de oportunidades perdidas.

Uma forma simples que eles usaram para minimizar o problema foi melhorar a comunicação entre as equipes. O gerente de produto do site encontraria regularmente com o gerente de produto do sistema de reservas que comparariam notas sobre os próximos lançamentos, com atenção especial para as dependências. Eles encontrariam formas de alterar seus calendários para ajudar um ao outro a entregar as funcionalidades no prazo ou abandonar as funcionalidades que não pudessem ser entregues.

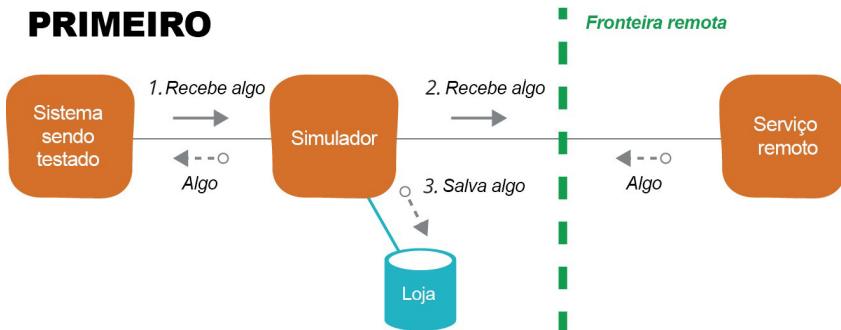
A solução de médio prazo é encontrar maneiras de simular o sistema de melhor frequência de alteração com o qual precisamos integrar. Uma técnica é usar versões virtualizadas destes sistemas. Outra alternativa é criar dublês de teste que simulam o sistema remoto para propósitos de teste — veja figura a seguir (FOWLER, 2009b). O importante a se ter em mente é que não queremos reproduzir fielmente o verdadeiro ambiente de produção. O que queremos é descobrir e consertar logo a maior parte dos problemas de integração, antes de lançar em um ambiente de pré-produção.

Ao simular sistemas remotos em um ambiente virtual, podemos integrar e executar testes de nível de sistema para validar nossas

mudanças de forma regular (por exemplo, diariamente). Isto reduz a quantidade de trabalho que temos de fazer em um ambiente adequadamente integrado.

A solução de longo prazo é arquitetar nossos sistemas de tal forma que possamos mover rapidamente. Em particular, isto significa ser capaz de lançar partes de nosso sistema independentemente à vontade, sem ter de ir através de lançamentos complexos orquestrados. Contudo, isso requer rearquitetura cuidadosa utilizando o padrão de estrangulamento, descrito no *capítulo 10*.

## PRIMEIRO



## DEPOIS

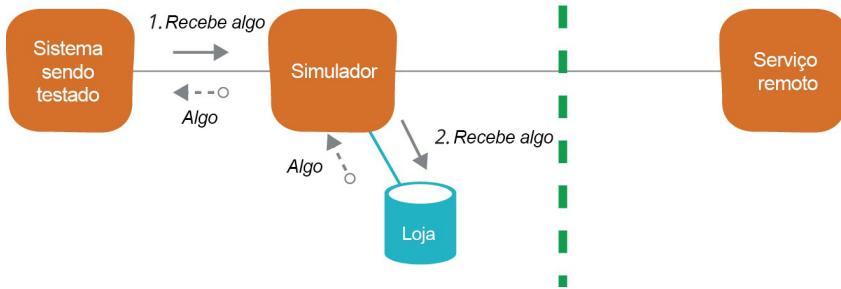


Figura 14.2: Simulando sistemas remotos para propósitos de testes, cortesia de Martin Fowler (2009b)

Quando começar neste processo, um primeiro e importante passo é mapear seus serviços e a conexão entre eles. Com base no

ciclo de vida das inovações (veja o *capítulo 2*) e no valor que cada serviço fornece para nossa organização, podemos desenhar em dois eixos o valor e o ciclo de vida, e então criar um mapa da cadeia de valor para visualizar cada produto e suas dependências (veja a figura seguinte).

Para criar um mapa da cadeia de valor, pegue um produto e posicione-o adequadamente no alto do novo diagrama. Então mapeie os serviços dos quais ele depende e as conexões entre eles. Este exercício pode ser realizado de forma rápida e barata em um quadro branco utilizando post-its (WARDLEY, 2013).

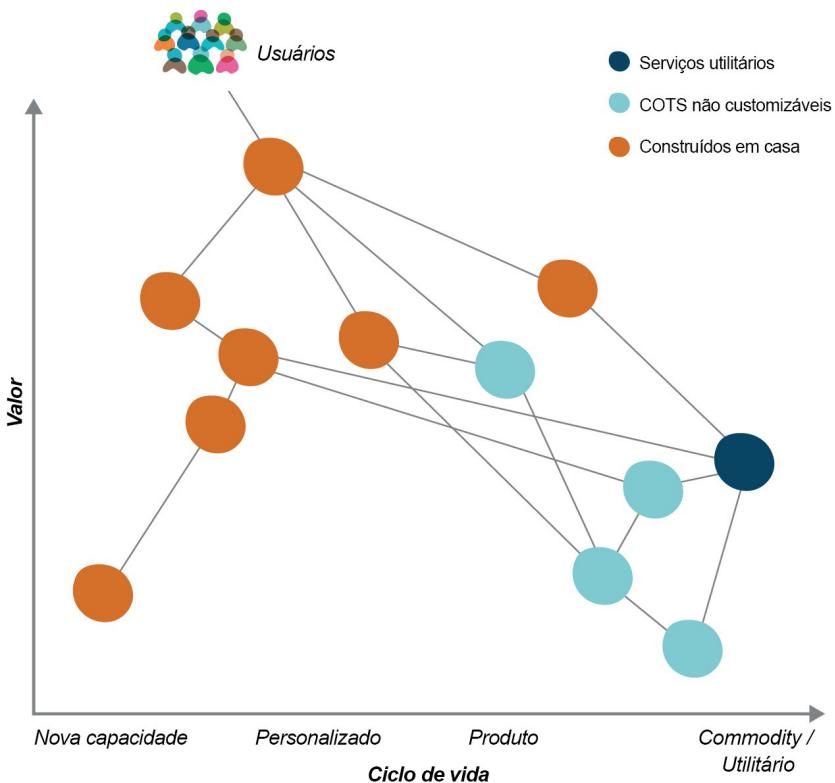


Figura 14.3: Mapa de cadeia de valor, cortesia de Simon Wardley (2013)

O próximo passo é criar uma versão "futura" deste diagrama,

segundo os princípios a seguir:

- Utilize fornecedores de software como serviço (SaaS) para todos os serviços de "utilidade", tais como folha de pagamento, gestão de fornecedores, e-mail, controle de versão e assim por diante. Se existirem sistemas que não podemos mover para a nuvem, deveríamos utilizar pacotes de software comerciais de prateleira.
- Para serviços estratégicos e aplicações que fornecem uma vantagem competitiva, deveríamos desenvolver software de forma customizada, como descrito no resto do livro. Evite a todo custo a tentação de utilizar pacotes para estas capacidades.
- Sistemas de registro serão normalmente os mais difíceis de mudar, e serão uma combinação de pacotes de prateleira e sistemas mais antigos, incluindo mainframes. Estes frequentemente exigem a consolidação e alguma quantidade de abstração para reduzir o custo de manutenção e de integrar-se com eles. Ao longo do tempo, eles podem ser estrangulados, caso necessário.

Quando usarmos pacotes de prateleira, é de extrema importância não customizá-los. Não temos como enfatizar o suficiente os problemas e riscos associados à customização desses pacotes. Quando as organizações começam a customizar, é difícil parar — as customizações de pacotes de prateleira são extremamente caras de fazer e de manter ao longo do tempo. Uma vez passado um determinado ponto, o fornecedor original frequentemente não dará mais suporte ao pacote. Atualizar os pacotes customizados é incrivelmente doloroso e é difícil fazer mudanças de forma rápida e segura a sistemas customizados de

prateleira.

Em vez disso, faça mudanças em seus processos de negócio para se adequar ao que os pacotes de prateleira oferecem naturalmente. Sempre que você escolhe uma solução de prateleira, você terá uma lista de funcionalidades a implementar ou defeitos a corrigir. Estes deveriam sempre ser tratados como entradas para a atividade de gestão de mudanças dos processos de negócio. Mudanças nos processos de negócio são muito mais baratas e mais efetivas do que mudar pacotes de prateleira para ajustá-los aos processos existentes.

Caso você já tenha trilhado esse caminho da customização, aproveite a próxima grande atualização do seu pacote de prateleira como uma oportunidade de migrar para a versão nova e sem customizações do pacote, como a Telstra (a maior Telecom da Austrália) fez quando abriu mão de uma versão altamente customizada do Remedy para uma completamente padrão (BARR, 2014).

Migrar do seu estado atual para seu estado "futuro" provavelmente levará anos. Como toda mudança de larga escala, o caminho correto a seguir é incrementalmente quebrar grandes programas de trabalho em passos pequenos que ofereçam o maior retorno para o menor investimento em termos de melhora nos resultados para o cliente e o negócio.

#### O PROGRAMA DE SIMPLIFICAÇÃO DA SUNCORP

*por Scott Buckley e John Kordyback*

O Suncorp Group da Austrália possui planos ambiciosos de desativar seus sistemas gerais de políticas seguras, melhorar a plataforma central de serviços bancários e começar um programa de excelência operacional. "Ao desativar sistemas

duplicados ou antigos, a Suncorp pretende reduzir custos operacionais e reinvestir estas economias em novos canais digitais", diz Matt Pancino, novo CEO da Suncorp Business Systems.

Práticas Lean e melhoria contínua são estratégias necessárias para entregar o programa de simplificação. A Suncorp está investindo com sucesso em frameworks automatizados de teste para apoiar o desenvolvimento, a configuração, a manutenção e a atualização rápidas dos sistemas. Estas técnicas são familiares para quem usa novas plataformas tecnológicas, especialmente no espaço digital, mas a Suncorp é bem-sucedida na aplicação de abordagens ágeis e Lean para o mundo dos computadores grandes, caros e super-rápidos, que é o mundo dos sistemas mainframe conhecido como "big iron".

### **Práticas de entrega**

Em seu negócio de seguros, a Suncorp está lançando mão da combinação de grandes e complexos sistemas mainframe de políticas de seguro com um sistema para apoiar processos de negócio comum pela organização e direcionar mais vendas de seguro pelos canais diretos. Algumas peças chave faziam parte do programa "Peças básicas", que forneceu um framework de testes funcionais para o sistema mainframe central de políticas, práticas ágeis de entrega e uma abordagem comum para a integração de sistemas com base em serviços web.

Durante o primeiro ano do programa de simplificação, os testes foram estendidos para apoiar a integração do sistema mainframe de política com os novos canais digitais e os sistemas de precificação. Critérios de aceite automatizados foram desenvolvidos enquanto diferentes sistemas eram desenvolvidos. Isto reduziu enormemente o tempo de teste para a integração de novos sistemas de precificação e de

avaliação de riscos com múltiplos tipos de políticas. Testes automatizados também deram apoio à gestão e às verificações de políticas para os cliente por meio de diferentes canais, como on-line ou *call center*.

Testes noturnos de regressão sobre as funcionalidades centrais mantiveram o passo com o desenvolvimento e apoiaram tanto os testes funcionais quanto a integração entre sistemas. Conforme os defeitos eram encontrados em cenários de ponta a ponta, resoluções responsivas eram gerenciadas em horas ou dias, não semanas como é típico para grandes sistemas empresariais.

## Resultados

No processo, a Suncorp reduziu de 15 para 2 os sistemas complexos de seguro pessoal e de vida, e desativou 12 sistemas legados. As atualizações são feitas uma vez e implantadas por todas as marcas. Eles têm uma base única de código para sites da internet para todas suas diferentes marcas e produtos. Isto permite resposta mais rápida às necessidades dos clientes e torna cada equipe responsável por um site redundante.

Por uma perspectiva de negócio, o sistema mais simples permitiu que 580 processos de negócio fossem redesenhadados e racionalizados. As equipes agora podem oferecer serviços novos ou melhorados de acordo com a demanda, em vez de melhorar cada marca isoladamente. Também houve redução do tempo de implantação dos novos produtos e serviços, tais como coberturas médica para clientes APIA ou assistência na estrada para clientes AAMI.

O investimento na simplificação e gestão dos sistemas centrais da Suncorp quer dizer que eles podem aumentar os investimentos em todos seus pontos de contato com os

clientes. Tanto nas práticas tecnológicas e de negócio, a Suncorp aumentou o passo da simplificação, com a maioria das marcas agora usando infraestrutura, serviços e processos comuns.

O relatório anual de 2014 da Suncorp aponta que "a simplificação permitiu ao Grupo operar uma base de custos mais variável com a habilidade de escalar recursos e serviços de acordo com as demandas do mercado e do negócio. Espera-se que a atividade de simplificação resulte na economia de \$225 milhões em 2015 e \$265 milhões em 2016" (<http://bit.ly/1v73OC3>).

## 14.5 CONCLUSÃO

Se queremos competir em um mundo com ciclos de produto cada vez mais curtos, a TI central precisa ser uma parceira com a confiança das unidades de negócio, não um centro de custos tirador de pedidos. Em contrapartida, a TI precisa atingir níveis mais altos de vazão ao mesmo tempo em que melhora a estabilidade e qualidade, além de reduzir custos. A complexidade dos ambientes empresariais existentes combinada com a quantidade de trabalho planejado e não planejado que deve ser feito para mantê-las rodando são as principais barreiras para atingir tais resultados.

Nós podemos somente começar a endereçar esses problemas quando considerarmos os efeitos do novo trabalho nas operações de TI e tratá-los como parte integrante do ciclo de vida do desenvolvimento de produto. Para gerenciar a complexidade adicional introduzida por novos produtos, serviços e features, precisamos começar movendo-nos de um modelo baseado em projeto para um centrado em produtos, como descrito no *capítulo*

10.

Equipes de produto precisam ser responsáveis pelos custos e acordos de nível de serviço dos sistemas que constroem. Em contrapartida, elas ganham a liberdade de escolher as tecnologias e recursos dentro da TI para focar em reduzir a complexidade de seus sistemas e de suas infraestruturas, além de construir uma cadeia de ferramentas e uma plataforma que promova o ciclo de vida de desenvolvimento de produtos que descrevemos neste livro.

Nós geralmente tratamos vazão e estabilidade como forças opostas — aumente a vazão e você reduzirá a qualidade e a estabilidade. Entretanto, estes objetivos podem ser complementares se a estratégia correta for colocada em prática. Como todo esforço de melhoria, precisamos começar articulando claramente nosso objetivo e encontrar os indicadores chave que mais importam para nós. Então, usamos a Melhoria Kata para trabalhar convergindo ao objetivo.

## Questões para os leitores

- A TI se considera como provedora de serviço, uma parceira para as áreas de negócio, ou uma promotora de inovação? O que outros líderes dentro da organização pensam a respeito?
- Você está medindo o tempo de aplicação da mudança, frequência de releases, o tempo para retornar o serviço, e a taxa de falha das mudanças de todos seus produtos e serviços? Está mantendo-os visuais para todos os times?
- Quantos serviços você aposentou no último ano? E quantos você adicionou? Quanto tempo levaria para você identificar quantos produtos e serviços você está gerindo? Quão certo você estaria desta resposta?

Quantos estão rodando sobre sistemas que já não têm o suporte do fornecedor?

- Quanto tempo leva para aprovar uma solicitação de mudança? Quanto tempo leva para ter um novo componente opensource aprovado para uso em ambiente de produção?
- De quanto em quanto tempo você realiza um exercício real de recuperação de desastres em seus sistemas em produção? Qual seu processo de acompanhamento de recomendações de melhorias que se originam deste exercício?
- Todos os desenvolvedores, líderes de desenvolvimento e arquitetos revezam no atendimento, com frequência, para os sistemas que constroem?

## CAPÍTULO 15

# FIQUE ONDE ESTÁ

*“Se você fizer algo e o resultado for muito bom, então você deve tentar fazer algo extraordinário. Descubra o que está por vir.” — Steve Jobs*

*“Daqui há um ano, você vai desejar ter começado hoje.” — Karen Lamb*

Nosso objetivo neste livro é inspirar você a visualizar um futuro alternativo para grandes organizações. Um futuro que coloca empregados, clientes e produtos no coração de sua estratégia. Um futuro em que uma cultura e um ambiente renovados permitam que a organização se adapte rapidamente às alterações nas demandas de mercado.

Compartilhamos histórias e lições de um conjunto diverso de organizações com backgrounds e circunstâncias variados para destacar que, mesmo em ambientes complexos, você pode prosperar e enfrentar os problemas mais desafiadores. Contudo, o caminho para o sucesso provavelmente não será linear, com instruções, marcos e KPIs definidos.

As organizações precisam se sentir confortáveis para seguir em frente com incerteza e informações imperfeitas, ao mesmo tempo em que aprendem, se ajustam e desenvolvem seu pessoal ao longo

do caminho.

A maior barreira para o sucesso ao mudar o seu modo de trabalhar é a convicção de que sua organização é muito grande ou burocrática para mudar, ou que seu contexto o impede de adotar as práticas que discutimos. Sempre se lembre de que cada pessoa, time e negócio que começaram sua jornada não estavam certos de quais caminhos seguir e como terminariam. A única verdade aceita foi a que, se não tomassem nenhuma atitude, um final mais certo e negativo estava por vir.

## 15.1 PRINCÍPIOS DA MUDANÇA ORGANIZACIONAL

Toda mudança é arriscada, especialmente a mudança organizacional, que inherentemente envolve mudança cultural — a mais difícil de todas, já que você está lidando com as forças que dão a identidade à organização. Ainda ficamos atônitos quando líderes planejam programas de “mudança organizacional” que esperam terminar em meses. Tais programas falham em reconhecer que transformar inovação ou mudança em um fato, em vez de em uma parte de seu trabalho diário, pode nunca produzir resultados significantes ou duradouros.

Periodicamente, financiar um programa de mudança em resposta a questões atuais, mudanças de liderança ou tendências de mercado, sem incutir uma cultura de experimentação, apenas resultará uma mudança incremental de curto prazo, se alcançar (figura a seguir). Organizações rapidamente voltarão ao seu estado anterior. Em vez disso, devemos criar uma cultura de melhoria contínua por meio da prática deliberada e constante de todos na organização.



Figura 15.1: A realidade dos programas de mudança “baseada em um fato”

Se a sua organização está esperando por um evento para estimular a mudança, você já está com problemas. No ambiente atual e na economia competitiva, um senso de urgência deveria ser um estado permanente. A ansiedade de sobrevivência sempre existe em organizações líderes, como descrevemos no *capítulo 11*. Contudo, como notou Schein, usá-la como motivador para mudança contínua é ineficaz. O único caminho para uma cultura de melhoria contínua é criar um ambiente no qual aprender novas habilidades e ser melhores no que fazemos sejam considerados valiosos por si só e apoiados pela gerência e pela liderança, dessa maneira reduzindo a ansiedade de aprendizado. Podemos usar a Melhoria Kata apresentada no *capítulo 6* para criar essa cultura e conduzir à melhoria contínua (figura seguinte).

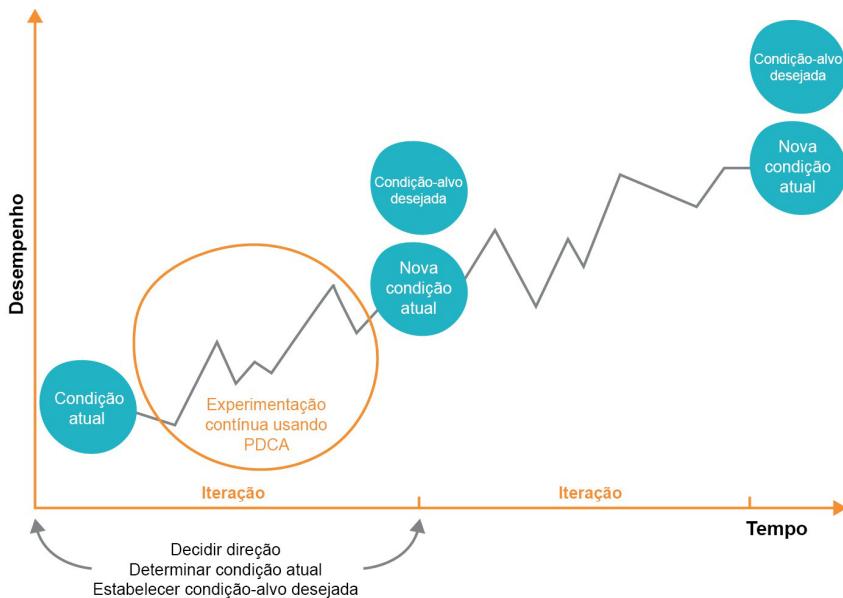


Figura 15.2: Evolução contínua e adaptação à mudança

Para propagar a Melhoria Kata nas organizações, gerentes devem aprender e implementar uma prática complementar conhecida como Coaching Kata. Para começar a jornada, um time ADVANCE, incluindo um patrocinador executivo — idealmente, o CEO —, deve guiar o Coaching Kata e a Melhoria Kata. Como esse time vai guiar sua implementação mais abrangente dentro da organização, é imperativo que eles entendam como ela funciona.

Para materiais gratuitos sobre a Melhoria Kata e o Coaching Kata, veja <http://bit.ly/1v73SSg>.

Preste atenção aos seguintes obstáculos:

- Adotar a Melhoria Kata requer mudanças substanciais

no comportamento em todos os níveis da organização. O Coaching Kata é usado para ensinar as pessoas a Melhoria Kata, mas o problema de como implantar o Coaching Kata dentro de uma organização permanece significativo.

- Fazer experimentos é difícil e requer grande disciplina. Encontrar bons experimentos requer talento e reflexão. Por natureza, as pessoas tendem a ir logo para a parte da solução em vez de primeiramente concordarem sobre objetivos-alvo (resultados) mensuráveis e então trabalhar em ciclos rápidos — e por rápidos queremos dizer horas ou dias — para criar hipóteses, teste e aprendizado. O corpo de conhecimento sobre como projetar e fazer experimentos no contexto do desenvolvimento de produto ainda está nos estágios iniciais, e as habilidades e técnicas necessárias não são amplamente conhecidas ou entendidas.
- Assegurar que há capacidade para executar a Melhoria Kata. Um dos maiores obstáculos que os times enfrentam quando tentam programar o trabalho de melhoria é que ele é frequentemente visto como uma distração para a entrega do trabalho. Isso é uma falácia, e o objetivo deve ser explicado antecipadamente e vigorosamente. No caso do HP FutureSmart, a razão pela qual o trabalho de entrega estava progredindo tão vagarosamente era porque o trabalho sem valor agregado estava guiando 95% dos custos. É vital para executivos do nível da diretoria ou vice-presidência assegurarem-se de que os times limitem seu WIP (trabalho em processo, ou *Work In Progress*, em inglês) como descrito no *capítulo 7*, para criar tempo para a melhoria.

- Como em todos os métodos, é provável que o progresso seja instável no começo, enquanto as pessoas aprendem a trabalhar de novas maneiras. As coisas vão piorar antes de melhorar. É provável que haja resistência das pessoas para aprenderem novas habilidades e alguns ficarem frustrados quando há conflito com seus hábitos e comportamentos existentes.

## Vise a implantação da estratégia

Apesar de termos falado sobre a Melhoria Kata como uma maneira de guiar a melhoria contínua em nível de programa, ela pode ser usada em qualquer nível, de times individuais até planejamento estratégico. Para aplicar a Melhoria Kata no nível do planejamento estratégico, comece com um consenso sobre o propósito da organização. O que queremos fazer para nossos clientes? Então, aqueles que participarem do exercício de planejamento estratégico devem definir e concordar sobre a direção global da empresa — identificar nosso “verdadeiro norte”.

O próximo passo é entender e deixar clara a situação atual da organização. Os participantes no exercício de planejamento estratégico devem identificar quais problemas precisam ser abordados e reunir dados para melhor entender cada problema. Tipicamente, mesmo grandes organizações têm capacidade limitada e podem gerenciar apenas uma certa quantidade de iniciativas por vez; escolher o que não focar e se certificar de que o time siga sua decisão é importante. Um framework econômico como o Custo de Atraso (ver *capítulo 7*) é útil para estimular a discussão sobre priorização do trabalho.

Uma vez que decidimos em quais problemas focar, precisamos definir nossas condições-alvo. Estas devem claramente comunicar o que é alcançar o sucesso. Elas também devem incluir KPIs, para que

possamos medir nosso progresso em direção ao objetivo. A abordagem tradicional de *balanced scorecard* para KPIs tem quatro perspectivas-padrão: financeira, de mercado, de operações e pessoas e organização.

A Statoil usou a abordagem de *balanced scorecard* em sua estrutura *Ambition to Action* (capítulo 13) e adicionou saúde, segurança e ambiente. O movimento enxuto nos ensina a focar em reduzir custo e aumentar qualidade, entrega, moral e segurança (essas cinco “métricas enxutas” são às vezes abreviadas como QCDMS, de *Quality, Cost, Delivery, Morale, Safety*, em inglês).

Bjarte Bogsnes, vice-presidente de desenvolvimento de gestão de performance na Statoil, recomenda escolher de 10 a 15 KPIs e designar metas relativas que conectem inputs com resultados (por exemplo, custo unitário em vez de custo absoluto) e são baseados em uma comparação com um patamar (por exemplo, “retorno de capital investido 10% maior do que nosso concorrente principal”) (BOGSNES, 2009, p. 125-126).

As metas-alvo em nível estratégico formam a direção para o próximo nível organizacional, que então passa pelo seu próprio processo de Melhoria Kata. As metas-alvo neste nível então formam a direção para o próximo nível organizacional abaixo, como mostrado na figura mais adiante. Esse processo, que nos permite definir alvos e gerenciar recursos e performance criando um alinhamento entre os níveis da organização, é chamado de implantação de estratégia (também conhecido como *Hoshin* ou *Hoshin Kanri* — Ambição para Ação é uma variação da implantação de estratégia).

Para uma descrição detalhada da Ambição para Ação, veja Bogsnes (2009, p. 114-169).

O processo de criar alinhamento e consenso entre os níveis é crítico. Em implantação de estratégia, esse processo é descrito como *catchball*, uma palavra escolhida para evocar um exercício colaborativo. As condições-alvo de um nível não devem ser transcritas diretamente para os times trabalhando em um nível abaixo; a *catchball* trata-se mais de uma tradução de estratégia, com “cada camada interpretando e traduzindo quais objetivos do nível acima querem para si” (BOGSNES, 2009, p. 124). Devemos esperar que o feedback dos times fará com que o plano do nível mais alto seja atualizado. Não subverta o *Hoshin* ao usá-lo simplesmente para cascatare metas pela organização: a chave para o *Hoshin* é que ele é um mecanismo para criar alinhamento baseado em colaboração e loops de feedback em múltiplos níveis.

Os horizontes temporais para cada nível devem ser claramente definidos e reuniões regulares de review devem ser agendadas, com metas-alvo atualizadas baseadas no progresso dos times do nível seguinte. Para ser verdadeiramente efetivo, essa conversa deve ser também multifuncional, promovendo a cooperação ao longo dos fluxos de valor, dentro e entre as unidades de negócio. Não é fácil, pois requer que as ideias e preocupações das pessoas responsáveis pelos resultados sejam ouvidas honestamente — e respondidas ajustando-se os planos de acordo com o feedback.

Saiba mais sobre implantação de estratégia no *capítulo 3* de *The Outstanding Organization*, de Karen Martin (2012) e leia o estudo de caso em <http://www.lean.org/Search/Documents/54.pdf>.

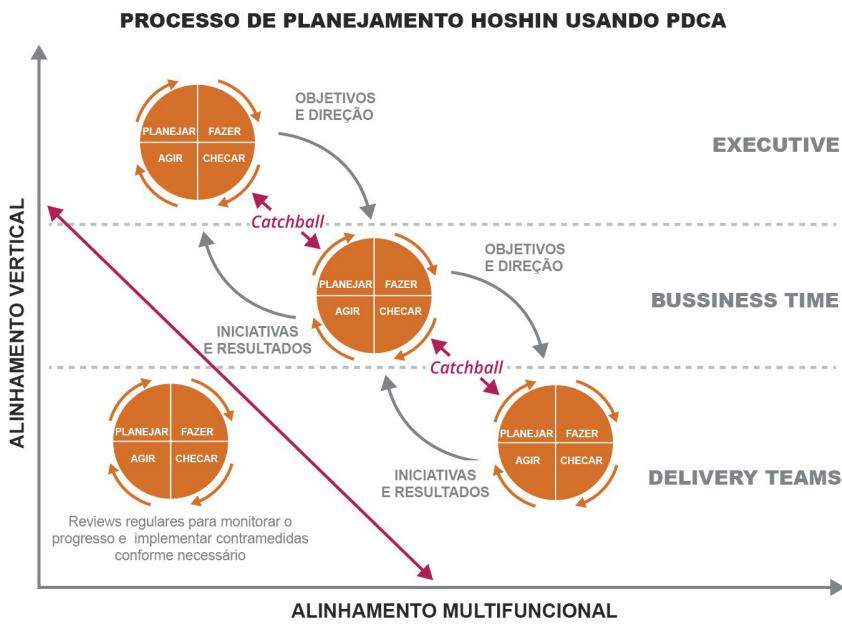


Figura 15.3: Usando a catchball para conduzir alinhamento estratégico de objetivos e iniciativas

Um exercício de planejamento de estratégia de alto nível pode ter um horizonte de seis meses a alguns anos, dependendo do quanto apropriado para seu negócio. Reuniões de review devem ser feitas mensalmente nas quais o time, junto com líderes de todos os times que se reportam a ele, se reúne para monitorar o progresso e atualizar as condições-alvo em resposta ao que descobriram. Times

em níveis mais baixos normalmente trabalharão com um horizonte mais curto, com reuniões de review mais frequentes.

A implantação de estratégia é uma ferramenta avançada que depende de comportamento e cultura alinhados, como descrevemos no *capítulo 11*. O objetivo principal é criar um consenso e um alinhamento, e permitir a autonomia dentro da organização, segundo o paradigma do Comando de Missão apresentado no *capítulo 1*. Vamos ver como o governo do Reino Unido aplicou uma versão da implantação de estratégia para transformar seu uso de plataformas digitais para fornecer serviços para os cidadãos, começando com pouco e crescendo iterativamente e incrementalmente.

## 15.2 O SERVIÇO DIGITAL DO GOVERNO DO REINO UNIDO

O governo do Reino Unido, como muitos outros, reconheceu o potencial da internet “tanto de comunicar e interagir com os cidadãos como entregar significantes ganhos de eficiência” (LANE-FOX, 2010). Contudo, projetos de governo envolvendo desenvolvimento de software têm um passado confuso. O governo do Reino Unido tinha vários projetos grandes de TI estourando o orçamento, ao mesmo tempo em que falhavam ao entregar os benefícios esperados, culminando no desastre do “Programa Nacional para TI”.

O maior programa de tecnologia da informação público, que supostamente entregaria uma infraestrutura de TI completamente nova para o Serviço de Saúde Britânico e um sistema computadorizado de registro do paciente, foi projetado para custar 2,3 bilhões de libras de acordo com seu planejamento em 2002. Sua entrega foi terceirizada para múltiplas empresas privadas, incluindo

a Accenture, Computer Sciences Corporation, Fujitsu e British Telecom. Apesar do cancelamento do programa em 2011, especula-se que ele tenha custado mais de 10 bilhões de libras.

O processo de aquisição do governo para grandes projetos de IT envolveu escrever uma especificação completa para o produto, criando vários casos de uso em níveis crescentes de detalhes, e então colocando o contrato em licitação — um processo que precisou de um a dois anos antes de começar o trabalho na criação do produto. “A essa altura”, comentou Francis Maude, chefe de gabinete do governo britânico, “certamente atrasará. Você está preso a um fornecedor, é muito caro fazer alterações” (<http://bit.ly/1F7yvbs>).

Como resultado da terceirização de projetos de TI, cada departamento do governo tinha sua presença na web projetada e operada de maneira independente, com diferentes experiências do usuário que refletiam na organização interna de cada departamento. Era complicado e extremamente difícil para os cidadãos usarem, então eles preferiram usar canais de serviço mais caros, como ambulatórios, correio e serviços telefônicos.

Em 2010, Martha Lane Fox, cofundadora da startup britânica lastminute.com, foi contratada para aconselhar o governo do Reino Unido em sua estratégia para entrega de serviços públicos online. Seu relatório recomendava criar um time central de servidores públicos responsável por projetar e entregar a presença online do governo, implementando uma política de dados abertos, pela qual todos os dados governamentais ficariam disponíveis por meio de APIs públicas, e nomear um CEO “com domínio absoluto sobre a experiência do usuário em todos os serviços online do governo (websites e APIs) e o poder de dirigir todos os gastos online do governo” (LANE-FOX, 2010).

Dessa maneira, surgiu o Serviço Digital do Governo do Reino Unido (*Government Digital Service*, GDS, em inglês). Martha Lane

Fox descreveu suas metas para o GDS: “Para mim, o teste rigoroso... é se pode empoderar e deixar a vida mais simples para cidadãos, e ao mesmo tempo permitir que o governo ‘desligue’ outras coisas. Um foco em um vasto aumento de alcance, uso e qualidade de transações online entregará o maior impacto: menos problemas para cidadãos e maior eficiência”.

#### **ESTUDO DE CASO DO SERVIÇO DIGITAL DO GOVERNO, POR GARETH RUSHGROVE**

O GOV.UK é o novo domínio único para todos os serviços governamentais no Reino Unido. Foi lançado em outubro de 2012 e substituiu inicialmente dois dos maiores websites existentes do governo no primeiro dia, e seguiu para substituir todos os sites de departamentos governamentais nos próximos meses. Em 2014, fechou milhares de websites e construiu um serviço único que é mais simples, mais claro e mais rápido, cobrindo tudo desde informação sobre benefícios aos quais o cidadão tem direito até tirar um passaporte.

Um aspecto do GOV.UK que o diferencia de um projeto típico do governo é que foi desenvolvido quase todo dentro do governo, por servidores públicos trabalhando no recém-formado Serviço Digital do Governo (GDS), que faz parte do Gabinete do Governo do Reino Unido. Foi também construído iterativamente, de maneira mais barata e usando métodos e tecnologias ágeis, mais comumente associados a startups em vez de grandes organizações. Aqui, temos uma descrição de como foi feito.

#### **Alfa e Beta**

Ao final de 2013, o time trabalhando no GOV.UK tinha mais de 100 pessoas — mas não começou assim. Na verdade, a

primeira versão não era nem mesmo chamada de GOV.UK. Uma versão Alfa foi construída por 14 pessoas trabalhando em uma pequena sala em um enorme prédio do governo. Seu objetivo não era ser um produto terminado, mas ilustrar como um único site governamental poderia ser, e como poderia ser construído rapidamente e de maneira barata. No total, a versão Alfa levou 12 semanas para ser feita e custou 261 mil libras.

O feedback dos usuários da Alfa levou diretamente ao trabalho da versão Beta, que escalou a proposta Alfa e envolveu mais pessoas do governo. O primeiro lançamento da Beta aconteceu seis meses depois do envio da versão Alfa, mas isso incluiu o tempo para montar o time. O primeiro lançamento público da versão Beta foi um website governamental real, mas na época faltava todo o conteúdo e as features necessárias para que substituisse os principais sites governamentais. Oito meses de constantes iterações depois, com o time com cerca de 140 pessoas e com novos conteúdos e features adicionados diariamente, o tráfego dos dois maiores websites governamentais foi redirecionado para o novo GOV.UK.

Todo este trabalho valeu a pena. Durante 2012-2013, o GDS economizou 42 milhões de libras ao substituir o Directgov e o BusinessLink pelo GOV.UK. Em 2013-2014, estima-se que o GDS economizou 50 milhões de libras ao fechar mais websites e trazê-los para o domínio único.

### **Times multidisciplinares**

O Serviço Digital Governamental é feito por especialistas em desenvolvimento de software, gerenciamento de produto, design, pesquisa de usuário, operações de web, design de conteúdo etc., assim como especialistas em políticas governamentais e outras áreas de domínio específicas. Deste grupo de especialistas, times foram formados para construir e

operar o GOV.UK. Estes times não tinham um foco estreito, contudo, a maior deles era multidisciplinar, feitos por pessoas com o mix certo de habilidades para as tarefas propostas.

Como, por exemplo, o time que trabalhou nos estágios iniciais da versão Beta do GOV.UK. Ele consistia de sete desenvolvedores, dois designers, um product owner, dois gerentes de entrega e cinco designers de conteúdo. Mesmo dentro dessas áreas, havia uma vasta gama de habilidades. Os desenvolvedores tinham habilidades que iam de front-end a administração de sistemas.

Ao empregar times multidisciplinares, a responsabilidade ponta a ponta por produtos inteiros ou tarefas individuais poderia ser delegada ao time, removendo a necessidade de comando e controle em larga escala. Tais times pequenos e autônomos tinham poucas dependências em relação a outros times, fazendo com que evoluíssem muito mais rapidamente.

Este modelo multidisciplinar também ajudou a minimizar problemas típicos em grandes organizações com estruturas organizacionais em forma de silos. Por exemplo, o Serviço Digital do Governo cresceu com o tempo, adicionando experts em informação governamental, aquisição e governança de TI para evitar gargalos e melhorar a priorização de recursos.

### **Entrega contínua**

Um aspecto importante do sucesso do GOV.UK tem sido a melhoria constante baseada no feedback do usuário, testes e dados de web analytics. O time do GOV.UK lança um novo software, em média, seis vezes por dia — com todos os tipos de melhorias, de pequenas correções de bugs a features completamente novas, para o site e plataformas de apoio.

Depois do lançamento da versão Beta do GOV.UK, um dos gerentes de produto, traumatizado com lançamento de software em outras organizações, perguntou se o mecanismo de deployments do software realmente funcionaria. A resposta foi “sim”. Àquela altura, o GDS havia feito mais de mil deployments, então o nível de confiança era alto. A prática da automação leva à perfeição.

Essa taxa de deployments não é normal para grandes organizações nas quais os processos existentes parecem projetados para resistir a mudanças. Os times de desenvolvimento do GOV.UK trabalharam extensivamente em automação, tiveram conversas profundas com pessoas preocupadas com uma mudança tão rápida. O termo-chave quando discutindo essa abordagem era o risco — especificamente, como releases regulares podem minimizar os riscos de mudança.

Muitas pessoas não se dão muito bem ao fazer tarefas repetitivas, mas computadores são perfeitos para automatizar essas tarefas. Deployment de software, principalmente se você vai fazê-lo de maneira regular, é um grande candidato à automação. Com o desenvolvimento e operação do GOV.UK, isso foi levado mais adiante: provisão de máquinas virtuais, configuração de rede, regras de firewall e configuração de infraestrutura foram todas automatizadas. Ao descrever grandes partes do sistema todo no código, os desenvolvedores usaram ferramentas como controle de versão e teste unitário para criar confiança em suas mudanças, e focar em um conjunto menor de processos bem-feitos em vez de em um processo separado (que requereria competências especializadas) para cada tipo de mudança.

Outras técnicas também ajudaram. Um rígido foco em

usuários e uma cultura de confiança desde o topo da organização colocou o GDS em uma posição de usar muito do que aprendeu construindo e operando o GOV.UK para transformar o resto do governo do Reino Unido.

A abordagem GDS foi adotada por todos os braços do governo, com resultados transformadores para os cidadãos. Para dar apenas um exemplo, o Time Digital do Ministério da Justiça do Reino Unido recentemente trabalhou com o Serviço Nacional de Gestão de Agressores e o Serviço de Prisão HM para mudar a maneira como as pessoas agendam visitas à prisão. Antes, os visitantes tinham de solicitar formulários de papel, enviá-los pelo correio e então ligar para agendar uma visita. As solicitações eram frequentemente rejeitadas, porque a data não estava disponível, forçando as pessoas a fazer tudo de novo. Agora, as visitas prisionais podem ser agendadas online em 5 minutos, selecionando-se até três datas (BARLOW, 2014).

Nem todo mundo está animado com a ideia de governos desenvolvendo suas próprias capacidades em TI. Tim Gregory, o presidente do Reino Unido da CGI, o maior fornecedor do site americano HealthCare.gov, que fechou um contrato de 292 milhões de dólares em 2013 antes de ser substituída pela Accenture em janeiro de 2014 (BEGLEY, 2013), afirma que a abordagem GDS não deixará rentável para grandes fornecedores terceirizados licitarem para projetos governamentais. O diretor executivo do GDS, Mike Bracken, descreve a visão de Gregory como “além da paródia” (PREEZ, 2013). Há várias observações a serem feitas sobre o estudo de caso do GDS.

Primeiro, começar aos poucos com um time multifuncional e aumentar gradualmente a capacidade do produto, ao mesmo tempo

em que entrega valor iterativamente e incrementalmente, é um modo extremamente eficaz de mitigar os riscos de se substituir sistemas de alta visibilidade, enquanto se alimenta uma cultura de alta performance simultaneamente. Fornece um alto retorno do investimento, economias de custo substanciais e funcionários e usuários mais felizes. Isso é possível mesmo em um ambiente complexo e altamente regulado como o governo.

Segundo, em vez de tentar substituir sistemas e processos existentes com um “big bang”, o GDS os substituiu incrementalmente, escolhendo começar por onde poderiam entregar valor mais rapidamente. Pegaram o padrão “aplicação estranguladora” apresentado no *capítulo 10* e o usaram para executar a mudança de arquitetura e organizacional.

Terceiro, o GDS buscava governança baseada em princípios. O time de liderança no GDS não diz a cada um o que fazer, mas providencia um conjunto de princípios-guias para as pessoas tomarem decisões alinhadas aos objetivos da organização. Os princípios de governança do GDS declararam (STEPHENS, 2014):

1. Não desacelere a entrega.
2. Decida, quando necessário, no nível certo.
3. Faça isso com as pessoas certas.
4. Vá ver com seus próprios olhos.
5. Só faça se agregar valor.
6. Confie e verifique.

Espera-se que as pessoas tomem as melhores decisões em seu contexto, mas elas são responsáveis por tais decisões — em relação aos resultados alcançados e saber quando é apropriado envolver outras pessoas.

Finalmente, o GDS mostra que níveis extraordinários de compensação e o uso de um modelo do setor privado não são

decisivos para criar uma cultura de inovação. O GDS é formado por funcionários públicos, não por empreendedores do Vale do Silício com opções de ações. Uma cultura de inovação é criada ao aproveitar a necessidade das pessoas de conhecimento, autonomia e propósito — e se certificando de que as pessoas estão profundamente comprometidas com o propósito da organização e com os usuários aos quais atendem.

Na verdade, a probabilidade relativamente baixa de uma startup "existir" com sucesso significa que, por razões puramente financeiras, você seria maluco em preferir um emprego em uma startup do que uma posição sólida como, por exemplo, na Google (BINETTI, 2011).

## 15.3 COMECE SUA JORNADA

Use os seguintes princípios para começar:

Esse princípio é parcialmente inspirado em John Kotter (2012), descritos em seu processo de 8 etapas: estabelecer um senso de urgência, criar a liga guia, desenvolver uma visão e uma estratégia, autorizar uma ação de base ampla, gerar sucessos em curto prazo, consolidar ganhos e produzir mais mudanças, abrigar novas abordagens em sua cultura.

- **Assegure-se de que você tem uma direção claramente definida.**

A direção deveria representar sucintamente o negócio

ou descrever os resultados organizacionais que você deseja atingir em termos mensuráveis, mesmo que pareçam um ideal inatingível. E, mais importante, deveria inspirar todo mundo na organização. Pense na meta da HP FutureSmart de melhorar em 10 vezes a produtividade.

- **Defina e limite seu escopo inicial.**

Não tente mudar a organização inteira. Escolha uma parte pequena dela — pessoas que compartilham da mesma visão que você e têm capacidade de perseguí-la. Comece com o GDS, comece com uma única parte multifuncional, talvez um único produto ou serviço. Certifique-se de que você tem apoio em todos os níveis, de executivos ao chão de fábrica. Crie objetivos-alvo, mas não pense demais sobre eles ou planeje como alcançá-los. Garanta que o time tem o que precisa para experimentar, siga a Melhoria Kata e itere.

- **Busque uma cultura de melhoria contínua de alta performance.**

Talvez o resultado mais importante de implementar a Melhoria Kata é criar uma organização na qual a melhoria contínua é um hábito.

- **Comece com as pessoas certas.**

Novas maneiras de trabalhar difundem-se pelas organizações do mesmo modo que outras inovações, como descrevemos no começo do \*capítulo 2\*. A chave é encontrar pessoas que têm uma mentalidade de crescimento (ver \*capítulo 11\*) e se sentem confortáveis em tentar novas ideias. Uma vez que você alcançou resultados positivos, passe para os primeiros seguidores (entusiastas), e depois para a maioria precoce. O resto é relativamente fácil, porque não há nada que a maioria tardia odeie mais do que estar em uma minoria. Essa abordagem pode ser aplicada para cada um dos três horizontes descritos n

o \*capítulo 2\*.

- **Encontre uma maneira de entregar resultados valiosos e mensuráveis desde cedo.**

Apesar de mudanças duradouras levarem tempo e nunca serem completas, é fundamental demonstrar resultados reais rapidamente, como o time GDS fez.

- **Crie momentum e credibilidade.**

Na verdade, a estratégia da Melhoria Kata é projetada para atingir essa meta, que esperamos que a torne atrativa para executivos que normalmente têm de demonstrar resultados rapidamente e consistentemente com um orçamento apertado.

Conforme você experimentar e aprender, compartilhe o que funciona e o que não funciona. Faça apresentações regulares, convidando os stakeholders-chave da organização e o próximo segmento de adoção de tecnologia. Faça retrospectivas para refletir sobre o que alcançou, e use-as para atualizar e refinar sua visão. Sempre siga em frente. O medo, a incerteza e o desconforto são suas bússolas em direção ao crescimento. Você pode começar agora mesmo, preenchendo o formulário simples mostrado na figura a seguir (veja o *capítulo 11* para mais detalhes sobre as condições-alvo). Para mais informações sobre como criar uma mudança sustentável, particularmente com a falta de apoio executivo, recomendamos *Fearless Change: Patterns for Introducing New Ideas* (MANNS; RISING, 2004).

<b>Seu Plano de Transformação</b>		
Objetivo de Negócio		
Plano de Gerenciamento de mudança		
<b>Iteração 1 Objetivos (data-alvo:</b>		<b>2-6 semanas</b>
Ranking	Tema	Condição-alvo

Figura 15.4: Projeto de plano de transformação

## 15.4 CONCLUSÃO

Criar uma empresa resiliente, enxuta, que possa se adaptar rapidamente a condições de mudança depende de uma cultura de aprendizado por meio da experimentação. Para essa cultura prosperar, toda a organização deve estar ciente de seu propósito e trabalhar continuamente para entender as condições atuais, definir condições-alvo de curto prazo e permitir que as pessoas façam experimentos para atingi-las. Nós, então, reavaliamos nossas condições atuais, atualizamos nossas condições-alvo com base no que aprendemos e continuamos em frente. Esse comportamento deve se tornar habitual e difundido. É assim que criamos uma mentalidade de melhoria contínua focada em níveis cada vez mais altos de serviço ao consumidor e qualidade, com custo cada vez mais baixo.

Esses princípios são as linhas que ligam todos os padrões científicos. Se você está procurando um modelo de negócio repetível por meio do loop de aprendizagem Lean Startup, trabalhando para melhorar seu produto por meio de pesquisa de usuário e entrega contínua, ou guiar a inovação de processo e mudança organizacional usando ciclos PDCA da Melhoria Kata, tudo isso é baseado em uma disciplinada e rigorosa busca de inovação em condições de incerteza. Que os mesmos princípios estão no coração tanto do desenvolvimento de produto enxuto quanto da mudança cultural e de processo eficaz foi uma epifania para os autores deste livro, mas talvez isso não devesse ser uma surpresa — em ambos os casos enfrentamos a incerteza e temos de lidar com um sistema adaptativo complexo cuja resposta a mudança é imprevisível. Ambas as situações pedem um progresso incremental e iterativo, alcançado por meio da criatividade humana aliada ao método científico.

As organizações devem continuamente revisitar a questão: “Qual é o nosso propósito e como podemos nos organizar para aumentar nosso potencial de longo prazo e de nossos consumidores e empregados?”. O trabalho mais importante para líderes é buscar a cultura de alta performance descrita neste livro. Dessa maneira, podemos prosperar em um ambiente de constantes avanços em design e tecnologia, e ampla mudança social e econômica.

## CAPÍTULO 16

# REFERÊNCIAS BIBLIOGRÁFICAS

## A

ADZIC, G. *Impact Mapping: Making a Big Impact with Software Products and Projects*. Provoking Thoughts, 2012.

ALLSPAW, John. *Counterfactual Thinking, Rules, and The Knight Capital Accident*. Out. 2013. Disponível em: <http://bit.ly/1e9idko>.

ANDERSON, D. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.

ARGYRIS, C.; SCHÖN, D. *Organizational Learning: A Theory of Action Perspective*. Addison Wesley, 1978.

ARNOLD, J.; YÜCE, Ö. Black Swan Farming Using Cost of Delay: Discover, Nurture and Speed Up Delivery of Value. *Proceedings of the 2013 Agile Conference*, IEEE Computer Society Washington, Nashville, USA, p. 101–116, 2013.

## B

BAGHAI, M.; COLEY, S.; WHITE, D. *The Alchemy of Growth*. Texere, 1999.

BARLOW, Jake. *You can now book a prison visit online*. Set.

2014. Disponível em: <http://bit.ly/1v73X8w>.

BARR, Philippa Nicole. *Data sovereignty: Up in the clouds.* Maio 2014. Disponível em: <http://bit.ly/1v73C5K>.

BATTARBEE, Katja; SURI, Jane Fulton; HOWARD, Suzanne Gibbs. *Empathy on the Edge: scaling and sustaining a human-centered approach in the evolving practice of design.* Jan. 2014. Disponível em: <http://bit.ly/1v6ZlPI>.

BATTELLE, John. *The 70 Percent Solution.* Dez. 2005. Disponível em: <http://cnnmon.ie/1v6YHBA>.

BEGLEY, Sharon. *As Obamacare tech woes mounted, contractor payments soared.* Out. 2013. Disponível em: <http://reut.rs/1v74loJ>.

BELL, S. C.; ORZEN, M. A. *Lean IT.* Productivity Press, 2011.

BERKUN, Scott. *The Dangers of Faith In Data.* Nov. 2013. Disponível em: <http://scottberkun.com/2013/danger-of-faith-in-data>.

BERTRAND, M.; MULLAINATHAN, S. Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination? *American Economic Review*, v. 94, n. 4, p. 991–1013, 2004.

BETZ, C. *Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler's Children.* Morgan Kaufmann, 2006.

BIDMEAD, Chris. *ARM creators Sophie Wilson and Steve Furber.* Maio 2012. Disponível: <http://bit.ly/1v6Ynmw>.

BINETTI, David. When and How to Pivot. *Startups Lessons Learned,* SLLCONF 2011, Maio 2011. Disponível em: <http://slidesha.re/1v6ZQZZ>.

BLANK, S. *The Four Steps to the Epiphany*: Successful Strategies for Products That Win. K&S Ranch Press, 2005.

BLANK, S. *Search versus Execute*. Mar. 2012. Disponível em: <http://steveblank.com/2012/03/05/search-versus-execute>.

BRODZINSKI, Pawel. *Portfolio Management: Stop Starting, Start Finishing*. Jun. 2014. Disponível em: <http://brodzinski.com/2014/06/portfolio-management.html>.

BROHAN, Mark. *Amazon sales top \$61 billion in 2012*. Jan. 2013. Disponível em: <http://bit.ly/1v701og>.

BROWN, Kristen V. *Tech shift: More women in computer science classes*. Fev. 2014. Disponível em: <http://bit.ly/1v72O0L>

BRYANT, Adam. *In Head-Hunting, Big Data May Not Be Such a Big Deal*. Jun. 2013. Disponível em: <http://nyti.ms/1v72xuz>.

BODEK, N. *Kaikaku: The Power and Magic of Lean*. PCS Press, 2004.

BOGSNES, B. *Implementing Beyond Budgeting*. John Wiley & Sons, 2009.

BOSSAVIT, L. *The Leprechauns of Software Engineering: How Folklore Turns into Fact, and What to Do About It*. Leanpub, 2013. Disponível em: <https://leanpub.com/leprechauns>.

BOTTCHER, Evan. *Projects Are Evil and Must Be Destroyed*. Ago. 2010. Disponível em: <http://bit.ly/1v73umC>.

BUNGAY, S. *The Art of Action: How Leaders Close the Gaps Between Plans, Actions, and Results*. Nicholas Brealey Publishing, 2010.

## C

CAGAN, Marty. *Inspired: How to Create Products Customers Love*. SVPG Press, 2008.

CAGAN, Marty. *Minimum Viable Product*. Ago. 2011a. Disponível em: <http://www.svpg.com/minimum-viable-product>.

CAGAN, Marty. *Product Market Fit vs. Product Vision*. Out. 2011b. Disponível em: <http://svpg.com/product-market-fit-vs-product-vision>.

CARR, Nicholas G. *IT Doesn't Matter*. Maio 2003. Disponível em: <https://hbr.org/2003/05/it-doesnt-matter>.

CARUSO-CABRERA, Michelle. *3M CEO: Research Is 'Driving This Company'*. Jun. 2013. Disponível em: <http://www.cnbc.com/id/100801531>.

CECI, S. J.; WILLIAMS, W. M. Gender Differences in Math-Intensive Fields. *Current Directions in Psychological Science*, v. 19, p. 275–279, 2010.

CÉDIEY, E.; FORONI, F.; GARNER, H. Discrimination à l'embauche fondée sur l'origine à l'encontre des jeunes français(e)s peu qualifié(e)s. *Premières Infos Premières Synthèses*, v. 06.3, 2008.

CHRISTIENSEN, Clayton. *The Innovator's Dilemma: The Revolutionary Book That Will Change the Way You Do Business*. Harper Business, 1997.

CRAWFORD, Jason. *Amazon's "two-pizza teams"*: The ultimate divisional organization. Jul. 2013. Disponível em: <http://blog.jasoncrawford.org/two-pizza-teams>.

CRISPIN, L.; GREGORY, J. *Agile Testing: A Practical Guide for Testers and Agile Teams*. Addison-Wesley, 2009.

CROLL, A.; YOSKOVITZ, B. *Lean Analytics: Use Data to Build*

a Better Startup Faster. O'Reilly, 2012.

CUNNINGHAM, Andrew; MYERS, Andrew. *A Deployment Pipeline for Infrastructure: A DevOps Case Study at NBN*. Dez. 2012. Disponível em: <https://puppet.com/blog/a-deployment-pipeline-for-infrastructure-a-devops-case-study-at-nbn>.

## D

DANISH DESIGN CENTER. *Evaluation of the Importance of Design* 2006.

DEKKER, S.; HOLLNAGEL, E.; WOODS, D.; COOK, R. *Resilience Engineering: New Directions for Measuring and Maintaining Safety in Complex Systems*. Lund University School of Aviation, 2008.

DEMING, W. E. *Out of the Crisis*. MIT Press, 2000.

DUHIGG, Charles. *The Power of Habit: Why We Do What We Do in Life and Business*. Random House, 2014.

## E

EDWARDS, S. Activity Based Costing: Topic Gateway Series. CIMA, n. 1, 2008.

ELMAN, Josh. *The only metric that matters*. Nov. 2012. Disponível em: <https://medium.com/what-i-learned-building/ab24a585b5ea>.

## F

FARRIS, P. W.; BENDLE, N. T.; PFEIFER, P. E.; REIBSTEIN, D. J. *Marketing Metrics: The Definitive Guide to Measuring Marketing Performance*. 2. ed. Pearson, 2010.

FRASER, Janice. *Crushing the Boulder: UX for Lean Startups*. Jul. 2011. Disponível em: <http://slidesha.re/1v715bL>.

FRIEDMAN, Thomas L. *How to Get a Job at Google*. Fev. 2014. Disponível em: <http://nyti.ms/1v72sHl>.

FORSGREN, N.; KIM, G.; KERSTEN, N.; HUMBLE, J. *2014 State of DevOps Report*. PuppetLabs, 2014. Disponível em: <http://bit.ly/2014-devops-report>.

FOWLER, Martin. *StranglerApplication*. Jun. 2004. Disponível em: <http://bit.ly/1v71DOH>.

FOWLER, Martin. *SelfInitializingFake*. Ago. 2009a. Disponível em: <http://martinfowler.com/bliki/SelfInitializingFake.html>.

FOWLER, Martin. *TechnicalDebtQuadrant*. Out. 2009b. Disponível em: <http://martinfowler.com/bliki/TechnicalDebtQuadrant.html>.

FOWLER, Martin. *DiversityMediocrityIllusion*. Jan. 2015. Disponível em: <http://bit.ly/1v72Rtt>.

FREEMAN, S.; PRICE, N. *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley, 2009.

## G

GILB, T. *Principles of Software Engineering Management*. Addison-Wesley, 1988.

GILB, T. *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*. Butterworth-Heinemann, 2005.

GITTLESON, Kim. *Can a company live forever?* Jan. 2012. Disponível em: <http://www.bbc.co.uk/news/business-16611040>.

GLADWELL, Malcolm. *The Talent Myth* Jul. 2002. Disponível em: <http://gladwell.com/the-talent-myth>.

GRAHAM, Paul. Do Things that Don't Scale: Want to start a startup?. Jul. 2013. Disponível em: <http://paulgraham.com/ds.html>.

GRAY, David; BROWN, S.; MACANUFO, J. *Gamestorming*. O'Reilly, 2010.

GRAY, David. *Elevator pitch*. Jan. 2010. Disponível em: <http://www.gogamestorm.com/?p=125>.

GRUVER, G. *A Practical Approach to Large-Scale Agile Development: How HP Transformed LaserJet FutureSmart Firmware*. Addison-Wesley, 2012.

GRUVER, Gary; YOUNG, Mike; FULGHUM, Pat. *A Practical Approach to Large-Scale Agile Development: How HP Transformed LaserJet FutureSmart Firmware*. New Jersey: Pearson Education, Inc. 2013.

GUALTIERI, Mike. *I Don't Want DevOps. I Want NoOps*. Fev. 2011. Disponível em: <http://bit.ly/1v73wLd>.

GUEST, David. The hunt is on for the Renaissance Man of computing. *The Independent*, Londres, set. 1991.

GUNTHER, Marc. *3M's innovation revival*. Set. 2010. Disponível em: <http://for.tn/11ixTko>.

GOLDIN, C.; ROUSE, C. Orchestrating Impartiality: The Impact of 'Blind' Auditions on Female Musicians. *American Economic Review*, v. 90, n. 4, p. 715–741, 2000.

GOTHELF, J.; SEIDEN, J. *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly, 2013.

## H

HAMMANT, Paul. *Facebook's Trunk Based Development (take 2)*. Mar. 2013. Disponível em: <http://paulhammant.com/2013/03/13/facebook-tbd-take-2>.

HASTINGS, Reed. *Netflix Culture*. Ago. 2009. Disponível em: <http://slidesha.re/1v71niL>.

HILL, Kashmir. *How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did*. Fev. 2012. Disponível em: <http://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/#253aaceb34c6>.

HOPE, J.; FRASER, R. *Beyond Budgeting: How Managers Can Break Free from the Annual Performance Trap*. Harvard Business School Press, 2003.

HUANG, Ryan. *Six Sigma 'killed' innovation in 3M*. Mar. 2013. Disponível em: <http://zd.net/1v71quY>.

HUBBARD, Douglas. *The IT Measurement Inversion: Are your IT investment decisions based on the right information?* Jun. 2007. Disponível em: <http://www.cio.com/article/2438748/it-organization/the-it-measurement-inversion.html>.

HUBBARD, Douglas. *How to Measure Anything: Finding the Value of "Intangibles" in Business*. 2. ed. Wiley, 2010.

HUMBLE, Jez; FARLEY, David. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley, 2010.

## I

ISACA. *COBIT 5 Framework*. ISACA & ITGI, 2012.

## J

JARVIS, Audley. *How Kodak invented the digital camera in 1975*. Maio 2008. Disponível em: <http://bit.ly/1v6YwGv>.

JENSEN, Michael C.; MECKLING, William H. *Teoria da Firma: comportamento dos administradores, custos de agência e estrutura de propriedade*. Journal of Financial Economics, v. 3, n. 4, 1976.

## K

KAHNEMAN, D. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

KANE, S. *Your Startup Is Broken: Inside the Toxic Heart of Tech Culture*. Model, View, Culture, 2014.

KAPLAN, Robert; NORTON, David. *The Balanced Scorecard — Measures that Drive Performance*. Fev. 1992. Disponível em: <http://bit.ly/1vt3X2Q>.

KAY, John. *Obliquity*. Penguin Books, 2010.

KENNY, Graham. *Your Company's Purpose Is Not Its Vision, Mission, or Values*. Set. 2014. Disponível em: <http://bit.ly/1zmWArB>.

KERTH, N. *Project Retrospectives: A Handbook for Team Reviews*. Dorset House, 2001.

KLEIN, Freida Kapor. *Giving Notice: Why the Best and Brightest are Leaving the Workplace and How You Can Help them Stay*. Jossey-Bass, 2007.

KLEIN, Laura. *UX for Lean Startups: Faster, Smarter User Experience Research and Design*. O'Reilly, 2013.

KOHAVI, Ronny; CROOK, Thomas; LONGBOTHAM, Roger; FRASCA, Brian; HENNE, Randy; FERRES, Juan L.; MELAMED, Tamir. *Online Experimentation at Microsoft*. Set. 2009. Disponível em: <http://stanford.io/130uW6X>.

KOHAVI, Ronny. *Online Controlled Experiments: Introduction, Insights, Scaling, and Humbling Statistics*. Dez. 2013. Disponível em: <http://bit.ly/1v6ZHpn>.

KONKEL, Frank. *The Details About the CIA's Deal With Amazon*. Jul. 2014. Disponível em: <http://theatl.tc/1v73AuB>.

KOTTER, J. *Leading Change*. Harvard Business Review Press, 2012.

KOTTKE, Daniel. *Macintosh Prototypes*. Jun. 1981. Disponível em: [http://folklore.org/StoryView.py?story=Macintosh\\_Protoypes.txt](http://folklore.org/StoryView.py?story=Macintosh_Protoypes.txt).

## L

LACEY, Mitch. *The Scrum Field Guide: Practical Advice for Your First Year*. Addison-Wesley, 2012.

LANE-FOX, M. *DirectGov 2010 and Beyond: Revolution Not Evolution*. Out. 2010. Disponível em: <http://bit.ly/11iByi9>.

LAPOUCHNIAN, A. *Goal-Oriented Requirements Engineering: An Overview of the Current Research*. University of Toronto Department of Computer Science, 2005.

LEE, Dave. *Nokia: The rise and fall of a mobile giant*. Set. 2013. Disponível em: <http://www.bbc.co.uk/news/technology-23947212>.

LESS — LEAN ENTERPRISE SOFTWARE AND SYSTEMS CONFERENCE. *What is Beyond Budgeting?* Nov. 2012. Disponível em: <http://www.slideshare.net/LESSConf/11-12-what-is-bb>.

LIKER, J. *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer*. McGraw-Hill, 2003.

LINDEN, Greg. *Early Amazon: Shopping cart recommendations*. Abr. 2006. Disponível em: <http://bit.ly/1v71kmW>.

LIMONCELLI, Tom. *Yes, you really can work from HEAD* Mar. 2014. Disponível em: <http://bit.ly/1v70LcY>.

LIMONCELLI, T. A.; CHALUP, S. R.; HOGAN, C. J. *The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems*. Vol. 2. Addison-Wesley, 2014.

## M

MANNS, M. L.; RISING, L. *Fearless Change: Patterns for Introducing New Ideas*. Addison-Wesley, 2004.

MARCH, J. Exploration and Exploitation in Organizational Learning. *Organizational Science*, v. 2, p. 71–87, 1991.

MARTIN, K. *The Outstanding Organization: Generate Business Results by Eliminating Chaos and Building the Foundation for Everyday Excellence*. McGraw-Hill, 2012.

MARTIN, K.; OSTERLING, M. *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation*. McGraw-Hill, 2014.

MAASSEN, Olav; MATTS, Chris; GEARY, Chris. *Commitment*. Hathaway Te Brake Publications, 2013.

MAURYA, Ash. *Running Lean: Iterate from Plan A to a Plan That Works*. O'Reilly, 2010a.

MAURYA, Ash. *3 Rules to Actionable Metrics in a Lean Startup*.

Jul. 2010b. Disponível em: <http://bit.ly/1v6ZG4L>.

MCCLURE, Dave. *Startup Metrics for Pirates: AARRR!*. Set. 2007. Disponível em: <http://bit.ly/1vt4925>.

MCCONNELL, Steve. *Origins of 10X – How Valid is the Underlying Research?* Jan. 2011. Disponível em: <http://bit.ly/1v72hvu>.

MCGREGOR, D. *The Human Side Of Enterprise*: 25th Anniversary Printing. McGraw-Hill, 1995.

MCKINLEY, Dan. *Design for Continuous Experimentation*. Dez. 2012. Disponível em: <http://slidesha.re/1v71gUs>.

MERINO, Faith. *How does Amazon make money? Really, how does it?...* Nov. 2013. Disponível em: <http://bit.ly/1v700QY>.

MOSS-RACUSIN, C. A.; DOVIDIO, J. F.; BRESCOLL, V. L.; GRAHAM, M. J.; HANDELSMAN, J. Science Faculty's Subtle Gender Biases Favor Male Students. *Proceedings of the National Academy of Sciences of the United States of America*, v. 109, n. 41, p. 16474–16479, 2012.

MOORE, G. A. *Escape Velocity: Free Your Company's Future from the Pull of the Past*. HarperCollins, 2011.

MOREHEAD, James. *Stanford University's Carol Dweck on the Growth Mindset and Education*. Jun. 2012. Disponível em: <http://bit.ly/1v72nmV>.

## N

NICOT, Anne-Marie. *Study reveals ethnic discrimination in recruitment of young workers*. Maio 2008. Disponível em: <http://www.eurofound.europa.eu/ewco/2008/02/FR0802019I.htm>.

## O

OHNO, T. *Taiichi Ohnos Workplace Management*: Special 100th Birthday Edition. McGraw-Hill Professional, 2012.

## P

PARNAS, D. L. On the Criteria to Be Used in Decomposing Systems into Modules. *Communications of the ACM*, v. 15, n. 12, p. 1053–1058, 1972.

PATTON, Jeff. *User Story Mapping: Discover the Whole Story, Build the Right Product*. O'Reilly Media, 2014.

PENIX, John. *Large-Scale Continuous Testing in the Cloud*. Maio 2013. Disponível em: <http://bit.ly/1v70NBG>.

PINK, D. H. *Drive: The Surprising Truth About What Motivates Us*. Penguin Group, 2009.

POLS, Andy; STEVENSON, Chris. *An Agile Approach to a Legacy System*. Extreme Programming and Agile Processes in Software Engineering, 5th International Conference, Jun 2004. Disponível em: <http://bit.ly/1v71GtR>.

POPPENDIECK, M.; POPPENDIECK, T. *Leading Lean Software Development: Results Are Not the Point*. Addison-Wesley, 2009.

POPPENDIECK, M.; POPPENDIECK, T. *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley, 2006.

PREEZ, Derek du. *CGI President slams government reforms – warns investment will be taken abroad*. Jul. 2013. Disponível em: <http://bit.ly/1v742ZT>.

PYNCHON, Victoria. *Oops! Your Implicit Bias is Showing*. Abr. 2014. Disponível em: <http://www.shenegotiates.com/blog/blog/2014/4/16/oops-your-implicit-bias-is-showing>.

## Q

## R

RAO, Venkatesh. *The Gervais Principle, Or The Office According to "The Office"*. Out. 2009. Disponível em: <http://bit.ly/1v71WJq>.

REINERTSEN, D. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas, 2009.

RICHARDS, Chet. *Boyd's OODA Loop: It's Not What You Think*. Mar. 2012. Disponível em: [http://www.jvminc.com/boydsrealooda\\_loop.pdf](http://www.jvminc.com/boydsrealooda_loop.pdf).

RIES, Eric. *What is customer development?* Nov. 2008. Disponível em: <http://bit.ly/1v6Y8YI>.

RIES, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Bussiness, 2011.

ROBBINS, Jesse. *Hacking Culture*. Fev. 2013. Disponível em: <http://www.infoq/presentations/Hacking-Culture>.

ROBBINS, Jesse; KRISHNAN, Kripa; ALLSPAWE, John; LIMONCELLI, Tom. Resilience Engineering: Learning to Embrace Failure. Queue - File Systems, v. 10, issue 9, set 2012. Disponível em: <http://queue.acm.org/detail.cfm?id=2371297>.

ROGERS, E. *Diffusion of Innovations* 5. ed. Free Press, 2003.

ROTHER, M. *Toyota Kata: Managing People for Improvement*,

Adaptiveness, and Superior Results. McGraw-Hill, 2010.

ROTHER, M. *Improvement Kata Handbook*. 2014. Disponível em: <http://bit.ly/11iBzLY>.

ROTHER, M.; SHOOK, J. *Learning to See: Value-Stream Mapping to Create Value and Eliminate Muda*. Lean Enterprise Institute, 2009.

RUBIN, Kenneth S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley, 2012.

## S

SACKMAN, H.; ERIKSON, W. J.; GRANT, E. E. Exploratory Experimental Studies Comparing Online and Offline Programming Performance. *Communications of the ACM*, v. 11, n. 1, p. 3–11, 1968.

SARASVATHY, Sara. *Effectuation: Elements of Entrepreneurial Expertise*. Edward Elgar Publishing, 2009.

SCHEIN, E. H. *The Corporate Culture Survival Guide: New and Revised Edition*. Jossey-Bass, 2009.

SCHPILBERG, D.; BEREZ, S.; PURYEAR, R.; SHAH, S. *Avoiding the Alignment Trap in Information Technology*. MIT Sloan Management Review Fall 2007, 2007.

SEDDON, John. *I Want You to Cheat!: The Unreasonable Guide to Service and Quality in Organisations*. Vanguard Consulting, 1992.

SEDDON, John. *Freedom from Command & Control: Rethinking Management for Lean Service*. Productivity Press, 2005.

SEMLER, R. *Maverick: The Success Story Behind the World's*

Most Unusual Workplace. Grand Central Publishing, 1995.

SHANE, Scott. *Start-up Failure Rates Vary — Choosing the Right Industry Matters*. Maio 2008. Disponível em: <http://bit.ly/1v6YeiN>.

SHANE, Scott. *Start Up Failure Rates: The Definitive Numbers*. Dez. 2012. Disponível em: <http://bit.ly/1v6YfTX>.

SHOOK, John. *How to Change a Culture: Lessons From NUMMI*. MIT Sloan Management Review, Jan. 2010. Disponível em: <http://bit.ly/1v720ZH>.

SHORE, James. *Continuous Integration on a Dollar a Day*. Fev. 2006. Disponível em: <http://www.jamesshore.com/Blog/Continuous-Integration-on-a-Dollar-a-Day.html>.

SOBEK, D. K., II; SMALLEY, A. *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*. Productivity Press, 2008.

STEPHENS, Ashley. *Governance for service delivery*. Jun. 2014. Disponível em: <http://bit.ly/1v747fT>.

SUTTON, Bob. *Strong Opinions, Weakly Held*. Jul. 2006. Disponível em: <http://bit.ly/1v72zTg>.

## T

TALEB, N. N. *Antifragile: Things That Gain From Disorder*. Random House, 2012.

TECHONOMY. *A Conversation with Mark Bertolini*. Nov. 2013. Disponível em: <http://bit.ly/1v6YM8m>.

THE ECONOMIST. *Special Report: Outsourcing and Offshoring*, 406, n. 8819, 19 de Janeiro 2013.

TURTON, Stuart. *Whatever happened to Hermann Hauser?* Jun. 2010. Disponível: <http://bit.ly/1v6YoH7>.

## U

## V

VELASQUEZ, Nicole Forsgren. *2014 State of DevOps Report: Statistics Class Edition.* Ago. 2014. Disponível em: <http://bit.ly/1v71SJ>.

## X

## W

WARDLEY, Simon. *Why Map?* Nov. 2013. Disponível em: <http://blog.gardenviance.org/2013/09/why-map.html>.

WESTRUM, R. The Study of Information Flow: A Personal Journey. *Safety Science*, v. 67, p. 58–63, 2014.

WIDENER, S. K. An Empirical Analysis of the Levers of Control Framework. *Accounting, Organizations and Society*, v. 32, n. 7–8, p. 757–788, 2007.

WILLIAMS, Branden. *PCI and the Art of the Compensating Control.* Mar. 2010. Disponível em: <http://bit.ly/1v732EU>.

WOMACK, J. P.; JONES, D. T. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation.* 2. ed. Simon and Schuster, 2010.

## Y

YU, E.; GIORGINI, P.; MAIDEN, N.; MYLOPOULOS, J. *Social Modeling for Requirements Engineering.* MIT Press, 2010.

**Z**