

Gestão de produtos

Como aumentar as chances de sucesso do seu software



© Casa do Código

Todos os direitos reservados e protegidos pela Lei nº9.610, de 10/02/1998.

Nenhuma parte deste livro poderá ser reproduzida, nem transmitida, sem autorização prévia por escrito da editora, a seja quais forem os meios: fotográficos, eletrônicos, mecânicos, gravação ou quaisquer outros.

Edição

Adriano Almeida

Vivian Matsui

Revisão e diagramação

Bianca Hubert

Vivian Matsui

[2015]

Casa do Código

Livro para o programador

Rua Vergueiro, 3185 – 8º andar

04101-300 – Vila Mariana – São Paulo – SP – Brasil

www.casadocodigo.com.br

SOBRE GESTÃO DE PRODUTOS DE SOFTWARE

Já chegamos àquele ponto no qual há software em lugares em que, há alguns anos, jamais imaginaríamos que haveria necessidade e utilidade de ter um software: televisão, geladeira, fogão, carro, relógio de pulso, óculos, roupa, fechadura, mesa, cadeira, equipamento médico e por aí vai. Isso sem falar no mais óbvio, o telefone que está em nossos bolsos ou bolsas e do qual estamos nos tornando cada vez mais dependentes.

A ubiquidade do software é hoje uma realidade, o software permeia inúmeras atividades e objetos de nosso dia a dia. Acredito que se possa dizer que 100% da população mundial tem suas vidas afetadas por software, e boa parte dessa população tem contato frequente com software. De acordo com o relatório *We Are Social* de 2015 (www.slideshare.net/wearesocialsg/digital-social-mobile-in-2015), mesmo regiões pouco desenvolvidas como a África já têm mais de 25% de sua população ativa na internet.

Mesmo com essa evidência de que o software está cada dia mais fazendo parte da vida de todas as pessoas do planeta, ainda tenho a impressão de que não damos a ele a devida atenção e cuidado. Pense em todas as vezes em que você usou um software nos últimos sete dias.

Você teve alguma experiência frustrante com ele? Tenho certeza de que sim.

Eu tenho experiências frustrantes com software diariamente. Mesmo softwares feitos por quem consideramos experts no assunto de desenvolver software – como Google, Facebook e outras empresas que nasceram e cresceram fazendo-os e que são frequentemente citadas como referências do processo do seu desenvolvimento – nos causam frustração.

POR QUE ISSO ACONTECE?

O desenvolvimento de software evoluiu muito ao longo dos anos. Em termos de processo, antigamente tínhamos o *waterfall*, em que cada fase do desenvolvimento de software acontecia de forma sequencial. A distância entre a necessidade que gerou a demanda de desenvolver o software e o software propriamente dito era enorme.

No início deste milênio, começamos a experimentar as metodologias ágeis, que transformaram o processo de desenvolvimento de software em um ciclo com interações curtas que promovem o feedback contínuo. Isso ajudou consideravelmente a diminuir a distância entre a necessidade que gerou a demanda de desenvolver o software e o software propriamente dito.

Do ponto de vista dos aspectos levados em consideração no desenvolvimento de software, também evoluímos bastante. Os primeiros

softwares eram desenvolvidos por times compostos exclusivamente por desenvolvedores de software. Aliás, não era raro esses times serem compostos de uma única pessoa. Cada vez mais vemos times multidisciplinares trabalhando juntos no desenvolvimento de software; o que é muito bom, pois traz novas visões para o software que está sendo desenvolvido.

De um lado, a preocupação com o usuário, seus objetivos ao usar o software e sua interação com esse software são cuidados por profissionais de experiência do usuário, ou UX (do inglês *User eXperience*).

Do outro lado, a preocupação com a operação do software – ou seja, onde esse software vai rodar e que performance e disponibilidade ele precisa ter – trouxe profissionais de administração de sistemas, os SysAdmins (do inglês *System Administrators*), mais próximos do processo de desenvolvimento de software. Essa aproximação da operação do software com o seu desenvolvimento é o que deu origem ao termo e à cultura DevOps.

Ou seja, entregamos software mais frequentemente, e trouxemos UX e SysAdmins para o desenvolvimento de software; mas será que isso é suficiente? Como vamos contar para o mundo, ou melhor, para as pessoas que podem ser beneficiadas com esse software, que esse software existe? Como vamos cuidar das suas questões jurídicas? Quando o usuário tiver um problema com o software, como vamos ajudá-lo? Como vamos gerenciar o retorno que ele trará? Como garantir que esse software atenda os objetivos de seu dono ao mesmo tempo em que atenda a necessidade de seus usuários?

GESTÃO DE PRODUTOS DE SOFTWARE

Pensando nessas questões, algumas empresas que são consideradas experts no tema desenvolvimento de software começaram a adotar uma nova função em seu processo de desenvolvimento de software, a **Gestão de Produtos de Software**. Essa função tem por objetivo garantir que um software sendo desenvolvido atenda aos objetivos de seu dono ao mesmo tempo em que atenda à necessidade de seus usuários.

Além disso, essa função tem de pensar em todos os aspectos do software que citei anteriormente. Algumas metodologias ágeis, como o Scrum, têm o papel do *Product Owner*, que tem como principal responsabilidade priorizar os itens a serem desenvolvidos. De uma certa forma, a **Gestão de Produtos de Software** é isso, mas ainda é um pouco mais.

É sobre isso que vou falar neste livro. :-)

PARA QUEM É ESTE LIVRO?

A resposta a essa pergunta é bem simples. Este livro é indicado para qualquer pessoa que trabalhe com software. Ele serve para pessoas que são gestoras de produto, pois todo gestor de produto sabe que sempre há muito por aprender. E mesmo aqueles que já conheçam bem todos os temas apresentados aqui poderão tirar proveito revendo algum assunto.

Este livro também é indicado para qualquer pessoa que esteja querendo entrar na carreira de gestor de produto. Acredito que ele possa

tirar um pouco da ansiedade de quem estiver pensando em se tornar gestor de produto, e não sabe ao certo o que fará e o que as outras pessoas esperarão dele.

Lembro uma vez de um amigo meu que era desenvolvedor de software e decidiu virar gestor de produtos. Ele disse que, nos primeiros meses, ele não entendia o que estava fazendo. Acostumado a medir o progresso do seu trabalho com código em produção, ao assumir a gestão de produtos, ficou perdido sem entender como medir se ele estava de fato entregando algo. Chegou inclusive a pensar em voltar a ser desenvolvedor de software. Já vi casos de pessoas que experimentaram por dois ou três meses e voltaram à função anterior.

Acredito que mesmo as pessoas que não são e não pretendem ser gestoras de produto também poderão tirar proveito deste livro, entendendo o que essa função faz e como ela deve se relacionar com as outras áreas.

Note que eu disse que este livro é indicado para qualquer pessoa que trabalhe com software. Mesmo empresas que não tem software como seu *core business* utilizam software no seu dia a dia e, não raro, desenvolveram algum software que tem interface com seus clientes como, um site ou um aplicativo mobile, por exemplo. É importante para essas empresas entenderem a função de gestão de produtos de software, para elas poderem gerir melhor esse software e aumentar suas chances de sucesso.

Vale lembrar de que este livro não tem a pretensão de cobrir de forma extensiva todos os temas que serão abordados, mesmo porque, se

o fizesse, provavelmente teria de ser em uma coleção de livros. Meu objetivo é falar sobre os principais temas relacionados com gestão de produtos de software, aprofundando em alguns temas específicos e fornecendo várias dicas de leitura adicional.

Em alguns lugares do livro, farei referências sobre as metodologias ágeis de desenvolvimento de software e o papel de PO (*product owner*). Acredito que o conhecimento desse processo de desenvolvimento e dos diferentes papéis envolvidos nele não seja necessariamente um pré-requisito para a leitura deste livro, mas é certamente um saber que ajudará a aumentar as chances de sucesso do seu software. Caso você queira se aprofundar no tema, recomendo o livro *Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio*, do André Faria Gomes. Além de explicar os princípios por trás da cultura ágil, ele também fala sobre Scrum, XP e Kanban (as três metodologias ágeis mais conhecidas), e sobre como espalhar essa cultura por toda a empresa. Leitura recomendada.

ESTRUTURA DO LIVRO

Escrevi o livro seguindo a seguinte estrutura:

- **Definições e requisitos:** para começar, farei uma rápida introdução às metodologias ágeis. Algumas das pessoas que leram os primeiros rascunhos acharam que poderia ser útil fazer esta introdução, já que falo sobre seus determinados aspectos em alguns pontos do livro. Em seguida, definirei algumas palavras-chave como software, produto, plataforma, gestão de produtos, entre outras. Nesta parte, falarei tam-

bém sobre as características de um bom gestor de produto, e darei algumas dicas para gestores de produto sobre liderança e cultura organizacional.

- **Ciclo de vida de um produto de software:** nesta parte, descreverei como é o ciclo de vida de um produto de software e quais são as fases deste ciclo: inovação, crescimento, maturidade e fim de vida. Ainda, falarei sobre o que é inovação, como encontrar um problema a ser resolvido, como descobrir se ele é de fato uma oportunidade a ser perseguida e como obter retorno com seu produto de software. Na fase do crescimento, quando o produto foi desenvolvido e lançado, devemos nos preocupar em como gerenciar o produto durante seu crescimento, ou seja, como gerenciar o feedback? O que é um *roadmap*? Como priorizar as demandas? O que fazer com pedidos especiais? Como dizer não? Que métricas acompanhar? Após esse crescimento, vem a maturidade. Nessa parte, vamos entender quando acontece a maturidade e o que fazer se o produto chega nessa fase. Depois da maturidade, ou quando o produto é desenvolvido mas não dá certo, chega a fase conhecida como fim de vida de um produto de software. Veremos como detectar e o que fazer nessa fase do ciclo. No final, quando já conhecermos todas as fases do ciclo de vida de um produto, mostrarei qual a diferença entre startup e gestão de produtos de software.

- **Relacionamento com as outras funções:** como o gestor de produtos deve se relacionar com as diferentes funções da empresa, como engenharia, UX, marketing de produtos, gestão de projetos, vendas, jurídico, financeiro e atendimento?

- **Gestão de portfólio de produtos:** por que algumas empresas decidem ter mais de um produto? Como elas fazem para gerenciar esse portfólio de produtos? Por que outras empresas preferem se focar em um único produto? Foco ou diversificação, qual é a estratégia mais apropriada? Como organizar o time de desenvolvimento de produtos de acordo com a estratégia escolhida? Esses temas são os quais considero tópicos avançados de gestão de produtos de software, e é o que veremos nos capítulos que compõem esta parte do livro.

- **Onde usar gestão de produtos de software:** será que gestão de produtos de software só pode ser usada por empresas que comercializam produtos de software e somente nos times de desenvolvimento que desenvolvem softwares comercializados como produtos? Ou será que outros tipos de empresa se beneficiariam ao pensar em seu software como um produto e ao usar as técnicas de gestão de produtos para aumentar as chances de sucesso dos softwares que desenvolvem? E onde devemos colocar a gestão de produtos de software em uma empresa? No marketing? Na área técnica? Esses serão os temas dos capítulos finais do livro.

Essa sequência tem uma ordem lógica de assuntos, e alguns capítulos referenciam temas abordados em capítulos anteriores. Por esse motivo, recomendo a leitura sequencial do livro. Contudo, eles também podem ser lidos isoladamente. Se você estiver muito curioso para ler sobre um determinado tema, pode pular direto para o respectivo capítulo.

Boa leitura!

QUEM É ESSE CARA PARA FALAR SOBRE GESTÃO DE PRODUTOS DE SOFTWARE?

Acho a sua pergunta bastante pertinente e apropriada; por isso, aqui vai um pequeno histórico. Acredito que minha experiência com gestão de produtos de software vem da época das primeiras linhas de código que escrevi, em meados da década de 1980. Desde desses primeiros programas, já me preocupava com facilidade de uso.

Naquela época, elementos de interação como menus, janelas e *mouse* estavam começando a ficar populares com as primeiras versões do Microsoft Windows e do Mac OS. Isso me mostrou que software não era composto apenas de um conjunto de instruções para a máquina executar. Para fazer um bom software, era (e ainda é) preciso pensar em como esse software deve interagir com seus usuários, se ele atende os objetivos de quem o desenvolveu, e assim por diante.

No final de 1992, quando estava terminando a faculdade de Engenharia da Computação no ITA, um tio meu me falou que ele conheceu um negócio muito bacana de computadores chamado BBS (*Bulletin Board System*). Ele não entendia nada de computadores, mas disse que tinha

algo a ver com redes, e que se eu achasse interessante, nós podíamos abrir um negócio juntos. Com mais dois sócios, meu tio e eu criamos a Dialdata BBS, que depois viria a ser um dos primeiros provedores de acesso à internet do Brasil, em 1995.

Durante esses anos na Dialdata, escrevi muitas linhas de código que se transformaram em softwares, que eram disponibilizados para os usuários da BBS usarem. Também escrevi o sistema de cobrança, utilizado pelos funcionários da Dialdata, para fazer a cobrança dos clientes. A interação com usuários internos e externos me ensinou muito sobre desenvolvimento de software. Não basta só ter uma ideia na cabeça e um computador na mão para sair codando o software. É preciso entender o que o usuário espera do software, e o que você e sua empresa planejam obter com ele.

Em 1998, a Dialdata foi vendida para uma empresa americana chamada VIA NET.WORKS, que estava comprando provedores de serviços de internet em vários lugares do mundo para criar um provedor global de serviços de internet, para fazer um IPO (*Initial Public Offering*, ou oferta pública inicial de ações em uma bolsa de valores). Nessa época, fui convidado para trabalhar com gestão de produtos na VIA NET.WORKS.

Foi a primeira vez em que tive contato com o termo e a função de gestão de produtos. Minha responsabilidade era criar um portfólio global de produtos a partir das diferentes ofertas de produtos das empresas que foram sendo adquiridas pela VIA NET.WORKS. Foi aí que comecei a entender a importância desse papel em empresas de tecnologia em geral e, especificamente, nas empresas de desenvolvimento de software.

Em 2005, Gilberto Mautner, que também estudou no ITA, me convidou para ajudá-lo a melhorar o processo de desenvolvimento de produtos na empresa dele, a Locaweb. Hoje, temos na Locaweb um portfólio de mais de 30 produtos e um time de mais de 10 gestores de produtos. O time completo de desenvolvimento de produtos – incluindo gestores de produto, designers de UX e engenheiros de software – conta com mais de 100 pessoas. Aprendemos muito ao longo desses anos, mas o processo de aprendizado não acabou e acho que nunca acabará; ele é constante e contínuo.

QUAL É O SEU PROPÓSITO?

Recentemente, li um livro intitulado *Como Avaliar Sua Vida (How will you measure your life?)*, do Prof. Clayton Christensen, professor de Harvard e criador do conceito de inovação disruptiva, que vou comentar mais à frente. Nesse livro, publicado em 2012, ele conta como percebeu que ao longo dos anos seus colegas de turma acabaram se tornando pessoas infelizes, com suas vidas pessoais e profissionais distantes da que haviam planejado na época da faculdade. Alguns tinham seus nomes atrelados a escândalos financeiros e fiscais. Outros se casaram, separaram e brigam judicialmente com os ex-cônjuges. Outros ainda mal conseguem acompanhar o crescimento dos filhos.

Essa constatação o fez refletir sobre como seria possível aumentar as chances de encontrar satisfação e felicidade ao longo da vida. No livro, ele propõe que uma forma de se fazer isso é aplicando algumas das ferramentas do mundo da administração à gestão da vida pessoal e profissional.

Uma dessas ferramentas é o propósito. O propósito empresarial é o motivo de existir de uma determinada empresa. Muitas publicam esse motivo de forma clara. O Google tem por propósito organizar toda a informação do mundo, e fazê-la universalmente acessível e útil. A Nike quer trazer inspiração e inovação para todos os atletas do mundo, e que se você tem um corpo, você também é um atleta.

Prof. Christensen propõe que as pessoas também tenham um propósito que deve nortear suas decisões ao longo da vida, da mesma forma que as empresas devem ter um propósito que norteie as suas. Achei essa ideia bem interessante e me provocou a pensar no meu propósito que, após analisar como invisto meu tempo e no que tenho prazer e satisfação em trabalhar, acabei definindo como sendo:

MEU PROPÓSITO É

ajudar as pessoas a criarem produtos de software melhores.

Por isso, compartilharei nas próximas páginas o que aprendi ao longo desses quase 30 anos de experiência. Acredito que o que vou compartilhar poderá ajudar na criação de softwares melhores, que atingirão os objetivos do dono do software, ao mesmo tempo em que atenderão as necessidades dos seus usuários.

Sei que ainda tenho muito a aprender e quero continuar aprendendo sempre. Como o aprendizado vem da troca de experiências e da conversa, convido-o a compartilhar suas experiências lá no meu blog (<http://guiadastartup.com.br>), ou via e-mail (jtorges@jig.com.br).

Vamos começar?



PREFÁCIO

Meu nome é Paulo Silveira e sou empreendedor há 13 anos. Ajudei a criar uma grande comunidade de desenvolvedores, o GUJ.com.br, além de três empresas de ensino e tecnologia, dentre elas a própria editora Casa do Código. Cada ano e mês que passa, tenho mais dificuldade em definir o que faço e quais são as minhas responsabilidades no trabalho. No término do expediente, muitas vezes tenho dificuldade em dizer que fui produtivo.

Ao ler o livro do Joaquim Torres, pude concluir o que já imaginava: sou realmente um gerente de produtos. Um que precisa melhorar muito. Como sei disso? Em um dos primeiros capítulos, Torres colocará uma lista de tarefas que fazem parte do dia a dia dos gerentes de produtos. É um *spoiler*, mas decidi colocar essa lista logo a seguir, com comentários meus, mostrando como todos são (deveriam ser) relevantes no meu cotidiano:

Com quantos clientes e usuários você falou essa semana?

fundamental e não posso deixar de lado.

Você tem uma estratégia e uma visão de longo prazo para o seu produto?

Parece fácil, mas esse exercício exige muito mais do que você imagina. Mesmo a visão de 2 anos, considerada um prazo relativamente curto, pode te dar dor de cabeça para imaginar. Se eu não sei onde quero estar em 2 anos, tenho certeza da motivação do que estou fazendo agora?

Como está e quais as últimas mudanças no cenário competitivo de seu produto?

Às vezes, me pego sem visitar os sites das empresas concorrentes por um longo tempo. Quando vou visitar, muitas vezes a surpresa é amarga. Essa situação pode ser ainda mais complicada no mercado de startups, tendo em vista que você pode ter concorrentes não muito bem definidos, seja pelo nicho específico, seja pelo mercado ainda incipiente.

Que insights você teve sobre o seu produto essa semana?

Certamente eu e meus sócios temos diversas ideias sobre o nosso produto, mas, na maioria das vezes, nunca comunicamos uns aos outros. Mais ainda: eu costumo ser muito crítico logo no início.

Você sabe qual a motivação e a métrica de cada item de seu roadmap?

De que adianta implementar tantas *features* se você não consegue mais dizer o porquê delas existirem? Porém, o mais grave: por que investir tanto esforço em uma feature que você não vai conseguir dizer

se agradou o usuário, se ajudou nas vendas, se resolveu um problema? Acredite: essa situação acontece todos os dias conosco e não percebemos.

Que novas tecnologias você vê aparecendo que podem influenciar, ou mesmo competir, com o seu produto?

A disrupção pode ser um pouco mais rara do que a mídia pinta por aí, mas ela existe. Mais importante: ela vem lá de baixo, normalmente de alguma outra tecnologia ou de outro mercado que você não dá muita bola. Fique atento.

Que novas habilidades você está procurando aprender?

Se a sua lista está vazia, você tem um problema! Cuidado para não ser absorvido pelo operacional. Claro que remover impeditivos é importante, como também é abordado no livro, mas o aprendizado deve ser contínuo. Bem, você já está no caminho certo ao encarar este livro.

Não é à toa que as perguntas são sempre muito relacionadas ao usuário. Enxergar-se como ele; *user centric*. Torres também gosta de evitar os termos belicosos que aparecem em livros negócios: ele sabe muito bem a importância que concorrentes possuem no mercado, e que eles têm papel importante no ecossistema.

No meio de dezenas de referências, livros e dicas, Joaquim Torres não vai trazer respostas simples para você. Muito pelo contrário. Assim como essa lista de tarefas, o autor nos apresenta perguntas, muitas perguntas. Dessa forma, você estará mais preparado para o trabalho.

Não mencionei o currículo do Joaquim Torres, mas esse é também um dos motivos pelo qual sempre o admirei e o considero como um dos meus mentores: foi o criador de uma das mais famosas BBSs, e é diretor da maior empresa de hosting do país, por onde passa por diversos desafios e amplia a gama de produtos da Locaweb. Essa experiência permeia o livro nos diversos exemplos dados. Aproveite!

Por Paulo Silveira

SUMÁRIO

PARTE I — DEFINIÇÕES E REQUISITOS

01	O QUE É UM PRODUTO DE SOFTWARE?	29
02	O QUE É GESTÃO DE PRODUTOS DE SOFTWARE?	39
03	GESTOR DE PRODUTOS OU PRODUCT OWNER?	51
04	PRINCIPAIS CARACTERÍSTICAS DE UM GESTOR DE PRODUTOS	59
05	DICAS DE LIDERANÇA PARA GESTORES DE PRODUTO	67
06	CULTURA ORGANIZACIONAL	75

PARTE II — CICLO DE VIDA DE UM PRODUTO DE SOFTWARE

07	COMO É O CICLO DE VIDA DE UM PRODUTO DE SOFTWARE?	91
08	INOVAÇÃO: O QUE É INOVAÇÃO?	101
09	INOVAÇÃO: FOCO NO PROBLEMA	107
10	INOVAÇÃO: O TRABALHO A SER FEITO	119
11	INOVAÇÃO: MUITAS OPORTUNIDADES	125
12	INOVAÇÃO: COMO OBTER RETORNO COM SEU PRODUTO DE SOFTWARE?	133
13	INOVAÇÃO: PRÓXIMOS PASSOS	149
14	CRESCIMENTO: FEEDBACK	157

15	CRESCIMENTO: O QUE É UM ROADMAP?	167
16	CRESCIMENTO: COMO PRIORIZAR O ROADMAP?	185
17	CRESCIMENTO: SEJA UM “DATA GEEK”	199
18	CRESCIMENTO: ENGAJAMENTO E CHURN	207
19	CRESCIMENTO: MÉTRICAS FINANCEIRAS	215
20	CRESCIMENTO: A MÉTRICA DA LEALDADE	223
21	CRESCIMENTO: CONSIDERAÇÕES SOBRE MÉTRICAS	231
22	MATURIDADE	241
23	FIM DE VIDA	255
24	QUAL A DIFERENÇA ENTRE STARTUP E GESTÃO DE PRODUTOS?	265

PARTE III — RELACIONAMENTO COM AS OUTRAS ÁREAS

25	ENGENHARIA DE PRODUTOS E GESTÃO DE PRODUTOS	273
26	UX E GESTÃO DE PRODUTOS	281
27	QUAL A DIFERENÇA ENTRE GESTÃO DE MARKETING DE PRODUTOS E GESTÃO DE PRODUTOS?	297
28	QUAL A DIFERENÇA ENTRE GESTÃO DE PROJETOS E GESTÃO DE PRODUTOS?	305
29	E AS OUTRAS ÁREAS?	313
30	DEFININDO RESPONSABILIDADES	325

PARTE IV — GESTÃO DE PORTFÓLIO DE PRODUTOS

31	JÁ ESTÁ PENSANDO EM SEU NOVO PRODUTO? NÃO? ENTÃO JÁ ESTÁ ATRASADO	333
32	COMO DIVERSIFICAR SEU PORTFÓLIO DE PRODUTOS?	341
33	COMO GERIR UM PORTFÓLIO DE PRODUTOS?	349
34	FOCO OU DIVERSIFICAÇÃO?	363
35	ORGANIZANDO PARA O FOCO E PARA A DIVERSIFICAÇÃO	373

PARTE V — ONDE USAR GESTÃO DE PRODUTOS DE SOFTWARE

36	QUE TIPO DE EMPRESA PRECISA DE UM GESTOR DE PRODUTOS?	385
37	O PROBLEMA DA TI	393
38	ONDE FICA A GESTÃO DE PRODUTOS DE SOFTWARE EM UMA EMPRESA?	403
39	CONCLUSÃO	411

PARTE I
DEFINIÇÕES E
REQUISITOS

Antes mesmo de definir gestão de produtos de software, vamos começar do começo, ou seja, definindo o que é software e o que é um produto de software. Em seguida, falarei sobre o que é gestão de produtos de software e quais são as principais características para ser um bom gestor.

Abordarei também as diferenças entre gestor de produtos e *product owner*, e quando é o momento certo para uma empresa ter um gestor. Por fim, darei algumas dicas de liderança para gestores de produtos que têm responsabilidade de liderar sem ser “chefe” de ninguém.

Preparado? :-)

CAPÍTULO 01

O QUE É UM PRODUTO DE SOFTWARE?

Antes mesmo de definir produto de software, acho que seria bom definirmos o que é software. De acordo com a Wikipédia (<https://en.wikipedia.org/wiki/Software>):

SOFTWARE

Software é qualquer conjunto de instruções que direciona o processador de um computador (o hardware) a executar operações específicas.

Nesse artigo da Wikipédia, existe até uma comparação do software com a música, em que os instrumentos musicais estão para o hardware como a música produzida nesses instrumentos está para o software. Achei essa comparação interessante.

Ótimo, já temos então uma definição de software, mas o que é então um produto de software?

Você certamente já usou algum produto de software, afinal, a internet é feita de produtos de software. Google, Facebook e Twitter são bons exemplos que você certamente já utilizou, e talvez use diariamente. Quando você faz compras na Amazon ou no Submarino, também está utilizando um produto de software. O sistema de internet banking do seu banco também pode ser considerado um, como o Netflix, que você pode acessar do computador, do smartphone, do tablet ou mesmo direto de sua TV.

O iOS e o Android, sistemas operacionais de smartphones, são produ-

tos de software. Existem também os produtos de software não online: os mais conhecidos, como o Office, Autocad, SAP e outros; e os menos conhecidos, aqueles softwares embarcados, que vêm dentro de hardwares que não são nem computadores, nem tablets, nem smartphones, mas sim dentro da impressora, da TV, de consoles de videogame, da urna eletrônica, de equipamentos de rede como roteadores, switches, hubs e firewalls etc.

PRODUTO DE SOFTWARE

Um produto de software é qualquer **software** que tenha **usuários**.

Então, todo software pode ser considerado um produto de software? Não, pois existem softwares que não têm usuários. São os softwares que fazem interface com outros softwares. Alguns exemplos são:

- Drivers de dispositivos de hardware que fazem a tradução entre o dispositivo e as aplicações ou o sistema operacional.
- Firmware, que é aquele conjunto de instruções operacionais programadas diretamente no hardware de um equipamento eletrônico.
- Camada de compatibilidade, que permite softwares rodarem em um ambiente no qual não foram originalmente programados para rodar.

Contudo, mesmo esse tipo de software se beneficiaria dos conceitos e das práticas da gestão de produtos de software que abordarei neste livro.

TIPOS DE PRODUTOS DE SOFTWARE

Podem-se classificar os produtos de software de diferentes formas, dependendo de como os enxergamos. Podemos olhar, como descrito anteriormente, para a forma como o produto é entregue aos usuários (como online, não online e embarcado), ou de acordo com o que ele faz: e-mail, comércio eletrônico, pagamento, e-mail marketing, gestão de conteúdo, educação, comunicação, colaboração, relatório, entretenimento, sistema operacional, ERP, CRM etc.

Outra forma – que é a qual prefiro, pois coloca o usuário no centro – é olhar e classificar produtos entendendo para quem o produto resolve o problema. Desse ponto de vista, podemos ter três tipos de produto de software:

Para o consumidor final: nesse tipo de produto, resolve-se um problema para o consumidor final que está disposto a pagar uma taxa para usar os serviços. Alguns exemplos são Netflix, LinkedIn e jogos. Pode-se também pensar em produtos web para consumidores finais que paguem a taxa pelo uso de forma indireta, ou seja, a taxa paga é por um serviço maior e o produto web está embutido nesse serviço. Alguns exemplos são internet banking, intranet de colégio ou faculdade, acesso a resultado de exames laboratoriais e softwares embarcados de forma geral. Os sites de comércio eletrônico são também dessa categoria, pois o uso do site é gratuito e a taxa paga é pelos produtos comprados no site que são entregues para o usuário.

Para as empresas: esse tipo de produto resolve o problema de empresas, que normalmente têm mais disposição de pagar que o consumidor final. Alguns exemplos são os produtos da Locaweb, Google AdWords, SAP, Autocad e Microsoft Office.

Mistos: nesse caso, o produto resolve tanto um problema para o consumidor final quanto para as empresas. Normalmente, esse tipo de produto não tem nenhum custo para o consumidor final; quem paga a conta são as empresas. O modelo mais comum de geração de receita nesse tipo de produto é o anúncio, pago pela empresa quando o consumidor final vê o anúncio, clica nele ou compra algo a partir dele. Alguns exemplos são Buscapé, Mercado Livre, Google Search + Google AdWords e UOL Conteúdo + anúncios.

Note que, na descrição de cada um desses tipos, tive a preocupação de deixar claro quem paga a conta. Isso é muito importante, pois todo produto de software tem custos. Por mais acessíveis que sejam esses custos, eles existem; logo, todo produto de software precisa gerar receita de alguma forma, para que ele cumpra sua missão de resolver o problema de um grupo de pessoas.

Mesmo eu preferindo a classificação dos produtos de software entendendo para quem o produto resolve o problema, as outras formas de se classificar também são válidas, e não são excludentes. Por exemplo, o Netflix é um produto de software para o usuário final, mas também é um produto de software online de entretenimento.

PRODUTO OU PLATAFORMA?

Cada vez mais vemos produtos de software que podem ser classificados como plataformas. Exemplos não faltam, desde grandes empresas de tecnologia como:

- Google que, com dois produtos (Search e AdWords), criou uma plataforma conectando pessoas que buscam informações na internet com pessoas que querem anunciar coisas na internet.
- Facebook, que começou como uma plataforma em que amigos se encontravam e trocavam informações, e tornou-se uma plataforma onde anunciantes podem falar com pessoas por meio de anúncios e páginas de empresas.
- LinkedIn, uma plataforma para profissionais, empresas e, mais recentemente, anunciantes.
- Apple, que conecta desenvolvedores de software com usuários de iPhone, iPad e Macs com sua AppStore. Outra plataforma da Apple é o iTunes, conectando produtores de mídia com pessoas interessadas em música, filmes, séries e livros.
- Amazon Kindle, plataforma para que editoras ou autores possam publicar livros para pessoas interessadas nesses conteúdos.

Aliás, algumas dessas empresas têm mais de uma plataforma, como Google, Apple e Amazon.

Além dessas grandes empresas de tecnologia, existem também os exemplos mais recentes:

- Uber, conectando motoristas com pessoas querendo transporte.
- Airbnb, conectando quem tem imóvel para alugar por períodos curtos com quem quer alugar um imóvel nessas condições.
- Bitcoin; quanto mais usuários tiver e quanto mais empresas aceitarem Bitcoin, melhor.
- Existem também exemplos de plataformas que não são necessariamente baseadas em tecnologia como os shoppings, que colocam lojas, restaurantes e cinemas próximos às pessoas que querem comprar, comer e se divertir.

O QUE SÃO PLATAFORMAS?

PLATAFORMA

Sistemas que são mais valorizados quanto mais pessoas usam.

Ou seja, são sistemas fortemente baseados no conceito de efeito de rede (*network effect*). Efeito de rede é o valor que os usuários de um determinado software obtêm quando outros usuários também utilizam o software.

Existem dois tipos de plataforma:

Plataformas de um único lado (*single-side*): são aquelas que, quanto mais usuários tiver, melhor. Usando um exemplo mais antigo, a máquina de FAX. De nada adianta apenas uma pessoa ter FAX; quanto mais pessoas usando, melhor. O mesmo vale para redes sociais (Facebook, Twitter etc.).

Plataformas com múltiplos participantes (*cross-side* ou *multi-side*): são aquelas em que são necessárias dois ou mais grupos distintos de pessoas que fazem uso da plataforma de forma diferente, e que se beneficiam dessa forma diferente de cada grupo usá-las. Esse tipo pode ser classificado em 3 categorias:

a) *Plataforma tecnológica:* onde a plataforma é o sistema operacional e, de um lado, temos usuários e, do outro, temos desenvolvedores, como: Linux, Windows, AppStore e Android.

b) *Plataforma de troca*: plataforma que reúne compradores e vendedores: MercadoLivre, Uber e Airbnb.

c) *Plataforma de conteúdo*: plataforma onde o conteúdo é o foco, e a monetização é feita normalmente por meio de anúncios (Google, Facebook e portais de notícias).

Veja a seguir um exemplo de plataforma tecnológica com 5 lados:

É importante não confundir o conceito de plataforma no contexto de produto, com o conceito de plataforma técnica ou plataforma computacional. Uma plataforma computacional é qualquer ambiente computacional onde um software será executado. Já no contexto de produto,

como definimos anteriormente, chamamos o produto de plataforma quando existem ganhos para os usuários e quanto mais usuários estiverem usando o produto.

Feitas as definições de software, produto de software e plataforma, vamos ver agora o que é gestão de produtos de software.

CAPÍTULO 02

O QUE É GESTÃO DE PRODUTOS DE SOFTWARE?

Já temos a definição de produto de software, vimos vários exemplos e várias formas de classificar esses produtos. Agora, vamos definir a função de gestão de produtos de software:

GESTÃO DE PRODUTOS DE SOFTWARE

Gestão de produtos de software é a função responsável **por todos os aspectos de um produto de software**, durante **todo o ciclo de vida desse produto**, desde a sua concepção até o fim de sua vida.

É a função responsável por fazer a conexão entre a **estratégia da empresa** e os **problemas e necessidades dos clientes** por meio do produto de software. Este deve, ao mesmo tempo, ajudar a empresa a atingir seus objetivos estratégicos, e solucionar os problemas e as necessidades dos clientes.

Essa definição deixa bem claros três pontos bem importantes:

O primeiro é a responsabilidade **por todos os aspectos de um produto de software**. Isso significa que um gestor de produtos de software deverá se preocupar com a experiência do usuário e com a engenharia de seu produto, incluindo sua arquitetura, infraestrutura e operação. Também deverá se preocupar com questões legais e financeiras, suporte ao cliente, e marketing e venda do produto.

Preocupar-se não significa fazer todas essas coisas. Na sua empresa, existem pessoas e áreas dedicadas a cuidar desses temas. Por isso, preocupar-se significa entender esses aspectos, quais são as suas relações com o produto, e como o produto impacta cada uma dessas

áreas. Esse será o tema da *Parte III – Relacionamento com as outras áreas*, onde falarei sobre a relação entre gestão de produtos de software, e sobre as outras áreas da empresa.

O segundo ponto é que a responsabilidade ocorre durante **todo o ciclo de vida do produto**. Como vamos ver na *Parte II – Ciclo de vida de um produto de software*, o ciclo de vida de um produto tem diferentes fases, e cada uma delas requer atenção especial.

O terceiro ponto é a conexão que a gestão de produtos deve fazer entre os **objetivos estratégicos da empresa** e os **problemas e necessidades dos clientes**, que é o que veremos a seguir.

ALINHANDO ESTRATÉGIA DA EMPRESA COM NECESSIDADES DE CLIENTES

O terceiro ponto bem importante da definição de gestão de produtos de software é a responsabilidade por garantir a conexão entre a **estratégia da empresa** e os **problemas e necessidades dos clientes**. É na interseção entre os objetivos do negócio e a solução dos problemas ou necessidade dos clientes que se encontra a gestão de produtos de software, como vemos na figura a seguir:

Essa é a teoria, e tudo parece simples na teoria. Contudo, como todos nós sabemos, na prática a teoria é outra. A gestão de produtos de software está bem melhor representada pela figura:

Nessa imagem, vemos Louis Cyr (http://en.wikipedia.org/wiki/Louis_Cyr), considerado em 1890 o homem mais forte do mundo em sua época, levantando 227 kg com apenas 3 dedos e 1.967 kg em suas costas!

Ela representa melhor a gestão de produtos de software, pois nem sempre é simples conciliar objetivos da empresa e a solução de um problema ou necessidade de clientes. Um exemplo simples é o Facebook que, como qualquer empresa, precisava de receita para pagar seus custos e dar algum retorno aos seus investidores. Esse era o objetivo de negócio do Facebook. Do outro lado, estão os usuários do Facebook, que acessam o sistema sem pagar nada e não estavam interessados em pagar para acessar.

O gestor de produtos do Facebook teve de encontrar uma forma de obter receita, mas sem cobrar dos usuários. A solução foi encontrar um outro tipo de cliente, os anunciantes, dispostos a pagar para exibir anúncios para os usuários do site.

ESSA IMAGEM ESTÁ INCOMPLETA

A primeira imagem está incompleta. Ela fala em **objetivos estratégicos da empresa** e em **problemas e necessidades dos clientes**. Contudo, um gestor de produtos não pode olhar apenas para esses dois itens. Há um terceiro item muito importante que é a **tecnologia disponível**.

O gestor de produtos precisa conhecer a tecnologia disponível para saber se é possível resolver o problema ou a necessidade do cliente, atendendo aos objetivos estratégicos da empresa.

Veja a figura a seguir:

CORE TEAM DE DESENVOLVIMENTO DE PRODUTOS DE SOFTWARE

As três preocupações vistas anteriormente nos dão a dica do que compõe o sucesso de um produto. Um produto de sucesso deve ser:

- **Desejável:** resolve problemas ou necessidades de clientes;
- **Viável:** atende os objetivos estratégicos da empresa;
- **Possível de ser construído:** existe tecnologia disponível para desenvolvê-lo.

Esses três quesitos definem as três funções essenciais para se criar um produto de sucesso: designer, gestão de produto e desenvolvedor. Esse trio é considerado o *core team* (time principal) para o desenvolvimento do produto, e deve estar bem alinhado em todas as fases do seu desenvolvimento.

VIÁVEL – O QUE SUSTENTARÁ O NEGÓCIO?

O gestor de produto tem duas responsabilidades principais: avaliar as oportunidades do produto e definir o produto que será construído. Depois de avaliado e decidido que vale a pena desenvolver o produto, ele inicia a fase de descobrir exatamente como ele deve ser (junto com o core team), incluindo as funcionalidades necessárias, a experiência do usuário e os critérios para o lançamento.

Além disso, está em suas mãos determinar o modelo de negócio que deverá ser seguido, e interagir com praticamente todas as outras áreas da empresa para definir questões jurídicas, contábeis, financeiras, de marketing, de distribuição etc.

DESEJÁVEL – O QUE AS PESSOAS PRECISAM?

É aqui que entra a Experiência do Usuário (UX). Há vários papéis em um time de UX, porém, o que trabalha em maior colaboração com o gestor de produtos é o designer de interação. Ele é responsável por buscar um profundo entendimento dos usuários, descobrindo suas motivações, comportamentos e habilidades; ajudar na definição dos requisitos e, assim, desenhar uma interface que torne a interação do usuário com o produto a mais simples e eficiente possível, ao mesmo tempo em que atenda aos objetivos do negócio.

POSSIBILIDADE – O QUE PODEMOS CONSTRUIR?

O Engenheiro ou Desenvolvedor de Software é o responsável por construir o produto efetivamente. Seu papel é importante na fase de descoberta do produto para dizer ao seu gestor e ao designer de interação o que é possível ser feito, avaliar o custo das diferentes ideias propostas e ajudar a identificar as melhores soluções. É sua responsabilidade definir a tecnologia e arquitetura mais apropriada para desenvolver um produto de qualidade.

QUAL A DIFERENÇA ENTRE GERENCIAR UM PRODUTO OU UMA PLATAFORMA?

Com produtos de software, a preocupação é apenas com um único tipo de cliente. Em uma plataforma, se ela for *single-side*, além de se preocupar em entender um único tipo de cliente, é preciso entender como ele se relaciona entre si.

Já se a plataforma for *multi-sided*, você deve se preocupar com dois ou mais tipos diferentes de usuários, e a relação entre usuários de mesmo tipo e de tipos diferentes. Ou seja, as preocupações, tanto do gestor de produtos como de todas as pessoas que trabalham no desenvolvimento da plataforma, podem ser bem mais complexas do que em um produto com um único tipo de cliente.

Uma estratégia de plataforma deve mudar as prioridades da empresa dona dessa plataforma, uma vez que o cliente não enxerga valor unicamente nas funcionalidades do produto que estão 100% sob controle da empresa. Além das funcionalidades, o cliente busca valor nas interações com terceiros, e é responsabilidade da empresa (dona da plataforma) gerenciar essas relações, para obter os melhores resultados tanto para os participantes da plataforma quanto para si mesma.

DUAS NOVAS PREOCUPAÇÕES NA GESTÃO DE PLATAFORMAS

Quem gerencia plataformas, além de todas as preocupações de gestão de produtos que descrevo aqui no livro, deve também se preocupar com dois novos aspectos:

As funcionalidades dependem da participação dos usuários: em inglês, usa-se o termo *tipping* para essa preocupação, ou seja, como fazer a plataforma ganhar usuários para ela poder ser útil para quem participa dela? Algumas estratégias para fazer isso são:

a) *Primeiro usuário*: conseguir um primeiro usuário que, por si só, já atrai outros usuários. Essa é uma tática muito usada por shoppings quando fecham com uma loja de departamentos, que já atrai compradores suficientes por si só. Depois, é só falar com outras lojas, que certamente terão mais interesse em estar no shopping.

b) *Social*: outra forma de conseguir usuários é usar redes e mecanismos sociais para conquistar mais usuários. Algo como “chame seus amigos do Facebook”.

c) *Usuário líder*: descobrir qual o perfil do usuário que vai ser fortemente atraído pela ideia ao ponto de ser o primeiro a adotar a plataforma. Bitcoin atraiu várias pessoas inicialmente de tecnologia, que se apaixonaram pela ideia de um dinheiro não atrelado a um governo, e são ferrenhos defensores da ideia.

d) *Benefícios como produto*: o próprio produto tem benefícios

suficientes por si só. O Instagram, antes da funcionalidade de compartilhar, era capaz de fazer fotos ficarem legais. O OpenTable, antes da funcionalidade de reservas, é um ótimo ERP para restaurantes.

e) *Reduzir preços*: é uma estratégia válida para atrair usuários, mas vale lembrar que é difícil aumentar o preço depois, principalmente se você reduzir o preço para zero. Claro, você pode subsidiar com anúncios, mas você precisa saber se seus usuários vão aceitar anúncios e se você vai conseguir anunciantes que queiram pagar.

As funcionalidades dependem de que os usuários se comportarem bem: em inglês, o termo usado para isso é *coring*, ou seja, como garantir que os usuários não vão tirar proveito um do outro, garantindo sempre que todos os participantes tenham benefícios? Algumas estratégias para cuidar do *coring*:

a) *Promova confiança*: sites de leilão e de pagamento online costumam fazer isso, segurando o dinheiro do comprador até que ele diga que recebeu o produto que lhe foi vendido.

b) *Ofereça informação de qualidade*: normalmente, aqueles *ratings* feitos pelos usuários. O grande risco aqui é gerenciar os ratings falsos; positivos feitos pela própria pessoa ou empresa que está sendo analisada, e negativos pelos concorrentes.

c) *Restrinja o uso*: torne a associação e o uso mais restrito, o que trará sim menos usuários, mas mais usuários de qualidade. É o que fez um site chamado eHarmony, de procura de namorado(a). Ele cobra uma mensalidade razoavelmente cara (US\$ 50.00) e tem um questio-

nário bem extenso para ser preenchido. Além disso, mesmo que seu algoritmo de *matchmaking* encontre várias opções, ele só apresentará um número limitado para facilitar o processo de escolha.

CONCLUINDO

É importante entender se você está trabalhando em um produto ou em uma plataforma, pois existem algumas diferenças em gerenciar cada um deles.

Uma plataforma precisa de uma estratégia para atrair os primeiros usuários, isso é tão ou mais importante do que as funcionalidades. Como gestores de um produto de software, temos a tendência em ficar animados com funcionalidades técnicas, entretanto, nas plataformas, o foco é ainda maior nos usuários, em suas relações e em como atrair os primeiros usuários. Além disso, gerir uma plataforma requer controle e governança dos relacionamentos dentro da plataforma, para garantir que todos os participantes estejam se beneficiando com ela. Uma vez que já entendemos o que é a gestão de produtos de software, e quais as diferenças entre gerir um software e gerir uma plataforma, precisamos agora entender o que é um gestor de produtos de software. Esse é o tema do próximo capítulo!

CAPÍTULO 03

GESTOR DE PRODUTOS OU PRODUCT OWNER?

Gestor de produtos ou *product owner*? Que termo usar? São papéis diferentes? São complementares? Há sobreposição? É melhor ter duas pessoas distintas, uma para cada papel? Ou é melhor combinar os dois papéis em uma única pessoa?

DEFINIÇÕES

Antes de mais nada, vamos ver mais um pouco de definições.

PRODUCT OWNER, SEGUNDO O GUIA DO SCRUM

O **product owner** é a pessoa responsável por maximizar o valor do produto, o trabalho do time de desenvolvimento, e a gestão do *backlog* do produto. Ele é uma pessoa, e não um comitê.

O **product owner** pode representar o desejo de um comitê no backlog do produto, mas aqueles que quiserem uma alteração nas prioridades dos itens de backlog devem convencer o **product owner**.

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

PRODUCT OWNER, SEGUNDO A WIKIPÉDIA

O **product owner** representa os *stakeholders* e é a voz do cliente. Ele é responsável por garantir que o time entregue valor para o negócio.

O **product owner** escreve (ou faz o time escrever) itens centrados no cliente, e classifica e prioriza esses itens, adicionando-os ao backlog do produto.

Fonte: [http://en.wikipedia.org/wiki/Scrum_\(software_development\)#Product_Owner](http://en.wikipedia.org/wiki/Scrum_(software_development)#Product_Owner)

Pelas definições mostradas, fica claro o foco do product owner em:

- Gerenciar as prioridades do backlog, com base em *inputs* dos *stakeholders* e clientes; e
- Maximizar as entregas do time de desenvolvimento.

Por outro lado, no *Capítulo 02 – O que é gestão de produtos de software?*, defini gestão de produtos como sendo:

GESTÃO DE PRODUTOS DE SOFTWARE

Gestão de produtos de software é a função responsável **por todos os aspectos de um produto de software**, durante **todo o ciclo de vida desse produto**, desde a sua concepção até o fim de sua vida.

É a função responsável por fazer a conexão entre a **estratégia da empresa** e os **problemas e necessidades dos clientes** por meio do produto de software. Este deve, ao mesmo tempo, ajudar a empresa a atingir seus objetivos estratégicos, e solucionar os problemas e as necessidades dos clientes.

Ou seja, o **gestor de produtos** precisa conhecer bem tanto quem é o dono do software e quais os objetivos que este deseja alcançar com ele, como quem vai usar esse software e quais os objetivos que esse usuário pretende alcançar ao fazer isso. Somente conhecendo os objetivos tanto do dono do software como de seu usuário é que o gestor de produtos poderá definir como será esse software.

Dá para perceber que, por um lado, as definições de **product owner** se focam bastante no processo, ou seja, em priorização de backlog e em maximização da produção do time de desenvolvimento; enquanto a definição de **gestão de produto** é totalmente focada no resultado, ou seja, no software que está sendo feito e nos seus objetivos, tanto para o seu dono quanto para seu usuário.

As definições de **product owner** focam no processo, pois todas as metodologias ágeis têm seu foco no processo de desenvolvimento de software. O próprio Manifesto Ágil (<http://agilemanifesto.org/>) diz que “Estamos descobrindo maneiras melhores de desenvolver software”. Note que a preocupação é em descobrir maneiras melhores de desenvolvê-lo, e não em descobrir maneiras de desenvolver softwares melhores. É uma diferença sutil do ponto de vista gramatical, mas enorme no significado.

Enquanto “descobrir maneiras melhores de desenvolver software” se foca no processo de desenvolver software, quando falamos em “descobrir maneiras de desenvolver softwares melhores”, passamos imediatamente a nos focar no resultado do desenvolvimento de software: o software! Por isso, minha definição de **gestão de produtos** tem foco no software e nos objetivos de seu dono e de seus usuários, enquanto que as definições de **product owner** focam em como melhorar o processo de desenvolvimento de software.

ENTÃO SÃO PAPÉIS DIFERENTES?

Resposta curta: NÃO. Apesar de terem focos distintos, pode-se dizer que são dois lados da mesma moeda. Não dá para ter um sem o outro. Ou seja, não dá para focarmos em melhorar o processo de desenvolvimento de software sem pensarmos em melhorar o software que está sendo produzido; da mesma forma que não é possível pensar em melhorá-lo sem investir no melhoramento do processo de desenvolvimento de software.

Conversei com pessoas de algumas empresas sobre como eles desenharam sua organização de desenvolvimento de software, e vi com alguma frequência o uso desses papéis de forma separada, ou seja, existem **product owners**, que são parte do time de desenvolvimento de software, e são responsáveis por gerenciar o backlog e detalhar os itens desse backlog; e existem os **gestores de produto**, que não fazem parte do time de desenvolvimento, são responsáveis pela visão de negócio do software e passam para o time grandes épicos que deverão ser detalhados por um **product owner**.

Na Locaweb, optamos por usar os termos **gestor de produto** e **product owner** como sinônimos, pois, como dito anteriormente, para nós são dois lados da mesma moeda. Não dá para priorizar o backlog e maximizar as entregas do time de desenvolvimento se você não tiver profundo conhecimento dos objetivos do dono e dos usuários desse software. Assim como não dá para produzir o melhor software que atenda os objetivos tanto do dono como de seus usuários se você não priorizar corretamente o backlog e não maximizar as entregas do time de desenvolvimento.

Um é o “o quê”, e o outro é o “como” do desenvolvimento de software. Não existe um sem o outro.

Depois de ler que gestor de produtos e product owner são dois lados da mesma moeda, caso você esteja em uma empresa onde esses papéis estão separados em duas pessoas distintas, você pode ficar com a dúvida sobre o que fazer nessa situação.

O QUE FAZER SE NA SUA EMPRESA TIVER GESTORES DE PRODUTO E PRODUCT OWNERS?

Conheço algumas empresas que têm essa separação de papéis em pessoas distintas e que, ao lerem que esses papéis devem se executados pela mesma pessoa, podem pensar que estão com sobra de gente. :-O

Não estão! Muito provavelmente algum outro papel está faltando no seu time de desenvolvimento de produtos de software. Minha recomendação para esses casos:

- **Não seja radical:** não saia demitindo pessoas achando que há sobreposição de papéis. É preciso um olhar

mais cuidadosamente, pois outras funções podem estar faltando na sua organização.

- **Marketing de produtos:** provavelmente está faltando uma pessoa cuidando do marketing de produto, que tem objetivos complementares, mas bem diferentes do gestor de produto. No *Capítulo 28 – Qual a diferença entre gestão de marketing de produtos e gestão de produtos?*, escreverei sobre a diferença entre gestões de produtos e gestor de marketing de produtos.

- **Analise o que está sendo feito hoje:** é provável que o seu gestor de produtos, às vezes chamado de gestor de negócios, esteja fazendo mais coisas de um gestor de marketing de produto. Nesse caso, pode ser interessante que essa pessoa passe a ser de marketing de produtos de fato e deixe as atividades de gestor de produto, que está fazendo para o product owner eventualmente. Este poderá, assim, assumir a gestão de produtos.

- **Use um próximo produto para experimentar com a nova divisão de papéis:** outra forma de experimentar com essa nova divisão de papéis e responsabilidades é usá-la somente num novo produto. Quando você estiver pensando em fazer um novo produto, experimente essa nova divisão de papéis e veja como esse produto evolui. Se der certo, você pode estender para outros produtos existentes.

GESTOR OU GERENTE DE PRODUTOS?

Em inglês, a função é conhecida como *product manager* e, olhando no Wikipédia, vemos a seguinte definição para a palavra *manager*:

MANAGER

Inglês para “gerente” ou “gestor”, em assuntos relacionados à economia e administração de empresas, e “treinador”, para esportivos.

Fonte: <http://pt.wikipedia.org/wiki/Manager>

Então, usar gestor de produtos ou gerente de produtos é indiferente, pois ambas as palavras servem de tradução para *manager*. Aqui no livro, usarei sempre gestor por uma razão simples. Na cultura corporativa brasileira, a palavra gerente costuma denotar alguém que “chefia” pessoas, só que o gerente de produtos não é chefe de ninguém.

Essa função não deve gerenciar pessoas, mas gerenciar coisas. Da mesma forma que um gerente de contas e um gerente de projetos não deve gerenciar pessoas, mas sim contas e projetos, respectivamente. Por esse motivo, prefiro usar a palavra gestor, para tirar qualquer conotação de que essa função deva gerenciar pessoas.

Agora que já entendemos um pouco mais o que é um gestor de produtos de software, vamos conhecer quais são as principais características dessa função.

CAPÍTULO 04

PRINCIPAIS CARACTERÍSTICAS DE UM GESTOR DE PRODUTOS

O que uma pessoa precisa ter para ser um bom gestor de produtos? Existem algumas características bem importantes e falarei sobre elas aqui, mas a mais importante de todas é, sem dúvida alguma, a **empatia**.

Fonte: Facebook Start Empathy
(<https://www.facebook.com/photo.php?fbid=191811397619249&set=a.105965526203837.7401.100867456713644&type=3&theater>)

Empatia é a capacidade que uma pessoa tem de se colocar no lugar de outra para compreender os seus anseios, motivações, necessidade e problemas. Essa característica é importante para entender os clientes e usuários do produto, saber como estes se relacionam com ele, e que problema esperam resolver ou que necessidade querem ver atendidas. Isso ajudará o gestor de produtos a entender melhor seu usuário para poder, junto com o UX e a engenharia, desenhar o melhor produto.

Contudo, a empatia não deve ser usada apenas com o cliente ou usuário. O gestor de produtos deve usá-la também no seu relacionamento com todas as áreas da empresa, e deve entender o impacto que seu produto tem no trabalho delas. Será que aumentaram os problemas jurídicos devido a alguma funcionalidade nova no produto? Qual o impacto para a equipe de vendas, suporte, operações, financeiro e marketing? Até mesmo em relação ao time do produto, engenheiros e analistas de UX, como o produto interfere no trabalho desses profissionais?

A segunda característica mais importante é a **comunicação**.

Para poder ter empatia, é necessário que o gestor de produtos se comunique com as pessoas nos mais diferentes cenários: em conversas um a um e em pequenos grupos, ou fazendo apresentações para grupos pequenos e grandes de pessoas, estes que podem ser internos (dentro da empresa) ou externos (em conferências, grupos de usuários etc.).

Deve também ser bom de comunicação escrita (e-mail, blog, documentação, chat, redes sociais etc.), e ser capaz de discernir sobre qual a forma de comunicação mais apropriada para cada momento, público e meio de comunicação; e de se comunicar de forma a ser entendido pelos diferentes públicos: técnico e não técnico. Como se isso tudo não bastasse, ele também precisa ser capaz de se comunicar passando segurança e confiança no que está comunicando; afinal, o gestor de produto é o porta-voz do produto.

Porém, não é só de falar que vive o gestor de produtos. Comunicação é uma via de mão dupla, ou seja, o gestor de produto tem de ser muito

bom em saber ouvir e compreender o que os outros estão dizendo, e entender os anseios e as necessidades deles; o que tem a ver com a primeira característica, a empatia.

A terceira característica mais importante é a **gestão do tempo**.

O dia a dia de um gestor de produtos pode se encher rapidamente de tarefas, e ele precisa ser capaz de perceber e diferenciar o que é urgente do que é importante, para garantir que terá sempre tempo para conhecer mais sobre o cliente e o usuário de seu produto. É muito fácil um gestor de produtos ver sua agenda repleta de reuniões, com pessoas de diferentes áreas para tratar dos mais diversos assuntos: backlog do produto, atendimento, comunicação de marketing, problemas operacionais, revisão de previsão de crescimento, questões jurídicas, cobrança, régua de comunicação etc.

Isso acontece pois o gestor de produto deve cuidar de seu produto por inteiro. Para o usuário, não existe engenharia, operações, financeiro, jurídico, atendimento e cobrança. Ele enxerga tudo isso como parte do produto que você cuida; e você tem sim de se importar com tudo isso. Contudo, importar-se não significa que você deva ir a todas essas reuniões. Se você fizer isso, poderá tirar o foco do que é mais importante para o seu produto. Você, como gestor de produtos, precisa focar seu tempo em:

- 1) Com quantos clientes e usuários você falou essa semana?
- 2) Você tem uma estratégia e uma visão de longo prazo para o seu produto?
- 3) Como está e quais as últimas mudanças no cenário

competitivo de seu produto?

4) Que *insights* você teve sobre o seu produto essa semana?

5) Você sabe qual a motivação e a métrica de cada item de seu *roadmap*?

6) Que novas tecnologias você vê aparecendo que podem influenciar, ou mesmo competir, com o seu produto?

7) Que novas habilidades você está procurando aprender?

As reuniões que mencionei anteriormente são importantes, e aconselho você participar delas quando puder. Apesar disso, você terá muito pouco a contribuir para essas reuniões se você não se focar nos 7 itens que acabei de listar. Procure bloquear um tempo em sua agenda para focar nisso, e você verá como sua participação nas reuniões será muito mais útil e produtiva.

Além dessas três características (empatia, comunicação e gestão do tempo), existem mais quatro que vão ajudar o gestor de produtos a fazer seu trabalho melhor:

- **Novas tecnologias:** o gestor de produtos deve estar antenado sobre as novas tecnologias para saber como ela pode impactar o seu produto. Acesso via smartphone, como isso impacta o produto? O usuário quer acessar via smartphone? Para fazer o quê? Redes sociais, como o produto pode tirar proveito delas? Bancos de dados não relacionais, quais os benefícios e as deficiências? Go, a nova linguagem de programação do Google, no que ela é melhor do que a linguagem usada no produto? E no que ela é pior? *Smartwatches*, *smartglasses*, como isso impacta o produto? Como o usuário espera interagir com o produto nessas novas interfaces?

- **Business skills:** o gestor de produtos deve entender como cada área da empresa funciona e como o produto afeta essas áreas. Como o jurídico funciona? Como o produto impacta o departamento jurídico? E como o departamento jurídico impacta o produto? Essas perguntas podem ser repetidas para todas as áreas da empresa: suporte, operações, financeiro, RH, marketing, vendas, engenharia e UX.

- **Curiosidade aguçada:** o gestor de produto precisa ter a capacidade de aprender rápido para poder ter *insights* e fazer julgamentos sobre o produto. Deve ser capaz de aprender tanto o lado soft do produto (qual é a motivação de negócio, qual problema do cliente o produto resolve etc.) como o lado mais técnico (qual tecnologia usamos, qual o impacto dessa tecnologia, que métricas podemos obter etc.).

- **Tema do produto:** por fim, mas não menos importante, o gestor de produtos precisa conhecer sobre o tema do produto. Se for um produto médico, o gestor de produto deverá entender um pouco sobre medicina. Se for um produto financeiro, deverá conhecer um pouco de finanças. Por exemplo, na Locaweb, temos produtos mais técnicos (como o Cloud Server) e menos técnicos (como a Loja Virtual). A necessidade de conhecimento técnico é bem diferente nesses dois produtos. O gestor de produto do Cloud Server deverá conhecer bem a fundo questões técnicas, enquanto o gestor de produto da Loja Virtual não precisa ter tanto conhecimento técnico, mas deverá saber bastante sobre questões de venda online.

Dá para perceber que essa lista é um conjunto de características que nem todas as pessoas têm. É comum casos de pessoas de outras áreas que resolvem experimentar a carreira de gestor de produtos, mas

que, após algum tempo, percebem que não têm todas as características necessárias.

Se você é um gestor de produtos, ou deseja ser um, faça uma autoanálise para ver como você está em cada uma das características e, se alguma estiver aquém do desejado, foque em desenvolvê-la. Se você é responsável por identificar e contratar gestores de produto, use essa lista como guia para saber se o candidato tem as características necessárias para ter sucesso como gestor de produto.

CAPÍTULO 05

DICAS DE LIDERANÇA PARA GESTORES DE PRODUTO

Um gestor de produtos tem a difícil tarefa de liderar a evolução do produto, sem ser “chefe” de ninguém. Isto é, ele deve convencer todas as pessoas que trabalham com seu produto de que o caminho que ele definiu para o produto é o mais adequado.

Em vários textos sobre gestão de produtos, encontramos que o gestor é o CEO do produto. Não gosto muito dessa analogia, pois um CEO, em última instância, tem ao seu dispor a liderança direta de todas as pessoas da empresa e pode, apesar de na maioria das vezes não ser a forma mais adequada, utilizar a hierarquia para atingir seus objetivos.

Por outro lado, um gestor de produtos trabalha em uma relação matricial, ou seja, não tem liderança direta de nenhuma das pessoas envolvidas com o produto. Aliás, esse é um excelente exercício de liderança e uma qualidade extremamente importante para um gestor: liderar sem ter a “chefia” organizacional.

Mesmo que ele trabalhe em uma organização totalmente horizontal, sem nenhuma hierarquia, ele deverá exercer alguma liderança para convencer as pessoas a evoluir o produto na direção que visualizou.

Por isso, aqui vão duas dicas de liderança que um gestor de produto, ou qualquer líder, deve lembrar para liderar de forma eficiente.

SETAR O CONTEXTO

A primeira dica é uma de caráter estratégico. O gestor de produtos tem a obrigação de:

Entender, comunicar e explicar o contexto no qual se insere o que está sendo trabalhado.

Ajudar o time a entender onde o produto se insere na estratégia da empresa e no mercado.

Saber como os clientes utilizam o produto e o que estes esperam dele, e qual o objetivo do produto para o cliente e para a empresa.

Compreender como uma determinada funcionalidade (ou seja, um novo desenvolvimento que o gestor de produto pede para o time) se insere no produto e na estratégia desse produto.

Essa dica parece simples e óbvia, mas não é incomum encontrar pessoas que trabalham sem saber exatamente por que estão fazendo seu trabalho. Ajudar as pessoas a verem o impacto do seu trabalho faz com que elas entendam por que ele é necessário.

Experimente levar engenheiros de software em sua próxima conversa com cliente. Melhor ainda, leve-os a um teste de usabilidade, para que eles possam ver seus usuários utilizando o software que eles desenvolveram. Isso os ajudará a entender por que o software que eles desenvolveram existe, qual problema ele resolve e para quem ele resolve esse problema.

Setar o contexto ajuda muito no engajamento das pessoas que estão envolvidas com o produto. Elas vão entender a sua importância tanto para a empresa – que é dona do produto – quanto para os seus usuários. Esse engajamento é importante não só no *core team* do produto como também com todas as pessoas envolvidas com ele, como o pessoal de vendas, marketing, jurídico e suporte ao cliente. Todos vão se beneficiar se o gestor de produto sempre procurar setar seu contexto e onde o trabalho de cada área se insere no seu sucesso.

Na Locaweb, temos alguns sistemas antigos que, como todo legado, são bem difíceis de mexer: pouca cobertura de testes, linguagens de programação antigas, código feito com práticas de 10 anos atrás. Sempre que é preciso mexer no código legado é um sofrimento. Estamos já trabalhando há alguns anos para minimizar a quantidade de código legado, e essa quantidade tem de fato diminuído. Contudo, o negócio não para e, às vezes, a única saída é mexer nele. Sempre que aparece uma demanda dessas, os desenvolvedores perguntam se não dá para esperar o novo sistema que o substituirá.

No início de 2015, apareceu uma demanda desse tipo, em que era necessário mexer nos limites e preços de nossos planos de hospedagem para acompanhar as mudanças do mercado, que estava mais competitivo. Claro que, inicialmente, houve resistência dos desenvolvedores em mexer no legado, mas quando mostramos todo o racional por trás das mudanças, eles arregaçaram as mangas e “sujaram” as mãos no código antigo.

Assim que as mudanças foram implementadas, frequentemente contávamos para as pessoas que trabalharam nesse projeto sobre os seus resultados positivos. Essa compreensão de por que algo é pedido e deve ser feito é fundamental tanto para a motivação de quem for trabalhar na demanda quanto para a qualidade do que será entregue.

REMOVER IMPEDIMENTOS

Remover os impedimentos é fundamental para que as pessoas do time possam desenvolver o produto. Isso é importante para poder termos aquela gostosa sensação de progresso, de que estamos fazendo algo, construindo algo. O artigo *What Really Motivates Workers* (O que motiva os trabalhadores) – disponível em <http://hbr.org/2010/01/the-hbr-list-breakthrough-ideas-for-2010> –, da Harvard Business Review, mostra uma informação muito importante sobre satisfação no trabalho. Fizeram um estudo para encontrar o que acontece em um excelente dia de trabalho. A resposta estava em uma palavra: **progresso**.

O conselho no final do artigo resume bem a segunda dica:

Evitar escrupulosamente o que estiver impedindo o progresso, evitar alterar objetivos autocraticamente, evitar indecisões, ou segurar recursos. Eventos negativos geralmente têm um efeito maior sobre as emoções, percepções e motivações das pessoas do que os positivos, e nada é mais desmotivador do que um revés – o tipo mais proeminente de evento nos piores dias dos trabalhadores do conhecimento.

CONCLUINDO

Bom, então estão aí duas dicas de liderança para gestores de produtos e para qualquer pessoa que se veja liderando algum projeto:

- Setar o contexto;
- Remover impedimentos.

Tenho as usado ao longo de toda minha vida profissional e tem sido muito útil!

CAPÍTULO 06

CULTURA ORGANIZACIONAL

Cultura organizacional é um tema muito importante para gestores de produto; por isso, quero abordar um pouco dele aqui. De uma certa forma, esse assunto complementa o capítulo anterior, sobre dicas de liderança para gestores de produto.

Cultura organizacional é uma característica de toda empresa. Até mesmo dentro de uma empresa existem subculturas, ou seja, cada área ou time dentro de uma empresa pode ter uma cultura própria. Por exemplo, a cultura de um time comercial tem sempre algumas diferenças da cultura do time de engenharia de software.

Não existe cultura certa ou cultura errada. Empresas diferentes têm culturas diferentes e, mesmo assim, podem ter sucesso apesar dessas diferenças. A charge a seguir ilustra a diferença de cultura entre Amazon, Apple, Facebook, Google, Microsoft e Oracle. Mesmo com essas diferenças culturais, todas são empresas de sucesso.

Por Manu Cornet. Fonte: <http://www.bonkersworld.net/organizational-charts/>.

Mas o que é cultura organizacional? Edgar Schein, professor da escola de administração de empresas do MIT, foi uma das primeiras pessoas a falar sobre cultura organizacional nos anos 1970. Segundo ele, cada empresa tinha sua própria personalidade, e sua própria forma de agir e reagir às situações; forma esta que é passada de funcionário para funcionário desde os fundadores da empresa. A definição de Schein para Cultura Organizacional é:

CULTURA ORGANIZACIONAL

Cultura é um conjunto de premissas que foram aprendidas e compartilhadas por um grupo de pessoas enquanto resolviam problemas de adaptação externa e de integração interna. Esse conjunto de premissas funciona bem o suficiente para ser considerado válido e, consequentemente, ser ensinado aos novos membros do grupo como a forma correta de perceber, pensar e sentir em relação a esses problemas.

Fonte: SCHEIN, Edgar. Organizational Culture and Leadership. Jossey-Bass, 2010.

A cultura vem dos fundadores da empresa. Os fundadores têm sua própria cultura e é natural que imprimam essa cultura na organização que estão criando. Em função disso, é muito comum se pensar que ela é algo que emerge em uma organização.

O fundador traz sua cultura e, ao contratar novas pessoas, busca sempre pessoas com cultura similar à sua. Isso acaba criando uma cultura comum muito próxima daquela trazida pelo fundador da empresa. Esse conceito de cultura emergente dá a impressão de que não se pode alterá-la deliberadamente, e que ela se desenvolverá de forma orgânica.

Schein alerta que isso é um engano. Culturas podem e devem ser planejadas. Por esse motivo, vou compartilhar três valores de cultura organizacional que tenho visto em algumas empresas e que, a meu ver, são essenciais na criação de produtos de software de sucesso.

NÃO DESPERDIÇAR TEMPO BUSCANDO CULPADOS

Quando erros acontecem, algumas pessoas têm uma tendência natural de ter como sua primeira reação procurar quem é o culpado, especialmente em atividades de grupo. Como se ter alguém para culpar fizesse o erro, de alguma forma, menos prejudicial. Isso é um grande desperdício de tempo e energia. Deixe-me explicar o porquê.

Blame game, por Ron Tandberg.

Ocorreu um erro. Erros acontecem. Este é um fato da vida. Não importa o que você está fazendo – desenvolvendo software, publicando código em produção, operando um paciente, cozinhando o jantar, construindo uma casa, tocando guitarra, jogando futebol, etc. –, há boas chances de que erros venham a acontecer.

Quando você gasta tempo tentando descobrir quem foi o responsável pelo erro, você vai adiar suas tarefas mais importantes em relação a ele:

- Compreender o que aconteceu;
- Descobrir como corrigir;
- Encontrar formas de evitar que isso aconteça novamente.

Quando você gasta tempo tentando descobrir quem foi o responsável pelo erro, as pessoas podem, naturalmente, tentar esconder o erro por temer as consequências. Será que vou ser demitido? Será que vou ser excluído do grupo? Será que vou ser punido? Será que as pessoas vão zombar de mim?

Quando as pessoas tentam esconder quem foi o responsável, você vai acabar adiando as tarefas mais importantes que listei para tratar o erro, pois será mais difícil entender o que aconteceu. As pessoas não vão dizer toda a verdade sobre o erro e as circunstâncias em que ele aconteceu.

Se no processo de entender o que aconteceu, você descobrir que alguém foi responsável pelo erro, lide com ele em particular. O mais provável é que ele tenha o causado sem intenção de fazer mal. Por isso você precisa ajudá-lo a melhorar para que ele não cometa mais esse tipo de erro.

Por outro lado, você tem a responsabilidade de criar um ambiente onde é seguro falar sobre erros, para que estes sejam detectados o mais rápido possível. Contudo, se você descobrir que o erro foi feito com intenção de prejudicar a companhia, algum time ou alguma pes-

soa, nesse caso você deve ser direto na repreensão, dizendo que o comportamento é inaceitável e, na reincidência, convidar essa pessoa a sair do grupo.

GUERRA

É comum ouvir comparações entre o mundo dos negócios e situações de guerra, de combate e de luta. Aliás, a própria palavra “estratégia”, tão comum nas empresas de hoje em dia, vem do mundo militar. Vem da palavra grega *strategos*, junção das palavras *stratos* (exército) e *agos* (líder). Ou seja, *strategos* significa originalmente o líder do exército, o general, aquele que define como a tropa deverá agir frente às situações.

Um dos livros que constantemente aparece na lista de mais recomendados para administração é *A Arte da Guerra*, um tratado militar escrito durante o século IV a.C. por Sun Tzu, general chinês.

Quem me conhece sabe que sou uma pessoa pacífica. Odeio brigas. Aliás, se acidentalmente entro em uma, estou disposto até a pagar para sair. Por isso, sempre que vejo essas comparações do mundo de negócios com guerra, combate, luta ou competição, me sinto profundamente incomodado. Veja a seguir algumas imagens para relembrar o que costuma acontecer durante uma guerra:

Buscando na Wikipédia, encontramos a seguinte definição para empresa:

EMPRESA

Empresa é uma instituição jurídica despersonalizada, caracterizada pela atividade econômica organizada, ou unitariamente estruturada, destinada à produção ou circulação de bens ou de serviços para o mercado ou à intermediação deles no circuito econômico.

Fonte: <http://pt.wikipedia.org/wiki/Empresa>.

Essa definição deixa claro que a empresa existe para produzir e/ou servir. Como pode algo que tem por objetivo produzir e/ou servir, ter por analogia algo que tem como objetivo a destruição? A maneira correta de olhar uma empresa e seus objetivos é pensando em construção, em relação ganha-ganha, ou seja, onde o cliente, os funcionários, os donos da empresa e a sociedade onde a empresa está inserida saem ganhando. Sempre que direcionamos energia para derrotar o “oponente” (cliente, competidor, funcionário etc.), estaremos desperdiçando energia que poderia ser usada em algo construtivo.

AR, COMIDA E LUCRO

Não raro vemos acionistas, investidores e funcionários de uma empresa 100% focados nos resultados financeiros dela. Quando em período de crise, esse foco consegue ir além dos 100%... :-/

Certa vez ouvi uma frase, popularizada por Dick Costolo, CEO do Twitter, que comparava receita a oxigênio:

Receita é como oxigênio, é necessário, é vital para a saúde e para o sucesso de uma empresa, mas não é propósito da vida. Você não acorda de manhã e a primeira coisa que pensa é “como eu posso conseguir mais ar?”

Gosto bastante dessa comparação. Toda empresa deve ter um propósito, e esse propósito não pode ser derrotar os inimigos (como explicado anteriormente), nem conseguir a maior quantidade de dinheiro possível. O resultado financeiro deve sempre ser usado como uma das métricas que indicam que a empresa está tendo sucesso, que está cumprindo com o seu propósito. Contudo, mesmo como métrica, ela não deve ser olhada de forma isolada, pois existe a boa e a má receita.

A má receita é toda receita obtida às custas do detrimento do relacionamento com o cliente. Por exemplo, imagine uma empresa que presta um serviço com pagamento mensal; quando um cliente quer cancelar, ela dificulta sua saída para manter aquela fonte de receita por mais alguns meses. Isso é uma má receita. Quando uma companhia aérea cobra uma taxa para adiantar o horário de um voo, mesmo sabendo que o avião tem espaço de sobra; isso é má receita. As taxas de roaming internacionais exorbitantes

também são um bom exemplo disso, como locadoras de veículos que cobram a taxa de gasolina quando você devolve o carro sem estar com o tanque cheio, com preço de gasolina bem mais caro que o do mercado. Se uma empresa vende algo com o preço acima do adequado, aproveitando-se do fato de que você precisa daquele item como, por exemplo, o custo absurdo da garrafa de água em um hotel, isso também é uma má receita.

Ou seja, apesar de a comparação de receita e lucro com oxigênio ser boa, ela é incompleta, pois não capta o efeito da má receita. Raramente você pensa no oxigênio, a não ser que esteja com falta de ar. Eu não acho que a receita só deva ser lembrada quando está faltando. Receita é o que mantém a empresa viva, capaz de executar seu propósito. Se for uma boa receita, a empresa vai continuar cumprindo seu propósito por muitos anos. Já se for uma má, ela terá dificuldades no médio prazo.

Por esse motivo, prefiro comparar receita e lucro com comida. Da mesma forma que as pessoas saudáveis não pensam em oxigênio o dia inteiro, pessoas saudáveis também não pensam em comida o dia todo. Contudo, diferentemente do oxigênio, quando falamos de comida, existe a comida boa, saudável, que faz bem para o seu corpo; e existe a comida má, que vai prejudicar seu organismo, com possibilidade de fazer você ficar doente. É muito importante que a pessoa saiba o que é boa comida e má comida, e que pense sobre como obter a boa e como evitar a má.

Pegando a frase de cima e aprimorando-a trocando o oxigênio pela comida, teremos:

Receita é como comida, é necessária, é vital para a saúde e para o sucesso de uma empresa, mas não é propósito da vida. Você não acorda de manhã e a primeira coisa que pensa é “como eu posso conseguir mais comida?”. Contudo, tanto uma empresa quanto uma pessoa devem estar sempre atentas à qualidade da comida que está ingerindo, para ter certeza de que ela não vai causar nenhum mal à saúde.

CONCLUINDO

Vimos neste capítulo o quão importante é a cultura organizacional para o sucesso do seu produto de software, e que cultura não é um tema para deixar acontecer, ele pode e deve ser planejado. Compartilho três valores culturais que, a meu ver, são essenciais para a criação de um software de sucesso:

- Não desperdiçar tempo buscando culpados;
- Não comparar fazer negócios com guerra, combate ou luta;
- Pensar em receita como comida, ou seja, algo necessário para viver, mas não é a razão do viver.

PARTE II

**CICLO DE VIDA DE
UM PRODUTO DE
SOFTWARE**

Já falei um pouco sobre o que é gestão e gestor de produtos de software, suas principais características, e até mesmo algumas dicas de liderança e de cultura organização para ajudar gestores a liderar sem serem “chefes”.

Agora, vamos falar sobre o ciclo de vida de um produto de software e suas diferentes fases: inovação, crescimento, maturidade e fim de vida.

Inovação: de todas as fases do ciclo de vida de um produto de software, acredito que a de inovação é a que tem a maior quantidade de dúvidas. É também a fase que tem a maior quantidade de livros sobre o tema, que são os livros sobre inovação e startup. Os temas que abordarei nas próximas páginas serão: o que é inovação; como encontrar um problema a ser resolvido; como descobrir se esse problema é, de fato, uma oportunidade a ser perseguida; e como obter retorno com seu produto de software.

Crescimento: nessa fase, quando o produto foi desenvolvido e lançado, devemos nos preocupar em como gerenciar o produto durante seu crescimento, ou seja, como gerenciar o feedback? O que é um roadmap? Como priorizar as demandas? O que fazer com pedidos especiais? Como dizer não? Que métricas acompanhar?

Maturidade: após o crescimento, vem a maturidade. Nessa parte, vamos entender quando ela acontece e o que fazer se o produto chegar a essa fase.

Fim de vida: depois da maturidade, ou quando o produto é desenvolvido mas não dá certo, chega a fase conhecida como fim de vida de um produto de software. Vamos ver como detectar e o que fazer nela.

Vamos começar?

CAPÍTULO 07

COMO É O CICLO DE
VIDA DE UM PRODUTO
DE SOFTWARE?

Antes de vermos como é o ciclo de vida, precisamos entender a curva de adoção de tecnologia. Esse conceito apareceu pela primeira vez em um livro de 1962, chamado *Diffusion of Innovations*, escrito em 2013 por Everett M. Rogers, sociólogo e professor da Universidade Estadual de Iowa. Nesse livro, Rogers explica que as inovações tecnológicas são adotadas conforme a curva mostrada na figura a seguir:

No começo, os inovadores são os primeiros a se interessar por novos produtos e inovações. Topam até produtos incompletos e com defeitos, pelo prazer de serem os primeiros a utilizar esse novo produto. Em seguida, estão os *early adopters*, também conhecidos como visionários ou entusiastas, que aceitam os riscos de testar um novo produto, mas não pelo prazer de ser o primeiro, mas sim porque enxergam o potencial dele. Normalmente, têm influência nas organizações e comunidades de que fazem parte.

A *early majority*, também chamados de pragmáticos, compram novos produtos somente depois de ter referências. *Late majority* são os conservadores, ou seja, aqueles que compram somente depois que o pre-

ço caiu consideravelmente. Por fim, há os *laggards*, que só compram um novo produto se essa for a única opção disponível.

Fazendo a integral dessa curva (quem se lembra das aulas de cálculo?), obteremos a famosa curva em S de adoção de tecnologia.

Essa curva em S pode ser quebrada em 3 fases: o início mais lento, que é a fase de inovação; em seguida vem a fase de crescimento, quando *early majority* e *late majority* adotam o produto; e, por fim, a fase de maturidade, na qual o produto já conquistou praticamente todo o mercado.

Essa curva realmente acontece na vida real. Na sequência, veja alguns exemplos.

CURVA S NA VIDA REAL

Nem sempre é tão perfeita quanto a curva teórica, mas se aproxima bastante. A curva da TV é a que mais se aproxima, e explica por que toda hora os fabricantes de televisões estão inventando algo novo para nos fazer comprar uma nova. Primeiro, eram TVs em preto e branco; depois, as coloridas. Aí vieram as com controle remoto, tela plana, tela de plasma, LCD, LED, 3D e SmartTV. Tudo isso para que os fabricantes pudessem continuar tendo nova receita de seus clientes, uma vez que o mercado da TV amadureceu uns 30 anos depois que ela foi inventada.

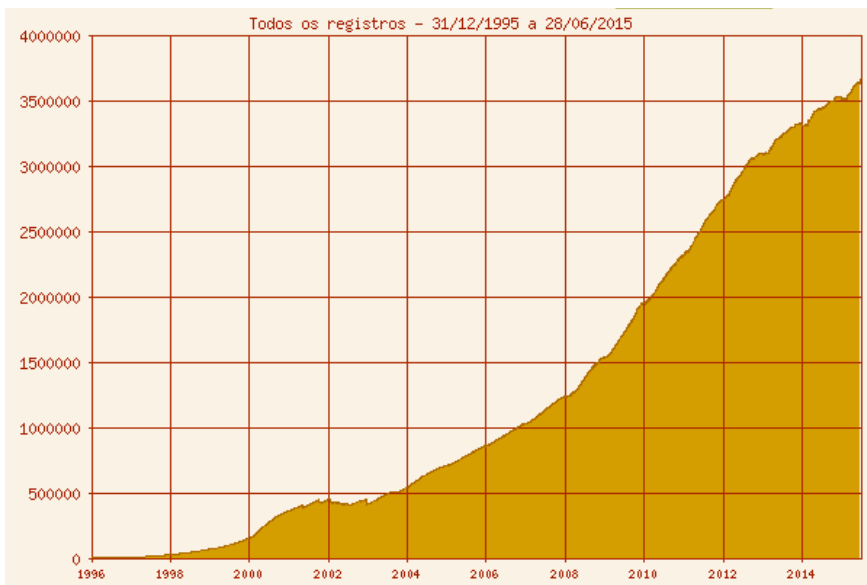
As curvas de internet e de celulares parecem crescer da mesma forma. Já as curvas de PC, eletricidade, aviões, telefone e automóveis têm

algumas alterações em seu desenho; mas, de forma geral, se assemelham bastante à curva S teórica.

Outro exemplo de curva, mais próximo de quem está envolvido com desenvolvimento de software, é a curva de quantidade de registro de domínios .br feitos.

REGISTRO DE DOMÍNIOS .BR

Dá para notar nessa curva a aceleração típica da fase da inovação que aconteceu entre 1996 e 2008. A partir desse ano, entramos na fase de crescimento. Parece que, a partir de 2013, está acontecendo uma desaceleração dessa curva, mas ainda é cedo para podermos dizer se estamos ou não na fase de maturidade.



O ABISMO

Sempre tem um “mas”. Em 1991, Geoffrey Moore escreveu um livro intitulado *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*.

Nesse livro, ele explica que entre os *early adopters* (entusiastas) e a *early majority* (pragmáticos) existe um abismo que muitos produtos não conseguem cruzar. Isso acontece pois os pragmáticos precisam de boas referências para poder comprar um novo produto, e os entusiastas normalmente não são boa referência. Daí a dificuldade de alguns produtos cruzarem esse abismo.

No livro, Moore também apresenta estratégias para cruzar esse abismo; só que, infelizmente, as estratégias propostas são todas baseadas em estratégias de guerra que, como expliquei no capítulo anterior, não acho que fazem muito sentido para o mundo dos negócios.

A estratégia proposta, tirando a referência à guerra do livro, resume-se em **foco**. Ou seja, procure se focar em um único tipo de cliente e resolver muito bem o problema dele com seu produto. Quando esse cliente estiver muito satisfeito, aí é o momento de você procurar novos tipos de clientes.

AUDIBLE

Como você pode ver, estou indicando uma quantidade considerável de livros, e você pode estar se perguntando como fazer para ler tantos. Eu gosto muito de ler, e costumo ler antes de ir dormir. Contudo, com a correria do dia a dia, acabo ficando cansado e leio poucas páginas antes de cair no sono. Isso faz com que um livro médio, de umas 250 páginas, leve quase 3 meses para ser concluído, ou seja, média de leitura de 4 a 5 livros por ano.

Uma alternativa é buscar resumos na internet. Existem até serviços que vendem resumos de livros de negócios. Já usei esses resumos, mas sempre fico com a impressão de que estou deixando algum conteúdo interessante de lado. Há um ano, descobri o Audible (<http://audible.com>), uma empresa que faz parte da Amazon, e que permite ouvir os livros. Tenho ouvido-os no caminho de ida e volta para o trabalho. Com isso, consigo ler, quer dizer, ouvir um livro, a cada duas semanas! *#FicaADica*

O abismo descrito por Moore mostra um dos dois possíveis caminhos de um produto de software:

- **Não vai cruzar o abismo:** a empresa não consegue fazer seu produto ir além dos entusiastas e, conseqüentemente, não terá clientes para sobreviver. Esse é um dos motivos da morte prematura de muitas startups.
- **Amadurecer:** seu produto vai dar certo e a empresa vai eventualmente chegar ao topo da curva S, e desacelerará até que alguma outra empresa invente um produto que substitua o seu. Veja a Kodak que, até hoje, não se recuperou da invenção das máquinas digitais, pois tinha sua receita vinda primariamente da venda de filmes e material fotográfico.

Com isso, chegamos à quarta fase do ciclo de vida de um produto de software: o fim – em inglês, usa-se um termo mais elegante, *sunset*.

Temos, então, quatro fases no ciclo de vida de um produto de software: a inovação, o crescimento, a maturidade e o fim.

Vamos agora conhecer cada uma dessas fases em mais detalhes e entender o papel da gestão de produtos em cada uma delas.

CAPÍTULO 08

INOVAÇÃO: O QUE É?

De todas as fases do ciclo de vida de um produto de software, a fase de inovação é a que costuma ter a maior quantidade de dúvidas. Como já mencionei, há vários livros sobre o tema da inovação e startups. Mas o que é inovação? Como encontrar um problema a ser resolvido? Como descobrir se esse problema é, de fato, uma oportunidade a ser perseguida? E como obter retorno com seu produto de software? Esses são os temas que vou abordar nas próximas páginas.

Inovação é um termo muito usado, mas para cada pessoa que você perguntar, cada uma terá uma definição diferente. Algumas pessoas vão dar uma definição focada na criatividade, ou seja, vão dizer que uma inovação é algo criativo, algo que não existia antes, algo diferente do que se encontra no dia a dia.

Existem vários produtos, não só de software, que são bastante criativos. Existem lojas dedicadas a esses produtos criativos. Nos Estados Unidos, uma loja bastante conhecida de produtos criativos é a Sharper Image.

Sem dúvida, esse ar-condicionado portátil da Sharper Image é um produto bastante criativo, mas, a meu ver, não chega a ser uma inovação.

Quantas pessoas precisam de um ar-condicionado portátil? Isso resolve algum problema ou uma necessidade de um conjunto de pessoas?

Outra empresa que comercializa esse mesmo tipo de produto é a SkyMall, que distribui nos aviões de voo local nos Estados Unidos aquele catálogo cheio de produtos diferentes e criativos. Aqui no Brasil, a Imaginarium é uma loja que oferece esse tipo de produtos.

Além das pessoas que associam inovação à criatividade, existem aquelas que entendem inovação como sendo a última tecnologia. Computador quântico, transmissão de eletricidade sem fio, edição de genoma, realidade virtual, realidade aumentada, nanotecnologia e internet das coisas são alguns exemplos das tecnologias mais recentes. Só que, de novo, não chegam a ser inovações. Quantas pessoas precisam dessas tecnologias? Isso resolve algum problema ou uma necessidade de um conjunto de pessoas?

Inovar não é simplesmente ser criativo ou conhecer a última tecnologia. É preciso conhecer as tecnologias disponíveis e saber como usá-las para

CLARO QUE O CLIENTE SABE O QUE QUER!

É comum ouvir em conversas sobre produtos que o cliente não sabe o que quer. Em determinada altura, alguém soltará a famosa frase de Henry Ford, o inventor do automóvel:

“Se eu tivesse ouvido os usuários em vez do automóvel, eu teria inventado uma carroça mais rápida.”

— Henry Ford

Aliás, quem gostava de repetir essa frase à exaustão era o eterno CEO da Apple, Steve Jobs.

As pessoas sabem sim o que querem. Elas querem uma solução para os seus problemas. O que elas não sabem, ou às vezes acham que sabem, é qual é a solução para esses problemas. E é aí que entra Henry Ford, Steve Jobs e nós, o resto dos mortais, que queremos desenvolver produtos para resolver esses problemas.

Os primeiros passos para criar um bom produto são:

- Perceber que existem pessoas com um problema ou uma necessidade a ser resolvida;
- Entender muito bem qual é esse problema ou necessidade;
- Entender o que motiva as pessoas a querer que esse problema ou necessidade seja resolvido.

Quando você conversar com pessoas com problemas ou necessidades, algumas até dirão que acham que esse problema poderia ser resolvido assim ou assado; entretanto, nesse momento, o mais importante é descobrir se existe uma necessidade a ser resolvida. Você deve separar o problema da sugestão de solução que seu interlocutor está tentando passar.

As pessoas demoravam muito tempo para se locomover. Esse provavelmente era o problema que as pessoas queriam que fosse resolvido na época de Henry Ford. Não importava como. Podia ser com mais cavalos na frente da carroça, podia ser com cavalos treinados para andar de patins, podia ser com cavalos geneticamente modificados para andar mais rápido, podia ser com a invenção do automóvel, podia ser com a invenção do avião, podia até mesmo ser com a invenção do teletransporte.

Note que, para quem tinha o problema de demorar para chegar, a forma como seria resolvido não importava, desde que fosse resolvido. Algumas pessoas provavelmente devem ter sugerido soluções, como a carroça mais rápida da famosa frase de Henry Ford, mas isso é só uma sugestão de solução. O problema a ser resolvido é que as pessoas gastavam muito tempo se locomovendo. Note que o problema não é que as pessoas queriam se locomover mais rápido. Isso já é parte da solução.

Nos próximos capítulos, veremos algumas técnicas para ajudar a descobrir e a entender bem problemas ou necessidades.

CAPÍTULO 9

INOVAÇÃO: FOCO NO PROBLEMA

O primeiro passo para se criar um novo produto de software é entender o problema. É bastante tentador, à medida que se vai entendendo o problema, já ir buscando soluções. É da natureza humana solucionar problemas.

Todo desenvolvedor de software tem alguma história para contar sobre demandas mais ou menos assim: “então, eu preciso de um sistema que faça isso e aquilo, preciso ter um campo para eu digitar essas informações e ele deve ter um relatório que mostre isso aqui”. Aí, quando esse sistema é entregue, a pessoa fala “bem, já ajuda, mas agora eu preciso também de um campo para inserir esse outro dado, e preciso que o sistema exporta um arquivo **.txt** separado por vírgulas”. Ou seja, as demandas já vêm em forma de soluções e sequer mencionam qual o problema a ser resolvido.

Existem algumas técnicas para focar no problema:

ENTREVISTA

É uma ótima forma de entender mais sobre o cliente, o problema que ele tem, o contexto onde esse problema acontece, e o que o motiva a ter esse problema resolvido. Contudo, é preciso ter cuidado, pois o entrevistado vai tentar dar a solução que ele já pensou para o problema. Você deve ser cuidadoso para não menosprezar a sua sugestão, mas deve sempre manter a conversa focada no problema.

A entrevista, na qual você conversa diretamente com seu cliente, é considerada um método qualitativo, ou seja, você faz algumas poucas

perguntas, mas tem a oportunidade de se aprofundar nas respostas.

A seguir, está um conjunto de perguntas para ajudar a manter a conversa focada no problema:

- Conte-me sobre seu problema.
- Qual é a maior dificuldade que você tem em relação a esse problema?
- O que o motiva a querer ter esse problema resolvido?
- Onde esse problema costuma acontecer?
- Quando aconteceu pela última vez? O que aconteceu?
- Por que foi difícil / complicado / ruim?
- Você conseguiu achar uma solução? Qual(is)?
- O que você não gostou das soluções que achou?

Por exemplo, voltando à questão do carro e da carroça do Henry Ford, imagine um diálogo imaginário entre Henry Ford e um hipotético Sr. Smith, um possível comprador de seu futuro produto:

— Sr. Smith, o que mais o aflige?

— Sr. Ford, o que mais me aflige é que passo pouco tempo com minha família.

— E por quê?

— Porque passo tempo demais na carroça indo de um lado para o outro. Se eu pudesse colocar mais um cavalo puxando minha carroça, ela andaria mais rápido e eu passaria mais tempo com minha família.

— Ah, entendi seu problema, e tenho uma solução ainda melhor para você, chama-se automóvel.

Será mesmo que Henry Ford entendeu o problema? Ou ele entendeu a solução que o Sr. Smith apresentou para ele, e desenvolveu uma solução baseada na solução dada pelo Sr. Smith? Será que o problema real do Sr. Smith não era que ele passava pouco tempo com a família?

Vamos experimentar usar as perguntas mostradas anteriormente para ver se conseguimos entender melhor o problema:

— Sr. Smith, o que mais o aflige?

— Sr. Ford, o que mais me aflige é que passo pouco tempo com minha família.

— E qual é a maior dificuldade que o Sr. tem em relação a esse problema?

— Passo muitas horas no trabalho, indo de um lugar para outro, sem falar com as pessoas.

— O que o motiva a querer ter esse problema resolvido?

— Tenho crianças pequenas e, por causa do trabalho, passo muito tempo fora de casa. Quando chego em casa, meus filhos já estão dormindo.

— Onde esse problema costuma acontecer?

— No trabalho.

— Quando aconteceu pela última vez? O que aconteceu?

— Praticamente todos os dias. Aconteceu ontem. Acho que eu consigo chegar em casa a tempo de ver meus filhos acordados uma vez por semana...

- Por que foi difícil / complicado / ruim?
- Porque me deixou longe das crianças.
- Você conseguiu achar uma solução? Qual(is)?
- Saí mais cedo do trabalho.
- O que você não gostou das soluções que achou?
- O trabalho acumulou nos outros dias.

Note que essa conversa pode gerar como solução a invenção do carro para ajudar o Sr. Smith a chegar mais rápido em casa. Entretanto, poderia também ter sido a motivação para a criação de um processo mais eficiente de trabalho, ou de uma máquina que fizesse o trabalho pelo Sr. Smith.

PESQUISA

Outra forma de se obter informações dos clientes é pela pesquisa. Esse método é considerado quantitativo, pois procura-se pesquisar uma quantidade considerável de pessoas para poder obter o que é conhecido como relevância (ou significância) estatística. Isto é, para poder ter confiança de que o resultado obtido não tenha acontecido por acaso.

Por exemplo, caso você pergunte em uma pesquisa se uma pessoa tem iPhone ou Android, e só consiga duas pessoas para respondê-la, ou seja, só duas respostas – quer sejam as duas iPhone, as duas Android, ou uma iPhone e a outra Android –, é muito difícil concluir qualquer coisa. Já se você tiver mais respostas, maiores as chances de o resultado de sua pesquisa refletir a realidade.

Existem ótimas ferramentas para se fazer pesquisa online, como o Google Form e o Survey Monkey. Existe também uma ferramenta brasileira chamada OpinionBox que, além de ter uma ferramenta para você construir seu questionário e coletar respostas, permite também que você selecione pessoas para respondê-lo baseando-se em critérios de definição de perfil que você especifica. Pesquisas não precisam necessariamente ser feitas online. Pode-se também fazer uma por telefone ou ao vivo. Existem empresas que podem ajudá-lo a coletar as respostas.

Decidir fazer uma pesquisa é fácil, já fazer o questionário é difícil. O primeiro passo é ter bem claro qual é o seu objetivo com a pesquisa. Em seguida, crie suas perguntas tendo em mente duas regras principais:

A primeira é: seja breve. Quanto menor o seu questionário, maiores são as chances de obter um bom número de respostas e garantir maior relevância estatística.

A segunda é: seja claro. Principalmente em questionários online, quando você não interage com o respondente do questionário, são grandes as chances de a pessoa interpretar sua pergunta de forma diferente da

qual você havia originalmente pensado. Uma boa forma de você testar seu questionário antes de disponibilizá-lo para obter um grande número de respostas é testá-lo com algumas pessoas ao vivo. Verifique o que as pessoas entenderam de cada pergunta, ou se tiveram dificuldade em compreender alguma parte dele.

Veja alguns exemplos de perguntas ruins que podem gerar interpretações equivocadas ou atrapalhar a qualidade das respostas obtidas, e sugestões de como essas perguntas podem ser melhoradas:

Pergunta original: Quão baixo era Napoleão?

Versão melhorada: Qual era a altura de Napoleão?

Pergunta original: Pais preocupados devem usar cadeirinhas infantis no carro?

Versão melhorada: Você acha que o uso de cadeiras especiais para crianças no carro deve ser obrigatório?

Pergunta original: Onde você gosta de beber cerveja?

Versão melhorada: Quebrar em duas perguntas: Você gosta de beber cerveja? Se sim, onde você gosta de beber cerveja?

Pergunta original: No seu trabalho atual, qual é seu nível de satisfações ou insatisfação com o salário e benefícios?

Versão melhorada: Quebrar em duas perguntas: Qual é seu nível de satisfação com seu salário em seu trabalho atual? Qual é seu nível de satisfação com seus benefícios em seu trabalho atual?

Pergunta original: Você sempre toma café da manhã? Respostas

possíveis: Sim / Não.

Versão melhorada: Quantos dias por semana você costuma tomar café da manhã? Respostas possíveis: Todos os dias / 5-6 dias / 3-4 dias / 1-2 dias / Não tomo café da manhã.

OBSERVAÇÃO

Uma outra técnica bem útil para entender o problema é a observação. Ver o cliente enfrentando o problema ou tendo uma necessidade, no contexto em que isso ocorre, pode ser inspirador!

A observação é uma forma qualitativa de se entender um problema. Para fazer uma boa observação, se prepare; ou seja, tenha em mão um conjunto de hipóteses. A cada rodada de observação, reavalie as suas hipóteses e as ajuste, se necessário. Normalmente, após umas 6 ou 8 observações, você já começará a encontrar um padrão.

A observação pode ou não ter a interferência do observador. Por exemplo, se você estiver observando hábitos de consumo em uma farmácia, você pode observar as pessoas sem que estas notem que você as está observando. Já em um teste de usabilidade, no qual você convida pessoas a testarem seu software, as pessoas saberão que você as está observando. As observações podem ser complementadas com entrevistas, o que pode ajudar a melhorar a qualidade dos achados. Uma técnica muito útil para descobrir problemas no uso do software é observar o que o seu usuário faz 10 minutos antes e depois de usá-lo. Por exemplo, se ele pegar informações em algum outro sistema para

inserir no seu software. Talvez haja aí uma oportunidade para você já trazer essas informações do outro sistema de forma automática. E se, após usá-lo, ele copia informações para uma planilha para gerar algum gráfico, eventualmente seu software poderia poupar esse trabalho e já gerar esse gráfico para ele.

Da mesma forma que existe a entrevista (considerada qualitativa) e a pesquisa (considerada quantitativa), na observação existe também o modo qualitativo (descrito anteriormente) e o quantitativo.

O que seria uma observação quantitativa? É o acompanhamento de métricas como estatísticas de acesso e de uso, engajamento, NPS, receita, novos clientes, *churn* etc. Ou seja, é uma forma de observar o comportamento do seu usuário enquanto ele se relaciona com seu software.

CONCLUINDO

Para concluir este capítulo, vou fazer algumas considerações finais sobre essa fase de entendimento do problema, crucial para o sucesso do seu produto de software.

Deixei de lado, propositadamente, um técnica chamada de *focus group* (ou grupo focal), em que reunimos um conjunto de pessoas (entre 6 a 15) para discutir sobre um determinado problema. É uma técnica tida como qualitativa, pois permite se aprofundar nas questões que estão sendo discutidas.

tante investir tanto tempo e esforço em entender bem o problema?”. A resposta é simples: desenvolver um produto de software é caro. Investir em ter uma boa compreensão do problema, das pessoas que o têm, do contexto onde ele ocorre e das motivações que as pessoas têm para vê-lo resolvido é essencial para não gastar tempo e dinheiro construindo a solução errada.

Por fim, como mencionei antes, os métodos não são excludentes, podem e devem ser usados em conjunto pois se complementam. Uma observação é bem complementada com uma entrevista. Os achados da observação e da entrevista podem ser confirmados (ou confrontados) com resultados de pesquisa e análise de métricas de uso.

Ah sim, tem mais um ponto. Isso tudo que falei sobre o entendimento do problema para se criar um novo produto de software vale também para a implementação de uma nova funcionalidade em um software existente. Afinal, qual desenvolvedor nunca ouviu uma demanda do tipo “é só você colocar uma *flaguezinha* no sistema que já resolve o meu problema”?

CAPÍTULO 10

INOVAÇÃO: O TRABALHO A SER FEITO

O Prof. Clayton Christensen, professor de Harvard e autor de vários livros – dentre eles *The Innovator's Dilemma: the revolutionary book that will change the way you do business*, livro obrigatório para todas as pessoas que trabalham com tecnologia –, desenvolveu uma técnica para descobrir problemas a serem resolvidos, chamada “o trabalho a ser feito”.

A ideia por trás do “trabalho a ser feito” é que precisamos entender para qual trabalho nossos clientes contrataram nossos produtos. Ou seja, qual o trabalho que nossos clientes esperam que nossos produtos façam.

Clayton ilustra essa técnica com um exemplo muito interessante. Ele havia sido contratado para avaliar um produto específico de uma lanchonete, o milkshake. A empresa já tinha feito pesquisas com os clientes perguntando o que eles queriam: se era milkshakes mais densos, com pedaços de biscoito, de fruta ou de chocolate, ou com mais calda. A resposta da pesquisa apontou para uma preferência, que foi implementada. Contudo, essa preferência em nada mudou as vendas, que era o objetivo principal da empresa.

Clayton decidiu fazer, então, a pesquisa de forma diferente. Em vez de perguntar o que os clientes queriam, procurou entender qual era o trabalho para o qual as pessoas contratavam o milkshake. Após várias conversas com clientes, descobriu que os clientes passavam na lanchonete antes de ir para o trabalho de carro e que ficavam bastante tempo no

trânsito. O milkshake, por ser grosso e ser tomado de canudo, demorava para acabar. Demorava o tempo todo da viagem, ou seja, entretia o cliente durante toda a viagem até chegar ao trabalho. As pessoas contratavam o milkshake de manhã antes de ir para o trabalho de carro para ter com que se entreter no caminho.

Elas já haviam tentado com outros “concorrentes”, como bananas, mas acabava muito rápido. Tentaram também com bagels, mas o trabalho e a sujeira para passar manteiga não compensava. O milkshake era perfeito para esse trabalho a ser feito!

Entendendo o trabalho a ser feito, a lanchonete pode melhorar o milkshake, deixando-o mais denso e colocando pequenos pedaços de fruta e / ou cereais para adicionar pequenas surpresas enquanto seus clientes os degustavam. Além disso, preparou um sistema de atendimento que minimizou as filas para garantir um tempo mínimo de espera, por entender que seus clientes estavam com pressa e não queriam ficar esperando na lanchonete. Esses ajustes, sim, trouxeram melhorias nas vendas.

O ALCANCE DA TÉCNICA DO “TRABALHO A SER FEITO”

O próprio Clayton comenta que, no marketing mais tradicional, é comum associar um determinado produto a um determinado conjunto demográfico. Por exemplo, no caso anterior, se fôssemos responsá-

veis pela gestão de produtos da lanchonete, poderíamos associar milkshakes com pessoas de certa idade que trabalham e que, conseqüentemente, sempre procuram milkshake densos que demorem para acabar. Acontece que, usando a técnica do “trabalho a ser feito”, temos uma visão mais ampla do contexto onde o produto está inserido.

Clayton estendeu sua pesquisa para outros períodos do dia e pegou as mesmas pessoas voltando à lanchonete no fim da tarde, só que agora com seus filhos, para um lanche rápido. Com certa frequência, os filhos pediam aos pais por um milkshake. A lanchonete servia o mesmo milkshake: denso, grande, e que demorava para acabar. Só que os pais não queriam esperar muito tempo para que os filhos terminassem seus milkshakes; era para ser um lanche rápido.

Aqui, apesar de o cliente ser o mesmo, o trabalho a ser feito era outro: agradar o filho do cliente de forma rápida. Sendo assim, o milkshake poderia ser menor, talvez metade do tamanho, e menos denso, para acabar mais rápido.

Ou seja, o mesmo cliente quer que o mesmo produto faça trabalhos diferentes, a depender da situação. Daí a importância de entender o trabalho a ser feito.

CONCLUINDO

Neste capítulo e no anterior, aprendemos como entender mais sobre um problema de um grupo de pessoas; entender a fundo seus problemas e o contexto em que eles acontecem; e entender a motivação que as pessoas têm para querer que ele seja resolvido.

Eventualmente, ao usarmos a técnica do trabalho a ser feita, fica a dúvida: será que temos uma boa oportunidade em mãos? Ou ainda, pode acontecer de encontrarmos mais de um problema. Como escolher, dentre esses problemas, e qual é a melhor oportunidade a ser explorada?

CAPÍTULO 11

INOVAÇÃO:
MUITAS
OPORTUNIDADES

É provável que, após se focar no problema e entender o trabalho a ser feito, como descrito nos capítulos anteriores, você descubra não só uma, como algumas oportunidades de desenvolvimento de novo produto, ou de funcionalidades novas para seu produto de software.

Nunca vi em nenhuma organização uma situação em que abundavam os recursos e havia escassez de oportunidades. Via de regra, sempre há muitas oportunidades pela frente, e não sabemos qual ou quais perseguir, pois não conseguimos perseguir todas já que não temos os recursos necessários (tempo, dinheiro ou pessoas) disponíveis.

Nessas situações, gosto de usar o modelo de **análise de oportunidade**, proposto por Marty Cagan em seu livro *Inspired: how to create products customers love*, ex-vice-presidente de gestão de produtos no eBay e, atualmente, consultor sobre gestão de produtos.

Respondendo a um conjunto de 10 perguntas, podemos ter mais informação para decidir, como se vale perseguir uma determinada oportunidade agora, ou se é melhor deixar para reavaliá-la no futuro. As perguntas são bem simples:

- 1) Qual problema vai resolver?
 - proposição de valor;
- 2) Para quem esse problema será resolvido?
 - mercado alvo;
- 3) Qual o tamanho dessa oportunidade?
 - tamanho do mercado;
- 4) Que alternativas existem?

- cenário competitivo;
- 5) Por que somos os melhores qualificados para perseguir essa oportunidade?
 - nossa diferenciação;
- 6) Por que agora?
 - janela de oportunidade;
- 7) Como levaremos essa oportunidade ao mercado?
 - estratégia de lançamento;
- 8) Como vamos medir o sucesso e ganhar dinheiro com esse produto?
 - métricas e receita;
- 9) Que fatores são críticos para o sucesso?
 - requisitos essenciais;
- 10) Dado o acima, qual a recomendação?
 - *go* ou *no-go*.

As perguntas de 1 a 4 servem para entender melhor a oportunidade. Já as 5 e 6 são bem interessantes, pois não basta só entender a oportunidade e ela ser muito boa. É preciso ter certeza de que esse é o momento certo e que nós somos os mais qualificados para persegui-la.

De 7 a 9 são perguntas do tipo “e se...”, ou seja, supondo que se decida por perseguir essa oportunidade, como ela vai ser levada ao mercado, como será medido o sucesso e quais fatores são críticos para o sucesso.

Responder a esse questionário é bom não só para decidir se vale a pena desenvolver um determinado produto, ele também pode ajudar com oportunidades de parceria, oportunidades de desenvolver novas funcionalidades, e oportunidade de atender um cliente especial que

requer mudanças no seu produto. Enfim, todas as oportunidades que vão requerer esforço seu ou do seu time podem e devem ser analisadas para ver se o retorno compensa o investimento de esforço requerido.

QUANTAS OPORTUNIDADES PERSEGUIR AO MESMO TEMPO?

Outra dica importante é não perseguir muitas oportunidades ao mesmo tempo; caso contrário, você e seu time perderão o foco e acabarão não obtendo retorno de nenhuma das oportunidades perseguidas.

No mundo ideal, o indicado é perseguir uma oportunidade por vez. Porém, como sabemos que isso é utópico, acabaremos perseguindo mais de uma oportunidade simultaneamente. Procure limitar essa quantidade a poucas unidades (2, 3 ou 4, no máximo), lembrando de que, em qualquer momento, você e seu time só conseguem dar atenção a uma coisa por vez.

Ou seja, se você estiver trabalhando em mais de uma oportunidade ao mesmo tempo, toda vez que quiser trabalhar em uma delas, deixará de trabalhar momentaneamente nas outras. Por isso, restrinja a poucas oportunidades para perseguir, preferencialmente, uma por vez.

TESTANDO OPORTUNIDADES

Depois de analisar as oportunidades, o próximo passo é testá-las, ou seja, ver se realmente existem pessoas interessadas nesse seu produto que resolverá o problema que você pesquisou. Existem algumas maneiras para fazer isso. A primeira é voltar para a rua e conversar com possíveis clientes. Essa será uma conversa exploratória na qual você expõe o problema para pessoas que provavelmente o tenham. Em seguida, você conta sobre a solução que será desenvolvida, e avalia a reação. Simples assim. O problema de contar sobre uma solução é que, por melhor que seja a descrição, ela ainda não é um produto pronto, e força a pessoa a imaginar algo que pode ser diferente daquilo que você está concebendo.

Em 2011, resolvi fazer um teste um pouco mais realista. Nessa época, conduzi um experimento de startup, fiz minha pesquisa inicial e acabei com 3 ideias produto para desenvolver. Com dificuldade para escolher em qual das 3 oportunidades investir meu dinheiro, minha energia e meu tempo, resolvi fazer um teste bastante simples. Escolhi um nome e um domínio para cada uma das ideias e registrei esses domínios. Em seguida, criei uma página para cada uma das ideias de produto web que tive. Como não sou designer, nem tenho qualquer habilidade para fazer páginas bonitas, usei um site chamado *unbounce* (<http://unbounce.com>), que permite criar páginas simples e até testes A/B de forma bem fácil. Com o *unbounce*, criei a seguinte página:

Note que essa página tem só 3 pontos que descrevem o produto, sendo: um ponto para falar do problema; outro para falar da solução; e um terceiro para falar do preço.

O formulário de e-mail serviu para captar a quantidade de interessados. Rodei uma campanha no Google AdWords de R\$10,00 por dia por produto que queria testar durante um mês.

Veja um print de um resultado parcial:

Esse print é de uma tela do unbounce. Depois dos 30 dias de testes, terminei com 216 e-mails interessados no ContaCal – que foi o produto que acabei lançando e está no ar até hoje – e 1.043 *pageviews*, o que dá 20,7% de taxa de conversão.

As outras duas ideias também mantiveram a mesma tendência desse resultado parcial. Preferi deixá-las ilegíveis para não influenciar ninguém a apostar em alguma delas sem fazer testes.

Para quem não conhece, o ContaCal é um diário alimentar virtual no qual o usuário registra sua alimentação e o sistema informa a quantidade total de calorias com um *twist*. Além da quantidade total de calorias, ele também informa a qualidade das calorias por meio de um sistema de cores. As cores servem para indicar quão recomendável é ingerir o alimento. Se é um alimento de calorias vermelhas, é melhor evitá-lo ao máximo. Calorias amarelas podem ser ingeridas com moderação. Já as calorias verdes podem ser ingeridas sem restrição.

O custo total desse teste foi de R\$990,00:

- Campanha AdWords: R\$900,00.
- Domínios: R\$90,00.
- Site unbounce: grátis por 30 dias; se passasse, pagaria R\$85,00 por mês para até 5 domínios diferentes que eu quisesse testar ao mesmo tempo.

Cada nova ideia custa R\$30,00 por registro de domínio .br, e mais R\$300,00 por mês de campanha no Google AdWords se fizermos um investimento de R\$10,00 por dia.

Um ponto importante a observar é que não necessariamente o ContaCal era a melhor das três oportunidades, mas sim a que eu melhor consegui comunicar. Por isso, esse teste é interessante, ajuda não só a testar a ideia de um produto como também a sua capacidade de comunicar.

CAPÍTULO 12

**INOVAÇÃO:
COMO OBTER RETORNO
COM SEU PRODUTO DE
SOFTWARE?**

Uma das perguntas mais importantes do questionário de **análise de oportunidade** que vimos no capítulo anterior é a pergunta “Como vamos medir o sucesso e ganhar dinheiro com esse produto?”. Ela trata de dois assuntos: métricas e receita. Veremos sobre métricas no *Capítulo 17 – Crescimento: seja um “data geek”*. Agora vamos falar sobre receita.

QUAIS SÃO OS CUSTOS DE DESENVOLVER, DISTRIBUIR E OPERAR UM SOFTWARE?

Todo desenvolvimento de software tem custo. Quando esse software é um produto de software – ou seja, é um software que tem usuários –, o produto tem, além do custo de desenvolvimento, o custo de fazer esse software chegar até os usuários e, a depender do tipo de produto, de armazenar dados desses usuários.

Antes da internet, os produtos de software eram vendidos em caixas com manual e disquetes, ou CDs de instalação. A receita vinha da venda dessas caixas com o software. Algumas empresas chegam a cobrar uma taxa anual de manutenção, que dá aos clientes o acesso a versões mais novas à medida que ficam disponíveis.

Esse produto de software roda nos computadores do cliente, ou seja, todo custo de operação é de sua responsabilidade. O fabricante recomenda uma configuração mínima de equipamento para

rodar esse software e o cliente é que se preocupa em ter, configurar e manter esse equipamento.

Além disso, administrar esse produto de software também é de sua responsabilidade, como também garantir que ele esteja rodando em equipamento com espaço em disco suficiente, com memória suficiente, e que os dados gerados estejam seguros e tenham backups. A receita proveniente da venda desse produto de software serve para pagar seus custos de desenvolvimento e de distribuição.

Com a internet, veio a possibilidade de oferecer softwares para serem usados de forma remota, ou seja, passou a ser possível usar um software que não está mais rodando no equipamento do cliente e que não precisa ser administrado por ele. É o que estamos chamando aqui de produto de software. Nesse novo modelo de comercialização, não há somente a venda do software, mas também há a venda do serviço de sua operação. Mesmo as *apps* dos *smartphones*, que rodam nos telefones dos usuários, têm em sua maioria algum componente online, ou seja, buscam e armazenam dados em servidores remotos, que também trarão custos de operação.

Ou seja, todo produto de software tem custos para desenvolver o software mais custos de operá-lo. Para cobrir esses custos, é necessário ter algum tipo de retorno com seu produto de software que os compense.

TIPOS DE RECEITA DE PRODUTO WEB

Basicamente, existem 3 formas de se ter receita com um produto web:

RECEITA PAGA PELO USUÁRIO

É o modelo usado na maioria dos produtos web para empresas, tais como os produtos oferecidos pela Locaweb, Google AdWords, Mail-Chimp, entre outros. Nesse caso, a receita vem do pagamento periódico (mensal, anual etc.) pelo uso do software via web. Outra opção bastante comum é o pagamento por uso (mais adiante falarei um pouco mais sobre essas formas de pagamento).

A receita paga pelo usuário é também uma forma usada por alguns produtos de software para consumidor final, mas é mais difícil cobrar desse tipo de usuário, sendo alguns exemplos o Netflix, o LinkedIn e o Dropbox. Para entender como é difícil cobrar de um consumidor final, imagine-se pagando para usar a busca do Google. Ou seja, mesmo o consumidor final percebendo muito valor em um produto web, é difícil justificar para esse tipo de usuário que ele deve pagar para usar.

RECEITA PAGA POR ALGUÉM INTERESSADO NO SEU USUÁRIO

Nesse modelo, normalmente não se cobra do usuário do produto, mas sim de alguém que está interessado nele. Esse modelo é bas-

tante usado em produtos para consumidor final. Normalmente, o modelo de negócio é a venda de anúncio. Um exemplo é o Google, que permite que qualquer pessoa use a busca, e cobra de empresas a possibilidade de colocar anúncios junto com os resultados da pesquisa.

Outro exemplo similar é o Facebook, que também oferece acesso gratuito aos usuários, e cobra por anúncios de empresas interessadas em fazer propaganda para seus usuários. Outra forma de receita é a venda de relatórios baseados nos dados de uso.

Um ponto importante a ser ressaltado é que, para conseguir ser interessante para alguém ao ponto de ele querer pagar para ter acesso ao seu usuário, você precisa ter uma quantidade muito grande de usuários. Pense em centenas de milhares, que retornem com frequência para usar seu produto web.

Para atrair centenas de milhares de usuários de forma paga, você provavelmente vai investir bastante dinheiro, então terá de buscar formas gratuitas de divulgar seu produto. Além disso, este deve promover o retorno frequente de usuários, pois assim você terá uma audiência que será de interesse para quem quiser pagar para falar com eles.

Para tornar sua base de usuários ainda mais atraente para possíveis anunciantes, você deve buscar conhecer bastante sobre seus usuários, como dados demográficos, comportamento e preferências. Assim, você pode oferecer maior assertividade e eficiência para os anunciantes.

RECEITA INDIRETA

É a receita que você obtém como resultado do uso do software, mas que não é paga pelo seu uso. Existem basicamente dois tipos de receita indireta:

- **Receita de venda ou aluguel de itens físicos ou virtuais:** é o caso das lojas virtuais que usam os software e serviços de loja online como produto de software para vender ou alugar itens físicos. Amazon e Submarino são bons exemplos.

Há também as lojas que vendem ou alugam itens virtuais, como o serviço de *streaming* da Netflix ou a venda de livros Kindle da Amazon. No caso dos livros, a venda é por item. Já no serviço da Netflix, eles cobram uma mensalidade para ter acesso ao conteúdo de *streaming*. Vale notar que os sites de comércio que fazem a intermediação da venda de itens físicos, como o eBay, Mercado Livre e Elo7, não são desse tipo. Eles são do tipo em que a receita é paga pelo usuário do produto de software, ou seja, por quem coloca o item físico para vender, pagando uma comissão pela venda. Esses sites de intermediação de vendas estão na categoria de plataformas, como definido no *Capítulo 01 – O que é um produto de software?*.

- **Redução de custos:** é o caso do internet banking, intranet de colégio ou faculdade, sistema de acesso a resultado de exames laboratoriais, entre outros produtos de software que não comercializam nada e nem cobram pelo acesso. Nesse tipo de produto, não existe receita, mas sim redução de custo. O internet banking diminui os custos de atendimento presencial de clientes nas agências; o intranet permite um fluxo de comunicação mais ágil entre escola e aluno ou pais de

aluno, economizando em visitas e reuniões no colégio ou faculdade; e o acesso a resultado via internet reduz custos de outras formas de entrega de exames, como impressão e envio pelo correio.

TIPOS DE PAGAMENTO PELO USO DO PRODUTO DE SOFTWARE

Quando há pagamento pelo uso de um produto, este pode ser feito de duas formas:

Pagamento periódico: é o pagamento de um valor fixo para poder usar o produto, como uma assinatura de uma TV a cabo ou uma mensalidade de uma academia. Se o usuário deixa de pagar, ele deixa de poder usar o produto e de ter acesso a todas as informações que podem estar armazenadas nele. As periodicidades mais comuns são a mensal e a anual.

Pagamento por uso: nesse caso, paga-se pelo uso do produto que deve ter uma forma muito fácil de ser medida e acompanhada. Por exemplo, em um produto de e-mail marketing, que permite o disparo de mensagens de e-mail para uma lista de endereços, pode ser cobrado pela quantidade de mensagens enviadas. Outro bom exemplo é o pagamento de anúncios, que podem ser pagos por exibição, por cliques ou ainda pela conversão do anúncio. A comissão cobrada pelos sites de intermediação de venda de itens físicos como eBay, MercadoLivre e Elo7 é outro exemplo, como também a cobrança feita pelo

Skype para fazer ligações para telefones fixos e celulares.

É possível ainda ter um modelo misto, com pagamento periódico mais pagamento por uso. Um bom exemplo é um produto para telefonia via internet, em que você pode cobrar uma mensalidade para ter acesso ao produto, mais uma cobrança por ligações feitas, sendo o caso do PABX Virtual da Locaweb. Outro exemplo é um produto que oferece a possibilidade de seu usuário ter uma loja virtual. Nesse caso, pode-se cobrar uma mensalidade mais uma taxa de uso baseada na quantidade de vendas que seu cliente faz usando essa loja.

OBTENDO RETORNO DE UMA PLATAFORMA

Nos *Capítulos 01 e 02*, falei sobre um tipo diferente de produto de software, as plataformas, sistemas que são mais valorizados quanto mais pessoas usam. Expliquei o que as diferenciam de um produto, e quais as diferenças entre gerenciar um produto e uma plataforma.

Agora que vimos como obter retorno com um produto de software, perguntas que ficam são: como funciona a precificação da plataforma, onde tenho interesse que muitas pessoas usem para que ela seja mais valorizada? E, em alguns casos de plataforma, tenho pessoas de grupos diferentes (compradores e vendedores, desenvolvedores de apps e usuários de smartphones). Ou seja, o ideal em uma plataforma seria não cobrar nada de ninguém, para garantir o maior número de usuários;

entretanto, se eu não cobrar nada, como cobrirei os custos do seu desenvolvimento e de sua operação?

São quatro pontos que devemos levar em conta sobre precificação de plataforma: quem, o quê, quanto e quando devemos cobrar.

QUEM DEVEMOS COBRAR?

A primeira resposta a essa pergunta é simples: quem for menos sensível ao preço. Se você tem uma plataforma de dois lados, identifique qual dos dois é menos sensível a um preço; este será o lado com maior disposição para pagar por uma plataforma. O lado mais sensível é o que queremos subsidiar.

Por exemplo, se você estiver gerenciando uma plataforma que conecta advogados com possíveis clientes, é bem provável que o lado menos sensível ao preço sejam os advogados, por terem margens altas. Por outro lado, suponha que você esteja gerenciando uma plataforma que conecta fornecedores de equipamentos e materiais químicos – que têm pouca margem em seus preços – com possíveis compradores; talvez faça mais sentido cobrar dos compradores, devido à baixa margem dos fornecedores.

Contudo, como disse anteriormente, essa é só a primeira resposta. Há outros fatores a serem considerados:

Sensibilidade à escala: se um dos lados perceber que sua plataforma é mais valiosa a partir da quantidade de usuários do outro lado, este será o sensível à escala e com maior disposição para pagar pela plataforma. Por exemplo, no Google, quanto mais pessoas buscando, mais interessante a plataforma fica para os anunciantes.

Sensibilidade à competição: se um dos lados perceber que a plataforma é mais valiosa quanto menor for o seu lado e, conseqüentemente, menor for a competição existente, este será o lado sensível à competição e com mais disposição para pagar. Um exemplo fora da internet são os bares que cobram entrada, que decidem ou cobrar uma entrada mais barata das mulheres ou não cobrar, mas cobram entrada cheia dos homens, que estarão dispostos a pagar por entender que haverá menos competição.

Ou seja, há vários fatores para se levar em consideração. Em alguns casos, os dois lados podem estar dispostos a pagar; em outros, somente um. Casos como Google e Facebook são mais simples de analisar:

- 1) São sempre dois lados: quem anuncia e quem consome a plataforma.
- 2) Para quem anuncia, quanto mais pessoas consumindo a plataforma, melhor.
- 3) Para quem consome a plataforma, os anúncios ou o próprio conteúdo podem ser uma intrusão se o Facebook e o Google não conseguirem fazê-los ser relevantes.
- 4) Para quem anuncia, cada negócio iniciado ou

- 5) Para quem consome a plataforma, cada negócio iniciado ou fechado através da plataforma representa um investimento com um possível desembolso de dinheiro.
- 6) Por essas razões, principalmente pelas duas últimas, faz sentido que sejam os anunciantes a pagar.

Por outro lado, casos como sites de leilão, de emprego, de busca de táxi, e de imóveis para compra ou aluguel já são mais complexos:

- 1) Também são sempre dois lados: quem anuncia e quem tem interesse no que está sendo anunciado.
- 2) Para quem anuncia, quanto mais pessoas consumindo a plataforma, melhor.
- 3) Para quem tem interesse no que está sendo anunciado, quanto mais ofertas disponíveis, melhor, pois há mais chances de encontrar o que se procura.
- 4) Para quem anuncia, cada negócio iniciado ou fechado através da plataforma representa um ganho.
- 5) Para quem tem interesse no que está sendo anunciado, cada negócio iniciado ou fechado através da plataforma representa a possibilidade de conseguir o que estava procurando.
- 6) Por essas razões, principalmente pelas duas últimas, pode fazer sentido que os dois lados

paguem para usar a plataforma. Por outro, considerando os itens 2 e 3, também faz sentido que os dois lados possam usar a plataforma sem nenhum custo. Nesse caso, pode fazer sentido analisar quem é menos sensível ao preço e quais são as práticas do mercado.

O QUE DEVEMOS COBRAR?

Normalmente, as pessoas estão dispostas a pagar por algo quando enxergam o valor e o benefício que esse algo traz para elas.

Existem 3 benefícios que podem ser cobrados em uma plataforma:

Acesso: pode-se cobrar para as pessoas terem acesso à plataforma. Algo como uma taxa mensal, por exemplo. Esse é a forma de cobrar menos comum, uma vez que um dos seus principais benefícios é a quantidade de pessoas nos diferentes lados. Pode-se encontrar esse modelo em plataformas que trabalham a exclusividade, em que a qualidade dos membros é sua principal proposta de valor, e a quantidade não é relevante. Algumas plataformas, após atingirem um certo volume crítico, podem decidir cobrar pelo acesso para garantir qualidade e exclusividade. É possível também implementar dois níveis de acesso, um grátis e um pago, com diferentes funcionalidades e níveis de serviço como, por exemplo, quem é grátis tem suporte somente via site de perguntas e respostas, e quem paga tem direito a um suporte personalizado por telefone.

Uso: pode-se cobrar por cada vez que as pessoas utilizam a plataforma. Por exemplo, em um site para anúncio de vagas de trabalho, pode-se cobrar por cada vaga anunciada. Outro exemplo é o Google AdWords, em que a cobrança é feita a cada vez que alguém clica em seu anúncio, ou seja, cada vez que seu anúncio é usado.

Taxa/percentual: nesse modelo, o valor a ser cobrado é um percentual ou uma taxa variável do benefício que um dos lados da plataforma tem com cada negócio realizado nela. Normalmente, sites de leilão e de intermediação de pagamento (PayPal, PagSeguro etc.) costumam usar esse modelo de cobrança.

É possível fazer combinações dessas formas de cobrança. Por exemplo, cobrar uma taxa mensal de acesso mais uma taxa de uso, ou um percentual.

QUANTO DEVEMOS COBRAR?

Para responder a essa pergunta, precisamos pensar no *lock-in*, que significa que um usuário de sua plataforma tem chances menores de parar de usá-la quanto mais ele usar e enxergar benefícios no seu uso. O custo de troca é o que explica o lock-in; quanto maior o custo de troca, maior será o lock-in. Outro ponto importante para levar em consideração quando estamos definindo quanto cobrar pela plataforma é o efeito de rede, ou seja, quanto valor geramos para o usuário ao termos mais pessoas usando a plataforma. Normalmente, o valor a ser cobrado por uma plataforma, quer seja acesso, uso e/ou taxa, reflete no lock-in e no efeito de rede.

Geralmente, esses valores mudam ao longo do ciclo de vida de uma plataforma. No começo, quando há poucos usuários, e o lock-in e o efeito de rede ainda são pequenos, é muito provável que se tenha que subsidiar o seu uso.

O Dropbox pode ser visto como uma plataforma de um lado só (*single-sided platform*), onde o benefício do efeito de rede aparece quanto mais usuários o usarem para troca de arquivos. O custo de troca aumenta à medida que você coloca mais arquivos no Dropbox e tem mais conhecidos com quem trocar esses arquivos por lá. Esse é o motivo por ele cobrar por GB e incentivar a convidar amigos para também o usar. Esse incentivo, dando GBs de graça para você e para os amigos que convidar, é a forma de subsídio que o Dropbox usa para poder aumentar o lock-in e o efeito de rede de sua plataforma.

Outro exemplo interessante é o de um serviço de monitoração de avalanches, que lançou uma plataforma na qual resorts de esqui compartilhariam dados sobre condições meteorológicas para poder prever avalanches com maior acurácia. Para poder participar desse sistema, o resort teria de instalar um monitor, e esse processo de instalação tinha um custo considerável. A plataforma pretendia cobrar também uma mensalidade para os resorts poderem usá-la. O problema é que eles não se sentiam confortáveis em pagar a mensalidade tendo já pago o custo da instalação do equipamento de monitoração. Além disso, não queriam pagar mensalidade no verão, quando há poucas avalanches e os resorts têm baixa ocupação. A solução encontrada foi subsidiar a instalação dos monitores nos resorts e fazer a cobrança anual em vez de mensal, com preço variando por profundidade da análise feita.

QUANDO DEVEMOS COBRAR?

Para cobrança por acesso à plataforma, pode-se cobrar uma única vez ou periodicamente. Por periodicidade, pode-se variar do mensal até a cobrança por múltiplos anos. Não é incomum ver casos em que há múltiplas opções de período (por exemplo, mensal e anual), em que se oferecem descontos nos preços com período maior.

Em alguns casos, períodos maiores podem ser a única forma de mostrar o benefício do longo prazo de sua plataforma, ou de afastar preocupações com a sua utilidade sazonal, como no caso do serviço de monitoração de avalanches.

Para cobrança por uso, o mais comum é fazer essa cobrança de forma periódica, ou seja, a cada período é avaliado quanto foi usado e é feita a cobrança. O período mais comum é o mensal.

Um cuidado deve ser tomado em relação a modelos de cobrança mistos, com cobrança por acesso mais cobrança por uso. Se você fizer cobrança por acesso anual, avalie se você poderá fazer também a cobrança por uso anual, ou se é melhor cobrar pelo uso mensal ou trimestralmente.

Já na cobrança de taxa ou percentual, o mais indicado é que ela ocorra no evento da transação. Caso haja uma frequência alta de eventos de transação em um mês, uma outra opção é cobrar essa taxa mensalmente.

CONCLUINDO

Acabamos de entender que todo produto de software tem custos de desenvolvimento e de operação, e que é preciso cobri-los de alguma forma. Vimos diferentes formas de se obter retorno com o produto, e entendemos as diferenças entre produto e plataforma no que se refere à obtenção de receita.

CAPÍTULO 13

INOVAÇÃO: PRÓXIMOS PASSOS

Uma vez que você descobriu um problema de um conjunto de pessoas, aprofundou-se no entendimento desse problema, no contexto onde ele acontece e nas motivações que as pessoas têm para ter esse problema resolvido, analisou a oportunidade em mais detalhes para avaliar se vale a pena desenvolver seu produto de software, e avaliou como você poderá cobrir os custos do seu desenvolvimento e de sua operação, chegou a hora de realmente desenvolvê-lo.

Existe uma quantidade enorme de livros que falam sobre o assunto e, se eu fosse também falar do assunto aqui, este livro iria provavelmente dobrar de tamanho. Por isso, prefiro dar algumas referências de livros que acho interessantes sobre esse tema. Aliás, desenvolvimento de um produto de software tem sido um tópico muito explorado em inúmeros textos, palestras e livros que falam sobre startups. De uma certa forma, startup e desenvolvimento de produto de software podem ser considerados sinônimos.

GUIA DA STARTUP

O primeiro livro que vou indicar é o **Guia da startup: como startups e empresas estabelecidas podem criar produtos web rentáveis**, publicado em 2012 pela Casa do Código, de autoria deste que vos escreve. Nesse livro, falo sobre inúmeras técnicas de desenvolvimento de produto de software e as

ilustro com exemplos práticos baseados na minha experiência com o ContaCal e com a Locaweb, e em conversas que tive com pessoas responsáveis pelo desenvolvimento de produtos de software de outras empresas. Nele, falo de vários assuntos relacionados a desenvolvimento de produto de software, tais como:

Problema ou necessidade?

Qual é a diferença entre problema e necessidade? Devo me focar em resolver um problema ou atender uma necessidade?

Produto web, mobile ou social?

As pessoas têm passado cada vez mais tempo em mobile e em redes sociais. Devo focar meu produto de software na web ou em mobile primeiro? Ou será que é melhor fazer logo uma aplicação para o Facebook?

O que é MVP?

O que caracteriza um produto ser minimamente viável? Viável para quem e para quê?

Por que é preciso lançar logo?

Você já deve ter ouvido falar que quanto mais rápido você lançar seu produto de software, melhor. A primeira explicação que vem à mente quando pensamos nisso é o custo de desenvolvimento, ou seja, quanto mais rápido lançarmos, menos vai nos custar. Contudo, na minha visão, essa não é a principal razão. Existem outras mais importantes.

Como fazer rápido seu produto web?

Depois de entender por que é importante desenvolver seu produto de

software rapidamente, a próxima preocupação é como acelerar esse desenvolvimento.

Como definir o preço certo?

Você vai cobrar pelo seu produto? Quanto? Quando?

Plano grátis vale a pena?

Existem vários produtos de software que oferecem plano grátis. Será que vale a pena para qualquer tipo de produto de software? Se não, que tipo de produto de software se beneficia da oferta de um plano grátis?

Checklist de lançamento.

Ok, está tudo pronto para lançar seu produto de software. Está mesmo? É sempre bom checar se está tudo em ordem antes de “abrir as portas”.

MAIS SUGESTÕES DE LEITURA

A seguir, há mais dicas de leitura sobre desenvolvimento de produto de software, em ordem de importância:

Direto ao ponto: criando produtos de forma enxuta, do Paulo Caroli, onde ele compartilha uma sequência de atividades rápidas e efetivas para entender e planejar a criação de produtos enxutos, baseadas no conceito de produto mínimo viável.

Getting real: the smarter, faster, easier way to build a successful web application, de Jason Fried, David Heinemeier Hansson e Matthew Linderman. Esse livro conta como o pessoal da 37signals fez seus produtos de sucesso. Tem várias dicas práticas muito bacanas e versão em português na web, *Caindo na real*, em http://gettingreal.37signals.com/GR_por.php.

The entrepreneur's guide to customer development: a cheat sheet to The Four Steps to the Epiphany, de Brant Cooper e Patrick Vlaskovits. Steve Blank, empreendedor em série do Vale do Silício, escreveu um livro intitulado *The Four Steps to the Epiphany: successful strategies for products that win*, que trata de startup de forma genérica, mas que cria um conceito muito importante, o de *customer development* (desenvolvimento do cliente). De acordo com sua experiência, startups não morrem pela dificuldade em fazer um bom produto, mas sim pela dificuldade em encontrar clientes para esse ele. Daí a ideia de buscar e desenvolver o cliente antes de desenvolver o produto. O problema é que o livro do Steve Blank é bem denso, então Brant e Patrick fizeram um excelente resumo de 104 páginas onde explicam em detalhes o conceito de *customer development*.

Where good ideas come from: the natural history of innovation, do Steven Johnson, autor de vários livros interessantes sobre ciência e conhecimento. Nele, ele explica quais os principais ingredientes da inovação, sendo um dos mais importantes a necessidade de equipes multidisciplinares para que seja possível ver o mesmo problema com diferentes perspectivas.

The little black book of innovation: how it works, how to do it, de Scott D. Anthony, sócio junto com o Prof. Clayton Christensen de uma empresa de consultoria em inovação chamada Innosight. Nesse livro, ele define inovação como *algo diferente que tem impacto*. A partir daí, ele mostra um guia passo a passo para encontrar e testar oportunidades de inovação.

Crossing the chasm: marketing and selling high-tech products to mainstream customers. Escrito em 1991, Geoffrey Moore escreveu o já mencionado livro onde ele explica que entre os *early adopters* (entusiastas) e a *early majority* (pragmáticos) existe um abismo que muitos produtos não conseguem cruzar, já que estes precisam de boas referências para poder comprar um novo produto e aqueles normalmente não são boa referência. É neste livro que Moore também propõe estratégias baseadas em estratégias de guerra, como descrito no *Capítulo 07 – Como é o ciclo de vida de um produto de software?*.

Running lean: iterate from Plan A to a plan that works, do Ash Maurya. Em 2010 Alexander Osterwalder e Yves Pigneur apresentaram um novo framework para analisar modelos de negócio, o Business Model Canvas (BMC). Gosto bastante desse framework, só que o BMC me parece mais focado para empresas já andamento e não para startups. Em 2012 Ash Maurya cria um framework a partir do Business Model Canvas, só que mais aplicável a novos negócios pois fala em problema, solução e métricas.

The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses, de Eric Ries,

muito amigo do Steve Blank. Esse livro foi resultado do blog *Startup Lessons Learned* que ele escreveu e continua escrevendo sobre as suas experiências com sua startup. Também é focado em startups de forma geral, não só em startups de produto de software. Chega até a falar sobre startup de uma ONG, de conceitos bastante conhecidos como o MVP (*Minimal Viable Product*, ou Produto Mínimo Viável) e do ciclo de feedback *Build-Measure-Learn* (Construa-Meça-Aprenda). Foi lançado em português recentemente com o título *A startup enxuta*.

Boa leitura, bom desenvolvimento de produto e prepare-se para a próxima fase: o crescimento (ou o abismo...), que é o próximo assunto que veremos.

CAPÍTULO 14

CRESCIMENTO: FEEDBACK

Você descobriu um problema de um conjunto de pessoas, e entendeu a fundo esse problema e o seu contexto. Descobriu o que motiva as pessoas a terem-no resolvido, e analisou a oportunidade em mais detalhes para avaliar se vale a pena desenvolver seu produto de software. Por fim, você desenvolveu sua primeira versão de seu produto de software e o lançou no mercado, seguindo as recomendações dos vários ótimos livros que existem falando sobre startup, inovação e desenvolvimento de software.

Sucesso! Só que não...

Na verdade, esse foi o seu primeiro passo de uma jornada bem longa, que vamos conhecer nos próximos capítulos.

Passada a fase de inovação, quando você desenvolve e lança a primeira versão de seu produto de software, chega o momento em que você tem de investir mais energia para entender se seu produto de fato resolve o problema dos clientes, e atende aos objetivos de seu dono. Esse é o momento em que você, ou entra na fase de crescimento, ou então pode não cruzar o abismo.

Aliás, esse é um dos momentos mais importantes do ciclo de vida de seu produto de software, o “momento da verdade”, o momento no qual seu software encontra o mundo real. Você certamente deixou alguma maneira fácil de seus usuários entrarem em contato, e agora está começando a receber seus *feedbacks*. São esses *feedbacks* que vão dizer se você está ou não na direção certa para resolver o problema deles.

LIDANDO COM FEEDBACKS DE USUÁRIOS

Aqui vão algumas dicas de como lidar com esses *feedbacks*:

RESPOSTA RÁPIDO

É importante responder rapidamente aos *feedbacks*. Isso criará uma percepção de que quem está por trás do produto de software se importa com os comentários e com os usuários de seu sistema. Isso ajuda a criar uma imagem positiva de seu produto.

NÃO INVENTE

Não diga que você vai implementar alguma determinada funcionalidade em algum momento do futuro se você não tem planos de fazer

isso, ou se isso é só uma possibilidade remota. Se esse for caso, apenas agradeça a sugestão.

SEJA EDUCADO

Essas pessoas que estão dando feedback estão fornecendo uma informação muito valiosa. Mesmo que elas não sejam muito educadas com você, o que elas estão lhe dizendo serve para você entender como elas estão percebendo seu produto.

Você deve sempre ser educado com seus usuários, mesmo com aqueles que não forem muito educados com você. Sua forma educada de tratá-los pode eventualmente ajudar a reverter a má impressão que ele tinha de seu produto.

NÃO IMPLEMENTE TODAS AS SUGESTÕES QUE RECEBER

Principalmente no começo, você receberá muitas sugestões de funcionalidade: versão para celular, funcionamento mais preditivo, conhecendo o usuário e já preenchendo os dados automaticamente, e assim vai.

Nesse começo, o recomendado é não implementar nada: você ainda está conhecendo seus usuários e entendendo se seu produto resolve um problema real deles. Se você implementar todas as sugestões que receber, poderá estragar a solução que criou e, pior, começará a dei-

nar seu produto complicado, com muitas funcionalidades desnecessárias. Para lidar com todas as sugestões, não é necessário criar um sistema para anotá-las. Depois de algum tempo recebendo sugestões, você vai se lembrar de quais são as mais populares.

Se você quiser implementar um sistema de sugestões, existe o User-Voice (<http://uservoice.com>), que tem versão gratuita.

FEEDBACK NÃO É SÓ O QUE O USUÁRIO LHE MANDA

Apesar de seus usuários que lhe mandam feedback já lhe dizerem muita coisa, você não deve considerá-los como sua única fonte de feedback. Você deve usar as estatísticas de uso do produto de software como ferramenta para entender como ele está sendo utilizado. A quantidade de vezes que as pessoas acessam, a quantidade de dados que elas entram no sistema, depois de quanto tempo elas voltam, tudo isso você deve ser capaz de extrair de sua base de dados e do relatório de estatísticas de acesso.

Para relatório de estatísticas de acesso ao site, uma boa solução é o Google Analytics. Com um pequeno pedaço de código JavaScript em seu site e em sua aplicação web, o Google Analytics colhe várias informações das pessoas que navegam por eles, como: por qual página entraram; de que página saíram; quanto tempo ficaram em cada página; de que localidade são (cidade, estado, país); se estão acessando via computador, tablet ou celular; além de informar quantidade de visitas, e várias outras informações muito úteis, principalmente nesse começo.

Uma outra ferramenta muito útil para ver como as pessoas estão usando seu produto é o ClickTale, que também com um pequeno JavaScript colocado em sua página, é capaz de dar informações sobre, não só os cliques, como também da movimentação do mouse das pessoas enquanto elas usam seu produto. Ver isso pode lhe dar várias informações úteis sobre sua aplicação.

INTERAJA COM SEUS USUÁRIOS POR DIFERENTES MEIOS

Existem outras formas além de você obter feedback de seus usuários. Seu site tem um blog para você contar novidades de seu produto, não tem? Nos comentários dos posts, você certamente receberá muita informação bacana.

Você também pode criar uma página no Facebook, que pode ser o lugar onde os usuários de seu produto se encontram e trocam experiências. Se você tiver a oportunidade, também converse ao vivo com as pessoas que estão usando seu sistema. Conversas ao vivo são muito ricas por permitirem maior interação e troca de informações.

EXEMPLO DE PRESSA EM AGIR DEVIDO AO FEEDBACK

Logo após o lançamento do ContaCal, produto de software que comentei no *Capítulo 11 – Inovação: muitas oportunidades*, muitos usuários comentaram que seria bacana ter a possibilidade de controlar não só a quantidade de calorias ingeridas como também a quantidade de calorias gastas. De tanto ouvir pessoas pedindo por essa funcionalidade, ela acabou ficando na minha cabeça como uma funcionalidade importante para ser implementada. Talvez eu pudesse até estar perdendo inscrições de novos usuários por não tê-la, ou usuários estivessem desistindo de utilizar o sistema pela falta dela.

Eu estava me organizando para decidir como implementar cobrança no sistema mas, em função desse feedback, decidi colocar a funcionalidade de registro de atividades físicas no ContaCal, dois meses e meio após seu lançamento. Fiz isso por ser simples de implementar e por achar que poderia ajudar a aumentar o número de pessoas que continuariam utilizando o produto após o primeiro uso.

Anunciei para a base de usuários existentes, passei a destacar isso como uma das funcionalidades que o sistema tem na comunicação do produto, e passei a ensinar novos usuários mais sobre essa funcionalidade. Várias pessoas acharam bacana, mas a curva de novos usuários que se

cadastraram para testar o sistema, bem como de usuários que decidiram continuar usando-o após o primeiro contato, não mudou em nada!

Para confirmar essa percepção de esforço com pouca utilidade, bastou olhar na base de dados para ver que apenas 2,4% dos registros feitos são de atividades. Acabei postergando em um mês a implementação de cobrança para colocar essa funcionalidade. Hoje vejo que não precisava ter a implementado antes da funcionalidade de cobrança. Acho até que não precisava ter implementado essa funcionalidade e poderia ter deixado o ContaCal como sistema apenas para registro de calorias ingeridas, sem me preocupar com o gasto de calorias em atividades físicas.

Recentemente, recebi um e-mail de uma usuária reclamando de que o gráfico que apresento como relatório não mostra as atividades físicas, sendo que o objetivo deste é somente mostrar a evolução das calorias ingeridas. Ou seja, coloquei mais uma complexidade no sistema com a qual tenho de lidar até hoje e com praticamente nenhum ganho, nem para os usuários, nem para mim, que sou o dono do software.

Muito cuidado quando for implementar uma nova funcionalidade. Sua criação cria complexidade de código, de regra de negócio e de interface. Se não estiver muito claro o que seus usuários e o dono do software vão tirar de positivo dela, talvez seja melhor esperar um pouco antes de investir.

CRUZANDO O ABISMO

Cruzar o abismo que separa os primeiros clientes, aqueles entusiasmados que adoram testar todo e qualquer produto novo, do restante do mercado não é uma tarefa simples. Tanto que existe um livro inteiro falando sobre o assunto, que já mencionei algumas vezes, o *Crossing the Chasm*, de Geoffrey A. Moore.

Apesar de eu já ter falado várias informações a respeito desse livro, neste momento em que estamos considerando o crescimento do produto e a importância de recolher feedback, vale a pena nos determos em algumas observações sobre ele.

A primeira parte do livro é muito boa; uma leitura recomendada para qualquer pessoa que trabalha com tecnologia. Nela, Moore mostra a curva de adoção de tecnologia em detalhes, a existência do abismo nessa curva e explica o motivo de este abismo acontecer.

Na segunda e última parte do livro, Moore sugere como cruzar o abismo. O problema é que ele usa analogias de guerra para isso; o que, volto a ressaltar, não é a maneira mais apropriada para endereçar questões de negócios. O primeiro capítulo da segunda parte está intitulado como “A analogia do dia D”, e vem seguido de capítulos com o mesmo tipo de títulos (“Mire o ponto de ataque”, “Monte sua força de invasão”, “Defina a batalha” e “Lance a invasão”).

Apesar da péssima analogia, as ideias são boas. Em resumo, ele recomenda um profundo entendimento do usuário para garantir que seu produto esteja, de fato, resolvendo um problema dele. Ele re-

comenda foco nesse único empecilho para um grupo de usuários. Melhor ser algo muito bom para um grupo de pessoas do que tentar ser tudo para todos.

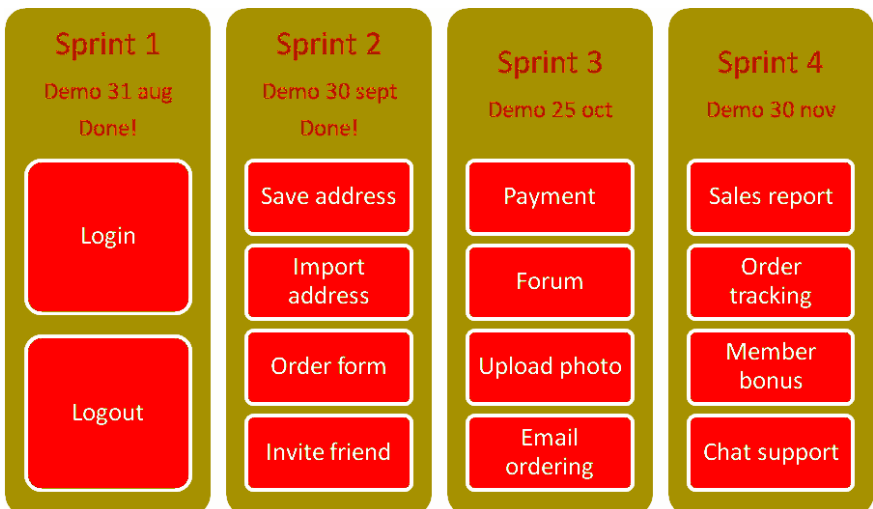
A partir do momento em que seu produto for uma ótima solução de um problema de um pequeno grupo, você poderá procurar mais pessoas que possam estar com um problema igual ou parecido para você adaptar seu produto, e assim ganhar mais mercado.

Claro que é muito mais fácil falar do que fazer, mas não se esqueça: o bom entendimento de seu usuário, de seu problema, do contexto onde este acontece e a motivação que ele tem para querer tê-lo resolvido são as suas melhores ferramentas para cruzar o abismo, e evitar a morte prematura de seu produto de software.

CAPÍTULO 15

CRESCIMENTO: O QUE É UM ROADMAP?

O *roadmap* é uma ferramenta muito útil para gestores de produto. Com ele, é possível planejar e comunicar a visão de futuro que você tem para o seu produto. Veja a seguir alguns exemplos de roadmap:



Repare que nos dois primeiros exemplos de roadmap são apresentadas as funcionalidades que vão constar em cada versão do software e que são roadmaps de curto prazo, ou seja, mostram alguns poucos meses. Já no roadmap do Windows Server, vemos uma visão mais abrangente, sem grandes detalhes, mas sendo um roadmap de longo prazo, com quase uma década.

Ao preparar o roadmap de seu produto, você deve adequá-lo à sua audiência. Isto é, você deve colocar mais ou menos detalhes, dependendo de para quem você apresentará esse roadmap.

DE QUANTO EM QUANTO TEMPO TENHO DE ATUALIZAR O ROADMAP?

Se você está em um time que utiliza boas práticas de engenharia de software, vocês estarão fazendo entregas frequentes e, ao fazer entregas frequentes, você terá bastante feedback de seus usuários sobre o software e as funcionalidades que vocês estão entregando. Isso provavelmente vai mudar seu roadmap, pois quando seus usuários começam a usar uma nova funcionalidade, eles terão novas sugestões para o seu software. Mas, mesmo que você não receba nenhuma sugestão, ao vê-los utilizando, você provavelmente terá novas ideias para seu produto.

Se você for um gestor de um produto de hardware – como um servidor, um notebook, um smartphone, um tablet, ou mesmo um sistema operacional para rodar nesses aparelhos –, seu roadmap será bem menos flexível. Muitas decisões deverão ser tomadas meses antes de o produto estar na frente do usuário.

Felizmente, as entregas contínuas em produtos web permitem muito mais flexibilidade. É interessante ter um roadmap de um produto web de pelo menos 12 meses. Entretanto, não se esqueça de que esse road-

map mudará frequentemente, de acordo com o que você e seu time aprenderem com os usuários do seu produto web e com a forma como o mercado reage às suas novidades. Por isso, a cada mudança de rumo, atualize seu roadmap e comunique todas as pessoas interessadas.

DEVO GUARDAR SEGREDO SOBRE MEU ROADMAP?

Muitas empresas publicam seus roadmaps para seus usuários e para o mercado, enquanto outras preferem guardá-los a sete chaves temendo que concorrentes copiem seus passos. Acredito que o roadmap de curto prazo (1 a 3 meses) deve ser conhecido pelos seus usuários, até para que eles possam dar feedback sobre ele.

Já o mercado, você pode responder de forma reativa, ou seja, quando perguntado, você pode responder se está ou não no seu roadmap de curto prazo. Já o de médio e longo prazo (3 ou mais meses), não faz sentido ser divulgado, nem tanto para guardar segredo de seus concorrentes, mas porque há grandes chances de ele mudar e, se ele for público, essas mudanças acabarão frustrando seus usuários.

CONE DA INCERTEZA

O cone da incerteza é um conceito usado em gestão de projetos que descreve a quantidade de incerteza ao longo da vida de um projeto.

No começo, pouco se sabe e a incerteza é grande. À medida que progredimos no projeto, aprendemos mais e a incerteza diminui.

Pesquisadores da indústria de software chegaram a concluir que, antes do início de um projeto de desenvolvimento de software, a incerteza pode variar de 4 vezes a até 1/4 do inicialmente estimado quanto ao tempo e ao custo de desenvolver esse software.

COMO FAZER UM ROADMAP?

Após aprender o que é um roadmap, a pergunta que fica é: como fazer um roadmap? Ou seja, como definir que itens vão nele e em qual ordem? A resposta é composta de três elementos: a empresa, os usuários e o que dá para fazer.

Quais os objetivos da empresa?

O primeiro elemento que um gestor de produtos deve conhecer para fazer o roadmap são os objetivos da empresa. O principal objetivo de uma empresa não é receita ou margem. Receita e margem são indicadores da sua saúde, que podem até mostrar se seus objetivos estão sendo atingidos.

Contudo, algumas vezes receita e margem não estão necessariamente atreladas aos objetivos. Aliás, esses objetivos costumam mudar com o tempo. Por exemplo, no começo de qualquer rede social, o objetivo não é receita, mas sim obter a maior quantidade de usuários e garantir que estes retornem. Somente depois de ter uma base considerável de usuários ativos é que faz sentido pensar em como obter receita, quer seja dos usuários, quer seja de empresas interessadas em falar com eles. Por isso, é importante o gestor de produtos saber qual o objetivo da empresa e, periodicamente, validar se ele continua o mesmo.

O que os usuários querem?

Sabendo quais os objetivos da empresa, o gestor de produtos precisa pensar em novos produtos ou em evoluir produtos existentes para atender a esses objetivos. Para fazer isso, ele precisa conhecer:

- *Seus usuários*: é preciso conhecer os usuários ou possíveis usuários de seu produto, e quais problemas ou necessidades eles têm que seu produto pode resolver. Existem inúmeras ferramentas e métodos para conhecer o cliente. Alguns exemplos são pesquisa, entrevistas, análise de dados de acesso, teste A/B, protótipos, teste de usabilidade etc.

- *Contexto*: o contexto em que seus usuários estão inseridos no dia a dia e, especificamente, quando usam o produto. O contexto é o conjunto de condições físicas e de eventos que circundam seu usuário. Por exemplo, seu usuário acessar seu produto de um desktop ou de um smartphone faz parte do contexto.

- *Mercado e concorrentes*: tanto concorrentes diretos como indiretos. Os concorrentes diretos são aqueles que entregam o mesmo produto ou um produto similar. Já os indiretos são aqueles que de alguma forma substituem o seu produto. Por exemplo, suponha que você fez uma ferramenta de gestão de ordens de serviço para pequenos prestadores de serviços. Um dos principais concorrentes é o e-mail, pois esses pequenos prestadores podem usá-lo em vez de usar sua ferramenta. Ou ainda, podem usar telefone, papel e caneta!

Dá para fazer?

Muito bem, você já conhece os objetivos da empresa, entendeu o seu usuário e agora definiu como vai ser seu produto ou aquela nova funcionalidade que vai, ao mesmo tempo, atender os objetivos da empresa e ser útil para o seu usuário. Agora, você precisa saber se dá para fazer e qual seria o custo. Pode até ser que seja possível fazer, mas se demorar muitos meses e custar muito, pode ser que não valha a pena. Aí entra a conversa com o time que vai produzir o novo produto ou a nova funcionalidade; é o pessoal de UX e de desenvolvimento. São eles que dirão o custo, tanto em tempo quanto em dinheiro e, se ele não for aceitável, vocês terão de conversar para buscar alternativas.

TRADUZINDO TUDO ISSO EM UMA IMAGEM

Após ler o que é preciso levar em conta ao fazer um roadmap, dá para traduzir gestão de produtos em uma imagem, que já vimos no *Capítulo 02 – O que é gestão de produtos de software?*:

Para fazer seu roadmap, você precisa conhecer os objetivos da empresa, os usuários, suas necessidades e problemas, e o que dá para fazer. Com isso em mãos, você consegue construir seu roadmap. Mas não se esqueça de que os objetivos da empresa podem mudar, como também os problemas e necessidades de seus usuários, e o que é possível fazer. Por isso, é fundamental fazer revisões periódicas de seu roadmap para mantê-lo sempre em linha com esses 3 elementos.

ROADMAP = MOTIVAÇÃO + MÉTRICA

É comum ver roadmaps como uma lista de funcionalidades a serem entregues. Os exemplos que apresentei anteriormente mostram exa-

tamente isso, roadmap como uma lista de funcionalidades. Esse tipo de roadmap funciona bem quando você estiver apresentando-o para uma audiência externa à sua empresa, ou seja, para os usuários e para o mercado em geral.

Contudo, para usar o roadmap como ferramenta de comunicação interna – para mostrar tanto para o time que trabalha com você no produto como para outros times da empresa –, não é o ideal apresentar um roadmap como uma simples lista de funcionalidades, pois está incompleto. Faltam dois elementos muito importantes para explicar por que essas funcionalidades estão no roadmap nessa ordem de prioridade.

QUAL A MOTIVAÇÃO?

Como dito anteriormente, devemos levar em conta três aspectos ao fazer um roadmap:

- Objetivos estratégicos da empresa;
- Problemas e necessidades do cliente;
- Tecnologia disponível.

Esses são os insumos que o gestor de produtos tem de levar em conta ao construir um roadmap, ou seja, para definir quais funcionalidades colocar nele e em que ordem. Por esse motivo, para o roadmap que será usado para comunicação interna, além da funcionalidade propriamente dita, é importante constar a motivação que levou o gestor de produtos a colocá-la lá. Mais clientes? Mais receita? Menos

contatos de cliente pedindo suporte? Aumentar o engajamento? Etc. Ter a motivação da funcionalidade no roadmap vai ajudar o gestor de produtos a setar o contexto para o time que trabalhará no desenvolvimento dessa funcionalidade.

COMO MEDIR O RESULTADO?

Além da motivação, o roadmap deve também ter alguma indicação de como medir se o que motivou a funcionalidade está sendo atingido. Se a motivação for aumentar o número de clientes, como será medido esse aumento de clientes? Se for obter mais receita, como será medido esse aumento de receita? Se for menos contato de suporte, como será medida essa diminuição? Se for aumentar o engajamento, como isso será medido?

É importante definir como medir se uma determinada funcionalidade cumpriu a sua motivação, e efetivamente fazer essa medição. É muito provável que a forma de fazer essa medição deva ser considerada no desenvolvimento da funcionalidade, pois pode haver a necessidade de incluir código de programação específico para permitir essa medida.

EXEMPLIFICANDO

Para ilustrar o uso de motivação mais métrica na construção de um roadmap, usarei um exemplo da Locaweb. Temos um produto de *e-mail marketing* que serve para fazer envio de e-mails.

Quem usa e-mail marketing sabe que é preciso seguir algumas boas práticas para conseguir uma boa taxa de entrega de e-mails disparados. Primeiro, é preciso ter o consentimento do destinatário para poder enviar e-mails para ele. Em seguida, é fundamental ter uma frequência de envio. Se enviar uma mensagem uma única vez e nunca mais mandar, você não está criando um relacionamento com o destinatário. Além disso, é importante manter sua lista de destinatários limpa, ou seja, sempre que receber mensagens de erro ou reclamação de *spam* de alguém, esse endereço deve ser removido de sua lista.

Quem não seguir essas regras simples acabará tendo uma taxa baixa de entrega de e-mails, vai se decepcionar com o produto e, eventualmente, deixará de usar o e-mail marketing por achá-lo ineficaz.

Pensando nisso, decidimos na Locaweb criar o conceito de “reputação” na forma de um percentual que tem por objetivo dizer ao cliente quão bem ele está seguindo essas boas práticas. Com isso, conseguimos educá-los sobre as boas práticas de envio de e-mail marketing. Sendo assim, a funcionalidade “reputação” do e-mail marketing da Locaweb teve:

- **Motivação:** educar os clientes sobre boas práticas de envio de e-mail marketing para que eles obtivessem maior taxa de êxito em suas campanhas e, conseqüentemente, não cancelassem o serviço.

- **Métrica:** o resultado dessa funcionalidade está sendo medido de duas formas: quantidade de entregas (entregas no inbox, taxa de abertura de e-mails, taxa de reclamação de envio de spam) e diminuição de cancelamentos.

DETALHAMENTO VERSUS AUDIÊNCIA

Como dito no início deste capítulo, o roadmap do seu produto de software é a sua ferramenta para comunicar a visão de futuro que você tem para o seu produto. Só que sabemos que existem diferentes públicos que precisarão de diferentes níveis de detalhamento do seu roadmap. Em qual nível de detalhamento devemos entrar para cada público?

A figura dá uma sugestão de nível de detalhamento de acordo com cada audiência. O primeiro aspecto que você tem de pensar é se o público é interno ou externo. Como disse anteriormente, para públicos externos, você normalmente falará de funcionalidades. Já com públicos internos, seu foco deverá ser mais em funcionalidade e métrica do que na funcionalidade em si.

O segundo aspecto a se preocupar é com o nível de detalhamento que você precisa dar.

- **Público em geral:** para eles, o detalhamento não precisa ser grande. O ideal é falar em termos de funcionalidades e você pode só comentar as funcionalidades de destaque que estão mais próximas de serem entregues.
- **Clientes próximos:** para os clientes mais próximos, já é possível dar um pouco mais de detalhes, dando uma visão de médio prazo. Com esses clientes, é importante criar uma boa relação para que eles lhe ajudem a direcionar a visão de seu produto baseado nos problemas que eles têm e que seu produto se propõe a resolver.
- **Parceiros:** com eles, vale a pena entrar em mais detalhes, principalmente aqueles que ajudam o produto a chegar ao cliente. Por exemplo, o produto de hospedagem de sites da Locaweb tem as empresas que hospedam seus sites na Locaweb como clientes finais, mas tem também os parceiros que são os desenvolvedores e agências que desenvolvem esses sites. Aqui devemos continuar falando em roadmap como lista de funcionalidades.
- **CEOs, vice-presidentes e diretores:** nesse ponto, movemos para o público interno. Para esse público, tão ou mais importante do que saber quais funcionalidades vêm pela frente é saber a motivação de essas funcionalidades estarem no roadmap. Para eles, essa visão pode ser bem macro.

- **Vendas e marketing:** já para os times de marketing e vendas, é preciso entrar em um pouco mais de detalhes. Para vendas, os detalhes são importantes pois eles podem já ter identificado a demanda no mercado e, para marketing, a necessidade de mais detalhes vem do trabalho de divulgação que farão.
- **Marketing de produtos:** esse grupo é praticamente o quarto elemento daquele *core team* que comentei no *Capítulo 02 – O que é gestão de produtos de software?*. Eles devem participar muito próximo do desenvolvimento do produto, e devem conhecer toda a motivação e as métricas das funcionalidades que estão no roadmap.
- **SAC:** esse grupo talvez não precise saber tanto sobre a motivação das funcionalidades quanto os outros grupos internos, mas ele, sem dúvida, é o que precisa conhecer mais detalhes dessas funcionalidades futuras, pois eles falam diariamente com os seus clientes.
- **Engenharia e UX:** fazem parte do *core team* do produto. Precisam de todos os detalhes, motivações e métrica para poderem fazer seu trabalho.

ROADMAP OU BACKLOG?

Por fim, uma pergunta que ouço com frequência é: qual a diferença entre roadmap e backlog? Roadmap é o seu “mapa da estrada”, isto é, as coisas que você tem de fazer; já backlog é o termo usado em metodologias ágeis para o seu “registro de coisa a fazer”. Ou seja, esses termos são equivalentes e podem ser usados como sinônimos.

No próximo capítulo, vamos ver técnicas de priorização de roadmap.

CAPÍTULO 16

CRESCIMENTO: COMO PRIORIZAR O ROADMAP?

Essa é uma pergunta frequente de todo gestor de produtos. Quer seja um produto novo – que está sendo criado agora –, quer seja um produto já em pleno funcionamento – cheio de sugestões de clientes e usuários –, como fazer para priorizar, ou seja, para decidir o que fazer primeiro?

Existe uma quantidade grande de técnicas. Falarei sobre algumas delas aqui e, no final, vou dizer qual é a melhor. Só lhe peço que não vá agora até o final do capítulo para não estragar a surpresa... :-P

VALOR VERSUS CUSTO

Uma das formas mais simples de priorizar um roadmap é fazendo uma análise de todos os itens, procurando estimar: o valor (benefício) de cada um para o negócio e para os usuários, e o custo de implementar cada item. Com esses dados em mãos, é possível até fazer um gráfico com dois eixos e colocar cada um dos itens nesse gráfico, baseado no valor e no custo. A ideia é priorizar sempre o que tiver maior valor e menor custo, pois o benefício será obtido mais rapidamente.

ANÁLISE KANO

A análise Kano foi criada pelo Professor Noriaki Kano, da Universidade de Tokyo, para classificar os itens de um produto com base também em duas dimensões, na necessidade de um item e na satisfação que este causa no cliente. Com isso, é possível classificar os itens em três tipos: *mandatórios*, *OK* e *encantadores*.

Por exemplo, em um carro, roda é um item mandatório, pois não há carro sem roda. Teto-solar é um item OK, se seu cliente não o valoriza muito. Já ser muito silencioso é um item que encanta um cliente que aprecia essa característica. A recomendação é ter todos os mandatórios, alguns OK, mas não deixar de fora alguns encantadores para poder impressionar positivamente o cliente.

ÁRVORE DE PRODUTO

A ideia é mais ou menos parecida com a análise de Kano: classificar os itens do roadmap de acordo com as partes de uma árvore. Raízes são a infraestrutura; caule é o que dá o suporte; galhos são os diferentes caminhos que você pode colocar no seu produto; as folhas são as funcionalidades propriamente ditas; e as flores e os frutos são as funcionalidades que realmente vão encantar o cliente. Todo produto tem de ter raízes, caule e alguns galhos com suas respectivas folhas, mas é importante sempre incluir algumas flores e frutos para deixar seu produto encantador.

COMPRE SUAS FUNCIONALIDADES

Nessa técnica, você põe o seu cliente para trabalhar. Você mostra todos os itens que estão em seu roadmap e dá um valor para cada um deles baseado no quanto vai lhe custar fazer cada um. Em seguida, convide alguns clientes e diga para eles que eles têm X para gastar. Esse X tem de ser consideravelmente menor que a soma do valor de todos os seus itens.

Com esse X, cada cliente tem de “comprar” as funcionalidades que são mais importantes para ele e, como o dinheiro é limitado, ele é forçado a fazer escolhas do tipo “Será que pego essas duas funcionalidades, ou as troco por essa outra mais cara?”. É um exercício muito interessante e dá ao gestor de produtos um bom conhecimento sobre o comportamento do cliente.

USERVOICE

UserVoice é um sistema de sugestões que você pode colocar dentro do seu produto. Com isso, seus usuários poderão fazer sugestões sobre ele, e poderão também votar em sugestões feitas por outros usuários. Você ainda pode limitar a quantidade de votos, forçando-os a fazer escolhas como no método anterior.

A QUE VOCÊ LEMBRAR PRIMEIRO

Jason Fried, fundador da 37signals – que agora se chama Basecamp –, disse no seu livro *Getting Real* que, na sua empresa, a opção foi por priorizar baseado na lembrança. Eles recebem várias sugestões todos os dias, e decidiram simplesmente não anotar para não ter que depois gastar tempo contando e classificando-as.

Como sugestões aparecem todos os dias, eles as ouvem todos os dias. De tempos em tempos, eles se reúnem e discutem sobre as sugestões que se lembram, e essas são as que são tratadas e eventualmente priorizadas no roadmap dos produtos.

A MELHOR TÉCNICA DE PRIORIZAÇÃO

Como deu para perceber, existem várias maneiras de se priorizar um roadmap, todas elas bastante úteis. Ou seja, o que dá para concluir é que, se há tantas maneiras de se priorizar um roadmap e se todas as maneiras de se priorizar podem ser úteis, isso significa que **a priorização de um roadmap não é uma ciência exata.**

Temos um anseio de querer achar um método de priorização para justificar nossas escolhas. Entretanto, sempre que esse método falhar em um determinado item, que temos certeza de que seria melhor fazer antes (ou depois) do que o método diz, acabamos tentados a seguir essa nossa certeza, pondo abaixo o método.

Por isso, por mais que existam várias técnicas e métodos de priorização de roadmap, o melhor método é o **bom senso**. Ou seja, a capacidade do gestor de produtos de analisar as opções disponíveis e, usando-se de sua **empatia**, priorizar essas opções levando em conta os objetivos da empresa e as necessidades dos usuários.

O QUE FAZER COM PEDIDOS ESPECIAIS?

Ao longo da vida de seu produto, você certamente encontrará pedidos de novas funcionalidades vindas de clientes (ou da equipe comercial) que vêm acompanhados, de forma explícita ou não, de uma ameaça de que, se não fizermos essa funcionalidade, vamos perdê-los. Por outro lado, se você atender a essa demanda, em detrimento de todo o planejamento de roadmap feito, corre o risco de atrasar a entrega de funcionalidades importantes para o resto dos clientes e para os objetivos estratégicos do dono do produto de software.

O QUE FAZER?

A resposta a essa pergunta depende muito do seu produto e de sua base de clientes. Vou explicar melhor essa resposta com um exemplo.

Na Locaweb, temos dois produtos de e-mail marketing. Um deles se chama *E-mail Marketing Locaweb* (nome bem original, né?), e o outro é o *All In Mail*. Para dar um pouco da dimensão de cada produto e do tipo de cliente que cada um deles atende, aqui vão alguns números.

O PROBLEMA DE FALAR SIM PARA TUDO

Quando falamos em priorizar um roadmap, uma coisa que acaba acontecendo é que acabamos atendendo absolutamente todos os pedidos, de todo mundo. Ou seja, tudo é importante, pois tudo entra no roadmap, só que o que é menos importante fica para depois. A pessoa que pediu tem o alento de que “está no roadmap”, apesar de ter ficado lá para a frente, e ter boas chances de ser empurrado ainda mais para a frente se aparecer algum item mais importante.

POR QUE ISSO ACONTECE?

O gestor de produtos acabando dizendo “sim” para pedidos que recebe por vários motivos:

Precisa agradar a todos: esse é um problema bastante comum do gestor de produtos, a necessidade de agradar a todos. Quando o gestor de produtos vê que a resposta “vou colocar no backlog” acalma as pessoas que estão pedindo algo, ele começa a usar essa resposta de forma indiscriminada. Com isso, o roadmap ficará enorme e bastante complexo de gerenciar. Além disso, quando a pessoa que lhe pediu algo ver que “estar no backlog” não é garantia de que será feito logo, ele não ficará feliz... :-/

Os dados mostram que temos de fazer: cada vez mais vejo empresas focadas em tomar decisões exclusivamente baseadas em dados.

Com isso, em breve poderemos todos nós ser substituídos por algoritmos que tomarão as decisões, já que basta tomar as decisões baseadas nos dados. Acontece que os dados nem sempre estão certos. Pode haver dados insuficientes, ou mesmo dados errados. Outro problema de decisões baseadas em dados é o risco de se cair em um ótimo local. A sugestão é usar dados como um dos *inputs* para a tomada de decisão, mas não esquecer também dos dados qualitativos, sua intuição e sua capacidade de julgamento.

Mas é uma funcionalidade tão pequena: toda funcionalidade, por menor que seja, implicará em mais código e mais interação. Mais código significa complexidade de código, e quanto mais complexo o código, mais difícil de manter. Mais interação significa mais complexidade para seu usuário lidar, ou seja, chances de oferecer uma experiência ruim para ele. Nenhuma funcionalidade é tão pequena que não traga complexidade de código e de interação, por isso, pondere bem se essa complexidade adicional trará benefícios para seu usuário e para o dono do software.

O cliente pediu e, se não fizermos, ele vai embora: esse é o momento de tomar decisões difíceis. Se você quiser agradar todos os clientes, acabará agradando nenhum. Você precisa escolher quem é o cliente para quem você está fazendo seu produto. Um produto não pode ter mais do que duas ou três personas primárias. Aliás, o recomendado é ter apenas uma persona primária; ter duas ou três já dará um trabalho razoável para conseguir gerenciar. Se o que esse cliente pediu não atende sua persona primária, você terá de ter coragem de dizer *NÃO*.

Nós sempre podemos fazer dessa nova funcionalidade mais uma opção nas configurações: mais uma vez, estamos criando complexidade sem necessidade. Como já dito, toda funcionalidade implica em complexidade de código e de interação. Colocar todas as novas funcionalidades como opcionais a serem configuradas em uma tela de configuração fará dessa tela algo bem difícil para seu usuário.



Nossos competidores já têm: esse é um erro bem comum, basear-se nos competidores e não no seu cliente/usuário. Lembre-se: um gestor de produtos tem de se preocupar em fazer o software atender os objetivos da empresa dona do software, ao mesmo tempo que resolve um problema ou atende uma necessidade de seus clientes. É importante saber o que o competidor está fazendo, mas se o que ele estiver fazendo não tem a ver com o objetivo de sua empresa, nem

com os problemas ou necessidades de seus clientes, então você não precisa fazer igual.

O chefe quer: existem chefes e chefes. Se o seu chefe for extremamente antenado em relação aos clientes, é importante ouvi-lo. Ele será sempre um grande aliado. Agora, se seu chefe quer uma determinada funcionalidade simplesmente porque viu em algum competidor, ou alguém falou para ele que deveria fazer assim, você deve lembrá-lo dos objetivos da empresa e dos problemas e necessidades de seus clientes que você está tentando atender com o seu produto de software.

APRENDENDO A DIZER NÃO!

Apesar das razões vistas, para dizer sim a todo pedido de funcionalidade que um gestor de produtos recebe, ele tem de aprender a dizer **NÃO**.

Dizer **NÃO** pode parecer difícil, mas se você tiver muito claro os objetivos de seu produto – ou seja, quais objetivos da empresa o seu produto deve alcançar, quem é seu cliente principal e qual problema desse cliente você está procurando atender –, você terá os argumentos necessários para dizer **NÃO** a certas demandas.

Seja gentil com a pessoa que está trazendo a demanda e diga algo como:

COMO DIZER NÃO

Sua sugestão é interessante e consigo entender por que você a está pedindo. Contudo, deixe-me lhe mostrar o que já temos planejado em nosso roadmap, e quais são os objetivos de cada item que está nele. Por este motivo, não conseguiremos dar a devida atenção ao seu pedido nesse momento. Você me lembra de conversarmos novamente sobre isso no futuro?

Deixe para quem está pedindo a nova funcionalidade a responsabilidade de lembrar-se de ter essa conversa novamente no futuro. Se ele não se lembrar, é porque o pedido dele não era tão importante assim. Se ele lembrar, avalie novamente o pedido e, se necessário, diga **NÃO** novamente.

CAPÍTULO 17

**CRESCIMENTO:
SEJA UM “DATA GEEK”**

Além de conversar com seus usuários e ouvir seu feedback, uma forma obrigatória de conhecer seu produto e como seus usuários interagem com ele é por meio dos dados que seu produto gera. Para poder tirar proveito desses dados, você deverá se transformar em um *data geek*, uma pessoa que conhece profundamente os dados gerados pela sua aplicação e seus significados.

Toda nota de dólar americano tem escrita a frase *In God We Trust* (Em Deus Nós Confiamos). É tão grande a importância de usar os dados como fonte de informação para entendermos melhor as coisas, que acabou sendo criado o bordão *In Data We Trust*, ou seja, “Nos Dados Nós Confiamos”.

William Edwards Deming é um engenheiro, estatístico e professor americano bastante conhecido pelo seu trabalho no Japão, logo após a II Guerra Mundial, onde ele ensinou sobre gestão estatística da qualidade, e ajudou os japoneses a se transformarem em 10 anos na segunda maior economia do mundo. A ele, é creditada a frase:

“Em Deus nós acreditamos. Todos os outros devem trazer dados.”
—W. E. Deming

QUAIS DADOS SÃO IMPORTANTES?

Todos os dados são importantes, mas, dependendo do que você está querendo entender, uns são mais importantes do que outros. Conhecer seus dados é uma tarefa contínua, pois a cada novo conhecimento que você adquire, aparecem novas questões, que vão precisar de mais dados para serem respondidas.

Uma das primeiras informações que você vai querer conhecer é quantas visitas você recebe no site de seu produto. Para conhecer esses números, você pode usar algum relatório de estatísticas que o provedor de hospedagem oferece. Outra opção muito usada é o Google Analytics.

Com um relatório como esse, você consegue algumas informações importantes, tais como: quantidade de visitas, quantidade de visitantes únicos, quantidade páginas vistas (*pageviews*) e várias outras. Dependendo do sistema de estatísticas que você estiver usando, você também conseguirá ver: qual a primeira e a última página visitadas durante um acesso ao seu site; de onde (que país e cidade) vêm seus visitantes; se eles acessaram seu site partindo de uma campanha de Google AdWords, do Facebook, ou de alguma outra campanha online que você está fazendo; ou se acharam seu site de forma orgânica, ou seja, digitando diretamente o endereço; ou buscando por algo em algum sistema de busca. Vale lembrar de que é importante ter esses relatórios de acesso não só para o seu site como também para seu produto software.

Cuidado, pois esses sistemas de relatório de visitas normalmente dão uma quantidade muito grande de informação, e é fácil se perder nesse mar de dados.

Junto com a quantidade de visitas e acessos que seu site tem, outros dados importantes que você precisa conhecer de seu produto web são:

Quantidade de pessoas que ficam sabendo que seu produto existe: é possível diferenciar as formas como as pessoas ficam sabendo que seu produto existe classificando-as em duas categorias, as pagas e as gratuitas. As pagas são aquelas em que você tem de investir algum dinheiro como, por exemplo, Google AdWords, anúncios no Facebook, anúncio em sites de conteúdo (preferencialmente ligados ao tema de seu produto) e anúncios em revistas (também preferencialmente ligados ao tema de seu produto). Já as gratuitas são aquelas em que você investe tempo e trabalho para ficar conhecido como, por exemplo, criar conteúdo relevante sobre o tema do seu site, interagir em blogs sobre o tema do seu produto, facilitar que as pessoas recomendem seu produto para seus conhecidos etc. O retorno nesse caso é mais lento, mas tem a vantagem de não ter custo financeiro, só de tempo e trabalho.

Quantidade de cliques gerados pelos seus anúncios ou por outros meios: essa é uma informação um pouco mais difícil de obter, pois, dependendo de sua es-

estratégia para atrair pessoas para o seu site, essa informação não estará disponível. Os sistemas de anúncio online como Google AdWords, Facebook e anúncios em sites de conteúdo normalmente têm essa informação disponível, e o preço que cobrarão normalmente será baseado em um preço por clique.

Quantidade de visitantes únicos: são os novos visitantes que seu site recebe. É diferente da quantidade de visitas, pois uma mesma pessoa pode visitar seu site mais de uma vez até decidir comprar.

Quantidade de visitantes que se tornaram usuários: desses visitantes únicos, alguns vão se cadastrar para se tornar um usuário do seu sistema. Se você oferecer um período de experiência gratuito ou uma versão grátis sem prazo de expiração, esse número pode ser razoavelmente grande.

Quantidade de usuários que se tornaram clientes: findo o período de experiência, alguns de seus usuários vão querer se tornar um cliente, ou seja, vão querer pagar para usar o seu serviço. Se você estiver oferecendo uma versão gratuita de seu produto, sem prazo de expiração, você deverá ter uma versão paga que motive seus usuários a saírem da versão gratuita e a pagarem pelo uso de seu produto.

FUNIL DE CONVERSÃO

Napoleão Bonaparte, líder político e militar francês conhecido pelas Guerras Napoleônicas – por meio das quais foi responsável por estabelecer a hegemonia francesa sobre a maior parte da Europa no início do século XIX –, teve uma grande derrota em 1812, na Campanha da Rússia. Essa campanha foi uma gigantesca operação militar tentada pelos franceses e seus aliados, que teve grande impacto sobre o desenrolar das Guerras Napoleônicas, marcando o início do declínio do Primeiro Império Francês. Nessa campanha, Napoleão usou 580.000 combatentes. Desses, sobreviveram apenas 22.000 combatentes, o restante pereceu no caminho da França até Moscou devido às dificuldades encontradas nesse caminho (frio, chuva, rios etc.).

Essa imagem lembra muito um funil de conversão de um site, que pode ser feito com os dados que discutimos anteriormente. O funil de conversão nos mostra quantos potenciais clientes estamos perdendo no caminho entre atrair pessoas para o site até o ponto em que uma pessoa paga para se tornar seu cliente:

O funil apresenta várias oportunidades para você entender melhor como seus usuários interagem com seu produto. Cada pedaço do funil tem suas características específicas e pode ser alargado de diferentes formas. Concentre-se em um pedaço por vez e faça seus testes. Na pior das hipóteses, se o teste for ruim, você sempre pode voltar para a situação anterior. Para um *data geek*, o funil deve ser o primeiro foco de dados a se obter e analisar.

No próximo capítulo, vamos ver mais duas métricas que são a consequência natural do funil de conversão: o engajamento, que mostra como seu usuário utiliza seu produto; e o *churn*, que mostra quantos usuários deixam de usá-lo, ajudando a identificar por que isso acontece.

CAPÍTULO 18

CRESCIMENTO: ENGAJAMENTO E CHURN

Uma vez que você conseguiu trazer usuários (gratuitos ou pagos) para utilizarem seu produto, sua próxima preocupação tem de ser com o engajamento desses usuários, ou seja, será que eles estão conseguindo utilizar o produto? Será que eles estão conseguindo resolver o problema que o produto se propôs a resolver? Quantas vezes por dia (semana ou mês) seu produto é usado, e durante quanto tempo? Como ele é usado?

ENGAJAMENTO

É muito importante encontrar métricas para medir o engajamento. Por exemplo, em um produto de disparo de e-mail marketing, algumas métricas de engajamento e uso são quantas vezes por dia a pessoa acessa, quantas campanhas o usuário dispara por mês, quantas

Facebook? Instagram? Dá para se dizer que você está bastante engajado com essas aplicações?

Fomentar o engajamento deve ser uma das preocupações do gestor de produto. Em 2013, foi lançado um livro chamado *Hooked: How to Build Habit-Forming Products*, de Nir Eyal, no qual ele explica a teoria por trás desses produtos que acabam entrando no nosso cotidiano. É um ótimo livro para entender mais sobre esse tema.

Existem algumas estratégias que podem ajudá-lo a aumentar o engajamento e o uso de seu produto. Em inglês, essas técnicas são chamadas de técnicas de *lock-in*:

APIs – *Application Programming Interface* (ou, em português, Interface de Programação de Aplicativos): é uma maneira de dar acesso ao seu produto, aos dados que estão armazenados lá e às rotinas que ele executa para outros softwares. Quando alguém cria um novo software usando as APIs de seu produto, existe grande chance de aumentar o engajamento com ele.

Incentivo ao uso: você pode fazer promoções que incentivem o uso do seu produto. Por exemplo, se seu produto tem quota de uso, você pode aumentar essa quota à medida que o tempo passa.

Treinamento: ensinar seus usuários a tirar todo o potencial de seu produto é uma forma de engajá-los. Quanto mais eles aprenderem, mais rápido entenderão como ele pode-

rá ajudá-los. Não é necessário treinamento formal, em sala de aula. Você pode fazer treinamento virtual via *webinars*, ou até mesmo usar em seu produto aqueles *tooltips*, mostrando passo a passo como o usuário pode tirar proveito dele.

Historic data: os dados de utilização de seu produto, como logs e relatórios, podem ser uma ferramenta muito útil para o seu usuário. Ajude-o a tirar proveito desses dados. Por exemplo, você pode mandar resumos diários (ou semanais) via e-mail para ele, convidando-o a acessar seu produto para ver mais.

Integração com outros produtos: outra forma de aumentar o uso do produto é por meio de integrações com outros produtos que o seu cliente já usa. Por exemplo, uma loja virtual pode ter integração com *gateways* de pagamento, com sistemas de nota fiscal eletrônica e com sistemas de envio de pacotes via correio.

CHURN

Outra métrica muito importante é *churn*, ou seja, a quantidade de usuários e clientes que deixaram de ser usuários ou clientes. Eventualmente, alguns deles vão deixar de ser seu usuário ou cliente. É importante saber quantos são, e os motivos por que isso aconteceu, pois aqui você descobrirá muita informação para melhorar seu produto de software.

Esse número é muito importante em qualquer empresa que tem por modelo de negócio o uso contínuo, principalmente aqueles baseados em assinatura. Ele costuma ser medido como um percentual da seguinte forma:

CHURN

$$\textit{Churn mensal} = \frac{\text{quantidade de clientes que cancelou no mês}}{\text{total de clientes do último dia do mês.}}$$

Existe também o churn anual, que se calcula da mesma forma, só que dividindo a quantidade de clientes que cancelaram em um determinado ano pelo total de clientes do último dia do ano anterior.

É um número que contém muita informação mas, por ser somente um único número, ele deixa várias perguntas em aberto. Já vi discussões do tipo: “se o churn está em 20%, em cinco meses não teremos mais clientes, então não vale a pena investir em divulgar esse produto”. Uma frase como essa não tem muito sentido, pois mesmo que o churn se mantenha a 20% durante vários meses, até mesmo mais de 5 meses, a quantidade total de clientes pode continuar crescendo. Como? Basta ter uma quantidade maior de ativações do que de cancelamentos, e a divulgação ajuda bastante nisso. Veja o exemplo:

Apesar de em todos os meses o churn ser maior do que 20%, o crescimento no ano foi de 73 novos clientes.

Por que mesmo com um churn mensal alto é possível crescer?

São dois os motivos. O primeiro, que já comentei, é que é preciso ter uma entrada de clientes maior do que a quantidade que sai.

O segundo é que o churn varia de acordo com a idade do cliente. É comum ter casos nos quais o churn é alto no primeiro mês, pois o cliente não gostou do serviço e decidiu cancelar logo de cara. Ou no terceiro ou sexto mês, se sua cobrança for trimestral ou semestral. Algumas pessoas chamam de churn prematuro.

O churn prematuro, apesar de ser comum, é algo que pode e deve ser diminuído. Você faz isso:

Alinhando a expectativa que você criou no cliente por meio da sua divulgação do seu produto com o que ele vai encontrar quando passar a usá-lo.

Garantindo que as primeiras experiências de uso de seu produto sejam muito boas e que o seu cliente consiga atingir seus objetivos nessas primeiras experiências.

Mantendo seu produto útil para seu cliente ao longo dos meses e dos anos, investindo em entender seu cliente e seus problemas, e em atualizar seu produto para que ele continue resolvendo os problemas de seu cliente.

Os conceitos de churn e de engajamento andam de “mãos dadas”, pois, quanto mais engajado estiver um usuário, menores são as chances de ele cancelar o serviço. Assim, uma boa maneira de prever o churn de um determinado cliente é acompanhar seu engajamento. Por exemplo, se você lançou um produto de ensino a distância e acompanha a utilização desse produto, você provavelmente verá que a taxa de cancelamento é maior nos clientes que nunca assistiram aula. Reveja o tópico anterior sobre *lock-in* para ver táticas para aumentar o engajamento e diminuir o churn.

No próximo capítulo, vamos continuar com o tema de métricas, com foco nas métricas financeiras e de longo prazo. Vamos descobrir o conceito do churn negativo, o “Santo Graal” dos produtos com modelo de negócio baseado em assinatura.

CAPÍTULO 19

CRESCIMENTO: MÉTRICAS FINANCEIRAS

Quando falei sobre a importância de ser um *data geek* no capítulo anterior, expliquei o funil de conversão. Este é composto por um conjunto de dados que podemos chamar de curto prazo, pois em questão de poucos dias (ou mesmo horas) você já poderá perceber tendências, tirar conclusões, criar hipóteses e validá-las, quer seja conversando com seus usuários, ou fazendo experimentos e medindo os resultados.

Falei também sobre métricas de engajamento, que mostram como seus usuários interagem com seu produto e de churn, tanto de cliente como de receita. Isso o ajudará a entender quantos clientes deixam de ser seu cliente, por que eles decidiram deixar de ser, e qual o impacto disso em sua receita.

Além desses dados, existem outros que levam mais tempo para se consolidarem e começam a mostrar alguma tendência, mas que devem ser monitorados desde o primeiro mês de vida de seu produto. São as métricas financeiras, que podem ser classificadas em número globais (receita e custos) e números individuais: CAC (*Customer Acquisition Cost*), LT (*Lifetime*) e LTV (*Lifetime Value*).

NÚMEROS GLOBAIS: RECEITA E CUSTOS

A receita é o dinheiro que você recebe quando as pessoas usam seu produto. Como explicado no *Capítulo 13 – Inovação: como obter retor-*

no com seu produto de software?, há várias formas de se obter receita com seu produto. Essa receita será usada para pagar seus custos. Quando você conseguir pagar seus custos mensais, a sobra servirá para compensar o investimento dos meses em que a receita mensal não cobria os custos mensais.

Tanto receita quanto custos devem ser controlados mês a mês. É interessante você classificar seus custos em algumas categorias, para lhe ajudar a entender onde você está gastando mais e onde você pode economizar. Costumo classificar custos em 3 categorias:

Infraestrutura: são todos os custos necessários para manter o serviço operando. Nessa categoria, incluem custo de hospedagem do site e da aplicação, de registro de domínio, de ferramentas de e-mail marketing, de SEO, de teste A/B, de sistema de chat online para atender clientes etc. Normalmente, esses gastos são recorrentes e, por isso, requerem muita atenção sempre que você contratar um novo serviço desses.

Desenvolvimento: aqui entram todos os custos para desenvolver e implementar novas funcionalidades no seu site e na sua aplicação, incluindo programação, desenvolvimento de interface, design visual, design de logotipo etc.

Marketing: todo investimento que você faz para atrair clientes, como anúncio AdWords, anúncio Facebook, anúncios em sites, revistas, jornais e TV. Também devemos incluir aqui os custos com impressão e distribuição de panfletos, cupons promocionais, folders etc. No começo, você muito provavelmente precisará de investimento, mas

é importante também investir tempo em formas gratuitas para atrair clientes que, apesar de serem de longo prazo, ajudarão ao longo dos meses a economizar um pouco nos custos de marketing ou, se você decidir continuar investindo na divulgação paga, ajudará a aumentar sua receita.

Para você ter um produto rentável, é preciso ter receita mensal maior que os custos mensais. Simples assim. Porém, é mais simples falar do que fazer.

NÚMEROS INDIVIDUAIS: CAC, LT E LTV

A receita e o custo são indicadores globais da saúde de seu produto web. É importante você também ter indicadores individuais, ou seja, indicadores por cada cliente que você tiver. Existem três indicadores que são bem importantes:

CAC: é o *customer acquisition cost*, ou custo de adquirir um cliente. É a soma dos custos associados com descobrir e conseguir a atenção de potenciais clientes, e de levá-los para seu site, convertendo-os em um usuário de seu produto web e, posteriormente, em usuário pagante. Por exemplo, imagine que você só tenha investido em Google AdWords para atrair seus clientes e que, em um determinado mês, você tenha gastado R\$1.000,00 e conseguido 10 novos clientes nesse mês. Com isso, dividindo R\$1.000,00 por 10, você terá um CAC de R\$100,00. Ou seja, seu custo para adquirir cada cliente é de R\$100,00.

LT: é o *lifetime*, ou o tempo de vida de seu cliente. Isto é, quanto tempo, em média um cliente seu fica como seu cliente. Esse número só faz sentido quando você tem um fluxo de receita recorrente. Continuando o exemplo anterior, vamos imaginar que o LT dos clientes que você adquiriu é de 20 meses.

LTV: é o *lifetime value*, ou o valor de um cliente durante o tempo em que ele permanecer seu cliente. É a receita que esse cliente gera enquanto ele é seu cliente. Seguindo ainda no exemplo anterior, vamos imaginar que esse cliente gere uma receita mensal de R\$8,00. Com isso, o LTV será a multiplicação dos 15 meses pelos R\$8,00 por mês, que dá um LTV de R\$160,00.

Nessas definições é fácil ver que seu produto será rentável quanto maior for o LT e o LTV, e menor o CAC, e que você precisa ter LTV maior que CAC.

No exemplo que vimos, temos um LTV de R\$160,00 e um CAC de R\$100,00, o que mostra que temos uma situação rentável. É preciso acompanhar esses números de perto, mês a mês. Se em um determinado mês o LTV continuar sendo R\$160,00 e o CAC tiver aumentado para mais de R\$160,00, é preciso revisar os esforços de aquisição de cliente para ver se é possível reduzir esse custo. Também deve-se estudar formas de aumentar o LTV, aumentando o LT e/ou aumentando o valor mensal.

CHURN DE RECEITA

Uma métrica de receita bem interessante, que deriva da métrica de churn de clientes que vimos no capítulo anterior, é o conceito de churn de receita, bastante usado pelas empresas que têm um modelo de negócios baseado em assinaturas. Ele é bastante parecido com o conceito de churn de clientes, podendo ser calculado da seguinte forma:

CHURN

$$\textit{Churn mensal de receita} = \frac{\text{receita dos clientes que cancelaram no mês}}{\text{total de receita do mês.}}$$

Para o seu produto crescer, é preciso ter um aumento de nova receita mensal maior do que seu churn mensal de receita.

Existe uma diferença bem importante entre o churn de clientes e o de receita. O churn de clientes sempre será um número positivo, já o churn de receita poderá ser negativo. Como? Basta que o crescimento de receita de seus clientes existentes, sem contar a receita dos novos clientes daquele mês, seja maior do que o churn de receita dos clientes que estão cancelando o serviço naquele mês.

Veja na figura a seguir a diferença entre churn de clientes e churn de receita:

COMO ISSO É POSSÍVEL?

O churn negativo acontece quando seus clientes da base estão aumentando o uso de seu produto de software, e eles têm de pagar por isso. Por exemplo, quando eles fazem upgrade para um plano de serviço com mais recursos, quando eles contratam serviços adicionais, quando eles pagam por uso adicional, ou quando eles compram mais contas de acesso, se seu produto tiver precificação baseada em quantidade de contas de acesso.

Isso fará com que seu produto cresça mais do que a quantidade de novas vendas por mês, pois com o churn negativo, mesmo se você não tiver nenhuma venda nova, sua receita vai crescer devido ao aumento de receita dos clientes existentes. Por isso, o churn negativo é considerado o “Santo Graal” dos produtos com modelo de negócio baseado em assinatura.

No próximo capítulo, vamos ver a métrica que, na minha opinião, é a melhor de todas: a métrica da lealdade.

CAPÍTULO 20

CRESCIMENTO: A MÉTRICA DA LEALDADE

O resultado financeiro deve ser sempre utilizado como uma das métricas que indicam que a empresa está tendo sucesso, que está cumprindo com o seu propósito. Contudo, ela não deve ser olhada de forma isolada, pois existem a boa e a má receita – como já apontei no *Capítulo 06 – Cultura organizacional*.

Recapitulando, má receita é toda receita obtida às custas do detrimento do relacionamento com o cliente. Por exemplo, aquela empresa que dificulta a saída do cliente quando ele quer cancelar, ou quando vendem algo com preço acima do adequado, se aproveitando de que você precisa daquele item, como o custo absurdo da garrafa de água em um hotel, isso também é uma má receita.

A má receita pode até ajudar a empresa no curto prazo; entretanto, ao longo do tempo, os clientes vão se cansando desse seu comportamento e vão buscar alternativas. Ou seja, a má receita pode eventualmente matar sua empresa ao longo prazo.

O problema é que não é possível diferenciar a boa receita da má receita em um relatório financeiro. Tudo o que aparece ali são números de receita sem nenhuma informação sobre o nível de satisfação do cliente. Em função disso, torna-se necessário encontrar alguma métrica para medir essa satisfação, e correlacioná-la com os resultados financeiros para certificar-se da existência de uma relação de causalidade.

NPS – NET PROMOTER SCORE

Existem várias maneiras de se medir a satisfação de clientes. Eu gosto bastante de uma métrica chamada NPS (*Net Promoter Score*, ou Índice Líquido de Promotores). Essa métrica surgiu pela primeira vez em 2003, em um artigo da Harvard Business Review, escrito por Fred Reichheld, consultor da Bain & Company, renomada empresa de consultoria americana. Em 2006, Reichheld publicou um livro chamado *A pergunta definitiva* (em inglês, *The Ultimate Question*), que teve uma segunda edição publicada em 2011, com as lições aprendidas ao longo dos anos.

O NPS é calculado com base em uma única pergunta que você deve fazer ao seu cliente:

NPS

Em uma escala de 0 a 10, quão provavelmente você nos indicaria para um amigo ou colega?

Zero significa de jeito nenhum, e 10 significa com toda certeza. As pessoas que deram respostas de zero a 6 devem ser classificadas como detratores; as que deram 7 ou 8 devem ser classificadas como neutras; e as que deram 9 ou 10 são as promotores.

O cálculo do NPS é bastante simples, basta subtrair o percentual de detratores do percentual de promotores. Com isso, temos um núme-

ro que varia de -100% até 100%. Um número negativo significa que você tem mais detratores do que promotores, e um número positivo significa o contrário, ou seja, que você tem mais promotores do que detratores.

Claro que é melhor ter mais promotores do que detratores, principalmente se pensarmos no efeito de marketing boca a boca que tanto detratores quanto promotores podem gerar. Além de falar bem de sua empresa e seus serviços para os amigos, os promotores compram mais e são clientes por mais tempo. Isto é, têm *life time*, e sempre lhe darão ideias e sugestões bastante pertinentes ao seu negócio.

Várias empresas usam o NPS para medir a satisfação e a lealdade de seus clientes. Dentre elas estão Apple, Zappos, Rackspace, Microsoft, Intuit e Facebook.

Quando medir NPS? Uma vez por ano? Todo trimestre? Todo mês? Toda semana? A recomendação do livro é medir NPS todos os dias,

de forma contínua. Isso pode ser feito da seguinte forma: todos os dias você faz o pergunta do NPS para os clientes que estão fazendo aniversário de X meses ou X anos naquele dia.

Por exemplo, você pode perguntar para seus clientes quando fizerem 1 mês desde que se tornaram seus clientes, aí você pode pegar algum problema de *on-boarding*. Você também pode perguntar para seus clientes com 6 meses de casa e, a partir daí, para todo aniversário de 1 ano do cliente com sua empresa. Você, então, poderá tirar relatórios de NPS por tempo em que o cliente está com você, e assim você saberá se está tratando bem tanto o novo cliente quanto aquele de muito tempo de casa. Para ver como seu NPS está evoluindo, você tira a média móvel dos últimos 30 dias.

FECHANDO O LOOP: DA INFORMAÇÃO A AÇÃO

O grande problema é que um número não resolve muita coisa. Precisamos saber por onde melhorar, o que fazer para que os promotores continuem promovendo nossa empresa e nossos serviços, e o que motivou os detratores a nos darem a nota baixa.

Para obter essas informações, Fred Reichheld recomenda adicionar mais uma pergunta à pesquisa:

A SEGUNDA PERGUNTA

Qual a principal razão para você nos dar essa nota?

Pronto, com essa resposta você tem o que precisa para ir da informação para a ação. Ao ler os comentários dos promotores, você vai entender o que os motivou a dar uma nota alta e manter essas ações. Com os comentários das notas baixas, você terá o primeiro insumo para começar a agir. É muito importante conversar com os detratores para ouvir mais detalhes sobre suas insatisfações.

A partir do momento em que eles virem que você se importa, a percepção deles sobre sua empresa já começará a melhorar. Buscar entender a motivação dos detratores e agir para solucionar os problemas que geraram sua insatisfação é o único caminho seguro para aumentar o NPS e, conseqüentemente, a lealdade de seus clientes.

Talvez você esteja se perguntando quando deve ser dado esse feedback. A resposta é bem simples e direta: quanto mais rápido, melhor. No livro, Fred Reichheld recomenda que se entre em contato com os detratores em, no máximo, 48 horas. A rapidez é importante para mostrar que você lê e se importa com o feedback de seu cliente.

DICA: NPS INTERNO

Uma sugestão interessante do livro é a medição de NPS dos funcionários, ou seja, em uma escala de 0 a 10, o quanto esse funcionário indicaria a sua empresa para um amigo trabalhar. Esse questionário teria de ser anônimo para dar liberdade ao funcionário de poder comentar qualquer problema que ele tenha com a empresa.

Da mesma forma que o NPS dos clientes, você pode rodar esse NPS por safra, ou seja, 1 mês, 6 meses e a cada aniversário de um ano com a empresa. Aí você poderá descobrir se ele teve um bom processo de entrada na empresa, e se ele continua motivado após alguns anos.

Pronto, já vimos vários tipos de métricas, começamos pelo funil de conversão, falamos de engajamento, de churn, de métricas financeiras globais e individuais, de churn de receita e sobre o “Santo Graal” dos produtos de assinatura, o churn negativo. Agora, concluímos o tema falando da métrica da lealdade, o NPS. No próximo capítulo, farei minhas últimas considerações sobre o tema métricas.

CAPÍTULO 21

CRESCIMENTO: CONSIDERAÇÕES SOBRE MÉTRICAS

Ter métrica é bom. Métricas são essenciais para conhecer mais a fundo seu produto e para poder tomar decisões mais acertadas. Mas é preciso tomar cuidado. Métrica em excesso pode atrapalhar, pode ofuscar o seu conhecimento pleno do seu produto.

Em inglês, existem dois termos que podem explicar bem essa minha preocupação. O primeiro termo é *data-driven*. Quando se gerencia um produto de forma *data-driven*, você roda experimentos que geram dados que alimentam suas decisões sobre próximos experimentos. É simples assim, e parece ser uma excelente forma de se gerir um produto. Só que existem alguns problemas nessa abordagem:

Medição de dados:

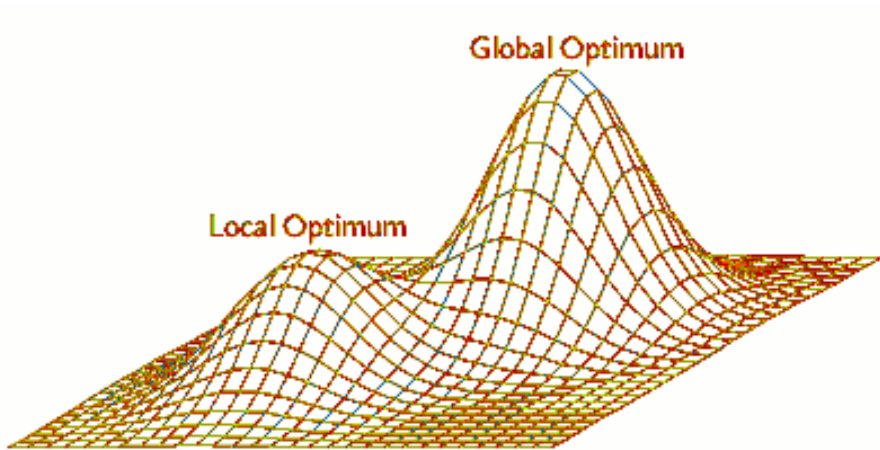
Existe uma tendência natural a medir o que é fácil de ser medido, e isso pode acontecer em detrimento de medir o que é mais relevante. Por exemplo, o que é mais fácil de medir: cliques em uma mensagem de uma campanha de e-mail marketing ou percepção do usuário (curiosidade, alegria, desdém etc.), ao receber essa mensagem de uma campanha de e-mail marketing?

Antes de sair tomando ações em cima dos dados existentes, é sempre bom perguntar se estes são os melhores dados para poder tomar a decisão.

Ótimo local:

Outro ponto de preocupação em relação às decisões baseadas em dados é o perigo de se ater às melhorias focadas em ótimos locais. Ou seja, você faz melhorias incrementais em seu produto, focando em melhorar determinada métrica, mas não percebe que, se você fi-

zer uma mudança mais radical, pode obter um aumento considerável nessa métrica, maior do que o maior aumento obtido com essas melhorias incrementais.



Qualidade dos dados:

Mesmo tendo dados, é preciso entender a qualidade dos dados, isto é, como eles são obtidos e processados antes de você recebê-los para analisar. Os resultados têm relevância estatística?

Você já deve ter ouvido falar em teste A/B, não? É um teste onde você monta duas de uma determinada página de um site. Em seguida, você divide o tráfego que vai para essas duas versões e começa a medir o comportamento das duas páginas em relação a um determinado objetivo como, por exemplo, cliques no botão "comprar". Após rodar esse teste por algum tempo, você vai saber qual das duas versões atinge mais o objetivo.

Experimente agora fazer um teste A/A, ou seja, faça um teste A/B onde as duas variações da sua página são idênticas. Depois de rodar esse teste, veja os resultados. Existem grandes chances de você encontrar uma das versões A de sua página atingindo o objetivo mais vezes do que a outra versão A. Como isso é possível? Variações estatísticas, sazonalidade e efeitos imprevisíveis.

Todo dado pode estar sujeito a variações estatísticas e, por isso, em dados quantitativos é sempre bom entender o tamanho da amostra e como ela foi coletada. A época em que a amostra de dados foi coletada também impacta na qualidade dos dados: hora do dia, dia da semana, dia do mês, período do ano, e assim por diante. Dados são uma foto com data e hora marcada, o mesmo dado coletado minutos depois pode dar informações diferentes.

Por fim, os efeitos imprevisíveis que podem afetar seus dados. A campanha presidencial de 2014 foi completamente mudada com a morte de um dos candidatos, o Eduardo Campos. Esse foi um evento imprevisível que mudou completamente a qualidade dos dados de pesquisa eleitoral.

Conhecimento qualitativo:

Nem só de dados quantitativos vive a gestão de produtos. Ao contrário, o gestor de produtos deve (como visto no *Capítulo 09 – Inovação: foco no problema*) usar diferentes ferramentas para aprender mais sobre seu usuário, o problema que o aflige, o contexto em que isso acontece e o que o motiva a ter esse problema resolvido. Muitas dessas ferramentas apresentam dados qualitativos, ou seja, dados de pesquisa exploratória, por meio da qual se está buscando entender

razões, motivações e opiniões. Esse tipo de entendimento é muito difícil, senão impossível, de se obter em análise de dados quantitativos.

Em função disso, em vez de *data-driven*, está cada vez mais claro que precisamos ser *data-informed*; em outras palavras, usar os dados como mais um insumo para as tomadas de decisão, e não como o único insumo. A experiência, a intuição, o julgamento e as informações qualitativas devem também ser levados em consideração, junto com as métricas, para aumentar a qualidade da tomada de decisão.

Acredito que um dos melhores exemplos que posso citar tem a ver com o produto de hospedagem de sites da Locaweb. Ao longo dos anos, de forma razoavelmente informada, mas contando sempre com bastante intuição, alteramos nossos planos de hospedagem para ter mais espaço em disco, transferência e quantidade de sites que poderiam ser hospedados em cada plano vendido. Em 2011, notamos que mais de 90% de nossos clientes escolhiam o plano mais básico, pois ele atendia a necessidade da maioria das pessoas que precisavam de um site.

Queríamos que os planos maiores tivessem uma participação maior nas vendas, mas, com os limites dos planos que tínhamos, não havia nenhuma motivação para os clientes os comprarem. Pensamos em mudar os planos para novas contratações, diminuindo os limites para incentivar os clientes a adquirirem planos maiores. Entretanto, como era uma mudança significativa e bastante sensível, trouxemos uma consultoria especializada em precificação, que nos ajudou a coletar e analisar vários dados, para sugerir qual a melhor mudança a ser feita.

Implementamos as mudanças sugeridas em 2012. Houve uma pequena variação na distribuição dos planos, mas a quantidade de planos contratados por mês não mudou. Aliás, chegou a cair um pouco, o que resultou em nenhuma alteração na quantidade de nova receita por mês. Ou seja, gastamos tempo e dinheiro coletando e analisando dados que nos fizeram tomar uma decisão que em nada mudou o resultado da empresa. Talvez se tivéssemos definido as mudanças de forma mais intuitiva, teríamos economizado dinheiro e saberíamos o resultado da decisão mais rapidamente.

UM POUCO SOBRE TESTES A/B

Existem duas ótimas ferramentas para fazer testes A/B, o Visual Website Optimizer (<http://visualwebsiteoptimizer.com>) e o Optimizely (<http://optimizely.com>). Usei o serviço Visual Website Optimizer, que oferece um mês grátis, para testar algumas hipóteses sobre a home do ContaCal:

Em menos de 30 minutos, consegui criar 4 variações e comecei a rodar o teste. Resolvi testar duas coisas. Uma era se a cor do botão “criar conta” faz diferença na quantidade de pessoas que clicam nele:

E a outra era se, mudando o vídeo explicativo para uma foto ilustrativa, aumentava ou diminuía a quantidade de pessoas que clicam no botão “criar conta”:

O resultado foi:

Ao ver esse resultado, fiquei com a impressão de que, se eu colocasse botão verde com a imagem da família saudável, eu iria aumentar mais ainda a conversão. Resolvi, então, fazer esse teste e o resultado foi:

Por isso, tome cuidado, as aparências enganam! Faça experiências antes de tirar conclusões!

ANALYSIS PARALYSIS

Por fim, além desses cuidados, é preciso tomar cuidado com o efeito *analysis paralysis*, ou seja, ficar o tempo todo analisando dados e não tomar nenhuma ação. Como bem ilustrado pela figura do xkcd.com (2014), a *analysis paralysis* pode custar bem caro:



Fonte: Efficiency,
por Randall Munroe, em
<https://xkcd.com/1445>

CONCLUINDO

As métricas não são a razão da existência de seu produto. Os usuários de seu produto, os problemas que ele resolve para esses usuários e os objetivos estratégicos da empresa que é dona do software são as razões de sua existência. Ou seja, métricas são um meio, e não o fim, o resultado e o objetivo. Por isso, use-as mais como uma das ferramentas para ajudá-lo a guiar seu produto na direção certa.

Com isso, fechamos o tema sobre a fase de crescimento do ciclo de vida de um produto de software. Entendemos como lidar com feedback dos clientes, e o que é e como priorizar um roadmap. Também vimos sobre os mais variados tipos de métricas, incluindo funil de conversão, engajamento, churn, métricas financeiras globais e individuais, churn de receita e o negativo, NPS, a métrica da lealdade, e vimos aqui algumas considerações sobre métricas.

No próximo capítulo, entenderemos melhor a próxima fase do ciclo de vida de um produto de software: a maturidade.

CAPÍTULO 22

MATURIDADE

Depois da fase da inovação, supondo que seu produto conseguiu passar pelo abismo, ele chegou ao crescimento. Só que vai chegar uma hora em que seu produto entrará na fase da maturidade. Isso é inevitável, basta revermos a curva S de adoção de tecnologia em vários casos práticos:

POR QUE ISSO ACONTECE?

Existem duas razões para um produto chegar à fase de maturidade de seu ciclo de vida:

Exaustão do mercado:

Esse é o caso da TV na figura anterior. O mercado de TV amadureceu uns 30 anos depois que a televisão foi inventada, ou seja, 100% do mercado endereçável, que podia comprar uma TV, já tinha uma. Qual

foi a solução que os seus fabricantes encontraram para sair da maturidade? Na verdade, eles não saíram da maturidade e, eventualmente, entraram na última fase: o fim de vida. A solução que eles encontraram foi criar novos produtos de TV que pudessem passar por todo o ciclo de vida novamente.

Hoje, os ciclos de vida de tecnologia da televisão são bem mais rápidos, e a tecnologia seguinte já começa sua fase de crescimento antes mesmo de a tecnologia anterior sair dessa mesma fase. Atualmente, os fabricantes de TV já não esperam mais a exaustão do mercado para criar uma nova TV. Eles já usam a inovação como forma de forçar o produto atual a ir para a maturidade e poder gozar da fase de crescimento do novo produto.

Inovação disruptiva:

É uma inovação que ajuda a criar um novo mercado e uma nova percepção de valor, e que eventualmente muda por completo um mercado existente e sua percepção de valor (ao longo de alguns anos ou décadas), tornando obsoleta a tecnologia antiga (https://en.wikipedia.org/wiki/Disruptive_innovation). Essa é a base do livro *The Innovator's Dilemma*, obrigatório para todas as pessoas que trabalham com tecnologia, do Prof. Clayton Christensen, professor de Harvard e consultor sobre inovação.

Inovação disruptiva é o que aconteceu com as câmeras com filme, com a chegada da câmera digital; ou com os telefones celulares que só faziam ligação de voz, com a chegada dos *smartphones* enciclopédias tradicionais, com a criação da Wikipédia; e com os CDs e DVDs com o lançamento de serviços de download e *streaming* de

áudio e vídeo. Às vezes, a inovação disruptiva pode ser criada pela mesma indústria, como uma estratégia para se defender do amadurecimento do mercado. É o caso da indústria de TVs que estão lançando as SmartTVs, e da indústria de telefone celulares, com o lançamento dos *smartphones*.

Maturidade programada:

Existe ainda uma situação em que o dono do produto, em vez de esperar a maturidade acontecer, gerencia-a de forma ativa. É o caso das fabricantes de TVs que mal lançaram as TVs de plasma, logo lançaram as de LCD e, em seguida, as de LED, não dando tempo de o mercado saturar e antecipando a maturidade com a nova versão. Isso também acontece em produto de software, principalmente em software instalado, como Windows, MySQL, Nginx, Asterisk e vários outros.

Antevendo a obsolescência de seu produto, seu gestor já planeja o lançamento das novas versões e o processo de fim de vida das versões anteriores. Esse processo costuma ser bem lento e, em vários casos, traumático. É o caso recente do Windows Server 2003, lançado em abril de 2003. Ele entrou em estado de maturidade programada assim que a versão seguinte do Windows Server, o Windows Server 2008, foi lançado em 2008. A partir de 2008, ele viveu na fase de maturidade; em 2010, ele deixou de ser vendido; 14 de julho de 2015 foi a data definida pela Microsoft para não dar mais nenhum suporte ao produto, ou seja, decretou o fim de vida do produto.

Para produtos de software online, a maturidade programada não faz tanto sentido, pois o produto é atualizado frequentemente e o usuário normalmente não sofre com as atualizações. Claro que existem

exceções, mesmo para produtos online. Existem casos em que o time de desenvolvimento de produto opta por reescrever o produto, por algum determinado motivo, e decide por lançar uma nova versão. Nesse caso, é importante entender que você estará colocando o produto atual em estado de maturidade programada e deverá gerenciá-lo como tal. Isto é, deverá planejar todo o ciclo de maturidade e de fim de vida dessa versão, incluindo pensar o que fazer com os clientes existentes da versão que entrará em maturidade programada e em como migrá-los para a versão mais nova de seu produto. Isso é algo que deve ser discutido como um dos fatores que podem influenciar na decisão do time de desenvolvimento de produto sobre partir ou não para desenvolver uma nova versão de seu produto de software.

QUANDO ACONTECE?

Não é muito difícil perceber quando se está chegando à maturidade. Caso se trate de uma maturidade programada, você mesmo vai definir quando ela vai acontecer. Se não for programada, basta olhar o crescimento de seu produto e notar que ele está crescendo mais devagar. Outro ponto que acontece é que o crescimento; quando houver, será sempre orgânico, ou seja, não adianta investir em divulgação e marketing que você não terá novas vendas.

É um pouco perturbador entrar na fase de maturidade do produto, principalmente se não for uma maturidade programada. Nessa hora, começam frases nostálgicas do tipo “nos bons tempos...”, mas o importante é não desanimar. Antes de mais nada, você precisa **ter**

certeza de que seu produto realmente chegou à maturidade, ou se há algum outro motivo para a desaceleração do crescimento. Para ter certeza de que seu produto chegou à maturidade, você deve se fazer algumas perguntas:

Ainda estamos com foco no problema?

Como vimos no *Capítulo 09 – Inovação: foco no problema*, todo produto existe para resolver um problema, e é fundamental que o gestor do produto nunca perca isso de vista. Quando um produto entra na fase de crescimento, é comum o foco mudar do problema a ser resolvido para como acelerar esse crescimento. Quando isso acontece, perde-se a principal ferramenta que pode ajudar a acelerar o crescimento, que é o foco no problema. Quando o gestor de produtos e o time de desenvolvimento de produtos perdem esse foco, as chances de o crescimento do produto começar a desacelerar são bem grandes, dando a impressão de que se chegou à maturidade.

Durante anos, tivemos um forte crescimento no produto de Hospedagem de Sites da Locaweb. Como contei no capítulo anterior, em 2012 trouxemos uma consultoria especializada em precificação, que após muita pesquisa e análise, nos sugeriu mudanças em nossos planos de hospedagem para otimizar e aumentar nossa receita e nosso lucro. As mudanças foram nos limites dos planos e nos serviços inclusos. A sugestão deles foi diminuir o número de sites por plano (já eram raros os casos de clientes que hospedavam mais de um site por plano) e, para compensar, dar serviços inclusos como o WebChat, para atendimento via chat. Como eu já disse, essas mudanças acabaram não afetando em nada a receita e o lucro, e acabamos perdendo um monte de tempo e dinheiro em análises para fazer essas mudanças inócu-

as. Mas isso não foi o pior. Com as mudanças feitas, nosso produto de Hospedagem de Sites ficou mais longe de atender ao problema de nossos clientes. Eles passaram a considerar que o produto tinha menos da funcionalidade principal, e que nós colocamos os serviços adicionais que encareciam o produto. Nossos clientes acabaram encontrando soluções melhores no mercado. Isso afetou nosso produto, que desacelerou consideravelmente em 2013.

Em função disso, decidimos, de forma mais empírica, fazer o que chamamos de reempacotamento do produto de Hospedagem de Sites, mudando os limites baseados não só em dados como também em nosso conhecimento dos clientes e de seus problemas, e em todos os nossos 15 anos de experiência com o produto. Com a mudança que fizemos, que nos custou muito menos tempo e dinheiro na sua fase de planejamento, ele voltou a crescer como no ritmo antigo. Por isso, é importante o gestor de produtos e o seu time de desenvolvimento nunca perder o foco no problema que ele resolve, e garantir que ele está resolvendo-o da melhor forma possível.

Esse exemplo ilustra bem o que comentei no capítulo anterior, quando falei que, além das métricas, é preciso usar outras informações, como a experiência, a intuição, o julgamento e as informações qualitativas, para tomar decisões relacionadas ao produto. A consultoria nos fez olhar exclusivamente para métricas, fazendo-nos desconsiderar todo o nosso conhecimento empírico com o produto.

O mercado está desacelerando?

Esse é um segundo ponto bem importante a ser analisado quando se observa uma desaceleração do crescimento. Como estão seus concorrentes, eles também estão desacelerando? E o mercado como um

todo, está desacelerando? Existe alguma questão da economia do país ou de seu estado que podem estar impactando o seu mercado? Todas essas questões devem ser avaliadas quando você começa a perceber uma desaceleração do crescimento do seu produto.

Como dá para ver na curva de registro de domínios, a partir de meados 2013, parece haver uma desaceleração na quantidade de domínios .br registrados por ano. Ainda é um pouco cedo para ter certeza de que há uma desaceleração acontecendo, mas essa informação já dá indícios. Isso nos ofuscou um pouco quando analisamos a desaceleração do produto de Hospedagem de Sites da Locaweb. Por isso, mesmo avaliando o mercado e chegando à conclusão de que ele está desacelerando, você e o time de desenvolvimento do produto nunca podem perder o foco no problema que ele resolve e em garantir que ele está resolvendo-o da melhor forma possível. Mesmo com mercado

desacelerando, é possível ter crescimento se seu produto for uma excelente solução para o problema de um conjunto de pessoas.

É preciso tomar cuidado para não confundir desaceleração circunstancial com desaceleração devido à maturidade. Como se pode ver na imagem da curva S na vida real, algumas tecnologias como o telefone, o automóvel e a eletricidade tiveram desacelerações circunstanciais antes de chegarem à maturidade. Cenários de crise econômica podem impor uma desaceleração circunstancial.

Existe substituto disruptivo?

Esse é o terceiro ponto a ser considerado quando um produto começa a apresentar sinais de desaceleração do crescimento. Existe no mercado alguma categoria de produto diferente que substitui o seu? No caso que contei da desaceleração do crescimento da Hospedagem de Sites da Locaweb, existia também uma categoria de produto com forte potencial para substituir o nosso; eram os de criação de site, como o Weebly, Wix e SitePX.

Desde os primeiros serviços de criação de site, sempre estivemos cientes desse potencial e, por isso, lançamos nosso produto de criação de sites para concorrer diretamente com a Hospedagem de Sites, como solução para o problema das pessoas que queriam ter um. Existe também um outro problema que a Hospedagem de Sites resolve, que é a necessidade de hospedar e rodar aplicações web com banco de dados. Para resolver isso, existem hoje no mercado soluções de hospedagem de aplicação, como o Heroku e o Google App Engine. Pensando nisso, também lançamos a nossa solução de hospedagem de aplicações, chamada Jelastic Cloud.

Vale notar que disrupção não existe somente na tecnologia usada pelo produto. Modelo de negócio também é uma forma de disrupção. A Locaweb lançou sua primeira versão de Cloud em 2008. Optamos por manter o mesmo modelo de negócio da Hospedagem de Sites, incluindo espaço em disco e transferência no preço, e cobrando um valor mensal pelo pacote. Em 2009, a AWS (*Amazon Web Services*) começou a ficar bastante conhecida por seus serviços de Cloud Computing, que eram muito similares à tecnologia que a Locaweb oferecia. A grande inovação vinha no modelo de negócio. A Amazon optou por cobrar os serviços por hora, e em fazer a cobrança de espaço em disco e transferência de forma separada. Apesar de ser uma forma mais complexa de se cobrar, tinha o atrativo de cobrar apenas pelo que era usado. Quando lançamos o Jelastic Cloud da Locaweb, optamos também por esse modelo de negócio.

Resumindo, antes de aceitar que seu produto realmente chegou à fase de maturidade, é muito importante ter certeza disso. No exemplo que dei da Locaweb, fica claro ainda havia espaço para crescimento se nos focássemos no problema do cliente.

Supondo que não seja uma maturidade programada, que você tenha feito a análise mostrada e que, mesmo voltando o foco do gestor de produto e do time de desenvolvimento para o problema que seu produto resolve, você não conseguiu retomar o crescimento: esta é hora de aceitar a situação, e começar a diminuir o investimento em desenvolvimento e em marketing do seu produto.

Você pode também decidir mover seus esforços de desenvolvimento e marketing para outro produto de seu portfólio. Veremos mais so-

bre gestão de portfólio de produtos e como alocar investimentos em diferentes produtos desse portfólio na *Parte IV – Gestão de portfólio de produtos* deste livro.

Outro ponto importante é a necessidade de diminuir custos de operação. Você não pode ter um produto de software na fase de maturidade que custe muito para operar. Custo de operação vem normalmente na forma de trabalho humano, ou seja, custo para corrigir problemas e custos de atendimento ao cliente. O ideal é que, desde a fase de crescimento, seu produto tenha custo de operação bem pequeno, com baixos índices de problemas e custos de atendimento.

Para fazer um produto assim, o time de desenvolvimento de produto deve zelar pela qualidade do produto que está sendo posto em operação, tanto do ponto de vista de qualidade técnica e preocupação do time de engenharia quanto do ponto de vista de facilidade de uso e preocupação do time de UX.

Uma vez que você faça esses desinvestimentos no desenvolvimento e marketing, provavelmente aolocá-los para outro produto de seu portfólio, e diminuir os custos de operação, seu produto pode ter ainda uma longa vida na maturidade. Tudo vai depender se o produto de software ainda dá retorno para o seu dono enquanto continua a resolver o problema dos clientes existentes.

Essa situação pode se sustentar por meses ou anos. Mesmo não havendo mais investimento em desenvolvimento, é preciso manter um mínimo de investimento em manutenção, mínimo este, que deve fazer sentido dentro da análise de retorno de seu produto. Esse mí-

nimo de investimento em manutenção deve ser coordenado pelo gestor do produto. Ele deve avaliar sua situação e decidir investir esforço somente quando houver necessidade de correções de bugs e de atualizações.

Vale lembrar de que, ao manter a manutenção do produto dessa forma, você estará periodicamente mudando o foco do time de desenvolvimento, que agora está com foco em outro produto. Por isso, essas manutenções devem ser mantidas ao mínimo. É provável que a necessidade de manutenção aumente com o tempo, devido ao envelhecimento do código que compõe o software e, ao longo do tempo, isso pode fazer com que o custo de manter esse produto seja maior que o retorno.

PRÓXIMA FASE

Se eventualmente você chegar ao ponto em que os custos de manter o produto não compensam mais o retorno que ele dá, muito provavelmente estará na hora de você se preparar para sua próxima fase: o fim de vida, tema do nosso próximo capítulo.

CAPÍTULO 23

FIM DE VIDA

O seu produto de software pode ter chegado aqui por três caminhos diferentes:

- 1) O crescimento do seu produto desacelerou, e você fez todas as análises e testes descritos no capítulo anterior para ter certeza de que ele, de fato, chegou à fase da maturidade.
- 2) Ou, então, você lançou o seu produto e não está conseguindo cruzar o abismo que existe no fim da inovação, a primeira fase do ciclo de vida de seu produto de software.
- 3) Ou ainda, você optou por uma maturidade programada, planejando o fim de vida da versão atual de seu produto de software em função de uma nova, que foi desenvolvida e lançada.

Independente do caminho seguido – maturidade não programada, não cruzar o abismo ou maturidade programada –, você eventualmente chegou ao que é conhecido como o fim de vida do produto. Em inglês, usa-se um termo mais elegante, o *sunset* do produto.

Assim como todas as fases anteriores, essa também terá de ser gerenciada. Engana-se o gestor de produto que acredita que, quando um produto chega ao fim de vida, termina-se o trabalho de gestão de produtos. Ao contrário, essa fase requer tanta atenção quanto as anteriores.

O primeiro passo é reconhecer que chegou ao fim de vida, e um ótimo indicador é avaliar se o custo de manter o produto é maior que o retorno que ele dá. Se isso acontecer, existe um forte indicativo de que ele está entrando nessa fase.

A decisão pelo fim de vida de um produto não é científica. Os números ajudam bastante nessa decisão; porém, como explicado no *Capítulo 21 – Crescimento: considerações sobre métricas*, além das métricas, é preciso usar outras informações como a experiência, a intuição, o julgamento e as informações qualitativas para se tomar a decisão de fim de vida do produto.

Cada um dos três possíveis caminhos que seu produto pode ter feito para chegar ao fim de vida vai trazer consigo fatos e considerações específicas para essa tomada de decisão. Normalmente, essa é uma tomada de decisão feita em comitê, com as pessoas que trabalham com o produto e os executivos da empresa.

Caso seu produto tenha de fato entrado no fim de vida, o gestor de produto precisará gerenciar essa fase. A decisão e seu gerenciamento

to dependem do caminho seguido por ele para chegar a ela, e é isso que vamos ver nas próximas seções.

FIM DE VIDA POR MATURIDADE NÃO PROGRAMADA

Esse é o pior dos casos pois, além de não planejado – como é o caso do fim de vida após a maturidade programada –, essa é uma situação difícil de identificar. Antes de decretar que é o fim de vida do produto, é preciso ter certeza de que isso não é só temporário. O mais comum nesses casos é deixar o produto dormente por um tempo.

O que significa dormente? Significa não investir em seu desenvolvimento e marketing, e observar como ele se comporta. Ele parou de crescer? Ou ele cresce muito devagar? Faz sentido parar de vendê-lo? Ou pode-se deixar o produto sendo comercializado por mais algum tempo? Após algum tempo dormente, deve-se reavaliar se vale investir novamente nele.

Esse foi o caso do produto PABX Virtual da Locaweb, que decidimos deixar dormente por uns 3 anos, sem nenhum investimento de marketing e de desenvolvimento, para podermos investir em outros produtos. Mesmo sem nenhum investimento, a quantidade de clientes não diminuiu. Inclusive, em certos momentos, houve até um crescimento. Esse crescimento, quando acontecia, era bem pequeno, mas não deixava de ser um crescimento. Por esse motivo, decidimos

voltar a investir no produto. Outro exemplo, já comentado no capítulo anterior, foi o caso do produto Hospedagem de Sites da Locaweb que, por nos focarmos demais em métricas e desconsiderarmos nosso conhecimento empírico, acabou entrando em uma situação de não crescimento que, felizmente, pôde ser corrigido.

Por isso quero dizer novamente para se ter muito cuidado para não tomar uma decisão precipitada, pois o fim de vida por maturidade não programada é extremamente difícil de ser identificada.

Se você realmente tiver certeza de que seu produto chegou mesmo ao fim de vida, você precisará gerenciar essa fase. A primeira decisão que você deverá tomar, junto com o comitê que mencionei – composto por pessoas que trabalham com o produto e executivos da empresa –, é se você vai descontinuar o produto ou deixá-lo dormente. Essa decisão será baseada no retorno que ele estiver dando para a empresa e no custo de mantê-lo.

Se o produto ainda der um retorno considerável, é provável que vocês decidam por mantê-lo e façam esforços para reduzir o custo de manutenção. O gestor de produtos será o responsável por coordenar os esforços de redução de custo de manutenção, e por avaliar se esses custos estão menores do que o retorno que o produto dá.

Por outro lado, se o retorno desse produto for pequeno, ou seus custos operacionais forem muito grandes e não passíveis de redução, é provável que vocês optem por descontinuí-lo. Nesse caso, o gestor deverá coordenar todos os passos necessários para descontinuar o produto, que incluem:

- Definir a data de fim de vendas e executar;
- Verificar se há um substituto no portfólio de produtos da empresa ou no mercado;
- Se houver substituto, definir como um cliente poderá migrar para este;
- Preparar a comunicação com os clientes, definindo prazos para o fim do produto;
- Testar essa comunicação com um grupo pequeno de clientes para avaliar impacto e fazer os ajustes necessários;
- Executar a comunicação com toda a base de clientes;
- Acompanhar e agir pontualmente, quando necessário.

FIM DE VIDA POR NÃO CRUZAR O ABISMO

Se seu produto entrar no fim de vida por não cruzar o abismo, apesar de essa não ser uma situação desejada pelo time de desenvolvimento, é uma situação importante que precisa ser identificada rapidamente, e o gestor de produto tem papel fundamental para detectar isso.

Essa situação é mais fácil de ser detectada, pois o produto não cresce. Ele conquista alguns poucos clientes, os *early adopters*, e depois para de crescer. Nesse ponto, você deve avaliar junto com o gestor de marketing se a estratégia de comunicação do produto está sendo eficiente e se está atingindo as pessoas que têm o problema que o seu produto se propõe a resolver.

É importante também entender com os possíveis clientes se o produto não só resolve o problema como também como eles estão dispostos a remunerar-lhe por esse produto. Sempre é possível fazer ajustes no produto e no seu modelo de comercialização para adequá-lo aos seus clientes.

Contudo, se mesmo após avaliar essas questões e tentar fazer ajustes no produto e/ou no seu modelo de comercialização, você não estiver conseguindo fazê-lo crescer, é hora de decretar seu fim de vida. Quanto antes você chegar a essa decisão, melhor, para não perder tempo e investimento em um produto que não vai vingiar.

Nesse caso, como o produto tem uma base pequena de clientes, as opções de mantê-lo dormente por ter uma receita considerável não faz sentido. Sendo assim, a única decisão viável é descontinuar o produto. Como no caso do fim de vida por maturidade não programada, aqui também o gestor de produtos deverá coordenar todos os passos necessários para descontinuí-lo, que são os mesmo descritos na seção anterior.

FIM DE VIDA POR MATURIDADE PROGRAMADA

Essa é a melhor das três opções para o dono do software, mas é a que dará mais trabalho para o gestor do produto. No caso de software instalado, isso acontece quando novas versões são lançadas. Para

software online, isso ocorre quando o time de desenvolvimento opta por reescrever o produto por algum motivo, e decide lançar uma nova versão. Essa decisão por reescrever um produto online deve ser muito bem pensada, pois o tempo gasto em reescrever seu produto é tempo não utilizado em sua evolução do ponto de vista de quem o usa.

Pode acontecer que seu produto não tenha sido devidamente planejado do ponto de vista técnico e agora você esteja em um ponto em que: ou reescreve, ou o produto não poderá mais evoluir. Entretanto, cair em uma situação como essa deveria ser exceção, e não parte normal do ciclo de vida de um produto online.

Resumindo: *para software instalado, o fim de vida por maturidade programada é intrínseco ao modelo, enquanto para software online, o fim de vida por maturidade programada deve ser sempre exceção.*

Como esse tipo de fim de vida pode ser programado, o gestor de produtos deve, em conjunto com UX e engenharia de produtos, desenhar como será essa fase, sempre visando ao mínimo de impacto para os clientes. Na Locaweb, no nosso produto de e-mail marketing, chegamos a um ponto em que tivemos de mudar o banco de dados. Usávamos MongoDB que, por problemas de projeto, não estava sendo capaz de aguentar o crescimento do produto; por isso, decidimos por mudar a base de dados para PostgreSQL.

Desenhamos essa reescrita do produto de tal forma para que o cliente não fosse impactado negativamente quando a nova versão com o PostgreSQL fosse implementada. Nada mudou na forma dos clientes usarem o produto. O único impacto percebido por eles foi que o produto estava com performance melhor, o que era esperado por nós.

Este é o ponto mais importante a ser pensado em relação ao fim de vida programado: o que vai acontecer com os clientes que estão na versão atual, que passará a ser a versão “velha”, quando lançarmos a nova? Isso tem de ser pensado com muito cuidado, e o foco tem de ser sempre em um impacto mínimo para o cliente.

É lógico que, quando a nova versão tem interface e interação diferentes da versão antiga, o cliente será impactado. É muito importante fazer testes com alguns deles para medir o impacto antes de forçar a mudança para toda a base de clientes. Em algumas situações, você pode até decidir por manter a versão antiga por algum tempo, dada a dificuldade de trazer os clientes para a versão nova. Mas, não se esqueça, o foco é sempre fazer seu cliente sofrer o mínimo possível.

CONCLUINDO

Vimos em detalhes nos capítulos anteriores todo o ciclo de vida de um produto de software, passando por todas as suas fases: a inovação, o crescimento, a maturidade – que pode ser programada ou não –, e o fim de vida – que pode vir após a maturidade ou quando o produto não cruza o abismo que separa a fase da inovação da fase

do crescimento. Para concluir esse tema sobre ciclo de vida, vamos falar agora sobre a diferença entre startup e gestão de produtos de software; diferença esta que tem muito a ver com o ciclo de vida de um produto de software.

CAPÍTULO 24

QUAL A DIFERENÇA
ENTRE STARTUP E
GESTÃO DE PRODUTOS?

Desde 2011, tem se falado bastante sobre startups, com muitos livros, palestras, grupos de discussão, eventos focados no assunto, que tem muito a ver com desenvolvimento de produto de software. Tem tanto a ver que algumas pessoas tratam os dois temas como um só. Contudo, existe uma diferença entre eles.

STARTUP

Como vimos no *Capítulo 07 – Como é o ciclo de vida de um produto de software?*, o ciclo de vida de um produto de software tem quatro fases: a inovação, o crescimento, a maturidade e o fim.

Tudo o que se discute sobre startups está focado na primeira fase do ciclo de vida de um produto de software, a fase de inovação. Em alguns casos, chega-se até a falar um pouco sobre crescimento e sobre como cruzar o abismo, mas o foco principal é sempre em como desenvolver o produto de software.

Como comentei no *Capítulo 13 – Inovação: próximos passos*, existe muita literatura boa a respeito do tema startup. Não deixe de pesquisar!

GESTÃO DE PRODUTOS DE SOFTWARE

Já a gestão de produtos de software envolve muito mais que somente a fase de inovação. A gestão de produtos de software cobre todo o ciclo de vida do produto. Além disso, como vimos no *Capítulo 02 – O que é gestão de produtos de software?*, a gestão é uma função do *core team* de desenvolvimento, que também tem engenheiros e designers de UX.

É importante entender quais os papéis e responsabilidades dessa função e como ela se relaciona com as outras áreas da empresa. Também

faz parte do conhecimento de gestão de produtos de software entender a diferença entre foco e diversificação de portfólio de produtos e, se a empresa optar pela diversificação, como gerenciar esse portfólio.

Por isso, como mostrado na figura, a gestão de produtos de software cobre muito mais temas do que quando falamos sobre o assunto startup. Para quem quiser ler mais livros sobre gestão de produtos de software, recomendo:

- **Inspired: how to create products customers love**, de Marty Cagan, ex-VP de gestão de produtos no eBay. É o manual de gestão de produtos de tecnologia, onde o autor explica todos os principais conceitos relacionados à gestão de produtos. Para as pessoas que vêm trabalhar comigo no meu time de coordenação de produtos da Locaweb, esse livro é leitura obrigatória.
- **Agile product management with Scrum: creating products that customers love**, de Roman Pichler, consultor de implementação de metodologias ágeis. Nesse livro, ele mostra como *product owners* podem criar produtos de sucesso usando metodologias ágeis.
- **42 rules of product management (2nd edition): learn the rules of product management from leading experts around the world**. Esse livro é uma coletânea de 42 ótimos artigos escritos por especialistas em gestão de produtos de software.

CONCLUINDO

Chegamos ao fim da Parte II sobre o ciclo de vida de um produto de software. Agora já conhecemos quais são as fases desse ciclo de vida, o que acontece em cada uma delas e o papel do gestor de produto em cada uma. Nos capítulos que seguem, vamos entender como é o relacionamento do gestor de produtos com as outras áreas da empresa.

PARTE III
RELACIONAMENTO
COM AS OUTRAS
ÁREAS

Como o gestor de produtos deve se relacionar com as diferentes áreas da empresa? Engenharia, UX, marketing de produtos, gestão de projetos, operações, vendas, jurídico, financeiro, atendimento, recursos humanos e administrativo.

Relembrando o que falei no *Capítulo 04 – Principais características de um gestor de produtos*, a característica mais importante que todo gestor de produto precisa ter é a **empatia**, ou seja, a capacidade de se colocar no lugar de outra pessoa para compreender seus anseios, motivações, necessidades e problemas. Essa característica deve ser usada não somente com o cliente do produto como também com as pessoas das diferentes áreas da empresa.

Como dito, o gestor de produtos deve entender o impacto que seu produto tem no trabalho do jurídico: será que aumentaram os problemas jurídicos devido a alguma funcionalidade nova no produto? E quanto à equipe de vendas, suporte, operações, financeiro, marketing, será que o novo produto complicou muito a vida deles? E em relação ao time do produto, os engenheiros e os analistas de UX que fazem parte do *core team* do produto, como as suas decisões em relação ao produto, qual o impacto em suas funções?

É o que veremos nos próximos capítulos!

CAPÍTULO 25

ENGENHARIA DE PRODUTOS E GESTÃO DE PRODUTOS

Vou começar falando sobre a relação entre engenharia de produto e gestão de produtos.

DEFINIÇÃO

Engenharia de produto é responsável por desenvolver o produto e mantê-lo operando. Com a visão de negócios trazida pelo gestor de produto, mais o desenho de solução feito pelo pessoal de UX – baseado no entendimento da necessidade ou do problema do cliente –, a engenharia de produto “constrói” o produto.

Para construí-lo, ela deve não só fazer a programação como também definir sua arquitetura técnica. Ou seja, qual é a infraestrutura onde ele rodará, qual a linguagem de programação mais adequada, qual o banco de dados mais apropriado, como garantir os requisitos não funcionais desse produto (velocidade de resposta, disponibilidade, escalabilidade etc.). Deve também validar com o gestor de produtos se o custo dessa infraestrutura cabe no seu modelo de negócio.

O fato de o gestor de produto ser responsável por definir o produto a ser feito pode dar a impressão de que a engenharia de produtos está subordinada à gestão de produtos. Entretanto, essa visão é incorreta e, se as áreas se comportarem dessa forma, a chance de fracasso do produto aumenta, pois quem se sente subordinado tem menos comprometimento com o resultado.

Engenharia de produtos, gestão de produtos e UX são um time, não há relação subordinação entre nenhum desses grupos. Eles devem funcionar como parceiros e sócios, cada um com sua especialidade e responsabilidade, colaborando para produzir o melhor produto possível.

INOVAÇÃO

No diagrama anterior, a engenharia de produtos é quem traz a tecnologia disponível. Como expliquei no *Capítulo 08 – Inovação: o que é inovação?*, inovar não é simplesmente conhecer a última tecnologia. É preciso conhecer não só esta como também todas as tecnologias disponíveis, e saber como usá-las. Esse é o papel da engenharia de produtos. A oportunidade e o potencial de produzir uma inovação estão em conhecer as tecnologias disponíveis e saber como utilizá-las para resolver um problema, ou atender a uma necessidade de um grupo de pessoas.

DICAS PRÁTICAS PARA GESTORES DE PRODUTO

Para facilitar o dia a dia com o time de engenharia de produto, aqui vão algumas dicas práticas:

Não se meta na solução técnica, a não ser que você tenha conquistado o direito de se meter. Um gestor de produto deve ter algum conhecimento técnico sobre seu produto, mas essa não é sua área de especialidade. Os especialistas estão na equipe de engenharia de produto. Por isso, evite apresentar soluções técnicas para o time de engenharia a não ser que esse time dê abertura para isso.

Leve engenheiros nas conversas com clientes e usuários.

Como parte do seu cotidiano como gestor de produto, você deve conversar sempre com seus clientes e usuários para entender como seu produto resolve os seus problemas ou atende às suas necessidades, e como você pode fazê-lo ficar ainda melhor. Os engenheiros gostam muito de ir a essas conversas. Para eles, é uma experiência muito bacana ver pessoas reais usando um software que eles desenvolveram. Eles ficarão ainda mais engajados na missão de fazer um produto melhor.

Sempre tome decisões baseadas em dados.

Quer seja dados de acesso e de uso do produto, ou dados de conversas com clientes e usuários, use dados para tomar suas decisões e apresente-os para o time. Isso dará maior consistência aos itens que você vai colocar no roadmap do seu produto.

Aprenda a tirar o excesso. Busque sempre o produto mínimo ou a funcionalidade mínima, ou seja, procure implementar o mínimo possível para atingir o seu objetivo de negócio.

Esteja presente. É fundamental estar junto com o time de engenharia de produto ou, pelo menos, acessível ao time o máximo de tempo possível. Durante o desenvolvimento do produto, inevitavelmente surgirão dúvidas e, se você não estiver presente, ou o time para, ou vai implementar como eles acham que deve ser, o que pode ser diferente do que você havia planejado.

Dê feedback para o time sobre o produto. Você, como gestor de produto, sabe como o seu produto está indo, quantos usuários tem, o que esses usuários estão achando dele, como esse produto está ajudando a empresa a atingir seus objetivos etc. Conte periodicamente sobre isso para o time de engenharia do produto. Como isso, você estará dando um contexto e um propósito para o time.

Negocie as histórias de reescrita e de manutenção. O time de engenharia, se for um time bom – antenado na evolução das boas práticas de engenharia de software e acompanhando sempre o que há de novo no desenvolvimento de sistemas –, sempre vai descobrir formas melhores de implementar cada pedaço do produto. Se depender do time de engenharia, o backlog do produto só terá histórias de melhoria técnica. Isso é bom, mostra que você está trabalhando com um ótimo time. Contudo, você deve usar o item anterior para dar o contexto do produto para o time, ou seja, mostrar que há determinados objetivos definidos para o produto, que vocês têm de atingir e que, por isso, vocês devem sempre colocar novas funcionalidades nele. Deve existir um balanço entre histórias de manutenção e reescrita e de novas funcionalidades. Já li em vários lugares algo como: *defina X% do tempo para histórias de manutenção e reescrita*. Eu não gosto de dar essa sugestão, pois acredito que **cada momento do produto requer um equilíbrio diferente**, e cabe ao gestor de produto e ao seu time de engenharia conversar e definir de comum acordo esse equilíbrio apropriado a cada fase. Vale lembrar da importância

de encontrar um bom equilíbrio, pois isso evitará o famoso débito técnico que, à medida que cresce, fará com que o time de engenharia de produto fique cada vez mais lento.

AH, E TEM MAIS UM PONTO!

Alguns engenheiros de produto acabam se tornando ótimos gestores. É importante ser capaz de perceber quando um engenheiro está procurando “outra coisa pra fazer”, e dar a ele essa opção de carreira.

Na Locaweb, temos (e tivemos) ótimos gestores de produto que eram engenheiros de produto. Em alguns casos, acabaram se tornando gestores do próprio produto em que era engenheiro. Por outro lado, existem alguns engenheiros de produto que experimentaram a gestão de produtos e não gostaram.

Para quem está acostumado a medir sua produtividade por funcionalidades entregues, pode ser estranho no começo perder essa referência de produtividade. Como já vimos, o dia a dia de um gestor de produto é composto de muita conversa com as várias pessoas envolvidas nele, e esse monte de conversa pode “assustar” o engenheiro que está acostumado a trabalhar focado no desenvolvimento de funcionalidade. Por isso, é preciso deixar a porta aberta para ele poder voltar a ser engenheiro de produto, sem nenhum prejuízo para a sua carreira.

CAPÍTULO 26

UX E GESTÃO DE PRODUTOS

A próxima área que quero comentar é a área de UX que, junto com engenharia e gestão de produtos, forma o *core team* de desenvolvimento de produtos.

O QUE É UX?

EXPERIÊNCIA DO USUÁRIO

A “experiência do usuário” engloba todos os aspectos da interação do usuário final com a empresa, seus serviços e seus produtos.

Fonte: Nielsen Norman Group The definition of user experience. <http://www.nngroup.com/articles/definition-user-experience>. 2015.

Essa é uma definição bastante simples, mas ela realmente contempla todos os aspectos de UX. Quem trabalha com software tem a tendência de achar que UX é a interface do software, tanto do ponto de vista do planejamento da interação do usuário com o software quanto do ponto de vista visual. Sim, UX é isso, mas não é só isso. UX também se preocupa com o que esse software causa para quem o usa.

De acordo com a ISO 9241-210, intitulada “Ergonomia da interação humano-sistema”, que fornece orientações sobre a interação humano-sistema durante todo o ciclo de vida de sistemas interativos:

EXPERIÊNCIA DO USUÁRIO

Experiência do usuário são “as percepções de uma pessoa e as respostas que resultam do uso ou uso antecipado de um produto, sistema ou serviço.” De acordo com a definição ISO, experiência do usuário inclui todas as emoções dos usuários, crenças, preferências, percepções, respostas físicas e psicológicas, comportamentos e realizações que ocorrem antes, durante e após o uso. A ISO também lista três fatores que influenciam a experiência do usuário: o sistema, o usuário e o contexto de uso.

Fonte: https://en.wikipedia.org/wiki/User_experience

UX NO PROCESSO DE DESENVOLVIMENTO DE PRODUTOS

No processo de desenvolvimento de produtos, UX é o responsável por entender a fundo o usuário e o problema que se deseja resolver para esse usuário. A figura a seguir ilustra bem o papel de UX bem como o de engenharia de produtos e o de gestão de produtos, no processo de desenvolvimento.

Na fase de concepção do produto, UX tem um papel central. O primeiro passo é entender o usuário, a pessoa que vai usar o produto de software. Para isso, UX utiliza uma ferramenta chamada **persona**, que representa um grupo de usuários com padrões de comportamento, atitudes e motivações similares em termos de decisão de compra, de uso de tecnologias ou produtos, de estilo de vida etc.

As **personas** são usadas para:

- Conhecer e entender clientes e usuários de produtos e serviços;
- Trazer o usuário para o foco do projeto;
- Tornar as decisões de *design* mais humanas e menos abstratas.

A figura a seguir mostra como construir uma persona:

O primeiro passo é definir um nome e algumas características dessa persona. Isso ajuda nas conversas sobre o produto. No nome, é bacana já dar uma dica da característica principal. Por exemplo, *Maria, a descolada* ou *Michelle, a ocupada*.

Se você estiver fazendo um produto para *Michelle, a ocupada* e quiser inserir uma nova funcionalidade, vale a pergunta: “Será que *Michelle, a ocupada* vai conseguir usar essa funcionalidade? Ela a achará útil o suficiente para achar tempo para aprender a usá-la?”. Além do nome e características básicas, é importante também descrever seus comportamentos e seus problemas. Os exemplos a seguir deixarão mais claro o conceito de **persona**.

Maria, a descolada é uma das personas que usamos na Locaweb. Dados os diferentes produtos que temos em nosso portfólio, acabamos construindo oito personas. Contudo, para cada produto de nosso portfólio, temos uns 3 ou, no máximo, 4 personas, sendo que uma destas é a persona primária do produto, ou seja, para quem todas as suas decisões de seu produto de software são tomadas.

Em seguida, baseado em muita pesquisa para entender a fundo os problemas e necessidades dessas personas, o UX constrói o protótipo que começará a materializar a ideia do produto. Esse protótipo deverá ser usado nas conversas com clientes e usuários, para validar se a ideia faz sentido, como também nas conversas com o time de engenharia de produto, para que eles já possam entender o que vem pela frente e se há tecnologia disponível para fazer.

É muito diferente ouvir a descrição de um produto ou funcionalidade com, por exemplo: “Você vai ver uma tela com login e senha, e um botão de entrar. Também verá um link caso você tenha esquecido sua senha”; e ver a tela onde essa funcionalidade acontece. A primeira versão pode ser um desenho em papel ou em quadro branco:

Page Setup

Margins Paper Size Paper Source Layout

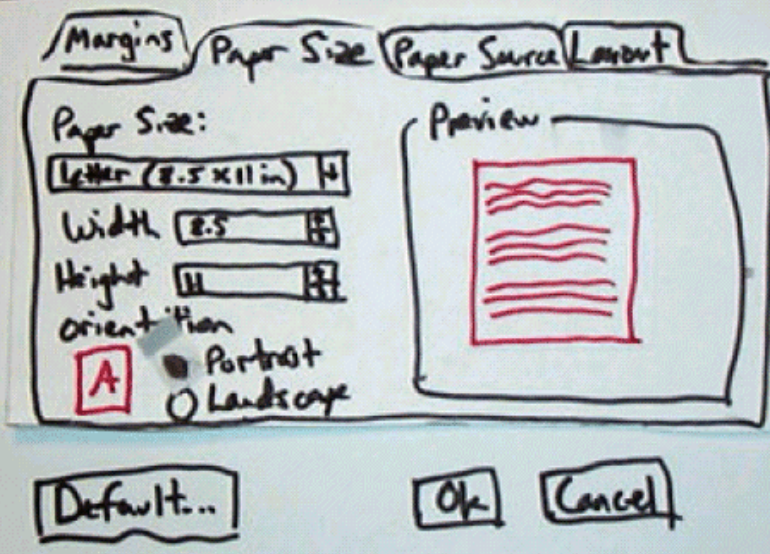
Paper Size:
Letter (8.5 x 11 in)

Width

Height

Orientation
 Portrait
 Landscape

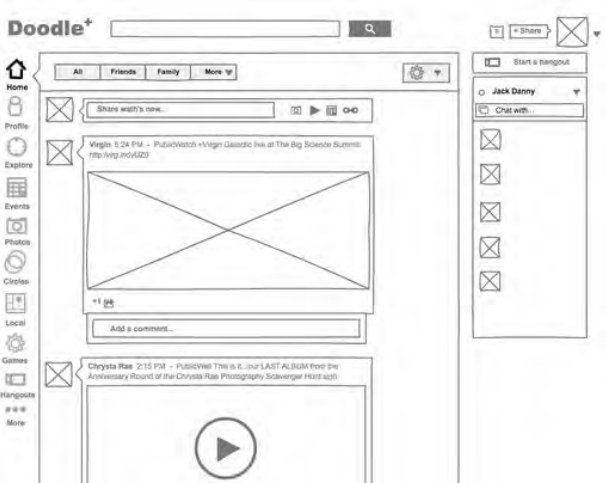
Preview



O protótipo também é muito importante como ferramenta de comunicação com a engenharia de produtos. Mostrando o protótipo, fica mais fácil para os engenheiros avaliarem a dificuldade de implementação de cada funcionalidade.

Quando fiz o ContaCal, como minhas habilidades de desenho não são as melhores, optei por usar o PowerPoint como minha ferramenta para desenhar o protótipo:

Depois de acordadas as funcionalidades básicas, esse protótipo pode ser feito em computador. A primeira versão do protótipo feita no computador é normalmente só em preto e branco, só com os contornos dos elementos. Essa versão é muitas vezes chamada de *wireframe*:



O próximo passo é o time de UX preparar um protótipo (ou wireframe) navegável, ou seja, um que já tenha o comportamento de interação para que vocês (gestor de produtos e analista de UX) possam mostrar para clientes e usuários, para que eles possam testar e interagir. Em seguida, o designer visual de UX começa a colocar cor e forma nessas telas, para transformá-las na versão que os usuários de fato verão.

Com um protótipo em mãos, é possível fazer testes de usabilidade para identificar problemas de usabilidade ou validar soluções de interface. Para fazer esse teste, convidam-se usuários para executar determinadas tarefas em seu protótipo. Durante o teste, é possível observar o comportamento do usuário ao realizar um conjunto de tarefas propostas, além de identificar as dificuldades e os motivos para algumas de suas decisões, ao interagir com o produto. O usuário é incentivado a verbalizar suas ações, opiniões e sensações para que, dessa forma, conheçamos o modelo mental criado por ele.

Esse protótipo servirá de guia para o time de engenharia desenvolver o produto. É a especificação do produto; mas lembre-se de que ela não é escrita em pedra. Por esse motivo, o time de UX não entrega o protótipo para você e para o time de engenharia, e depois vai fazer outra coisa. A pessoa de UX que desenhou o protótipo deve continuar junto com o gestor de produto e com o time de engenharia enquanto ele estiver sendo desenvolvido para ajustar o protótipo (se necessário), descobrir melhorias e aproximar o time de engenharia das necessidades do usuário.

O QUE MAIS O PESSOAL DE UX FAZ?

Outro ponto em que o pessoal de UX participa ativamente é na definição e acompanhamento das métricas. Como disse no *Capítulo 15 – Crescimento: o que é um roadmap?*, todo item de um roadmap deve ter sua motivação e sua métrica. O designer de UX deve saber muito bem qual é essa motivação quando ele vai desenhar o protótipo, e deve trabalhar junto com o gestor de produto e com o time de engenharia para definir qual(is) métrica(s) acompanhar para medir se a motivação foi atingida.

O profissional de UX também conversa muito com o usuário. Ele será seu principal companheiro nessas conversas e interações com o usuário, e vai estar sempre focado no que é melhor para o usuário. Seu papel será sempre colocar o resto do contexto, ou seja, além de ser bom para o usuário, precisa ser bom para a empresa que é dona do software que vocês estão desenvolvendo.

Ele também deve criar padrões de interação. Não dá para, toda vez que for desenvolvida uma nova funcionalidade, desenhar do zero. É importante ter um padrão pronto. Na Locaweb, criamos o Locaweb Style, com nossos padrões de interação, e o publicamos como *open source* (<http://open-source.locaweb.com.br/locawebstyle>).

QUAL É A RELAÇÃO ENTRE UX E GESTÃO DE PRODUTOS?

Como comentei no capítulo anterior, o fato de o gestor de produto ser responsável por definir o produto a ser feito, pode dar a impressão de que a engenharia de produtos está subordinada à gestão. Essa é uma visão incorreta e, se as áreas entenderem como certa, as chances de o produto fracassar aumentam, pois pessoas que se sentem subordinadas têm menor comprometimento com o resultado. A engenharia e a gestão de produtos como também os profissionais UX têm de ser vistos como um time, sem relação de subordinação e funcionando como parceiros que colaboram para produzir o melhor produto possível.

O UX sempre “puxará a sardinha” para o lado do usuário, e sempre vai querer fazer o que for melhor para este tanto do ponto de vista de função (interação) quanto do ponto de vista de forma (visual). E isso é bom! Se o time de UX for um time bom, ele sempre estará “puxando essa sardinha”, da mesma forma que o time de engenharia estará “puxando a sardinha” para o lado técnico, propondo sempre o trabalho de reescrita. Caberá ao gestor de produtos defender os interesses da empresa, que é dona do software, e encontrar e propor um balanço entre essas necessidades.

AH, E TEM MAIS UM PONTO!

Assim como os engenheiros de produto, alguns designers de UX acabam se tornando ótimos gestores de produto. É importante ser capaz de perceber quando um designer está procurando “outra coisa pra fazer” e dar a ele essa opção de carreira. Na Locaweb temos ótimos gestores de produto que eram designer de UX. Em alguns casos acabaram se tornando gestor de produto do próprio produto dos quais eram responsáveis por UX. Por outro lado, também existem alguns designers de UX que experimentam a gestão de produtos e não gostam.

Essa situação é menos comum de acontecer pois, diferentemente do engenheiro de produto, o designer de UX normalmente está acostumado a conversar bastante com o usuário e com as outras áreas relacionadas ao produto, por isso o trabalho de gestão de produtos não lhe é tão estranho. Mesmo assim, é preciso deixar a porta aberta para ele poder voltar a ser designer de UX, sem nenhum prejuízo para a sua carreira.

CAPÍTULO 27

QUAL A DIFERENÇA
ENTRE GESTÃO DE
MARKETING DE
PRODUTOS E GESTÃO
DE PRODUTOS?

Fora da indústria de tecnologia e de software, é comum encontrar a função de marketing de produtos como uma das funções do gestor de produto. Ou seja, a gestão de produtos é vista como parte do marketing. Tanto é assim, que o termo gestão de produtos na Wikipédia diz:

GESTÃO DE PRODUTOS

Gestão de produtos é uma função do ciclo de vida organizacional de uma empresa que lida com planejar, orçar e divulgar um produto ou conjunto de produtos em todas as fases do ciclo de vida desse produto. Essa função é composta por duas funções, desenvolvimento de produto e marketing de produto, que são esforços diferentes mas complementares, com o objetivo de maximizar receita, participação de mercado e margens. O gestor de produto é normalmente responsável por analisar as condições do mercado e por definir as funcionalidades de um produto.

Fonte: https://en.wikipedia.org/wiki/Product_management

Na indústria de tecnologia e de software, principalmente na indústria de produtos de software, essas funções, apesar de continuarem muito ligadas, são normalmente separadas. A gestão de produtos é responsável pela definição e desenvolvimento do produto, junto com os times de UX e de engenharia, enquanto a gestão de marketing de produtos é responsável por contar ao mundo sobre esse produto.

GESTÃO DE PRODUTOS

Para poder definir em detalhes o produto ou funcionalidade que vai ser desenvolvido, o gestor de produtos precisa conhecer muito bem os usuários de seu produto, seus problemas e necessidades. Com essa informação em mãos, ele pode dizer em detalhes como ele deve ser. Com a ajuda de um designer de interação, eles podem desenhar como ele será, e como vai resolver o problema ou atender à necessidade dos usuários.

Dá para imaginar que, ao longo do processo de definir e implementar a definição, muitas decisões terão de ser tomadas pelo gestor de produto, que é responsável por tomá-las pensando sempre no que é melhor para o seu usuário, fazendo-o atingir seus objetivos com aquele produto, e para a empresa que é dona do produto, que também tem seus objetivos. O produto é uma forma de a empresa atingir esses objetivos, como foi explicado no *Capítulo 15 – Crescimento: o que é um roadmap?*.

GESTÃO DE MARKETING DE PRODUTOS

Para poder contar ao mundo sobre o produto, o gestor de marketing de produto precisa também conhecer os seus usuários, seus problemas e necessidades. Esse conhecimento é fundamental para ajudar o gestor de marketing a definir o conteúdo e a forma de sua

mensagem sobre o produto. Outro elemento importante para ajudar a definir essa mensagem é o conhecimento do mercado onde o produto está inserido, ou seja, quais são os competidores que têm produtos similares, e quem tem produtos que, mesmo não sendo similar ao seu, podem substituí-lo. Quais os diferenciais do seu produto em relação a esses outros? Quais os diferenciais deles em relação ao seu produto? Por que seus usuários devem escolher o seu produto, e não os concorrentes? Essas são as questões com que o gestor de marketing de produto tem de se preocupar.

GESTÃO DE MARKETING DE PRODUTOS E GESTÃO DE PRODUTOS

Apesar de as funções serem bem distintas, existem muitas áreas de contato e até mesmo muita sobreposição. Em 1960, Edmund Jerome McCarthy, um professor de marketing da Universidade do Estado de Michigan, propôs o conceito de *Marketing Mix*, popularizado por Philip Kotler, que é um conjunto de ferramentas usadas pela função de marketing, com os famosos 4 Ps:

Produto (product): diz respeito ao produto em si e como ele resolve um problema ou atende uma necessidade de um conjunto de pessoas.

Preço (price): por qual preço esse produto será vendido.

Promoção / divulgação (promotion): de que forma vamos contar ao mundo sobre esse produto, e como ele é capaz de resolver o problema ou atender à necessidade de um conjunto de pessoas.

Praça / canal (place): onde esse produto será disponibilizado e vendido.

O produto é claramente de responsabilidade do gestor de produtos, mas isso não quer dizer que o gestor de marketing não pode acompanhar o processo do seu desenvolvimento. Aliás, acompanhar esse processo dará ao gestor de marketing de produtos muitos elementos importantes para ajudar a criar a contar ao mundo sobre ele.

Em algumas empresas, o gestor de produto é responsável pela definição de preço, mas normalmente essa definição, bem como a definição das condições comerciais, é de responsabilidade do gestor de marketing de produtos. Este deve trabalhar junto com o gestor nessas definições, uma vez que ele tem muita informação que pode ajudar nelas e que englobam não só definir o preço como também se haverá versões mais caras ou mais baratas, ou mesmo se haverá funcionalidades adicionais pagas. O conhecimento do cliente e de quanto ele valoriza a solução do seu problema, ou o atendimento de sua necessidade, são essenciais para a definição do preço e das condições comerciais.

A definição da forma e do conteúdo da divulgação do produto, que inclui também definir o nome, é de responsabilidade do marketing de produtos, assim como a definição dos canais de venda, ou seja, por onde o produto vai ser vendido: pela web, pela força de vendas, por canais de vendas, ou uma combinação dessas 3 formas.

Assim, usando os 4 Ps do Marketing Mix, poderíamos ilustrar essa divisão de responsabilidades da seguinte forma:

MÉTRICAS COMPARTILHADAS

Falei bastante sobre métricas desde o *Capítulo 17 – Crescimento: seja um “data geek”* até o *Capítulo 21 – Crescimento: considerações sobre métricas*. São 5 capítulos inteiros dedicados ao assunto. Essas métricas serão compartilhadas entre o gestor de produto e o gestor de marketing de produto.

O gestor de produto deve ter um foco grande nas métricas relacionadas ao uso do produto, tais como NPS, churn e engajamento. Já o de marketing terá um foco maior nas métricas de receita, como CAC,

LTV, receita e novas vendas. O importante é saber que o gestor de produtos e o de marketing de produtos têm focos distintos, mas as métricas são compartilhadas. Isto é, o gestor de marketing também deve acompanhar e se preocupar com as métricas de engajamento, assim como o gestor de produto deve acompanhar e se preocupar com as métricas de receita.

CONCLUINDO

Como visto ao longo deste capítulo, gestão de marketing de produtos e gestão de produtos são funções bem distintas, sendo a primeira responsável por definir como o produto será comercialmente oferecido e contar ao mundo sobre ele, enquanto a segunda tem a responsabilidade de definir em detalhes como ele será. Apesar de serem bem distintas, elas devem trabalhar muito juntas, pois o trabalho de uma é o insumo do trabalho da outra (e vice-versa).

Como comentei ao longo do livro, o *core team* do produto é um time multidisciplinar, contendo o trio de funções gestor de produto, pessoas de UX e engenheiros. Na Locaweb, inserimos um quarto elemento a esse time: as pessoas de marketing de produto, para que elas participem do processo de seu desenvolvimento, dando seus *inputs*, e tirem daí mais alguns elementos que serão úteis para fazer sua função de comunicar ao mundo sobre ele.

Minha recomendação é que você mantenha essas funções separadas – ou seja, tendo pessoas distintas cuidando dessas duas fun-

ções –, mas as mantenha trabalhando perto, uma vez que a colaboração entre elas é muito benéfica para o produto, para o time que o desenvolve e para toda a empresa.

CAPÍTULO 28

QUAL A DIFERENÇA
ENTRE GESTÃO DE
PROJETOS E GESTÃO
DE PRODUTOS?

Assim como com as funções de gestão de marketing de produtos e gestão de produtos que, conforme vimos no capítulo anterior, são bastante distintas, mas apresentam sobreposição; as funções de gestão de projetos e de gestão de produtos também são bem distintas, mas também apresentam bastante coisa em comum.

Antes de falar sobre a diferença entre essas funções, é preciso deixar claro a diferença de projeto e produto. Vou recorrer mais uma vez à Wikipédia:

PROJETO

Um projeto em negócio e ciência é normalmente definido como um empreendimento colaborativo, frequentemente envolvendo pesquisa ou desenho, que é cuidadosamente planejado para alcançar um objetivo particular.

Fonte: <https://en.wikipedia.org/wiki/Project>

PRODUTO

O termo produto é definido como “algo produzido pelo trabalho ou esforço” ou como “resultado de um ato ou processo” e tem sua origem no verbo produzir, do Latim *produce(re)*, ‘fazer existir’.

Fonte: [https://en.wikipedia.org/wiki/Product_\(business\)](https://en.wikipedia.org/wiki/Product_(business))

Ou seja, enquanto o **projeto é um processo** com começo, meio e fim; o **produto é o resultado** de um processo, de um esforço.

ENTÃO GERIR PROJETOS E PRODUTOS SÃO DUAS FUNÇÕES DISTINTAS?

Sim. Enquanto se está gerindo um projeto, a preocupação é com o processo e com tudo o que o cerca, ou seja, se está no prazo, se tem tudo o que é necessário e se está sendo feito com a qualidade esperada.

Por outro lado, quando se gera um produto, a maior preocupação é garantir que este resolva um problema do cliente a quem ele é destinado, e atenda aos objetivos da empresa.

Em um post de 2007 do blog *How To Be A Good Product Manager* – disponível em <http://www.goodproductmanager.com/2007/09/24/product-management-vs-project-management> –, o autor Jeff Lash lembra alguns pontos importantes que não devemos esquecer quando pensamos em gestão de projetos e gestão de produtos:

- 1) Assim como todo produto precisa de um gestor de produtos, todo projeto precisa de um gestor de projeto.
- 2) O fato de os gestores de produtos acreditarem que são capazes de gerir seus próprios projetos não significa que eles devam gerir seus próprios projetos.
- 3) As competências, talentos e conhecimento envolvidos em gestão de projetos são bem diferentes dos envolvidos em gestão de produtos.

4) Assim como é difícil encontrar uma pessoa capaz de fazer, ao mesmo tempo, gestão de produtos e marketing de produtos muito bem, é difícil encontrar uma pessoa capaz de fazer ao mesmo tempo gestão de produtos e gestão de projetos muito bem.

5) Gestão de projetos não é um passo na carreira de gestão de produtos, nem gestão de produtos é um passo na carreira de gestão de projetos.

6) Bons gestores de projetos são tão valiosos quanto bons gestores de produtos.

7) Ter um bom gestor de projetos para gerenciar seus projetos vai lhe ajudar a ser um gestor de produtos melhor.

8) Quanto menos tempo um gestor de produtos passar gerenciando projetos, mais tempo passará gerindo o produto.

9) Para evitar conflitos entre gestão de projeto e gestão de produto, os gestores de projeto, os gestores de produto e todo o time envolvido no projeto devem acordar sobre os objetivos compartilhados pelo time o máximo possível.

Já Marty Cagan deixa clara a necessidade de separação desses papéis em um de seus posts – disponível em <http://www.svpg.com/product-management-vs-project-management>:

“Para empresas de internet é realmente importante que os papéis sejam separados. Você vai ter problemas em gerenciar seus releases se você não separar esses papéis, e seus releases irão sempre atrasar e demorar mais do que deveriam.”

—Marty Cagan

E COMO AS METODOLOGIAS ÁGEIS VEEM ESSAS FUNÇÕES?

As metodologias ágeis, mais especificamente o Scrum, têm dois papéis claros no time: um focado mais no projeto, o Scrum Master; e outro focado mais no produto, o Product Owner (PO):

- **Product Owner:** pessoa responsável por manter o backlog do produto, que representa os interesses dos stakeholders.
- **Scrum Master:** pessoa responsável pelo processo Scrum, garantido que é usado corretamente e maximizando seus benefícios.
- **Team:** grupo de pessoas multifuncional responsável por se gerenciar para desenvolver o produto.
- **Scrum Team:** Product Owner, Scrum Master e o Time.

Há um artigo na InfoQ escrito por Mark Levison em 2008, chamado *Can Product Owner and Scrum Master be Combined?* (em uma tradução livre, “Product Owner e Scrum Master podem ser combinados?”) – disponível em <http://www.infoq.com/br/news/2008/12/scrum-master-product-owner> –, em que o tema de ter uma única pessoa gerindo projeto e produto é discutido. Tanto nas opiniões que compõem o texto – e que incluem testemunhos de pessoas como Mike Cohn e Ken Schwaber – quanto nos comentários feitos por leitores do artigo, é unânime que, apesar de ser possível combinar as duas funções – e, se o time for muito pequeno, é até aceitável –, o mais recomendado é que estas sejam desempenhadas por pessoas diferentes.

E NA VIDA REAL?

Todos os relatos vistos são baseados em fatos reais, mas sabemos que cada empresa tem sua própria realidade e seu próprio contexto. Então, o que é melhor fazer: deixar esses papéis separados ou combinados?

O ideal é você ir experimentando e, em algum determinado ponto, você encontrará uma combinação que seja a mais adequada para você, para o time com quem você trabalha e para a sua empresa. Note que cada grupo de pessoas tem sua dinâmica própria, e o que funciona em um grupo de pessoas pode não funcionar para outro.

Na Locaweb, temos vários times desenvolvendo diferentes produtos, e cada um tem sua dinâmica própria onde o gestor de produto assume responsabilidades diferentes em relação ao time. Em alguns, a responsabilidade pelas tarefas de gestão de projeto técnico – ou seja, cuidar de questões de desenvolvimento, deploy e operação do produto – é tocada por um gestor de projeto, enquanto que, em outros times, essa responsabilidade é compartilhada entre o líder técnico da equipe e o gestor de produto.

Por outro lado, em todos os times, o gestor de produto exerce o papel de gestor de projeto para todas as tarefas não técnicas. Isto é, coordena com o time de marketing a comunicação do produto, coordena com o jurídico e com o financeiro suas necessidades legais e fiscais, suporta marketing no treinamento para as equipes de vendas, e cuida de passar o conhecimento para a equipe de suporte técnico.

Enfim, procure encontrar um equilíbrio que faça sentido para você, para o seu time e para empresa que você trabalha, só tome cuidado para não absorver todas as funções de gestão de projeto. Procure dividi-las com alguém, principalmente as questões técnicas; caso contrário, não sobrar tempo para você gerir seu produto.

CAPÍTULO 29

E AS OUTRAS ÁREAS?

Nos capítulos anteriores, contei sobre a relação entre o gestor de produtos e as áreas de engenharia, UX, marketing de produtos e gestão de projetos. A pergunta que fica é: e as outras áreas da empresa? Como é o relacionamento do gestor de produtos com operações, atendimento, jurídico, vendas, financeiro administrativo e recursos humanos?

OPERAÇÕES

A área de operações é responsável por manter o produto operando. Ou seja, para produtos online, é a responsável por garantir que toda a infraestrutura necessária para que ele esteja disponível, e tenha a dimensão e a elasticidade necessárias para oferecer um serviço de boa qualidade operacional para o cliente. A qualidade operacional de um produto tem a ver com duas características operacionais do produto:

O produto deve ter boa performance: a cada requisição feita pelo cliente ao produto, ele deve responder em menos de X segundos. A definição desse X é a definição de performance desse produto. É importante saber que X não é um número inteiro; ele pode e, idealmente, deve ser um número fracionário menor que 1 (por exemplo, o produto deve responder requisições em 0,3 segundos). Diferentes funcionalidades do produto podem ter diferentes expectativas de performance.

O produto deve estar disponível para uso o máximo de tempo possível: esse é o SLA (*Service Level Agreement*), definido como um número percentual, por exemplo 99,9%. Esse número pode ser me-

dido em um dia, mês ou em um ano. Em um mês, 99,9% de disponibilidade significam 43,2 minutos de indisponibilidade, enquanto 99,5% significam 3 horas e 36 minutos de indisponibilidade. Quando analisamos a disponibilidade de um produto, devemos levar em conta dois fatores:

- *Frequência de problemas*: ou seja, de quanto em quanto tempo acontece cada indisponibilidade. De nada adianta ter uma disponibilidade de 99,9%, com um máximo de apenas 43,2 minutos por mês de indisponibilidade, se essa indisponibilidade acontecer em um determinado dia por 1 minuto a cada 10 minutos? Isso significa que o produto ficará instável por mais de 7 horas naquele dia. Existe uma métrica operacional que deve ser acompanhada e que ajuda a medir a frequência de problemas. É o MTBF, do inglês *mean time between failures*, ou tempo médio entre falhas.
- *Duração do problema de indisponibilidade ou mal-funcionamento*: aqui o racional é simples, o cliente deseja que, se houver problema no produto, que este dure o mínimo de tempo possível. Também existe uma métrica operacional que deve ser acompanhada e que ajuda a medir a duração dos problemas. É o MTTR, do inglês *mean time to resolve*, ou tempo médio para resolver.

A definição de boa qualidade operacional é uma definição que tem de ser feita pelo gestor de produto em conjunto com engenharia, UX e operações. Claro que nós sempre queremos oferecer o produto com a melhor qualidade possível para nossos clientes, porém, na

maioria das vezes, isso tem um custo alto. É importante o gestor de produtos definir, junto com engenharia, UX e operações, qual é o nível de qualidade ::versus:: nível de custo operacional aceitável para o produto. Isso é o que chamamos de requisito não funcional, pois ele não é uma funcionalidade do produto, mas sim uma característica que impacta no seu uso.

Para que operações possam operar o produto, é importante que o produto seja “operável”. Parece meio redundante o que acabo de escrever, mas não é raro ver produtos praticamente prontos para serem lançados que têm de voltar para o desenvolvimento, pois não foram levados em conta alguns pontos operacionais simples, como: o que deve ser monitorado, como deve ser monitorado e o que deve ser feito caso a monitoração acuse algo fora do comum.

O ideal é que isso seja pensado antes de se começar a desenvolver o produto. Se deixar para pensar nesses pontos somente depois que o produto ficar pronto, há grandes chances de se ter de refazer alguma coisa no produto para encaixar essas questões operacionais. Para que a ação de pensar nas questões operacionais aconteça antes e durante seu desenvolvimento, o mais indicado é que alguém de operações participe de forma mais próxima do desenvolvimento.

Apesar de a relação de operações dever ser mais próxima de engenharia do que do gestor de produtos, é papel deste último acompanhar essa relação e ajudar no que for possível, principalmente nas definições dos requisitos não funcionais de operação.

ATENDIMENTO

O ideal é que seu produto não precise de atendimento por telefone, chat ou e-mail. Entretanto, por mais que você trabalhe junto com UX para que isso seja verdade, sempre haverá clientes que querem falar com alguém da sua empresa por algum motivo relacionado ao seu produto. Por isso, você deverá preparar seu time de atendimento para falar com seu cliente e resolver seus problemas.

Antes de lançar seu produto, você deve preparar e ministrar um treinamento para o seu time de atendimento para que eles estejam preparados para responder as principais dúvidas. Assim que ele for lançado, você deve acompanhar todos os atendimentos feitos sobre ele para não só ver a qualidade das respostas dadas pelo time de atendimento, mas, principalmente, para entender quais são as principais dúvidas que eles têm quando usam o produto.

Seu objetivo deve ser minimizar esses contatos, tentando resolver essas dúvidas diretamente no produto. Essa é uma tarefa que você precisará conduzir em conjunto com as pessoas de UX que estiverem trabalhando no produto. Toda vez que você lançar novidades nele, é importante comunicar para o time de atendimento com alguma antecedência para deixá-los preparados, para que eles possam dar todo o suporte às dúvidas sobre essas novas funcionalidades.

JURÍDICO

O jurídico de sua empresa pode ser tanto interno quanto externo, ou até mesmo um híbrido dessas duas opções. Independente do modelo, é importante que este tenha bom conhecimento ligado à área digital, e que esteja constantemente se atualizando, uma vez que muitas leis e normas ainda estão sendo criadas, vide o Marco Civil da Internet (Lei nº 12.965) de 23 de abril de 2014 – disponível em http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm. Novas jurisprudências sobre casos relacionados a produtos de software, produtos digitais e produtos online aparecem com boa frequência, e é importante que seu jurídico esteja por dentro.

O jurídico tem duas funções principais ligadas ao seu produto:

Contrato com usuário: nesse contrato, são documentadas as principais características do produto, incluindo nível de serviço, tipo de atendimento, política de privacidade de dados e formas de pagamento. Além disso, esse contrato deve conter direitos e deveres tanto do seu cliente quanto da sua empresa, ou seja, as regras que devem ser respeitadas para o bom funcionamento do produto.

Por exemplo, na Locaweb definimos em um determinado momento fazer o nosso produto de Hospedagem de Sites com espaço em disco e transferência ilimitados. Para poder fazer isso, definimos em contrato o que pode e o que não ser feito em nosso produto de Hospedagem de Sites com espaço em disco e transferência ilimitados para garantir o bom funcionamento do produto para todos os clientes.

Colocamos na política de ilimitados (<http://www.locaweb.com.br/sobre-locaweb/contratos-politicas/politica-ilimitado.html>) que “ter espaço e transferência ilimitados não significa que você pode utilizar sua hospedagem de forma a prejudicar o funcionamento do servidor, para fins ilícitos ou que afronte por qualquer outra maneira a legislação em vigor”.

Lembre-se, além de fazer o contrato, o jurídico lida com todas as reclamações que seus clientes fizerem no âmbito jurídico, normalmente por meio de ações abertas contra a sua empresa; por isso, você deve usar da empatia para entender as preocupações que o seu jurídico certamente terá em relação ao seu produto e a forma como ele é usado.

Contrato com fornecedores: nesse contrato, o foco é definir como será gerenciada a relação da sua empresa com os fornecedores de software que vão fazer parte do seu produto. Por exemplo, na Locaweb temos o produto Hospedagem de Sites, que pode ser na plataforma Linux ou na Windows. No Linux, temos de entender os contratos de uso dos softwares que são de código aberto (*open source*) para ver se o produto de Hospedagem de Sites Linux pode ser feito usando esses softwares, e se há alguma restrição que devemos considerar. Na plataforma Windows, temos de verificar constantemente o contrato de licenciamento de softwares da Microsoft para ter certeza de que estamos ofertando o produto de Hospedagem de Sites em Windows de acordo com as políticas da Microsoft.

VENDAS

A equipe de vendas precisa ser treinada por você e pela pessoa do marketing de produtos sobre o produto que será lançado. Depois disso, é importante fazer atualizações de treinamento sempre que vocês lançarem novidades em seu produto.

Outro ponto importante em relação à equipe de vendas é que eles podem fornecer muito feedback importante em relação ao seu produto. Novas funcionalidades, ajustes nas funcionalidades existentes, ajustes em preço e em modelo de negócios são temas nos quais certamente a equipe de vendas poderá contribuir. Mas aqui vale de novo lembrar da empatia, ou seja, lembre-se de qual é o objetivo da equipe de vendas: vender. Eles enxergam o seu produto sob essa ótica, como vender mais. Você precisa usar seu discernimento e o conhecimento que você tem do cliente para julgar se o *input* de vendas é válido para o produto, ou se ele eventualmente poderá atrapalhar os seus objetivos.

Por exemplo, é razoavelmente comum aparecer um vendedor falando que tem um potencial enorme de fazer uma venda grande, mas, para o cliente fechar, ele precisa de certa funcionalidade e, se ela não for implementada, essa venda será perdida. Cabe ao gestor de produto ter o discernimento sobre a importância dessa nova funcionalidade para o produto e para os outros clientes do produto.

Como visto no *Capítulo 16 – Crescimento: como priorizar o roadmap?*, quando comentei sobre a diferença dos produtos E-mail Marketing Locaweb e da All In Mail, o contexto do produto deixa clara essa necessidade de discernimento do gestor de produtos. Aceitar esses pedi-

dos especiais no produto E-mail Marketing Locaweb não faz nenhum sentido. Já no produto All In Mail, esses pedidos do time de vendas é o que deve direcionar o seu roadmap.

FINANCEIRO

O financeiro é a área que dará subsídios para o gestor de produtos entender se o produto que ele cuida é um bom produto para a empresa; ou seja, se o retorno do produto está compensando os seus custos. Claro que você e o marketing de produtos vão acompanhar a receita de seu produto diariamente, mas os custos não são tão simples de serem acompanhados. O financeiro pode ajudar nos cálculos e no acompanhamento do custo do produto. Você pode dividir seu custo em dois grandes grupos: o custo de manter seu produto e o custo de atrair clientes para o seu produto.

O custo de manter seu produto web em operação é o que podemos chamar de **custo de operação**. Nessa categoria, estão custo com a infraestrutura e com pessoas necessárias para tocar o dia a dia do produto. Aqui nesse grupo, você deve também considerar o seu time de desenvolvimento, isto é, o pessoal de UX, os engenheiros de produto e você.

Outro custo de operação que deve também ser considerado aqui é o financeiro, ou seja, os impostos que você paga. O financeiro, junto com o jurídico, são os mais indicados para definirem a melhor estrutura de impostos para o seu produto.

Os investimentos de marketing e vendas são o **custo de atrair novos clientes**. Se você gastar com AdWords ou outras formas de campanha online ou offline, isso terá um custo e é muito importante você medir o retorno que este lhe dará. Quantos R\$ você precisa investir em marketing para trazer um cliente? Quanto de receita esse cliente traz? É suficiente para pagar o custo de atraí-lo, mais os custos de operação, e ainda sobrar um pouco para pagar o investimento feito em desenvolver o produto?

Nessa categoria entra também o custo das equipes de vendas e de marketing, e as comissões que forem dadas para a equipe de vendas, caso sua empresa opte por remunerá-la com comissionamento. Se você usar canais indiretos de venda, como venda por meio de parceiros, a remuneração desses parceiros também entrará nesse custo. Se o time de marketing resolver fazer promoções, como primeiro ano com desconto, isso também entra como custo de atrair novos clientes.

Quando você definir o preço de seu produto, é importante considerar sua estrutura de custos, pois a receita serve para cobri-los e ainda sobrar um pouco. Se você não conseguir receita suficiente para isso, seu produto não terá sucesso.

RECURSOS HUMANOS

A princípio, pode parecer que a área de RH não tem nada a ver com o produto e sua gestão, mas tem tudo a ver. O RH é essencial para o sucesso do produto, pois vai fazer parte do processo de encontrar e

atrair novos profissionais para trabalhar nele. Além disso, deverá ajudar no *onboarding* do novo funcionário, ou seja, nos primeiros dias dele na empresa. O RH também pode ajudar no treinamento e nas avaliações das pessoas dos times de desenvolvimento do produto. Daí sua importância no sucesso do produto.

ADMINISTRATIVO

Esse é mais um time que parece distante do produto e do time que desenvolve o produto, mas não é. O administrativo mantém a infraestrutura necessária para a empresa continuar funcionando. Escritório, salas, móveis, material de escritório, compras, serviços de escritório (limpeza, lanches, refeição, entregas etc.), tudo isso é necessário para sua empresa funcionar e, conseqüentemente, para o time de desenvolvimento de produtos poder fazer seu trabalho.

CONCLUINDO

Optei por colocar todas essas áreas em um único capítulo em vez de dedicar um para cada uma, como fiz para falar sobre engenharia, UX, marketing de produtos e gestão de projetos, pois a interação com essas áreas é menor. Contudo, vale lembrar de que essas áreas e o bom relacionamento do gestor de produtos com elas é essencial para o sucesso de seu produto.

Relembrando do que falei no *Capítulo 04 – Principais características de um gestor de produtos*, a característica mais importante que todo gestor de produto precisa ter é a **empatia**, ou seja, a capacidade de se colocar no lugar de outra pessoa para compreender seus anseios, motivações, necessidade e problemas. Essa característica deve ser usada não somente com o cliente do produto como também com as pessoas das diferentes áreas da empresa. O gestor de produtos deve entender o impacto que seu produto tem no trabalho de cada uma delas para poder tornar o trabalho conjunto o mais frutífero possível.

Como vimos neste capítulo e nos anteriores, o gestor interage praticamente com todas as áreas da empresa e, com várias delas, pode haver sobreposição de papéis e responsabilidades. No próximo capítulo, vamos ver uma valiosa ferramenta para ajudar a definir claramente os papéis de cada área nas diferentes tarefas relacionados ao ciclo de vida de um produto.

CAPÍTULO 30

DEFININDO RESPONSABILIDADES

No Capítulo 27 – *Qual a diferença entre gestão de marketing de produtos e gestão de produtos?*, comentei sobre como os 4Ps do marketing (Produto, Preço, Promoção e Praça) são distribuídos entre gestores de marketing de produto e gestores de produto. Essa é uma visão macro de divisão de responsabilidades. Nele, falei também sobre como esses times (marketing e gestão de produtos) devem trabalhar muito próximos para o sucesso dos produtos que eles desenvolvem e administram, sendo que o mesmo acontece em relação aos times de Engenharia e de UX, que, apesar de serem funções distintas, também devem trabalhar muito próximos com a gestão. Outra função que pode ter bastante sobreposição com o gestor de produtos é o gestor de projetos que, em times de desenvolvimento que trabalham com cultura ágil, são chamados de Scrum Master.

Essa proximidade costuma gerar algumas dúvidas do tipo: “Esta tarefa, quem é o responsável?”, “Esta outra tarefa, quem eu preciso consultar antes de levar ela adiante?” e “Para esta outra aqui, quem precisa estar junto comigo para eu conseguir concluí-la com êxito?”. Muitas vezes, acabam ficando bolas divididas: um acha que é responsável por determinada tarefa e o outro também acha. Ou pior, um acha que o outro é o responsável, e esse outro acha que o primeiro que era, e ninguém a faz.

RASCI

Para resolver esse tipo de situação, existe uma ferramenta muito útil chamada Matriz de Responsabilidade RASCI. Esse RASCI é uma abre-

viação das primeiras letras dos possíveis papéis que uma pessoa, área ou função pode ter em uma tarefa:

Responsible: é a pessoa responsável pela execução da tarefa, ou seja, quem tem de liderar o esforço de planejar, fazer e concluir a tarefa. Não pode existir mais de um responsável por tarefa. É aquela máxima de que cachorro com dois donos morre de fome.

Accountable: é a pessoa que responde pela tarefa, e que tem o poder de delegar para o responsável a tarefa a ser feita. Responsável e *accountable* podem ser a mesma pessoa. Também vale a regra de que não pode existir mais de um *accountable* por tarefa.

Support: são as pessoas ou equipes que trabalham junto e sob a coordenação do responsável para a execução da tarefa.

Consulted: são as pessoas ou equipes que não participam da execução da tarefa, mas que precisam ser consultadas antes ou enquanto a tarefa estiver sendo executado, pois elas podem dar *inputs* relevantes para sua execução.

Informed: são as pessoas ou equipes que não participam da execução da tarefa, nem precisam ser consultadas antes ou enquanto a tarefa estiver sendo executado, mas que precisam ser informadas quando a tarefa for concluída.

A seguir, está um exemplo de uma matriz de responsabilidade RASCI entre engenharia, UX, marketing de produtos e gestão de produtos que usamos na Locaweb:

COMO USAR?

O primeiro passo é fazer a matriz de responsabilidades. Minha recomendação é preencher essa tabela juntando todas as pessoas envolvidas em uma sala, assim pode-se discutir se a divisão de responsabilidade está boa para todos e se tem alguma tarefa faltando. Muito provavelmente, vão surgir algumas “bolas divididas”, mas esse é um ótimo momento para discuti-las e definir quem é o responsável.

Em seguida, deve-se experimentar fazer as tarefas seguindo a matriz responsabilidade por algum tempo, tipo um ou dois meses. Depois, é importante fazer uma retrospectiva para ver se está tudo certo, ou se é necessário algum ajuste.

Daí para frente, o uso passa a ser automático e as pessoas não precisarão mais se referir à matriz de responsabilidades. A cada 1 ou 1,5

anos, quando surgir alguma dúvida, ou mesmo quando surgir alguma tarefa nova, é bom revisitá-la.

Pronto, todo mundo sabe o seu papel e suas responsabilidades, e agora é só partir para a execução! :-)

PARTE IV
GESTÃO DE
PORTFÓLIO DE
PRODUTOS

Por que algumas empresas decidem ter mais de um produto? Como elas fazem para gerenciar esse portfólio de produtos? Por que outras empresas preferem se focar em um único produto? Foco ou diversificação, qual é a estratégia mais apropriada? Como organizar o time de desenvolvimento de produtos de acordo com a estratégia escolhida?

Esses temas são os que considero como sendo tópicos avançados de gestão de produtos de software, e é o que veremos nos próximos capítulos!

CAPÍTULO 31

JÁ ESTÁ PENSANDO EM
SEU NOVO PRODUTO?
NÃO?
ENTÃO JÁ ESTÁ
ATRASADO!

Você sabia que o Google tem 177 produtos ativos além de 79 produtos descontinuados? Essas são as estatísticas do Google na época em que estou escrevendo esse livro, um pouco antes dele anunciar o Alphabet, em agosto de 2015.

E quantos escritórios o Google tem espalhado pelo mundo? Tentei contar no mapa a seguir, mas não consegui. Parece mais um mapa do jogo de tabuleiro War, com o Google conquistando o mundo.

A Locaweb, guardadas as devidas proporções, também tem uma estratégia parecida. Temos mais de 25 produtos e já descontinuamos mais de 10. Além disso, a Locaweb tem escritórios em Porto Alegre, Marília e Blumenau.

Outro exemplo é a Caelum, que tem serviços de consultoria e de aulas presenciais de desenvolvimento de software em São Paulo, no Rio de

Janeiro e em Brasília. Tinha um serviço de desenvolvimento de software sob demanda que foi descontinuado, e criou um serviço de aulas online e uma editora.

POR QUE ESSAS EMPRESAS USAM ESSA ESTRATÉGIA DE DIVERSIFICAÇÃO DE PORTFÓLIO E DE CONQUISTA DE NOVOS MERCADOS?

Como já expliquei no *Capítulo 07 – Como é o ciclo de vida de um produto de software?*, precisamos rever o conceito de curva de adoção de tecnologia. Esse conceito apareceu pela primeira vez em um livro de 1962 chamado *Diffusion of Innovations*, escrito por Everett M. Rogers, sociólogo e professor da Universidade Estadual de Iowa. Nesse livro, Rogers explica que as inovações tecnológicas são adotadas conforme a curva vista a seguir.

Como visto naquele capítulo, os inovadores são os primeiros a se interessar por novos produtos e inovações (*innovators*), topando até pro-

dutos incompletos e com defeitos pelo prazer de serem os primeiros a utilizá-los. Em seguida, estão os *early adopters*, também conhecidos como visionários ou entusiastas, que aceitam os riscos de testar um novo produto, mas não pelo prazer de ser o primeiro, mas porque enxergam o potencial desse novo produto. Normalmente, eles têm influência nas organizações e comunidades de que fazem parte.

Os *early majority*, também chamados de pragmáticos, compram novos produtos somente depois de ter referências. Os *late majority* são os conservadores, ou seja, aqueles que compram somente depois que o preço caiu consideravelmente. Por fim, há os *laggards*, que só compram um novo produto se essa for a única opção disponível.

Fazendo a integral dessa curva (quem se lembra das aulas de cálculo?), obteremos a famosa curva em S de adoção de tecnologia.

Essa curva em S pode ser quebrada em 3 fases: o início mais lento, que é a fase de inovação; a fase de crescimento, quando *early majority* e *late majority* adotam o produto; e, por fim, a fase de maturidade, na qual o produto já conquistou praticamente todo o mercado.

O ABISMO

Essa curva em S pode ser quebrada em 3 fases: o início mais lento, que é a fase de inovação; a fase de crescimento, quando *early majority* e *late majority* adotam o produto; e, por fim, a fase de maturidade, na qual o produto já conquistou praticamente todo o mercado.

Sempre tem um “mas”. Em 1991, Geoffrey Moore escreveu um livro intitulado *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*. Nele, ele explica que, entre os *early adopters* (entusiastas) e a *early majority* (pragmáticos), existe um abismo que muitos produtos não conseguem cruzar. Isso acontece pois os pragmáticos precisam de boas referências para poder comprar um novo produto, e os entusiastas normalmente não são uma boa referência. Daí a dificuldade de alguns produtos cruzarem esse abismo.

Conseqüentemente, uma empresa de um único produto vai eventualmente:

- a) **Não vai cruzar o abismo:** a empresa não consegue fazer seu produto ir além dos entusiastas e, conseqüentemente, não terá clientes para sobreviver. Esse é um dos motivos da morte prematura de muitas startups.

b) **Amadurecer**: seu produto vai dar certo e a empresa vai eventualmente chegar ao topo da curva S, e vai desacelerar até que alguma outra empresa invente um produto que substitua o seu produto. Veja a Kodak que até hoje não se recuperou da invenção das máquinas digitais, pois tinha sua receita vinda primariamente da venda de filmes e material fotográfico.

Para citar mais uma vez a “Rainha das Ciências”, a matemática: **Q.E.D.**

Q.E.D.

Quod erat demonstrandum é uma expressão em latim que significa “como se queria demonstrar”. É usual aparecer no final de uma demonstração matemática com a abreviatura Q.E.D., ou na versão em português C.Q.D.

Fonte: https://pt.wikipedia.org/wiki/Quod_erat_demonstrandum

CONCLUINDO

Entendendo a curva S de adoção de tecnologia e o conceito de abismo que existe nessa curva entre os entusiastas e os pragmáticos, é fácil perceber a necessidade de diversificação de portfólio que as empresas têm. Contudo, perguntas ficam ainda sem resposta:

- Que opções existem para diversificar o portfólio de produtos?
- Como fazer para gerenciar um portfólio de produtos?
- Por que algumas empresas optam pelo foco em um único produto mesmo estando claro (como descrito anteriormente) a necessidade de diversificação de portfólio de produtos para a sua sobrevivência?

Essas perguntas serão respondidas nos próximos capítulos.

CAPÍTULO 32

COMO DIVERSIFICAR SEU PORTFÓLIO DE PRODUTOS?

No capítulo anterior, falei sobre o porquê de as empresas deverem diversificar seu portfólio de produtos. Agora, mostrarei como fazer isso.

NOVOS MERCADOS

A primeira opção é a busca por novos mercados. Quando pensamos nisso, lembramos logo daquele mapa do Google, que vimos no capítulo anterior, que mostra a presença de seus escritórios em vários países.

Essa é a forma mais comum de se pensar a busca por novos mercados, e – de novo, guardadas as devidas proporções – é o que a Locaweb e a Caelum também têm feito. Tanto a Locaweb quanto a Caelum ainda estão no âmbito nacional de busca de novos mercados no Brasil, mas não deixa de ser algo similar, ou seja, a busca de novos mercados geográficos.

Contudo, quando pensamos em novos mercados, não devemos nos restringir à geografia. Existem outras características que definem novos mercados:

Faixa etária: se você tem um produto para pessoas jovens, é possível adequá-lo para pessoas mais velhas? Será que estas vão ter interesse? E as crianças?

Pessoa física vs. pessoa jurídica: se o seu produto é feito para empresas, será que ele poderia também ser usado por consumidores finais?

Tamanho da empresa: seu produto é indicado para pequenas empresas? O que seria necessário para adequá-lo para grandes empresas? Ou seu produto é feito para grandes empresas? É possível fazer algo para adequá-lo para as pequenas empresas? Foi exatamente a busca desse novo mercado que motivou a Locaweb a adquirir a All In. Como expliquei no *Capítulo 16 – Crescimento: como priorizar o roadmap?*, a Locaweb tinha o produto Email Marketing Locaweb focado em pequenas empresas. Com certa frequência, recebíamos demandas de funcionalidades típicas de empresas grandes, que disparam grande volume de e-mail. Era um novo mercado que estava se abrindo para nós. Analisamos as opções que tínhamos que eram: não fazer nada, evoluir o produto, desenvolver um novo produto para poder atender esse novo mercado, ou adquirir uma empresa que já tivesse um produto que atendesse esse novo mercado. Optamos pela terceira opção e adquirimos a empresa All In, que tem um produto de e-mail marketing feito para grandes empresas.

Nichos: você tem um produto para dentistas? Será que ele pode servir também para médicos? E fisioterapeutas?

Ou seja, não são apenas diferenças geográficas que caracterizam novos mercados. Diferenças etnográficas, sociais e culturais também ajudam a definir mercados, e um gestor de produtos tem de ser capaz de entender e identificar essas diferenças para poder explorar novas oportunidades para seus produtos.

Vale lembrar de que, quando você vai explorar um novo mercado com seu produto existente, há boas chances de você precisar fazer adaptações nele para que ele possa ser aceito nesse novo mercado. Basta ver o

que descrevi sobre o E-mail Marketing Locaweb. Na busca de um novo mercado, de clientes maiores, ficou clara a necessidade de adaptações aos nossos produtos para atender a esses novos mercados. Daí vem a segunda opção para diversificação de portfólio de produtos.

NOVOS PRODUTOS

A outra opção é desenvolver novos produtos para atender o mercado que você já conhece, ou mesmo para atender um novo mercado, como descrevi no caso do E-mail Marketing da Locaweb. Os *Capítulos 08* até o *13* falam exatamente sobre o tema de desenvolvimento de um novo produto, e apresentam as técnicas para descobrir um problema ou necessidade de um grupo de pessoas e como desenvolver um produto para solucionar ou atendê-los.

Só lembrando, aqui entra novamente a importância da empatia, de entender bem seu cliente e os problemas e as necessidades que ele tem. Essa é a forma de encontrar oportunidades de novos produtos. Uma técnica bastante simples, mas muito poderosa de descobrir novas oportunidades, é acompanhar um ou alguns dias do seu cliente. O que ele faz durante esse dia? Quando ele se frustra para fazer uma determinada tarefa no seu dia? O que o deixa satisfeito? Que tarefas poderiam ser aceleradas, melhoradas ou até mesmo eliminadas com ajuda de um software?

A Caelum é um ótimo exemplo para ilustrar o desenvolvimento de novos produtos para os clientes existentes. Inicialmente, eles tinham

cursos presenciais de desenvolvimento de software em São Paulo. Em seguida, expandiram para Rio de Janeiro e Brasília, ou seja, buscaram a estratégia de diversificação de novos mercados geográficos. Depois de algum tempo, começaram a perceber demanda em outras cidades, mas era uma demanda muito dispersa, sem nenhum foco específico em uma cidade ou região. Daí veio a ideia de lançar o serviço de cursos online da Caelum, que mais recentemente foi rebatizado de Alura.

Ao longo desses vários anos dando treinamento para desenvolvedores, ou pessoas que queriam aprender sobre desenvolvimento de software, o pessoal da Caelum percebeu a demanda por livros de qualidade, em português, sobre desenvolvimento de software e assuntos correlacionados. Daí surgiu a ideia de fundar a Casa do Código, editora de livros de tecnologia.

Além das técnicas que citei nos capítulos sobre inovação, existe ainda uma outra estratégia usada por várias empresas para diversificação de portfólio de produtos, incluindo Google e Locaweb, que é a estratégia de aquisição de empresas. Vários dos produtos do Google não foram desenvolvidos dentro dele, mas sim por empresas pequenas que desenvolveram um produto que chamou a atenção dos gestores de produto do Google e os motivou a adquiri-las. Alguns casos bem conhecidos são o Waze, o YouTube e o Android.

A Locaweb também começou a fazer isso recentemente. A Locaweb sempre teve produtos para clientes que queriam ter sua loja online. Começamos com um produto de Gateway de Pagamentos, lançado no começo da Locaweb, em 2006, para que desenvolvedores de sites pudessem criar lojas virtuais com pagamento online. Em 2010, lança-

mos a WebStore, produto completo de loja virtual e, em 2012, conhecemos uma empresa chamada Tray, dos fundadores de uma outra empresa ligada ao e-commerce, a Pagamento Digital, que foi adquirida pelo Buscapé e que recentemente foi rebatizada de bCash.

A Tray era uma empresa completa de e-commerce, com produtos de loja virtual, gateway de pagamento, checkout e marketplace. Eles estavam muito à frente dos nossos de produtos de e-commerce, com produtos mais maduros e que sequer tínhamos em nosso portfólio. Por isso, no final de 2012, anunciamos a aquisição da Tray, empresa de soluções de comércio eletrônico. Outra área em que sempre tivemos interesse é a área de transmissão de vídeos online, um complemento importante para nossos clientes que querem ter presença online. Por isso, adquirimos no final de 2012 uma startup, a Eventials, de transmissão ao vivo de eventos.

CONCLUINDO

Então, existem duas formas de diversificar seu portfólio de produtos. Uma delas é buscando **novos mercados**. Não devemos aqui nos restringir a novos mercados do ponto de vista geográfico. Devemos também pensar em novos mercados pensando em outros parâmetros, tais como tamanho de empresa, idade do cliente, nicho de mercado etc.

A outra forma de diversificação é a criação de **novos produtos** para os clientes existentes.

Vale notar que essas duas estratégias não são excludentes. Muitas vezes é preciso criar um novo produto para poder atender um novo mercado.

Agora que já conhecemos a forma de diversificar seu portfólio de produtos, muito provavelmente você chegará a um ponto no qual você terá 4, 5 ou até mais produtos em seu portfólio e vai começar a se perguntar como gerenciar esses diferentes produtos. Como alocar esforço de desenvolvimento de produto em cada um deles? Como alocar o investimento em marketing em cada um deles? Esse é o tema do próximo capítulo.

CAPÍTULO 33

COMO GERIR UM PORTFÓLIO DE PRODUTOS?

Nos dois capítulos anteriores, expliquei por que uma empresa deve se preocupar em ter mais de um produto e como uma empresa de produtos de software pode diversificar seu portfólio. Neste capítulo, falarei sobre como gerir um portfólio de produtos.

Quando se tem dois ou três produtos, é razoavelmente fácil gerenciar seu portfólio de produtos. Porém, quando chega a uns 4 ou 5, é interessante ter alguma ferramenta para ajudar. Imagine a Locaweb, com mais de 35 produtos, entre produtos ativos e descontinuados, ou o Google, com mais de 250.

A MATRIZ BCG

Na Locaweb, usamos a matriz BCG. A matriz BCG é uma ferramenta de análise gráfica desenvolvida por Bruce Henderson em 1970. Em 1963, Bruce Henderson fundou a empresa de consultoria empresarial americana *Boston Consulting Group* (BCG), da qual foi presidente até 1980 e presidente do conselho até 1985. Ou seja, não é uma ferramenta nova, mas é muito útil, como você verá daqui a pouco.

Seu objetivo é suportar a análise de portfólio de produtos baseada no conceito de ciclo de vida do produto. Ela é utilizada para ajudar nas decisões de alocação de recursos nos diferentes produtos.

		Market share	
		High	Low
Market growth	High		
	Low		

Ela é composta por dois eixos. No eixo Y está representado o crescimento do mercado, e no eixo X está representada a participação de seu produto nesse mercado.

Com isso, divide-se a matriz em quatro quadrantes:

Apostas: no original, é chamado de “ponto de interrogação”, “em questionamento” ou “criança problemática”. Eu prefiro o nome “aposta” por ser um pouco mais positivo! :-P No quadrante das apostas, é onde devem estar todas as startups que estão experimentando soluções para um problema de um conjunto de pessoas. E é nesse quadrante que todos os novos produtos devem começar e estar, pois, de um lado, você sempre começa com uma participação pequena do mercado. Já em termos de mercado em que você está entrando, é sempre melhor entrar em um que está em franco crescimento. Se você estiver pensando em entrar em um mercado com pouco crescimento, pode ter certeza de que você terá muito mais trabalho. Produtos que estão nesse quadrante têm duas opções: ou viram estrelas, ou viram abacaxis quando não cruzam o abismo. Quando se segue uma estratégia de diversificação de portfólio de produtos, é importante ter uma boa quantidade de apostas, pois uma quantidade considerável de apostas não cruzará o abismo e virará um abacaxi.

Estrelas: quando a aposta dá certo, você começa a ter maior participação no mercado, e isso transforma essa aposta em uma estrela. As estrelas são o principal elemento de crescimento de sua empresa, pois elas são os produtos que apresentam o maior crescimento absoluto de número de usuários e de receita.

Vaca leiteira: normalmente, aqui ficam os produtos mais antigos de uma empresa. Como o mercado já não cresce tanto e a empresa tem um pedaço considerável desse mercado, não há necessidade de grandes investimentos, e a receita que vem desses produtos nor-

malmente financia o desenvolvimento dos que estão como apostas e estrelas.

Abacaxis: também conhecidos como “cão”, “vira-lata” ou “animal de estimação”, aqui ficam os produtos que não cruzaram o abismo da curva de adoção de tecnologia. É muito importante perceber quando um produto entra nesse quadrante, pois, na maioria dos casos, quando isso acontece, o produto deve ser descontinuado.

Colocando a curva S de adoção de tecnologia nos respectivos quadrantes da matriz BCG, teremos as apostas como o momento da inovação, as estrelas como o momento do crescimento, as vacas leiteiras como a maturidade dos produtos, e os abacaxis como os produtos que não cruzam o abismo entre os *early adopters* e a *early majority*.

Um ponto relevante a observar é a distribuição de produtos nos diferentes quadrantes. Na Locaweb, temos uma grande concentração de apostas, pois, por mais que existam várias metodologias para se desenvolver produtos melhores e mais acertados, a verdade é que mesmo assim muitos produtos não cruzam o abismo e viram abacaxis. Por isso a importância de testar novas ideias de produtos de forma rápida. Na sequência, mostrarei uma versão simplificada da Matriz BCG da Locaweb.

ALOCAÇÃO DE ESFORÇO E INVESTIMENTO

Do ponto de vista de recursos da empresa, devemos alocá-los da seguinte forma:

		Market share	
		High	Low
Market growth	High	Stars Dev: ↑ Ops: ↑ Mkt: ↑	Bets Dev: ↑ Ops: ↔ Mkt: ↑
	Low	Cash cows Dev: ↔ Ops: ↑ Mkt: ↔	Dogs Dev: ↓ Ops: ↔ Mkt: ↓

Ou seja, para cada fase do produto, o seu investimento em desenvolvimento, em operações e em marketing, será diferente.

APOSTAS

Durante a fase de aposta, você deve alocar seus esforços da seguinte forma:

Desenvolvimento: as apostas requerem esforço alto de desenvolvimento. O mercado está em franco crescimento, e seu produto precisa se adaptar e conquistar um espaço relevante nesse mercado, pois só assim ele poderá ir para a fase de estrela.

Operações: em uma aposta, as preocupações operacionais são menores. É até aceitável ter alguns processos manuais. E você não deve criar uma infraestrutura para milhões de clientes se você está fazendo uma aposta. Afinal, é só uma aposta que pode ou não dar certo.

Marketing: o investimento de marketing acompanha o de desenvolvimento. Em uma aposta, você deve investir para ganhar mercado.

ESTRELAS

Quando o produto deixa de ser aposta e passa a ser estrela, as preocupações mudam:

Desenvolvimento: as estrelas também requerem esforço alto de de-

envolvimento. O mercado continua em franco crescimento, e seu produto continua precisando se adaptar a esse mercado.

Operações: quando sua aposta vira uma estrela, você deve se preocupar com a escalabilidade de seu produto. Por isso, você precisa pensar em como dobrar, triplicar ou mesmo multiplicar por 10 sua base de clientes. O que você precisa fazer com sua infraestrutura para chegar nesse tamanho? E processos manuais devem ser minimizados ou, preferencialmente, eliminados. Processos manuais, além de serem sujeitos a erros aleatórios, não escalam.

Marketing: em uma estrela, você também deve investir, pois o mercado está crescendo rápido e, apesar de você já ter um bom *share* de mercado, você precisa continuar investindo para acompanhar seu crescimento.

VACAS LEITEIRAS

Eventualmente, sua estrela pode virar uma vaca leiteira. Nesse momento, você deve direcionar seus esforços da seguinte forma:

Desenvolvimento: as vacas leiteiras precisam de um esforço moderado de desenvolvimento, apenas o desenvolvimento necessário para manter o produto estável e com o mínimo possível de operação manual.

Operações: quando um produto vira vaca leiteira, a escalabilidade dele deve ser excelente. Sem processos manuais e sem dores de cabeça.

Marketing: já quando seu produto vira uma vaca leiteira, seu crescimento é orgânico, você já é o líder (ou um dos líderes) do mercado, e já não precisa mais investir tanto em marketing para esses produtos.

ABACAXIS

Quando um produto vira um abacaxi, quer seja por não cruzar o abismo, ou após a maturidade (como visto no *Capítulo 23 – Fim de vida*), as alocações de esforço e investimento deverão ser ajustadas, conforme vemos a seguir:

Desenvolvimento: nos abacaxis, você deve tirar todo o esforço de desenvolvimento. Se você ainda tem de desenvolver algo para um produto abacaxi, algo está errado. Provavelmente, ele ainda não é um abacaxi.

Operações: a principal origem de abacaxis são as apostas. Nesse caso, o produto não tinha grandes investimentos em escalabilidade. E assim deve continuar! Caso ele tenha chegado à fase de abacaxi após a maturidade, muito provavelmente sua base de clientes deve estar diminuindo e, conseqüentemente, suas preocupações com escalabilidade também.

Marketing: em um abacaxi, não é mais necessário investir em marketing.

EXEMPLOS PRÁTICOS

O primeiro exemplo é o Google, que citei no *Capítulo 31 – Já está pensando em seu novo produto? Não? Então já está atrasado*. Antes de mais nada, preciso deixar claro que:

- 1) Não sei se eles usam matriz BCG para a gestão de portfólio de seus produtos; e
- 2) Se usarem, não sei se a classificação que fiz a seguir bate com a forma que eles enxergam os seus produtos.

Bom, ressalva feita, vamos à matriz BCG. Não coloquei todos os 177 produtos ativos mais os 79 produtos descontinuados, pois não daria para enxergar nada. Selecionei alguns produtos em cada quadrante para dar uma ideia. Algumas apostas:

- **Google App Engine:** mercado de Cloud, competindo de frente com Amazon, e mercado de PaaS, competindo com Heroku e com o Jelastic Cloud Locaweb.
- **Waze:** mapa com rede social, Google tentando entrar no mercado de redes sociais por uma outra porta.
- **Google Driverless Car:** aqui não há um mercado com crescimento forte, pois sequer há um mercado. Google está tentando criar um mercado novo.

Dentre as estrelas, podemos citar YouTube, onde Google é o líder do mercado e este continua em franco crescimento. Além do YouTube, tem também o Android, que veio competir com iOS e hoje já tem posição relevante no mercado.

A grande vaca leiteira do Google é o Search com os anúncios de AdWords. O mercado de anúncio de busca é totalmente dominado pelo Google e, se você fizer uma busca no Google (!), dá para encontrar algumas matérias falando de crescimento desse mercado da ordem de uns 15%.

Por fim, alguns abacaxis descontinuados são o Google Wave, sistema de colaboração *real-time* que “ia matar o e-mail”, e o Google Health, que permitia às pessoas guardarem e gerenciarem suas informações médicas em um único lugar.

Agora ,algo que posso falar com conhecimento de causa, a Locaweb. Aqui usamos a matriz BCG e alguns de nossos produtos estão no exemplo visto a seguir. Preferi não colocar todos os produtos para simplificar a explicação, mas no nosso estão todos os mais de 25 produtos.

Como apostas temos:

- **Jelastic Cloud:** PaaS que lançamos para competir com Amazon, Google App Engine e Heroku.
- **Eventials:** startup de transmissão de eventos ao vivo.
- **TrayCheckout:** serviço de pagamento, nas linhas de PayPal, Pag-Seguro, bCash e MoIP.

Nossas estrelas são os produtos Cloud Server e E-mail Marketing, cujos mercados estão em franco crescimento. Nossa vaca leiteira é nosso primeiro produto, a Hospedagem de Sites, e um abacaxi que descontinuamos em 2015 é o WebChat.

CONCLUINDO

Pronto, agora você não só sabe o porquê é necessário diversificar o portfólio de produtos e como diversificá-lo como também sabe como gerenciar um portfólio de produtos com vários deles. A matriz BCG, apesar de ser uma ferramenta antiga, é bastante útil para entender em que fase do ciclo de vida seu produto está bem como que tipo de investimentos e esforços é preciso fazer em cada produto.

Temos usado a matriz BCG na Locaweb há uns 3 anos, e é uma ferramenta realmente muito útil. A cada trimestre, revisitamos nossa matriz BCG para entender se é necessário fazer alguma movimentação e, conseqüentemente, realocar nossos esforços e investimentos.

No próximo capítulo, vou falar sobre a opção de não diversificar – ou seja, de manter o foco em um único produto –, e veremos quais as vantagens e desvantagens de cada estratégia e quando cada uma é mais apropriada.

CAPÍTULO 34

FOCO OU DIVERSIFICAÇÃO?

Pronto, já lhe convenci de que quem não diversifica pode ficar em situação complicada, pois empresa de um único produto acaba morrendo cedo ou tarde. Expliquei também estratégias para diversificação de portfólio e mostrei como gerenciar um portfólio de produtos.

Por outro lado, existem vários exemplos de empresas que optaram pelo foco. Uma história bem interessante de foco é a empresa 37signals. Eles eram uma empresa de desenvolvimento de sites. Para ajudá-los no acompanhamento dos projetos de desenvolvimento de site, eles desenvolveram um software que permitia maior visibilidade do andamento do projeto para todas as pessoas envolvidas nele, incluindo o cliente. Os clientes gostavam de interagir nesse software e pediam ao pessoal da 37signals para usar esse software em outros projetos de suas empresas. Nesse momento, a 37signals decidiu transformar esse software em um produto que pudesse ser comercializado, e deram a ele o nome Basecamp. Em pouco tempo, eles pararam de desenvolver software sob demanda e focaram toda atenção no produto Basecamp.

Depois de algum tempo, seguindo a estratégia de diversificação de portfólio que descrevi nos capítulos anteriores, eles lançaram mais produtos: Highrise, um sistema de gestão de contatos; Backpack, sistema de comunicação interna; e Campfire, sistema de chat para empresas. No início de 2014, a 37signals decidiu adotar uma nova estratégia. Decidiu que, dali para a frente, se focaria 100% no Basecamp. Os outros produtos não iriam mais aceitar novos clientes. E a “cereja do bolo” dessa estratégia foi a mudança do nome da empresa, que deixaria de se chamar 37signals para passar a se cha-

mar Basecamp. Essas mudanças acabaram sendo tema de um artigo da Harvard Business Review intitulado “*Basecamp’s Strategy Offers a Useful Reminder: Less Is More*” (A estratégia da Basecamp oferece um lembrete útil: menos é mais) em 2014 – disponível em <https://hbr.org/2014/02/basecamps-strategy-offers-a-useful-reminder-less-is-more> –, no qual o autor diz:

“É uma tendência natural do ser humano querer fazer mais. A maioria de nós tem dificuldade em se distanciar de oportunidades tentadoras, quer seja na mesa de jantar, quer seja no trabalho. Por isso acabamos com indigestão em casa, ou sobrecarregados no trabalho. Por isso é preciso muita disciplina, e até mesmo coragem, para emagrecer, tanto fisicamente quanto estrategicamente.”

—Ron Ashkenas

Foco não é uma estratégia tão rara. Alguns outros exemplos do mundo do software:

- **Facebook:** apesar de ter perfil de pessoas, página de grupo e página de empresas, além de sistema de anúncios (que poderiam ser considerados como produtos), esses sistemas não deixam de ser todos visões diferentes de um único produto. Sim, eles têm diversificado seu portfólio de produtos, mas sempre por meio de aquisições, como Instagram e WhatsApp. Essas aquisições continuam funcionando como empresas independentes, cada uma delas também 100% focada em seus respectivos produtos.

- **Twitter:** mais uma empresa que tem um porte considerável e que continua 100% focada em seu único produto. Agora tem se focado em um novo grupo de clientes, os anunciantes, mas sempre com foco em seu único produto.
- **LinkedIn:** outra empresa que tem um porte considerável e que continua 100% focada em seu único produto. Agora também tem se focado em um novo grupo de clientes, os anunciantes, mas sempre com foco em seu único produto. Recentemente adquiriu o SlideShare, mas o manteve como empresa independente.
- **Spotify:** exemplo de empresa 100% focada em seu único produto, o *streaming* de músicas.
- **MailChimp:** empresa de software de e-mail marketing. Ela costuma fazer aquisições, mas todas as suas aquisições são dentro do mesmo tema, e-mail marketing e envio de e-mail.
- **DigitalOcean:** empresa de VPS (*Virtual Private Servers*, ou Servidores Privados Virtuais) que também é um bom exemplo de empresa 100% focada em seu único produto.
- **airbnb:** empresa de intermediação entre pessoas que querem alugar imóveis ou cômodos de imóveis por curtos períodos, e pessoas que buscam hospedagem. É uma plataforma 100% focada no seu único produto.

Por outro lado, já falamos do Google com seus 177 produtos e seus mais de 70 produtos descontinuados. Esse vasto portfólio acabou fazendo-os rever sua estratégia de marca, e criar uma “empresa mãe” chamada Alphabet, da qual o Google seria apenas uma das empresas, e várias empresas dos outros produtos do Google passariam a ser independentes. Essa não deixa de ser uma estratégia de aumento de foco, mas, mesmo assim, o Google continua sendo uma empresa de

vários produtos (Search, Adwords, Gmail, Google Apps, Google App Engine, Youtube, Android etc.).

Outro exemplo extremo de diversificação de portfólio é a 3M, que tem um mais de 55.000 produtos em seu portfólio. É isso mesmo, você leu certo, mais de 55.000 produtos em seu portfólio. Imagina só a matriz BCG da 3M. :-P

ENTÃO QUAL É A MELHOR ESTRATÉGIA?

Com os exemplos citados, fica a dúvida: afinal, qual é a melhor estratégia, diversificação ou foco?

Quando eu estava preparando uma apresentação sobre esse tema, chamou-me a atenção a grafia das duas palavras. **Foco** é uma palavra bem curta, de 4 letras, composta de 3 letras distintas, enquanto **diversificação** tem 14 letras, sendo 10 letras distintas, mais duas letras com mudança (o “ç” e o “ã”). Achei curioso que a complexidade da grafia das palavras tem tudo a ver com seu significado.

Para entender qual é a melhor estratégia, precisamos primeiro entender quais os pontos negativos de cada uma delas.

FOCO

Quando uma empresa é focada em um único produto, ela perde a oportunidade de resolver outros problemas de seus clientes, o que pode ser ruim por dois motivos. O primeiro (e bem óbvio) é o fato de perder oportunidade de obter nova receita. O segundo motivo (não tão óbvio) é o risco de perder o cliente, pois quando este procurar quem pode resolver esse outro problema, pode encontrar alguém que não só resolve esse outro problema, mas também o problema inicial que seu produto já resolve, e esse seu cliente pode decidir concentrar tudo nesse novo fornecedor.

Além disso, como já vimos nos *Capítulo 31 – Já está pensando em seu novo produto? Não? Então já está atrasado*, uma empresa de um único produto pode eventualmente morrer, quer seja porque seu produto não cruzou o abismo, ou porque o produto chegou a maturidade.

DIVERSIFICAÇÃO

Por outro lado, a diversificação também tem suas desvantagens. A primeira delas é o fato de se precisar de mais investimento para poder cuidar de mais de um produto. Será necessário ter uma equipe de desenvolvimento para cada um de seus produtos, e isso pode ter um custo alto.

Outra desvantagem são os desperdícios inerentes à estrutura de grupos diferentes trabalhando em coisas parecidas. Por exemplo, na Locaweb temos vários produtos que têm por funcionalidade permitir

que o cliente envie e-mails; o próprio produto de e-mail, a hospedagem de sites, a revenda de hospedagem, o e-mail marketing e o SMTP. Como são grupos diferentes que cuidam de cada um desses produtos, a arquitetura de cada um deles não necessariamente aproveita os aprendizados e a infraestrutura dos outros.

COMO ESCOLHER?

Para poder escolher entre essas duas estratégias, é preciso olhar tanto para fatores internos da empresa quanto para externos.

O fator interno a ser olhado e entendido é a cultura da empresa. Se a sua empresa tem uma cultura que valoriza muito o empreendedorismo, é muito provável que o mais indicado seja a diversificação. Por outro lado, se a cultura da empresa valoriza muito a excelência, o mais apropriado é adotar uma estratégia de foco em um único produto. No caso da Locaweb, sempre tivemos um espírito empreendedor muito forte, desde os fundadores. Apesar de a excelência ser importante para nós, o empreendedorismo é mais.

Contudo, olhar só os fatores internos não é suficiente. É preciso também olhar e entender os fatores externos à sua empresa, ou seja, o mercado. Se você estiver em um mercado pequeno, de baixo crescimento, ou de muita competição, a diversificação é o mais apropriado. Já se você estiver em um mercado mal servido, foco é a melhor opção.

No caso da Locaweb, enfrentamos competição em todas as nossas linhas de produto. Recentemente, temos começado a ter concorrência

internacional atuando no Brasil em todas as nossas principais linhas de negócio. Por isso, a diversificação é a estratégia mais apropriada para nós.

ENTÃO UMA EMPRESA DE UM ÚNICO PRODUTO VAL...

Como expliquei no final do *Capítulo 31 – Já está pensando em seu novo produto? Não? Então já está atrasado*, uma empresa de um único produto vai eventualmente morrer, pois, ou seu produto não cruzará o abismo, ou, se cruzar, vai eventualmente chegar ao fim de vida.

Contudo, como vimos, existem várias empresas que optam por foco em único produto em vez da diversificação de portfólio. Isso significa que essas empresas estão fadadas a morrer? Sim, mas, o fato de elas saberem disso as faz pensar melhor sobre o futuro, e o que acaba acontecendo é que elas não só se preparam para o fim de vida, como planejam o fim de vida e as novas versões do seu produto que virão a seguir.

É o que expliquei no *Capítulo 07 – Como é o ciclo de vida de um produto de software?*, onde contei como o mercado de TV amadureceu uns 30 anos depois que a TV foi inventada e que, por isso, toda hora os seus fabricantes estão inventando algo novo para nos fazer comprar uma nova TV: primeiro eram em preto e branco até chegarem na Smart-TV. Tudo isso para que eles continuassem tendo nova receita de seus clientes após o fim de vida do produto anterior.

Por isso, tanto foco como diversificação são estratégias válidas. O importante é entender bem os prós e os contras de cada uma delas, e entender em que contexto cada uma é a mais apropriada.

CONCLUINDO

Neste capítulo, vimos que nem só de diversificação de portfólio vive uma empresa de produto de software. É possível sobreviver mantendo 100% do foco em único produto. Aliás, várias empresas optam por esse caminho com sucesso.

Na Locaweb, optamos pela estratégia de diversificação, tanto devido a motivadores internos e a cultura empreendedora, quanto motivadores externos, como o constante aumento de concorrência em todas as nossas linhas de negócio.

Com certa frequência, recebemos questionamentos tanto de funcionários quanto de clientes e parceiros sobre nosso portfólio de produtos com mais de 25 produtos. Eles nos perguntam se essa realmente é a estratégia mais acertada para nós. Acreditamos termos adotado a estratégia mais adequada, não só pelas razões já explicadas (cultura empreendedora + mercado competitivo), mas principalmente pelos números. Se a Locaweb não tivesse diversificado seu portfólio de produtos e tivesse mantido o foco apenas no de hospedagem de sites, hoje teríamos apenas 40% de nosso tamanho atual.

No próximo capítulo, falarei sobre como uma empresa deve organizar seu time de desenvolvimento de produto de acordo com o tipo de estratégia de diversificação, ou de foco, adotada.

CAPÍTULO 35

ORGANIZANDO PARA O FOCO E PARA A DIVERSIFICAÇÃO

Quando uma empresa opta pela diversificação, a organização dos times tende a ser mais simples. Para cada produto, haverá um único time com engenheiros (de 2 a 8 pessoas, sendo 1 ou 2 de *front-end* e o resto de *back-end*), mais uma pessoa de QA e um SysAdmin (administrador de sistemas), ou seja, até 10 pessoas na engenharia. Além disso, terá de 1 a 2 pessoas de UX, sendo que uma terá mais foco em design visual, e, por fim, um gestor de produtos. Esse é o *core team* do produto que comentamos no *Capítulo 02 – O que é gestão de produtos de software?*.

Não raro pode acontecer de algumas pessoas do time serem compartilhadas com outros produtos. Os principais candidatos a serem compartilhados são o designer visual, o gestor de produtos, o QA, o SysAdmin e o engenheiro de front-end. Engenheiros de back-end não devem ser compartilhados.

É importante lembrar de que, ao compartilhar uma pessoa entre dois produtos, a sua atenção será compartilhada, o que tem pontos positivos e negativos. O ponto negativo óbvio é que ela dará menos atenção a cada um dos produtos que cuida. Por outro lado, o fato de ela viver duas realidades, ou seja, de cuidar de dois produtos, pode fazê-la trazer as boas práticas de um time para o outro (e vice-versa).

Contudo, mesmo existindo esse ponto positivo, existem outras formas de se trocar boas práticas, sem sacrificar o tempo e a atenção de uma pessoa compartilhada. Por isso, o mais recomendável mesmo é não compartilhar pessoas entre dois produtos. É claro que se você tiver restrições financeiras, essa divisão entre produtos é aceitável, mas procure considerar isso uma situação temporária.

ATENÇÃO

Compartilhar pessoas entre mais de dois produtos é extremamente desaconselhável. O máximo aceitável de compartilhamento é em dois, e isso já terá um impacto considerável em atenção, produtividade e qualidade.

Outro ponto importante desse tipo de estrutura é a importância de manter uma coesão funcional. Isso pode ser feito por meio de uma liderança funcional, ou de uma auto-organização funcional.

A coesão funcional é importante para garantir que exista consistência entre o trabalho dos times, e que cada um aprenda com as boas práticas do outro. Caso contrário, cada time fará um produto que parecerá, não só ser de um time diferente, como também de uma empresa diferente!

Na Locaweb, optamos por manter a coesão por meio de liderança funcional. Temos líderes de UX, de gestão de produtos, de QA e de front-end. Isso nos garante uma boa consistência entre os diferentes produtos. Por exemplo, criamos padrões de interação, o Locaweb Style, assim todos os produtos da Locaweb têm um padrão de interação igual. Um cliente do produto E-mail Marketing sabe que, quando for usar o produto Backup, não terá de aprender a interação tudo de novo. O Locaweb Style está disponível como *open source* (<http://opensource.locaweb.com.br/locawebstyle>).

COMO ORGANIZAR TIMES GRANDES?

Quando seu produto cresce – quer seja em uma empresa de um único produto, quer seja em uma com um portfólio diversificado –, começam as questões sobre como se organizar. Normalmente, isso demora mais em empresas com portfólio de produtos diversificado, pois, sempre que um determinado time cresce um pouco, existe a vontade de pegar algumas pessoas dele para focar em um novo produto.

Já em uma empresa com foco em um único produto, a necessidade de organizar times grandes acontece bem rápido. Não é difícil imaginar mais de 8 engenheiros disponíveis para trabalhar em um produto e, como vimos anteriormente, cada time de produto deve ter, no máximo, de 6 a 8 engenheiros. Como se organizar com times maiores?

O produto deverá ser quebrado em subsistemas. Estes terão certamente algum tipo de integração e interdependência, mas suas arquiteturas devem ser tal que a interdependência seja mínima para que a camada de integração seja a mais simples possível. Cada time desses precisará de engenheiros de back-end e de front-end, QA, UX, SysAdmin e gestor de produtos. Por se tratar de subsistemas de um mesmo produto, esses times podem eventualmente compartilhar o mesmo gestor de produtos, o mesmo UX, o mesmo SysAdmin, o mesmo QA e o mesmo engenheiro de front-end. Aqui o compartilhamento é um pouco mais aceitável e pode existir compartilhamento em até mais de 2 subsistemas.

Contudo, é preciso ficar atento para que essas pessoas compartilhadas entre mais de um subsistema não virem gargalos. O ideal é ter pessoas 100% dedicadas a cada um. Nesse cenário ideal, no qual cada membro dos times que cuidam de cada subsistema é 100% dedicado, é muito importante alguém ter uma visão geral para ajudar na coordenação dos trabalhos entre os times. Como mencionado, é importante minimizar a interdependência desses subsistemas, porém, alguma dependência sempre existirá, e isso precisará ser coordenado. Eventualmente, um gestor de gestores de produto pode ser necessário para ajudar nessa coordenação. Idealmente deveríamos ter um gestor de gestores de produto, um gestor de engenharia que acompanha o trabalho de todos os times de engenharia, e um gestor de UX para ajudar na coordenação do trabalho dos designers de UX de cada subsistema.

Por exemplo, o produto Hospedagem de Sites é dividido em 7 times de desenvolvedores que se focam em subsistemas diferentes do produto. Um time trabalha no desenvolvimento do painel de controle da

hospedagem; e um outro trabalha no subsistema de provisionamento, responsável por instalar, suspender e desinstalar os componentes que compõem a Hospedagem de Sites da Locaweb, isto é, a hospedagem propriamente dita – que pode ser Linux ou Windows –, o banco de dados e o e-mail.

Assim, um outro time cuida do painel de controle e do webmail do e-mail, e, por fim, mais 4 times cuidam dos subsistemas ligados diretamente à infraestrutura, que são os times de infraestrutura Linux, de Windows, de banco de dados e de e-mail.

Temos dois gestores de produtos para todos esses times: um focado nos subsistemas de hospedagem de sites, outro focado nos subsistemas de e-mail. Temos também dois designers de UX, com a mesma separação de foco, além de um gestor de gestores de produto, um de designers de UX e um dos times de engenharia.

Outro bom exemplo, em uma escala consideravelmente maior, é o Spotify, aplicativo de streaming de música criado na Suécia em 2008, que tem mais de 75 milhões de usuários de acordo com os dados de junho de 2015. A empresa tem mais de 1.500 funcionários, todos dedicados a um único produto, e boa parte deles faz parte do time de desenvolvimento do produto.

Recentemente, eles postaram dois vídeos contando um pouco sobre sua cultura e como eles se organizam. Vale a pena assisti-los! Eles encontram-se em <https://vimeo.com/85490944> e em <https://vimeo.com/94950270>.

CONCLUINDO

Vimos nos capítulos anteriores a importância da estratégia de diversificação de portfólio, as diferentes formas de se fazer essa diversificação de produtos, e aprendemos a usar a matriz BCG para a gestão dos portfólios de produtos e de alocação de esforços entre produtos em diferentes fases do ciclo de vida.

Depois de tê-lo convencido de que a sua diversificação é a melhor estratégia, mostrei que a estratégia de foco pode também ser tão boa quanto, a depender de características da cultura da empresa e do mercado no qual o produto estiver inserido. Por fim, neste capítulo, mostrei como organizar os times de desenvolvimento de produto de acordo com a estratégia escolhida. Com isso, concluímos a Parte IV do livro.

Agora, vamos para a última parte, na qual mostrarei a importância do gestor de produtos em diferentes cenários e tipos de empresa.

PARTE V
ONDE USAR GESTÃO
DE PRODUTOS DE
SOFTWARE

Será que gestão de produtos de software só pode ser usada por empresas que comercializam produtos de software, e somente nos times de desenvolvimento que desenvolvem softwares que serão comercializados como produtos? Ou será que outros tipos de empresa se beneficiariam por pensar em seu software como um produto, e de usar as técnicas de gestão de produtos de software para aumentar as chances de sucesso dos que desenvolveram?

E onde devemos colocar a gestão de produtos de software em uma empresa? No marketing? Na área técnica?

É o que veremos nos capítulos finais deste livro.

CAPÍTULO 36

**QUE TIPO DE EMPRESA
PRECISA DE UM GESTOR
DE PRODUTOS?**

Quando pensamos na função de gestão de produtos, sempre imaginamos esta sendo exercida em uma empresa cujo negócio principal é um software oferecido via internet. Contudo, na minha opinião, esse não é o único tipo de empresa que se beneficia em ter um ou mais gestores de produto para ajudar no processo de desenvolvimento de software. Existem outros 3 tipos de empresa que podem, e devem, se beneficiar com o trabalho de um gestor de produtos.

EMPRESAS QUE DESENVOLVEM SOFTWARE NÃO ONLINE

Ainda existe muito software que não é online, que precisa ser instalado em um computador para ser executado localmente, ou acessar dados de um servidor (cliente servidor). Mesmo com o forte crescimento de aplicações SaaS dos mais variados tipos – como ERP, CRM, BI, Supply Chain Management, entre outros –, ainda existe muito softwares dessas e de outras categorias que não são online, ou seja, que são executados no computador local, ou em rede no modelo cliente servidor, conhecidos como software *on-premises*, em oposição ao software online.

Esse tipo de software não vai deixar de existir tão cedo, quer seja por necessidades técnicas ou por questões de política de uso. Não é incomum encontrar empresas que poderiam usar um software online mas que, por política da empresa sobre segurança e privacidade da

informação, queiram manter esses dados e o software que o gerencia dentro da empresa.

No futuro, é muito provável que as políticas de uso e os receios das empresas vão abrandar ao ponto de nenhuma empresa mais querer ter software instalado por questões de segurança, da mesma forma que hoje é bastante raro encontrar empresas ou pessoas que gerem sua própria energia elétrica. Contudo, sempre haverá quem opte por ter software instalado por alguma razão específica, apesar do custo que isso possa representar.

Por outro lado, questões técnicas podem inviabilizar a possibilidade de usar um software online. Basta imaginar situações em que não é possível estar online como, por exemplo, em um avião ou um barco sem conectividade. Aqui também é possível imaginar um futuro onde a conectividade será boa e universal, mas mesmo assim pode haver situações em que rodar o software localmente faça mais sentido.

Ou seja, mesmo que exista esse movimento em direção ao software online, ainda existirá software on-premises por muito tempo. Este, da mesma forma que o software online, é um software que tem de atender as necessidades de seu dono, ao mesmo tempo em que atende as necessidades dos seus usuários.

Por esse motivo, empresas que desenvolvem software não online devem também ter gestores de produto de software em sua equipe de desenvolvimento.

EMPRESAS QUE NÃO TÊM DESENVOLVIMENTO DE SOFTWARE COMO SUA ATIVIDADE PRINCIPAL

Muitas empresas não tem software como seu negócio principal. Veja alguns exemplos na figura a seguir:

Contudo, a maioria é usuária de software. Tem computadores e sistemas internos para auxiliar nos mais variados processos da empresa. À medida que a familiaridade e utilidade dos computadores e sistemas se comprovam, é comum ver essas empresas começando a pensar em ter um ou mais softwares para ajudar na interação com seus clientes.

Nos exemplos da figura, o Fleury tem um sistema de resultados online, o Ipiranga tem um programa de fidelidade, o Magazine Luiza tem forte presença online com e-commerce, e o Colégio Santo Américo, assim como vários colégios, têm sistema de educação a distância para passar lição de casa para os alunos e se comunicar com os pais.

Mesmo esses softwares não sendo seu *core business*, eles fazem parte da estratégia dessas empresas. Por esse motivo, eles devem ser gerenciados por alguém que tenha conhecimento de gestão de produto de software, para garantir que eles atendam tanto os objetivos do seu dono quanto dos seus usuários.

EMPRESAS QUE DESENVOLVEM SOFTWARE SOB DEMANDA

As melhores empresas que desenvolvem software sob demanda estão sempre na fronteira no que se refere a desenvolvimento de software. Usam novas tecnologias, novas linguagens de programação, bancos de dados e arquiteturas; e propõem novas formas de se fazer software, as metodologias ágeis, Scrum, Kanban, Lean.

Aliás, o termo *Product Owner* vem das metodologias ágeis, responsável por construir o backlog, priorizando o trabalho a ser feito de acordo com as demandas do cliente. Ou seja, as empresas que desenvolvem software sob demandam sabem da importância de se ter um gestor de produto no time que desenvolve software. Tanto que usam essa função tanto nos seus produtos quanto nos softwares que fazem sob demanda.

Contudo, normalmente as empresas que fazem software sob demanda assumem que seu cliente sabe gerenciar um software, e apenas atendem suas demandas e requisitos. Elas fazem reuniões com seus clientes perguntando o que eles querem e esperam do software, coletam os requisitos, priorizam-nos de comum acordo com o cliente, e começam a desenvolver o software. Uma boa empresa que desenvolve software sob demanda vai procurar fazer entregas frequentes para que ele possa não só ver o progresso, como validar o que está sendo entregue.

O problema é que o seu cliente não sabe gerenciar um software! Se esse for o primeiro software dele, será ainda pior! Ele sabe gerenciar o seu próprio negócio, e pode até saber contratar software pronto; entretanto, ele não terá a mínima ideia do que é ser dono de um software, e de que o software é algo bastante flexível, devendo se adaptar para atender os objetivos da empresa e dos seus usuários. Isso tudo é novidade para ele.

Por este motivo, as empresas que desenvolvem software sob demanda têm a obrigação de, no pacote do seu desenvolvimento, incluir algum tipo de treinamento ou aconselhamento para preparar seu

cliente para gerenciá-lo. Só assim essas empresas poderão aumentar as chances de o software que está sendo desenvolvido sob demanda atender os objetivos do seu cliente e dos usuários do seu cliente.

CONCLUINDO

Em minha opinião, toda empresa que é dona de um software, ou que desenvolve software para si ou para outras empresas, deve ter um ou mais gestores de produto de software em seu time. Isso aumentará bastante as chances deste ter sucesso, ou seja, atender os objetivos tanto do dono do software quanto dos seus usuários.

Também na minha opinião, as empresas que desenvolvem software sob demanda têm ainda mais uma obrigação nesse ciclo de desenvolvimento de software: ensinar seus clientes sobre gestão de produtos de software, da importância dessa função no seu sucesso, e o que é necessário para fazer uma boa gestão de produtos de software.

CAPÍTULO 37

O PROBLEMA DA TI

Costumo conversar com alguma frequência sobre departamentos de Tecnologia da Informação (TI) de empresas e como eles parecem estar desconectados dessas empresas, muitas vezes assumindo um papel reativo frente às demandas do negócio. É comum ouvir reclamações da área de negócios sobre a TI dizendo que eles quase nunca entregam o que é pedido e que é difícil de entender o que eles falam.

Por outro lado (não menos comum), é normal ouvir o pessoal da TI falando que a área de negócios não sabe o que quer e que não dá para atender “trocentas” demandas de alta prioridade vindas dela. Esse falta de entendimento entre TI e a área de negócios da empresa é tão comum que virou até motivo de charges dos mais variados tipos:

Fonte: <http://vidadeprogramador.com.br/2014/02/19/o-que-ainda-falta/>

Fonte: <http://projectcartoon.com/>

Mas o que está errado? Qual é o problema da TI?

DESENVOLVIMENTO DE SOFTWARE

Para quem vive na parte de TI que tem a ver com desenvolvimento de software, esse problema tem sido endereçado há algum tempo. O Manifesto Ágil (<http://agilemanifesto.org/>), de 2001, deixa isso bem claro:

- Passamos a valorizar colaboração com o cliente mais que negociação de contratos.
- Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
- Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.

Como o pessoal que desenvolve software precisa colocar seu software em algum lugar, eles decidiram envolver o pessoal que cuida do ambiente de produção nessa forma de pensar, que aproxima TI e negócios. Com isso, em 2009 nasceu o movimento DevOps:

DEVOPS

DevOps (amálgama de Desenvolvedor e Operador) é uma metodologia de desenvolvimento de software que explora a comunicação, colaboração e integração entre desenvolvedores de software e profissionais de TI (Tecnologia da Informação). DevOps é a reação à interdependência entre desenvolvimento de software e operações de TI. Pretende ajudar organizações a produzir software e serviços rapidamente.

Fonte: <https://pt.wikipedia.org/wiki/DevOps>

Nesses times que desenvolvem software, é comum a figura do Product Owner ou do gestor de produtos de software, que está definida no *Capítulo 02 – O que é gestão de produtos de software?*.

O PROBLEMA DA TI

Imagine agora a área de TI da Magazine Luiza, do Posto Ipiranga, do Laboratório Fleury, do Colégio Santo Américo, e assim por diante. Essas áreas terão, dentre seu escopo, as seguintes funções:

- Inventário de hardware;
- Inventário e instalação de software;
- Monitoração e métricas de disponibilidade de servidores;
- Gestão de antivírus e antimalware;
- Monitoração de atividades dos usuários;

- Monitoração e gestão de capacidade;
- Gestão de segurança;
- Gestão de *storage*;
- Gestão e monitoração de redes.

Como dá para ver, essas áreas de TI já têm bastante coisa com que se preocupar, e dificilmente vão desenvolver software. Se optarem por desenvolver software, muito provavelmente usarão empresas terceiras para fazer esse desenvolvimento. Mesmo que decidam desenvolver internamente, desenvolvimento de software ainda será um pedaço pequeno da área de TI. A preocupação dessas áreas é de como integrar softwares de mercado e fazê-los funcionar para atender as necessidades do negócio.

O problema é que, diferentemente da função de desenvolvimento de software – que já descobriu a importância de ter um gestor de produto de software para ajudar a entregar resultado mais alinhado com a área de negócios –, todas as outras funções de uma área de TI não contam com essa ponte entre TI e a área de negócios.

UMA POSSÍVEL SOLUÇÃO

Gostaria de propor uma solução para o problema da TI: ter mais pessoas com a função de “gestor de produto”. Acho que esse nome não encaixa muito bem quando o que a área de TI está entregando não é um software, mas o que importa é o papel que essa pessoa terá de criar: a ponte entre as áreas de TI e negócios. Talvez um nome mais

apropriado seja “gestor de negócios”. Essa pessoa teria por responsabilidades:

- Levantar necessidades das diferentes áreas da empresa em relação à área de TI, inclusive identificar e propor soluções para eventuais conflitos entre essas necessidades.
- Quando essas necessidades tiverem impacto no cliente final da empresa, entender esse impacto junto a esses clientes.
- Negociar prioridades de implementação das necessidades das áreas da empresa.
- Trabalhar em conjunto com as equipes de TI para que as entregas feitas estejam de acordo com os requisitos levantados com as áreas da empresa.
- Atuar em conjunto com as áreas solicitantes e com a equipe de TI no relacionamento com eventuais fornecedores e parceiros (como bancos, consultorias etc.).
- Comunicar todas as áreas da empresa sobre novas implementações da área de TI bem como próximas implementações previstas. Preparar treinamentos, quando necessário.
- Trabalhar em conjunto com marketing para comunicar os clientes quando as novas implementações forem *customer facing*.
- Definir, acompanhar e analisar métricas de uso da TI, para utilizá-las como mais uma informação relevante para tomada de decisão.

Para poder assumir essas responsabilidades, essa pessoa precisa ter:

- Experiência em trabalhar com equipes de TI para entrega de projetos de qualidade, dentro dos prazos esperados.

- Entendimento de TI e conhecimento técnico para poder negociar opções de implementação. Ter sido da área de TI não é requerido, mas pode ajudar na função.
- Conhecimento geral sobre os produtos e serviços da empresa bem como sobre o funcionamento das suas diferentes áreas.
- Boa comunicação oral e escrita.
- Habilidade de negociação entre diferentes *stakeholders*, com diferentes prioridades.
- Capacidade de documentação de requisitos e cenários de uso.
- Liderança.

Como dá para ver nessa descrição, essa pessoa tem um perfil mais sênior. Ela será um par do gerente de TI.

Uma dúvida que pode surgir ao ler essa proposta de adicionar um “gestor de negócios” ao time de TI é: por que os gerentes de TI não podem assumir essa função? Até podem, mas ele tem outras preocupações.

Gerente de TI tem dois focos principais: tecnologia e pessoas. Ele precisa estar antenado sobre as tecnologias de sua área, para saber como melhor atender as necessidades que surgirem, e precisa gerenciar o time, pois encontrar e coordenar bons profissionais de TI não é uma tarefa fácil. Colocar no gestor de TI mais essa carga de negócios pode causar uma queda de qualidade nos focos atuais.

No desenvolvimento de software, já percebemos que é melhor ter uma função separada para cuidar da parte de negócios. Por que não aplicar essa mesma separação de papéis para as outras áreas de TI?

USANDO GESTOR DE NEGÓCIOS NA PRÁTICA

Na Locaweb, temos já há muitos anos um time que faz o desenvolvimento e manutenção do que nós chamamos de sistemas centrais da empresa. Esses sistemas têm os dados dos clientes, dos produtos e a relação entre cada cliente e os produtos contratados. Além disso, eles são responsáveis por fazer a cobrança dos serviços prestados e, quando não houver o pagamento, desativar o serviço do cliente. São os sistemas de cobrança da Locaweb.

Durante muitos anos, foram tratados como sistemas na maneira tradicional, ou seja, o próprio gerente da área, além de cuidar do time de desenvolvimento e das questões técnicas, era também responsável por interagir com todas as áreas da empresa para coletar e priorizar as demandas.

Em 2014, decidimos mudar a estrutura da área criando a figura de PO (Product Owner) de sistemas centrais, como par do gerente da área. Pouco tempo depois, percebemos que essa mudança foi essencial, não só para melhorar a qualidade das entregas desse time, como também para desafogar o gerente da área para que ele pudesse se focar nas questões técnicas de desenvolvimento de sistemas. O time ficou mais feliz com a nova forma de trabalhar, assim como todos os clientes internos e externos da área passaram a enxergá-la como uma facilitadora.

CAPÍTULO 38

ONDE FICA A GESTÃO
DE PRODUTOS DE
SOFTWARE EM UMA
EMPRESA?

Após ler o *Capítulo 27 – Qual a diferença entre gestão de marketing de produtos e gestão de produtos?*, sobre as diferenças entre gestão de produtos e gestão de marketing de produtos, pode ficar a impressão de que o melhor lugar para colocar a gestão de produtos em uma empresa é na área de marketing. Entretanto, na verdade, não é tão simples assim. A resposta para a pergunta que é o título desse capítulo é “depende”!

Depende principalmente de sua atividade principal.

EMPRESAS QUE DESENVOLVEM E COMERCIALIZAM SOFTWARE

Para esse tipo de empresa, até o final da década de 1990, a função de gestão de produtos de software ficava no marketing. Algumas empresas antigas de software ainda colocam a gestão de produtos dentro dessa área e, dentre estas, algumas sequer fazem a separação dos papéis de gestor de produtos e gestor de marketing de produtos, descrita no *Capítulo 27 – Qual a diferença entre gestão de marketing de produtos e gestão de produtos?*.

Contudo, esse não é o modelo mais apropriado. O modelo mais pertinente para empresas que desenvolvem e comercializam software requer, em primeiro lugar, a separação de papéis conforme já explicado. Em seguida, é importante que o gestor de produtos de software fique em outra área que não seja a de marketing, para realmente

não haver confusão entre os papéis e não dar ao responsável pela área de marketing a falsa percepção de que seria possível juntar os dois em uma única pessoa.

Uma área bastante comum para o gestor de produtos de software é a de tecnologia, ligada ao CTO (*Chief Technology Officer*) da empresa. Faz bastante sentido pois, como o gestor de produtos trabalha junto com o time de desenvolvimento de software, nada mais natural que eles estejam dentro da mesma área da empresa.

Por outro lado, as empresas mais modernas que fazem desenvolvimento e comercialização de software optaram por criar uma terceira área, a área de gestão de produtos, independente do marketing e da área de tecnologia, criando a figura do CPO (*Chief Product Officer*), algo como um diretor de produtos. A ideia dessa separação é dar isonomia às três áreas, para que cada uma possa “puxar a sardinha para o seu lado”, e o produto final se beneficiar desse equilíbrio.

Normalmente, a área de UX acaba reportando-se também ao CPO, ficando como par dos gestores de produtos. Seguindo nessa linha de isonomia entre as funções, há ainda algumas empresas que têm uma quarta área independente, a área de UX, criando a figura do CDO (*Chief Design Officer*) ou diretor de design.

STARTUPS

Para startups de software, normalmente um dos fundadores tem o papel de gestor de produtos, já que é ele quem tem a visão sobre aonde ele quer chegar com aquele software. Quando a startup cresce um pouco e o fundador que tem o papel de gestor de produtos começa a ser puxado para outras tarefas, é comum trazer alguém para ser o gestor de produtos.

Em alguns casos, traz-se alguém de fora; em outros, é comum escolher alguém do time de desenvolvimento de software para ser gestor de produtos. Essa pessoa terá constante contato com o fundador da empresa e terá de ter muita habilidade para conquistar sua confiança.

Nas primeiras semanas e até meses, é muito provável que o gestor de produto que comece a trabalhar em uma startup seja apenas um mero executor das ideias do fundador. Aos poucos, este deverá conquistar a confiança do fundador para questões mais estratégicas e, eventualmente, essas decisões serão, em sua maioria, tomadas em conjunto pelo gestor e o fundador da startup.

Fui contratado na Locaweb para ser gestor de produtos em 2005, quando a empresa, fundada em 1998, já tinha pouco mais de 100 funcionários. Durante o primeiro ano, meu trabalho era focado no processo de desenvolvimento de novos produtos, sem grandes interações em questões estratégicas. Aos poucos, fui ganhando a confiança do Gilberto Mautner, fundador da Locaweb, que tinha mais foco no produto e na tecnologia, para conversas mais estratégicas. Dessas conversas, saiu a linha de produtos SaaS (*Software as a Service*) da

Locaweb, como o Email Marketing, o PABX Virtual, a Loja Virtual e o SMTP, que hoje representam mais de 20% da receita da empresa.

EMPRESAS QUE NÃO TÊM DESENVOLVIMENTO DE SOFTWARE COMO SUA ATIVIDADE PRINCIPAL

Para empresas que não têm desenvolvimento de software como sua atividade principal, como as que citei no *Capítulo 36 – Que tipo de empresa precisa de um gestor de produtos?*, a decisão sobre onde colocar a gestão de produtos é um pouco diferente. Não faz muito sentido ter uma área independente de gestão de produtos. O software é um auxiliar à atividade principal.

Por exemplo, em um laboratório de exames clínicos, o software serve para facilitar o atendimento e o relacionamento com o cliente; então, pode fazer sentido a gestão de produtos estar ligada ao SAC. Já um colégio que quer implementar um sistema de ensino a distância para complementar a educação presencial de seus alunos pode pensar em ter um gestor de produtos de software na área de ensino. O posto Ipiranga, com seu programa Quilômetros de Vantagens, assim como as companhias aéreas e seus programas de fidelidade, têm esses programas visando aumentar fidelidade de seus clientes. Isso está bastante ligado tanto a SAC quanto a marketing, então pode fazer sentido ter o gestor de produtos em uma dessas duas áreas. Já o Magazine Luiza tem um site para vendas online, então faz sentido o gestor de produ-

tos, que cuida do desenvolvimento da loja virtual, ficar no marketing ou em vendas.

Ou seja, depende muito do tema do software e de quanto este está ligado à atividade principal da empresa. Porém, deve-se sempre procurar colocar a gestão de produtos o mais próximo possível da área que lida com o tema do software e com a atividade principal da empresa.

EMPRESAS QUE DESENVOLVEM SOFTWARE SOB DEMANDA

Para as empresas que desenvolvem software sob demanda, como as citadas no *Capítulo 36 – Que tipo de empresa precisa de um gestor de produtos?*, ao começarem a adotar a prática da gestão de produtos, o mais comum é deixar essa função junto com os desenvolvedores de software. À medida que essa prática de gestão de produtos ganha importância dentro da empresa, pode fazer sentido criar uma área independente para essa função, que pode, além de ser parte do serviço de desenvolvimento de software sob demanda, ser também um serviço independente de consultoria voltado especificamente para capacitar empresas na função de gestão de produtos de software.

CONCLUINDO

Com este capítulo, concluímos a seção sobre onde usar a gestão de produtos de software. Aqui, vimos quais são os lugares mais apropriados para essa função dentro das empresas, e que decidir onde colocá-la depende muito da atividade principal da empresa e da sua maturidade.

CAPÍTULO 39

CONCLUSÃO

No início de 2014, conversei bastante sobre gestão de produtos com o Fernando de la Riva, fundador e CEO da Concrete Solutions, uma empresa de consultoria de TI. Ele estava bastante fascinado com a descoberta que fez sobre a importância da gestão de produtos no ciclo de vida do desenvolvimento de software, tanto que acabou escrevendo um post bem interessante sobre como uma consultoria de software “descobriu” a gestão de produtos – disponível em <http://blog.concretesolutions.com.br/2014/01/its-product-management-stupid>. Essa conversa motivou-me a voltar a escrever sobre o tema.

Como Fernando disse no final de seu post:

“Temos um ecossistema de engenharia de software de primeira linha (pequeno, mas de classe mundial), temos investidores maduros que entendem como ninguém como lidar com volatilidade e temos um mercado grande e quase virgem. Nosso calcanhar de Aquiles é uma incrível incapacidade de entender como se faz produtos digitais, como ciclar rápido e produzir coisas como o Whatsapp, o Netflix e o Google Glass ou como manter times realmente multidisciplinares trabalhando de forma ótima durante 24 ou 26 meses até produzir produtos que as pessoas amem. O desafio é enorme, mas o prêmio também é.”

—Fernando de la Riva

Por isso, me vi com o dever de repassar o que aprendi ao longo desses anos de gestão de produtos. Minha ideia era termos um livro brasileiro sobre o tema, que reflita essa nossa realidade, mostre para nossa indústria de software qual o papel da gestão de produtos no ciclo de

desenvolvimento de software, e motive mais pessoas a estudar o tema e a incluir essa função em seus projetos de desenvolvimento de software.

Espero ter alcançado esse objetivo! :-)

RECAPITULANDO

Para fechar o livro, iniciarei com uma breve recapitulação do que escrevi. O livro foi dividido em 5 partes:

Definições e requisitos: para começar, fiz uma rápida introdução à metodologias ágeis. Algumas das pessoas que leram os primeiros rascunhos acharam que poderia ser útil fazer esse introdução, já que falo sobre determinados aspectos das metodologias ágeis em alguns pontos do livro. Em seguida, defini algumas palavras chave como software, produto, plataforma, gestão de produtos entre outras. Nessa parte falei também sobre as características de um bom gestor de produto e dei algumas dicas para gestores de produto sobre liderança e cultura organizacional.

Ciclo de vida de um produto de software: nessa parte descrevi como é o ciclo de vida de um produto de software e quais são as fases desse ciclo de vida: inovação, crescimento, maturidade e fim de vida. Sobre inovação, falei sobre o que é inovação, como encontrar um problema a ser resolvido, como descobrir se esse problema é de fato uma oportunidade a ser perseguida e como obter retorno com seu produto de

software. Na fase do crescimento, quando o produto foi desenvolvido e lançado, devemos nos preocupar em como gerenciar o produto durante seu crescimento, ou seja, como gerenciar o feedback? O que é um roadmap? Como priorizar as demandas? O que fazer com pedidos especiais? Como dizer não? Que métricas acompanhar? Após o crescimento vem a maturidade. Nessa parte expliquei sobre quando acontece a maturidade e o que fazer se o produto chegar nessa fase. Depois da maturidade, ou quando o produto é desenvolvido mas não dá certo, chega a fase conhecida como fim de vida de um produto de software. Vimos como detectar e o que fazer nessa fase do ciclo de vida de seu produto de software. No final dessa parte, quando já conhecemos todas as fases do ciclo de vida de um produto de software, mostrei qual a diferença entre startup e gestão de produtos de software.

Relacionamento com as outras funções: Como o gestor de produtos deve se relacionar com as diferentes funções da empresa? Engenharia, UX, marketing de produtos, gestão de projetos, vendas, jurídico, financeiro e atendimento.

Gestão de portfólio de produtos: Por que algumas empresas decidem ter mais de um produto? Como elas fazem para gerenciar esse portfólio de produtos? Por que outras empresas preferem se focar em um único produto? Foco ou diversificação, qual é a estratégia mais apropriada? Como organizar o time de desenvolvimento de produtos de acordo com a estratégia escolhida? Esses temas são os que eu considero como sendo tópicos avançados de gestão de produtos de software, e é o que vimos nos capítulos que compõem essa parte do livro.

Onde usar gestão de produtos de software: Será que gestão de pro-

duto de software só pode ser usado por empresas que comercializam produtos de software e somente nos times de desenvolvimento de software que desenvolvem softwares que irão ser comercializados como produtos? Ou será que outros tipos de empresa se beneficiariam de pensar em seu software como um produto e de usar as técnicas de gestão de produtos de software para aumentar as chances de sucesso dos softwares que desenvolvem? E onde devemos colocar a gestão de produtos de software em uma empresa? No marketing? Na área técnica? Esses foram os temas dos capítulos finais do livro.

PRÓXIMOS PASSOS

Depois de ler esse livro, vou recomendar alguns outros livros, em ordem de importância. Já falei sobre esses livros em diferentes capítulos do livro, mas deixo aqui essa lista completa. Primeiro os livros sobre gestão de produtos de software, em ordem de importância. Infelizmente não há tanta literatura sobre o tema, mas os livros a seguir são bem importantes:

Inspired: How To Create Products Customers Love, do Marty Cagan, ex-VP de gestão de produtos no eBay. É o manual de gestão de produtos de tecnologia. Explica todos os principais conceitos relacionados à gestão de produtos. Para as pessoas que vem trabalhar comigo no meu time de coordenação de produtos da Locaweb, esse livro é leitura obrigatória.

Agile Product Management with Scrum: Creating Products that

Customers Love, do Roman Pichler, consultor de implementação de metodologias ágeis. Nesse livro ele mostra como *product owners* podem criar produtos de sucesso usando metodologias ágeis.

42 Rules of Product Management (2nd Edition): Learn the Rules of Product Management from Leading Experts around the World, esse livro é uma coletânea de 42 ótimos artigos escritos por especialistas em getão de produtos de software.

E aqui estão os livros que ensinam técnicas para criação de produtos de software:

Guia da Startup, Como startups e empresas estabelecidas podem criar produtos web rentáveis, de minha autoria, publicado em 2012. Nesse livro eu falo sobre inúmeras técnicas de desenvolvimento de produto de software e as ilustro com exemplos práticos baseados na minha experiência com o ContaCal e com a Locaweb e em conversas que tive com pessoas responsáveis pelo desenvolvimento de produtos de software de outras empresas. Nesse livro falo de vários assuntos relacionados a desenvolvimento de produto de software.

Direto ao Ponto: Criando produtos de forma enxuta, do Paulo Caroli onde ele compartilha uma sequência de atividades rápidas e efetivas para entender e planejar a criação de produtos enxutos, baseadas no conceito de produto mínimo viável.

Getting Real: The smarter, faster, easier way to build a successful web application, de Jason Fried, David Heinemeier Hansson e Mat-

thew Linderman. Esse livro conta como o pessoal da 37signals fez seus produtos de sucesso. Tem várias dicas práticas muito bacanas. Tem versão em português na web, chamada *Caindo na real*, em http://gettingreal.37signals.com/GR_por.php.

The Entrepreneur's Guide to Customer Development: A cheat sheet to The Four Steps to the Epiphany, de Brant Cooper e Patrick Vlaskovits. Steve Blank, empreendedor em série do Vale do Silício, escreveu um livro intitulado *The Four Steps to the Epiphany: Successful Strategies for Products that Win*, que trata de startup de forma genérica, mas que cria um conceito muito importante, o de *customer development*, ou desenvolvimento

The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, do Eric Ries, que é muito amigo do Steve Blank. Esse livro foi resultado do blog *Startup Lessons Learned* que ele escreveu e continua escrevendo sobre as suas experiências com sua startup. Também é focado em startups de forma geral, não só em startups de produto de software. Chega até a falar sobre startup de uma ONG. Fala de conceitos muito bastante conhecidos como o MVP (*Minimal Viable Product* ou Produto Mínimo Viável) e do ciclo de feedback *Build-Measure-Learn* (Construa-Meça-Aprenda). Foi lançado em português recentemente com o título "A Startup Enxuta".

Existem também inúmeros livros sobre inovação. Basta digitar *innovation* na Amazon para encontrar mais de 67 mil títulos! Claro que não li todos, mas li alguns e desses que li, destaco:

Where Good Ideas Come From: The Natural History of Innovation, do Steven Johnson, autor de vários livros interessantes sobre ciência e conhecimento. Nesse livro ele explica quais os principais ingredientes da inovação, sendo um dos mais importantes a necessidade de equipes multi-disciplinares para que seja possível ver o mesmo problema com diferentes perspectivas.

The Little Black Book of Innovation: How It Works, Ho To Do It, de Scott D. Anthony, sócio junto com o Prof. Clayton Christensen, de uma empresa de consultoria em inovação chamada Innosight. Nesse livro ele define inovação como **algo diferente que tem impacto**. A partir daí ele mostra um guia passo a passo para encontrar e testar oportunidades de inovação.

Por fim, mas não menos importante, o livro que fala sobre a curva de adoção de tecnologia, que define o conceito de ciclo de vida de produtos tecnológicos e é leitura obrigatória para qualquer gestor de produtos de software:

Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. Em 1991, Geoffrey Moore escreveu esse livro onde ele explica que entre os *early adopters* (entusiastas) e a *early majority* (pragmáticos) existe um abismo que muitos produtos não conseguem cruzar. Isso acontece pois os pragmáticos precisam de boas referências para poder comprar um novo produto e os entusiastas normalmente não são boa referência. Daí a dificuldade de alguns produtos cruzarem esse abismo. No livro, Moore também apresenta estratégias para cruzar esse abismo só que, infelizmente, as estratégias propostas são todas estratégias baseadas em estratégias de guerra que, como expliquei no *Capítulo 06 – Cultura Organizacional*, não acho que fazem muito sentido para o mundo dos negócios.

SE FOR PARA GUARDAR UMA ÚNICA COISA DESSE LIVRO

Para terminar este livro, queria pedir um favor. Se for para você se lembrar de uma única coisa de todo o livro, gostaria muito que você guardasse o seguinte: a **gestão de produtos de software** é responsável por saber o “porquê” estamos fazendo software e de lembrar o time durante todo a vida do software desse “porquê”.

Quais os objetivos de negócio que o dono desse software tem de atingir? Quais problemas e necessidades do usuário esse software deve resolver? Durante toda a vida do software, desde a concepção, passando por cada nova funcionalidade, cada correção de bug, ajuste ou mudança em arquitetura, até a decisão de descontinuá-lo, a gestão de produtos deve sempre ter em mente esse “porquê” e usar isso como guia para todas as decisões. Para fazer isso, o gestor de produtos precisa ter **empatia**, a capacidade que uma pessoa tem de se colocar no lugar de outra para compreendê-la.

VAMOS CONTINUAR A CONVERSA?

Como disse no começo do livro, sei que ainda tenho muito a aprender e quero continuar aprendendo sempre. O aprendizado vem da troca de experiências e da conversa, então lhe convido a compartilhar suas experiências lá no meu blog (<http://guiadastartup.com.br>) ou via e-mail (jtorres@jig.com.br).

Estou esperando seus comentários! Até mais!

AGRADECIMENTOS

Antes de mais nada, queria agradecer à Isabel, minha esposa, e à Laura, minha filha, pelo incentivo e pela paciência. Escrever um livro realmente toma tempo, tempo esse que acabei, de uma forma ou de outra, tirando da minha família.

Agradeço também ao Paulo Silveira da Caelum por me convidar para fazer o workshop de gestão de produtos, que foi a forma que encontrei para organizar o meu conhecimento sobre o assunto. Quero agradecer também ao Adriano Almeida da Casa do Código pelo convite para escrever este livro e pela ajuda para que ele tomasse a forma que tem hoje.

Agradeço ao Gilberto Mautner, à família Gora e ao Flavio Jansen pelo muito que contribuíram para que eu pudesse aprender sobre gestão de produtos de software, provendo todo o ambiente e a infraestrutura necessária para que pudessemos fazer vários experimentos de desenvolvimento de produtos de software que, hoje, são serviços da Locaweb e de grande sucesso no mercado de internet.

Obrigado a todas as pessoas com quem tive a honra e o prazer de trabalhar. Cada interação sempre é para mim uma oportunidade de aprendizado e de melhoria, e continuamente aprendo muito com todos com quem trabalhei e trabalho.

Gostaria de agradecer também aos *alpha readers*, pessoas que se dispuseram a ler o livro à medida que ele era feito, e a me mandar seus comentários e sugestões, muitas delas incorporadas à versão final do livro.

Agradeço também a todas as pessoas com quem interagi em palestras, cursos, blogs ou conversas. Todas essas interações são, para mim, mais uma fonte de aprendizado contínuo.

Por fim, a você leitor, agradeço pelo interesse no assunto de gestão de produtos de software. A indústria de software é fundamental para a economia do Brasil, e a função de gestão de produtos de software é essencial para o sucesso dessa indústria. Seu interesse pelo tema, ao ponto de incentivar não só a escrita deste livro como também outros livros, palestras, blogs, encontros, debates e reportagens sobre esse tema é fundamental para que mais conteúdo relevante seja disponibilizado. Isso fará com que mais pessoas se interessem por esse tema e, conseqüentemente, que tenhamos cada vez mais softwares brasileiros de sucesso.

