Elogios Para *O Codificador Limpo*

"Tio Bob" Martin definitivamente eleva o nível com este seu último livro. Ele explica suas expectativas para que um programador profissional gerencie interações, tempo, a pressão, a colaboração e a escolha de quais ferramentas usar. Além de TDD e ADDT, Martin explica não somente o que todo programador que se considera profissional precisa saber, mas também o que precisa seguir, a fim de fazer com que a jovem profissão de desenvolvimento de software cresça.

— Markus Gärtner
Desenvolvedor Sênior de Software
it-agile GmbH/www.it-agile.de/www.shine.de (conteúdo em inglês)

Alguns livros técnicos inspiram e ensinam; outros encantam e divertem. Raramente, um livro técnico faz todas essas quatro coisas. Na minha opinião, os livros de Robert C. Martin sempre atendem a esses requisitos, e *O Codificador Limpo* não é exceção. Leia, aprenda e viva as lições deste livro e você poderá intitular-se um profissional de software.

— George Bullock
Gerente Sênior de Programação
Microsoft Corp.

Se o curso de Ciência da Computação tivesse "leitura obrigatória após a graduação", seria esta. No mundo real, o código ruim não desaparece quando o semestre acaba, você não recebe um A por ter feito uma maratona codificadora na noite anterior da devida tarefa e, pior de tudo, você precisa lidar com pessoas. Então, gurus da codificação não são, necessariamente, profissionais. *O Codificador Limpo* descreve a jornada ao profissionalismo... e faz um trabalho notavelmente divertido a partir dela.

— Jeff Overbey
Universidade de Illinois em Urbana-Champaign

O Codificador Limpo é muito mais que um conjunto de regras e diretrizes. Ele contém conhecimento e sabedoria conquistados a duras penas, que normalmente são obtidos por muitos anos de tentativa e erro, ou ao trabalhar como aprendiz de um mestre. Se você se intitula profissional de software, precisa deste livro.

— R. L. Bogetti
Projetista de Sistema de Liderança
Baxter Healthcare/www.RLBogetti.com

O Codificador Limpo

O Codificador Limpo

Um Código de Conduta para Programadores Profissionais

Robert C. Martin

ALTA BOOKS
EDITORA
Rio de Janeiro, 2012

Entre 1986 e 2000, trabalhei com Jim Newkirk, um colega de Teradyne. Partilhávamos uma paixão pela programação e pelo código limpo. Passávamos noites, finais de tarde e fins de semana juntos, brincando com diferentes estilos de programação e técnicas de desenvolvimento. Estávamos o tempo todo conversando sobre ideias de negócios. Finalmente, criamos juntos a *Object Mentor, Inc.* Aprendi muitas coisas com Jim na medida em que traçávamos nossos planos juntos. Mas uma das coisas mais importantes foi sua atitude de *trabalho ético*; era algo que eu lutava para emular. Jim é um profissional. Tenho orgulho de ter trabalhado com ele e chamá-lo de amigo.

Sumário

Prefácio

Você escolheu este livro, então presumo que seja um profissional de software. Isso é bom; eu também sou. E uma vez que tenho sua atenção, permita-me dizer por que eu escolhi este livro.

Tudo começou há pouco tempo em um lugar não muito distante. Abram as cortinas, luzes e câmeras, Charley...

Anos atrás, eu trabalhava em uma empresa de médio porte vendendo produtos altamente regulados. Você conhece o tipo: sentávamos em cubículos, em um prédio de três andares, diretores e seus superiores tinham escritórios privados, e juntar todo mundo que você precisava para uma reunião levava uma semana.

Operávamos em um mercado bastante competitivo quando o governo tornou um novo produto acessível.

De repente, tínhamos um conjunto de clientes em potencial completamente novo; tudo o que necessitávamos fazer era dar um jeito para que eles comprassem o produto. Isso significava que precisávamos pedir determinado prazo para o governo federal, passar uma auditoria de avaliação em outra data e chegar ao mercado em uma terceira.

Repetidamente, nosso gerente enfatizava a importância daquelas datas. Uma única escorregada nos deixaria de fora do mercado por um ano, e se os clientes não

pudessem se cadastrar no primeiro dia, então todos se cadastrariam com outro e estaríamos fora do negócio.

Esse é o tipo de ambiente no qual algumas pessoas reclamam e outras apontam que a "pressão produz diamantes".

Eu era gerente de projeto técnico, promovido do setor de desenvolvimento. Minha responsabilidade era a de colocar o site no ar diariamente, de forma que os clientes potenciais pudessem baixar informações e, mais importante, os formulários de cadastramento. Meu parceiro na empreitada era o gerente de projetos de negócios, a quem chamarei de Joe. O papel dele era de gerenciar o outro lado, lidando com vendas, marketing e as exigências que não fossem técnicas. Ele também era o cara que gostava do comentário "pressão produz diamantes".

Se você já trabalhou bastante no mundo corporativo, provavelmente já viu os dedos apontados, à procura de culpados e pessoas com aversão ao trabalho, o que é completamente natural. Nossa empresa tinha uma solução interessante para esse problema, comigo e com Joe.

Um pouquinho como Batman e Robin, nosso trabalho era de fazer com que as coisas acontecessem. Eu me encontrava com a equipe técnica diariamente em um canto; repassávamos a agenda todo santo dia, descobríamos o caminho principal e, então, removíamos cada possível obstáculo dele. Se alguém precisava de software; íamos buscá-lo. Se eles "adoravam" configurar o firewall, mas "Caramba, é hora do almoço", comprávamos almoço para eles. Se alguém quisesse trabalhar em nosso ticket de configuração, mas tivesse outras prioridades, Joe e eu falávamos com o supervisor.

Depois, com o gerente.

Depois, com o diretor.

Conseguíamos que as coisas fossem feitas.

É um pouco de exagero dizer que chutávamos cadeiras para o alto, gritávamos e dávamos escândalo, mas usamos cada técnica disponível para conseguir que as coisas fossem feitas, inventamos algumas novas ao longo do caminho e fizemos isso de uma forma ética, que me orgulha até hoje.

Eu pensava em mim mesmo como um membro da equipe, não passando por cima dela para escrever uma instrução SQL ou fazer um breve emparelhamento a fim de obter um código. Ao mesmo tempo, pensava em Joe da mesma maneira, como membro da equipe, não acima dela.

Finalmente, percebi que Joe não partilhava da mesma opinião. Aquele foi um dia muito triste para mim.

Era sexta-feira às 13h; o site estava programado para entrar no ar bem cedo na segunda-feira seguinte.

Havíamos acabado. *ACABADO*. Todos os sistemas estavam preparados, assim como nós. Eu tinha a equipe de tecnologia inteira convocada para a reunião final e estávamos prontos para ligar o interruptor. Mais do que "somente" a equipe técnica, tínhamos conosco as pessoas de negócios do marketing, os donos do produto.

Estávamos orgulhosos. Era um bom momento.

Então Joe apareceu.

Ele disse algo como, "Más notícias. O jurídico não tem os formulários de cadastro ainda, então não podemos entrar no ar".

Isso não era grande coisa; tínhamos sido detidos por uma coisa ou outra ao longo de todo o projeto e tínhamos a rotina Batman/Robin na ponta da língua. Eu estava pronto e minha resposta foi essencialmente: "Ok parceiro, vamos fazer isso mais uma vez. O jurídico fica no terceiro andar, certo?".

Aí as coisas ficaram estranhas.

Ao invés de concordar comigo, Joe perguntou, "Do que você está falando, Matt?".

Eu disse, "Você sabe. Nosso velho truque. Estamos falando de quatro arquivos em PDF, certo? Eles estão feitos; o jurídico só precisa aprová-los? Vamos aparecer em seus cubículos, dar uma olhada feia e conseguir que isso *seja feito*!".

Joe não concordou com o que eu disse e respondeu, "Nós entramos no ar um pouco depois na semana que vem. Não é grande coisa".

Você provavelmente pode adivinhar o resto de nossa conversa; foi algo mais ou menos assim:

Matt: "Mas por quê? Eles podem fazer isso em poucas horas."

Joe: "Pode levar mais do que isso."

Matt: "Mas eles têm todo o final de semana. Tempo o suficiente. Vamos lá!."

Joe: "Matt, eles são profissionais. Não podemos simplesmente encará-los e pedir que sacrifiquem suas vidas pessoais por nosso pequeno projeto."

Matt: (pausa) "... Joe... o que você acha que temos feito com a equipe de engenharia nos últimos quatro meses?."

Joe: "Sim, mas esses são profissionais."

Pausa.

Respire.

O. Que. Joe. Acabou. De. Dizer?

Na época, eu achava que a equipe técnica era profissional no melhor sentido da palavra.

Relembrando isso, contudo, não tenho tanta certeza.

Vamos dar uma olhada na técnica Batman e Robin uma segunda vez, de uma perspectiva diferente. Eu achava que estava extraindo o melhor desempenho da equipe, mas suspeito que Joe estava jogando, com a suposição implícita de que a equipe técnica era seu oponente. Pense no assunto: por que era necessário correr por todos os lados, chutar cadeiras e inclinar-se sobre as pessoas?

Não deveríamos ter sido capazes de perguntar à equipe quando eles teriam o trabalho feito, obter uma resposta firme, acreditar no que nos foi dito, e não sermos queimados por aquela crença?

Certamente, como profissionais, nós deveríamos... e, ao mesmo tempo, não podíamos. Joe não confiava em nossas respostas, e se sentia confortável

microgerenciando a equipe técnica – e ao mesmo tempo, por algum motivo, ele confiava na equipe jurídica e não estava disposto a microgerenciá-la.

O que isso quer dizer?

De alguma maneira, a equipe jurídica havia demonstrado profissionalismo de uma forma que a técnica não tinha.

De algum modo, o outro grupo havia convencido Joe de que eles não precisavam de uma babá, que não estavam jogando, e que precisavam ser tratados como colegas a serem respeitados.

Não, eu não acho que isso tenha algo a ver com certificados extravagantes pendurados nas paredes ou alguns anos a mais de faculdade, embora esses anos possam ter incluído uma pequena porção de treinamento social implícito sobre como se comportar.

Desde aquele dia, anos atrás, pergunto-me como a profissão técnica precisaria mudar para ser considerada profissional.

Assim, tive algumas ideias. Bloguei um pouco, li bastante, busquei aperfeiçoar minha vida profissional e ajudar alguns outros. Contudo, não sabia da existência de livro algum que expusesse um plano, que deixasse as coisas inteiramente explícitas.

Então um dia, do nada, recebi uma oferta para revisar o rascunho de um livro; este que você segura em suas mãos agora.

Este livro lhe dirá, passo a passo, exatamente como você deve se apresentar e interagir como um profissional. Não com um clichê banal, não com apelos a pedaços de papel, mas o que você pode fazer e como fazê-lo.

Em alguns casos, os exemplos são palavra por palavra.

Alguns desses exemplos têm respostas, contrarréplicas, esclarecimentos e até conselhos, sobre o que fazer se a outra pessoa tentar “simplesmente ignorá-lo”.

Ei, olha só, lá vem o Joe de novo, fora do palco desta vez.

Aqui estamos, de volta à Grande Empresa, comigo e com Joe, mais uma vez no grande site do projeto de conversão.

Só que, desta vez, imagine um pouco diferente.

Em vez de esquivar-se de seus compromissos, a equipe técnica de fato os cumpre. Em vez de esquivar-se de estimativas ou deixar alguma outra pessoa fazer o planejamento (e depois, reclamar dele), a equipe técnica de fato se organiza e torna os compromissos reais.

Agora, imagine que a equipe esteja realmente trabalhando junta. Quando os programadores são bloqueados pelas operações, eles apanham o telefone e o administrador de sistemas realmente começa a trabalhar.

Quando Joe vem apagar um incêndio para que o ticket 14321 funcione, ele não precisa fazer isso; poderá ver que o DBA está trabalhando diligentemente e não surfando na web. De forma parecida, as estimativas que ele obtém da equipe parecem consistentes, e ele não tem a sensação de que o projeto é prioritário em algum momento entre o almoço e a checagem de e-mail. Todos os truques e tentativas de manipular a agenda não vão de encontro a "Nós vamos tentar", mas sim, "Este é nosso comprometimento; se você quiser estabelecer suas próprias metas, vá em frente".

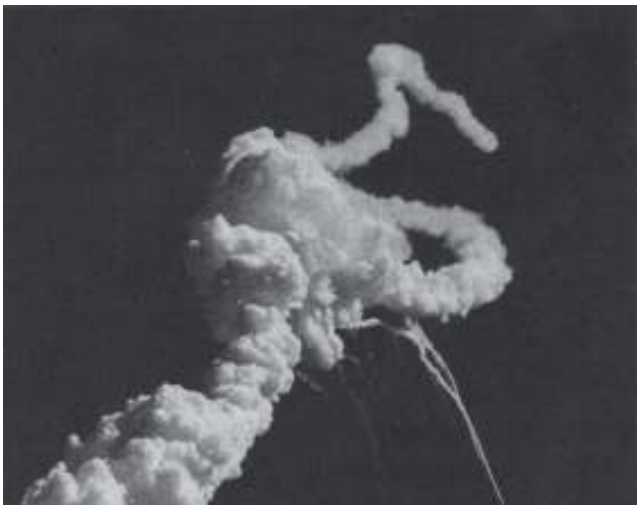
Após um tempo, suspeito que Joe começaria a pensar na equipe técnica como, bem, profissionais. E ele estaria certo.

Os passos para transformar nosso comportamento de técnicos para profissionais? Você os encontrará no restante deste livro.

Bem-vindo ao próximo passo em sua carreira; eu suspeito que você irá gostar.

— Matthew Heusser
Naturalista de Processo de Software

Introdução



Às 11h39min de 28 de janeiro de 1986, apenas 73.124 segundos após o almoço e à altitude de 48.000 pés, a nave espacial Challenger foi partida em pedacinhos por uma falha na turbina direita do foguete propulsor (SRB). Sete corajosos astronautas, incluindo a professora do Ensino Médio Christa McAuliffe, morreram. A expressão no rosto da mãe de McAuliffe, enquanto observava a morte de sua filha a nove mil milhas acima de sua cabeça, me assombra até hoje.

A Challenger se partiu porque gases quentes de escape, na turbina (SRB) danificada, vazaram entre os segmentos de seu casco, se espalhando pelo tanque de

combustível externo. O fundo do tanque principal de hidrogênio líquido explodiu, inflamando o combustível e jogando o tanque para frente, chocando-o contra o tanque de oxigênio líquido que ficava na parte de cima. Ao mesmo tempo, a SRB se destacou do suporte traseiro e rodou em torno do dianteiro. O nariz da nave foi rompido pelo tanque de oxigênio líquido. Esses vetores de forças anormais fizeram com que a nave inteira, se movendo bem acima do Mach 1.5, rodasse contra a corrente de ar. Forças aerodinâmicas rapidamente a despedaçaram.

Entre os segmentos circulares da SRB havia dois anéis concêntricos de borracha sintética. Quando os anéis foram aparafusados, eles foram comprimidos, formando uma vedação onde os gases de exaustão não deveriam ter sido capazes de penetrar.

Mas no anoitecer anterior ao lançamento, a temperatura na plataforma de lançamento chegou a -8ºC, 23 graus abaixo da temperatura mínima específica para os anéis, e 33 graus mais baixos do que qualquer lançamento já feito. Como resultado, os anéis ficaram muito rígidos para bloquearem os gases quentes. Com a turbina em chamas houve um pulso de pressão, à medida que os gases quentes acumulavam-se rapidamente. Os segmentos do dínamo incharam do lado de fora e relaxaram a compressão dos anéis. A rigidez evitou que eles mantivessem a vedação, então parte dos gases vazou e vaporizou os anéis em um arco de 70 graus.

Os engenheiros da Morton Thiokol que desenharam a SRB sabiam que havia problemas com os anéis e eles já teriam sido reportados aos gerentes da Thiokol e da NASA sete anos antes. De fato, os anéis de lançamentos anteriores foram danificados de maneiras semelhantes, embora não o suficiente para causar uma catástrofe. O lançamento mais frio teve a experiência mais prejudicial. Os engenheiros tinham desenhado um reparo para o problema, mas a implementação havia sido adiada por um longo tempo.

Os engenheiros suspeitavam que os anéis enrijecessem quando ficava frio. Também sabiam que as temperaturas para o lançamento da Challenger eram mais baixas do que em qualquer outro lançamento feito anteriormente, e bem abaixo da linha vermelha. Ou seja, os engenheiros *sabiam* que o risco era bastante alto. Eles agiram em conformidade a esse conhecimento. Escreveram memorandos levantando bandeiras vermelhas gigantescas. Recomendaram fortemente que os gerentes da Thiokol e da NASA não fizessem o lançamento. Em uma reunião de última hora, ocorrida algumas horas antes do lançamento, os engenheiros apresentaram seus

melhores dados. Eles se exaltaram, tentaram persuadir os gerentes e protestaram. Mas no final, foram ignorados.

Quando chegou a hora do lançamento, alguns dos engenheiros se recusaram a assistir a transmissão porque temiam uma explosão na plataforma de lançamento. Mas à medida que a Challenger começou a voar graciosamente pelo céu, começaram a relaxar. Momentos antes da destruição, enquanto observavam a nave passar pelo Mach 1, um deles disse que eles tinham "se esquivado de uma bala".

Apesar de todos os protestos, memorandos e pedidos dos engenheiros, os gerentes achavam que estavam certos e que os engenheiros estavam exagerando. Não confiaram nos dados e conclusões deles. Fizeram o lançamento porque estavam debaixo de uma pressão financeira e política imensa. E, no final, *esperavam* que tudo ficasse bem.

Esses gerentes não foram meramente tolos, eles foram criminosos. As vidas de sete bons homens e mulheres, e as esperanças de uma geração, olhando em direção às viagens espaciais, foram destruídas naquela manhã fria porque aqueles gerentes colocaram seus medos, esperanças e intuições acima das palavras de seus próprios especialistas. Eles tomaram uma decisão que não tinham o direito de tomar. Usurparam a autoridade das pessoas que realmente sabiam: os engenheiros.

Mas e quanto aos engenheiros? Certamente, eles fizeram o que tinham que fazer. Informaram seus gerentes e lutaram muito para fazer valer sua posição. Seguiram pelos canais apropriados e invocaram todos os protocolos certos. Fizeram o que podiam *dentro* do sistema – e, ainda assim, os gerentes os ignoraram. Então, pode parecer que esses engenheiros podem seguir sem qualquer culpa.

Mas, às vezes, eu me pergunto se algum desses engenheiros fica acordado à noite, assombrado pela imagem da mãe de Christa McAuliffe, e desejando que tivesse chamado Dan Rather*.

* Nota do tradutor: jornalista norte-americano que ficou famoso por conta da cobertura do assassinato de John Kennedy, em 1963.

Sobre Este Livro

Este livro tem a ver com profissionalismo de software. Contém muitos conselhos pragmáticos em uma tentativa de responder perguntas como:

- O que é um profissional de software?
- Como um profissional se comporta?
- Como um profissional lida com conflitos, prazos apertados e gerentes pouco razoáveis?
- Quando e como um profissional deve dizer "não"?
- Como um profissional lida com a pressão?

Porém, escondido entre os conselhos mais pragmáticos deste livro, você encontrará uma atitude de luta para se libertar. É uma atitude de honestidade, honra, respeito próprio e orgulho. É uma disposição para aceitar a terrível responsabilidade de ser um artesão e um engenheiro. Essa responsabilidade inclui trabalhar bem e de forma limpa. Ela inclui a boa comunicação e a estimativa confiável. Inclui gerenciar seu tempo e encarar as difíceis decisões de risco-recompensa.

Mas essa responsabilidade inclui outra coisa – algo assustador. Como engenheiro, você tem um profundo conhecimento sobre seus sistemas e projetos que nenhum outro gerente pode ter. Com esse conhecimento, vem a responsabilidade de *agir*.

Bibliografia

(McConnell87): Malcolm McConnelle, *Challenger 'A Major Malfunction'*, Nova York, NY: Simon & Shuster, 1987

(Wiki-Challenger): "Space Shuttle Challenger disaster", http://en.wikipedia.org/wiki/Space_Shuttle_Challenger_disaster

Agradecimentos

Minha carreira tem sido uma série de projetos e colaborações. Embora eu tenha tido muitos sonhos e aspirações pessoais, pareço sempre encontrar alguém com quem partilhá-los. Nesse sentido, me sinto um pouco como os Sith, “Sempre existem dois”.

A primeira colaboração que pude considerar profissional foi com John Marchese, aos 13 anos. Planejamos construir computadores juntos. Eu era o cérebro e ele os músculos. Mostrava onde soldar um fio e ele o soldava. Mostrava onde montar um transmissor e ele montava. Era bastante divertido e passávamos centenas de horas nisso. Na verdade, construímos alguns poucos objetos que tinham um aspecto impressionante, com transmissores, botões, luzes e até teletipos! Claro, nenhum deles fazia coisa alguma de fato, mas eram bastante impressionantes e trabalhávamos duro neles. Para John: Obrigado!

Quando eu era calouro no Ensino Médio, conheci Tim Conrad na aula de alemão. Tim era *esperto*. Nos unimos para criar um computador, ele era o cérebro e eu os músculos. Ele me ensinou eletrônica e deu minha primeira introdução ao PDP-8. Construímos de fato uma calculadora eletrônica binária de 18 bits que funcionava a partir de alguns componentes básicos. Ela podia somar, subtrair, multiplicar e dividir. Levou um ano de finais de semana e todos os feriados de primavera, verão e Natal. Trabalhamos incansavelmente nela. No final, ela funcionava muito bem. Para Tim: Obrigado!

Tim e eu aprendemos como programar computadores. Isso não era algo fácil de ser feito em 1968, mas demos um jeito. Pegamos livros sobre montagem do PDP-8, Fortran, Cobol, PL/1, entre outros. Devoramos todos. Escrevemos programas os quais não tínhamos esperança de executar, pois não tínhamos acesso a computadores. Mas os escrevemos da mesma forma, apenas pela paixão de fazê-lo.

Nosso colégio introduziu na grade um curso de ciência da computação quando estávamos no segundo ano. Eles conectaram um Teletipo ASR-33 a um modem discado de 110-baud. Tinham uma conta no sistema de partilha de tempo Univac 1108 do Instituto de Tecnologia de Illinois. Tim e eu imediatamente começamos a operar aquela máquina. Ninguém mais conseguia chegar perto dela.

O modem era conectado ao atender o telefone e discar o número. Quando você ouvia o sinal de resposta do modem, apertava o botão “orig” no Teletipo, o que fazia com que o modem de origem emitisse seu próprio sinal. Então, desligava o telefone e a conexão de dados estava estabelecida.

O telefone tinha uma trava na discagem. Somente professores tinham a chave. Mas isso não importava, pois descobrimos que era possível discar um telefone (qualquer um) ao digitar o número dele no interruptor do gancho. Eu era baterista, então tinha um bom timing e reflexo. Podia discar o modem com a trava no lugar em menos de dez segundos.

Tínhamos dois Teletipos no laboratório de computação. Um era a máquina online e o outro, uma máquina offline. Ambos eram usados por estudantes para escrever seus programas. Eles deveriam digitar seus programas nos Teletipos com a fita de papel do perfurador envolvida. Todos os toques no teclado eram perfurados na fita. Os estudantes escreviam seus programas em IITran, uma linguagem de interpretação notavelmente poderosa. Os estudantes deixavam as fitas de papel em uma cesta próxima aos Teletipos.

Depois da escola, Tim e eu discávamos o computador (teclando, claro), carregávamos as fitas no sistema do IITran, e então desligávamos. A 10 caracteres por segundo, esse não era um procedimento rápido. Mais ou menos uma hora depois, ligávamos de volta e obtínhamos as impressões, novamente a 10 caracteres por segundo. O Teletipo não separava as listas dos estudantes ao ejetar

as páginas. Ele apenas imprimia uma após a outra e assim por diante; então, nós as separávamos usando tesouras, juntávamos o input com a lista, com um clipe, e os colocávamos na cesta.

Tim e eu éramos os mestres e deuses desse processo. Até mesmo os professores nos deixavam em paz quando estávamos naquela sala. Fazíamos o trabalho deles e eles sabiam disso. Eles jamais pediram que o fizéssemos. Jamais disseram que poderíamos fazê-lo. Jamais nos deram a chave do telefone. Apenas entrávamos e eles saíam – e eles nos davam grande liberdade. Aos meus professores de Matemática, Sr. McDermit, Sr. Fogel e Sr. Robien: Obrigado!

Então, depois que todo o trabalho de casa estava feito, nós brincávamos. Escrevíamos programa após programa para fazer qualquer tipo de coisa louca e esquisita; escrevíamos programas que colocassem círculos e parábolas em gráficos de ASCII no Teletipo; escrevíamos programas de circuitos aleatórios e geradores de palavras aleatórias. Calculamos 50 fatoriais até o último dígito. Passamos horas e horas inventando programas para serem escritos e então colocados em funcionamento.

Dois anos depois, Tim, nosso parceiro Richard Lloyd e eu fomos contratados como programadores pela ASC Tabulating, em Lake Bluff, Illinois. Tim e eu tínhamos 18 anos na época. Decidimos que faculdade era perda de tempo e que tínhamos que começar nossas carreiras imediatamente. Foi aqui que encontramos Bill Hohri, Frank Ryder, Big Jim Carlin e John Miller. Eles deram a alguns jovens a oportunidade de aprender o que a programação profissional era na verdade. A experiência não foi inteiramente positiva, nem negativa. Foi, sem dúvida, educacional. Para todos eles, e para Richard, que comandou e dirigiu a maior parte do processo: Obrigado!

Depois de desistir e de fazer besteira aos 20 anos, fiz um bico como cortador de grama, trabalhando para meu cunhado. Era tão péssimo nisso, que ele teve que me despedir. Obrigado, Wes!

Aproximadamente um ano depois, acabei trabalhando na Outboard Marine Corporation. Na ocasião, estava casado e tinha um bebê a caminho. Eles também me despediram. Obrigado, John, Ralph e Tom!

Então, fui trabalhar na Teradyne onde encontrei Russ Ashdown, Ken Finder, Bob Copithorne, Chuck Studee e CK Srithran (agora Kris Iyer). Ken era meu chefe. Chuck e CK eram meus colegas. Aprendi muito com todos eles. Obrigado, pessoal!

E lá estava Mike Carew. Na Teradyne, nos tornamos uma dupla dinâmica. Escrevemos diversos sistemas juntos. Se você quisesse que algo fosse feito e com velocidade, chamava Bob e Mike. Nos divertimos bastante juntos. Obrigado, Mike!

Jerry Fitzpatrick também trabalhou na Teradyne. Nos conhecemos enquanto jogávamos Dungeons & Dragons juntos, mas rapidamente começamos a colaborar. Escrevemos software no Commodore 64 para dar suporte aos usuários de D&D. Também começamos um novo projeto na Teradyne chamado "A Secretária Eletrônica". Trabalhamos juntos por vários anos e ele tornou-se, e permanece, um grande amigo. Obrigado, Jerry!

Passei um ano na Inglaterra enquanto trabalhava para a Teradyne. Lá me juntei a Mike Kergozou. Juntos, tramamos sobre todo tipo de coisas, embora a maior parte dessas tramoias tivesse a ver com bicicletas e pubs. Mas ele era um programador dedicado e bastante focado em qualidade e disciplina (embora ele talvez discorde). Obrigado, Mike!

Ao voltar da Inglaterra em 1987, comecei a fazer planos com Jim Newkirk. Ambos deixamos a Teradyne (com meses de diferença) e nos juntamos a uma empresa iniciante chamada Clear Communications. Ficamos muitos anos juntos, lutando para conseguir os milhões que nunca vieram. Mas continuamos com nossos planos. Obrigado, Jim!

No final, fundamos juntos a Object Mentor. Jim é a pessoa mais direta, disciplinada e objetiva com quem já tive o privilégio de trabalhar. Ele me ensinou tantas coisas que não posso sequer enumerá-las. Em vez disso, dediquei este livro a ele.

Há tantos outros com quem trabalhei, tantos com quem colaborei, tantos que tiveram impacto em minha vida profissional: Lowell Lindstrom, Dave Thomas, Michael Feathers, Bob Koss, Brett Schuchert, Dean Wampler, Pascal Roy, Jeff Langr, James Grenning, Brian Button, Alan Francis, Mike Hill, Eric Meade, Ron Jeffries, Kent Beck, Martin Fowler, Grady Booch, e uma lista interminável de outras pessoas. Obrigado a todos.

Claro, a maior colaboradora de minha vida tem sido minha amada esposa, Ann Marie. Nos casamos quando eu tinha 20 anos, três dias após ela completar 18 anos. Por 38 anos ela tem sido minha companheira firme, meu leme e vela, meu amor e minha vida. Estou ansioso por outras quatro décadas ao lado dela.

E agora, meus colaboradores e parceiros de planejamento são meus filhos. Trabalho junto com minha filha mais velha, Angela, minha adorada protetora e intrépida assistente. Ela me mantém nos trilhos e nunca permite que eu esqueça um compromisso. Faço planos de negócios com meu filho Micah, o fundador da 8thlight.com. Sua cabeça para negócios é de longe melhor do que a minha jamais foi. Nossa última empreitada, a cleancoders.com, é muito empolgante!

Meu filho mais novo, Justin, acabou de começar a trabalhar com Micah na 8th Light. Minha filha mais jovem, Gina, é engenheira química e trabalha na Honeywell. Com esses dois, os projetos mais sérios acabaram de começar.

Ninguém em sua vida lhe ensinará mais do que seus filhos. Obrigado, crianças!

Sobre o Autor

Robert C. Martin ("Tio Bob") é programador desde 1970. Ele é fundador e presidente da Object Mentor, Inc., uma empresa internacional de desenvolvedores de software e gerentes altamente experientes que se especializaram em ajudar empresas a executarem seus projetos. A Object Mentor oferece consultoria para a melhoria de processos, consultoria de design de software, treinamento, e serviços de desenvolvimento de habilidade para grandes corporações em todo o mundo.

Martin publicou dezenas de artigos em diversas revistas especializadas e é palestrante regular em conferências internacionais e feiras.

Ele é autor e editor de diversos livros, incluindo:

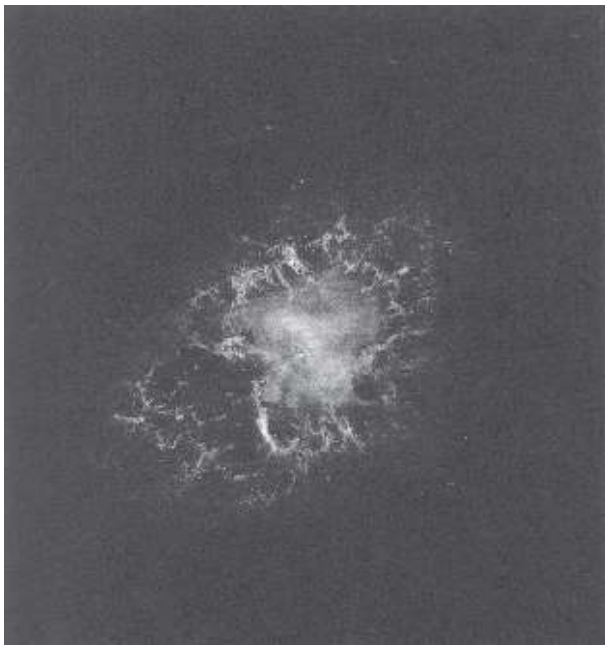
- *Código Limpo*, da editora Alta Books
- *Designing Object Oriented C++ Applications Using the Booch Method*
- *Patterns Languages of Program Design 3*
- *More C++ Gems*

- Extreme Programming in Practice
- *Princípios, Padrões e Práticas Ágeis em C#*
- *Para Entender a Linguística*

Líder na indústria do desenvolvimento de software, Martin foi editor chefe do C++ Report durante três anos e o primeiro presidente da Agile Alliance.

Robert também é fundador da Uncle Bob Consulting, LLC, e cofundador, juntamente com seu filho, Micah Martin, do The Clean Coders LLC.

Sobre a Capa



A maravilhosa imagem na capa, reminiscência do olho de Sauron, é M1, a Nebulosa do Caranguejo. M1 fica em Taurus, por volta de um grau à direita de Zeta Tauri, a estrela na ponta do chifre esquerdo do touro. A Nebulosa do Caranguejo é a sobra de uma supernova que soprou suas rajadas nos céus na data auspiciosa de 4 de julho, em 1054 DC. Distante 6.500 anos-luz de nós, a explosão pareceu para os chineses ser uma nova estrela, aproximadamente tão brilhante quanto Júpiter.

De fato, ela era visível *durante o dia*! Ao longo dos seis meses seguintes, ela lentamente foi se apagando da visão a olho nu.

A imagem da capa é um composto de raios-x e luz visível. A imagem visível foi tirada pelo telescópio Hubble e constitui o envoltório externo. O objeto interno, que se parece com um azul-alvo foi fotografado pelo telescópio de raios-x Chandra.

Essa imagem visível mostra uma rápida nuvem de pó e gás se expandindo, enlaçada a elementos pesados, deixados pela explosão da supernova. Essa nuvem tem agora 11 anos-luz de diâmetro, pesa 4,5 massas solares e está se expandindo na taxa furiosa de 1.500 quilômetros por segundo. A energia cinética daquela velha explosão é impressionante, para dizer o mínimo.

Bem no centro do alvo há um ponto azul brilhante. É onde fica o *pulsar*. Foi a formação dele que fez com que a estrela explodisse em primeiro lugar. Aproximadamente, uma massa solar de material no centro da estrela condenada implodiu em uma esfera de nêutrons, que tinha por volta de 30 quilômetros de diâmetro. A energia cinética daquela implosão, somada à incrível barragem de neutrinos, criados quando todos aqueles nêutrons se formaram, rasgou a estrela e ela se obliterou.

O pulsar está girando por volta de 30 vezes por segundo, lampejando na medida em que gira. Podemos vê-lo piscar em nossos telescópios. Esses pulsos de luz são o motivo pelo qual ele é chamado de pulsar, que é a abreviatura para Estrela Pulsante.

Introdução
Pré-Requisito

(não pule, você precisará dela)

Presumo que você tenha escolhido este livro por ser um programador e está intrigado com a noção de profissionalismo. E deve estar. Profissionalismo é algo que nossa profissão precisa terrivelmente.

Eu também sou programador. Tenho sido por 42 anos[1] e, durante esse período – *deixe-me dizer-lhe* –, já vi de tudo. Fui despedido. Fui laureado. Fui líder de equipe, gerente, funcionário comum e até CEO. Trabalhei com programadores brilhantes e com lesmas[2]. Trabalhei com sistemas de software/hardware de alta tecnologia e

1. Não entre em pânico.
2. Termo técnico de origem desconhecida.

com sistemas corporativos burocráticos. Programei no COBOL, FORTRAN, BAL, PDP-8, PDP-11, C, C++, Java, Ruby, Smalltalk, e uma grande quantidade de outras linguagens e sistemas. Trabalhei com ladrões de contracheque não confiáveis e com profissionais experientes. Esta última classificação é o tópico deste livro.

Nas páginas a seguir tentarei definir o que significa ser um programador profissional. Descreverei as atitudes, disciplinas e ações que considero essenciais a um profissional.

Como sei quais são essas atitudes, disciplinas e ações? Por que precisei aprendê-las da forma mais difícil. Veja, quando consegui meu primeiro trabalho como programador, profissional seria a última palavra que você usaria para me descrever.

O ano era 1969. Eu tinha 17 anos. Meu pai tinha atormentado uma firma local chamada ASC para me contratar como programador temporário meio período (Sim, meu pai fazia coisas assim. Certa vez o vi entrar na frente de um carro em alta velocidade, e com a mão erguida, ele ordenou 'Pare'. O carro parou. Ninguém dizia "não" para meu pai). A empresa me colocou para trabalhar na sala em que todos os manuais de computadores da IBM eram guardados. Eles fizeram com que eu colocasse anos e anos de atualizações nos manuais. Foi ali que eu vi pela primeira vez a frase: "Esta página foi deixada intencionalmente em branco".

Após alguns dias atualizando manuais, meu supervisor pediu que eu escrevesse um simples programa Easycoder[3]. Fiquei extasiado. Nunca havia criado um programa para um computador de verdade antes. Entretanto, eu havia devorado os livros sobre Autocoder e tinha uma vaga noção de como começar.

O programa era simplesmente ler gravações de uma fita e substituir as IDs dessas gravações por novas. As novas IDs começavam em 1 e eram incrementadas em 1 a cada nova gravação. As gravações com as novas IDs tinham que ser escritas em uma nova fita.

Meu supervisor me mostrou uma prateleira que tinha muitas pilhas de cartões perfurados vermelhos e azuis. Imagine que você comprou 50 baralhos de cartas, 25 vermelhos e 25 azuis. Então, você empilha esses baralhos uns sobre os outros. Era assim que essas pilhas de cartões se pareciam. Elas eram agrupadas em vermelho e azul, e cada cor tinha em torno de 200 cartões. Cada cor continha o

3. Easycoder era o montador para o computador Honeywell H200, que era similar ao Autocoder para o IBM 1401.

código-fonte para a biblioteca de sub-rotinas que os programadores normalmente usavam. Os programadores simplesmente apanhavam o baralho de cima da pilha, certificando-se de que não pegassem nada além de cartões azuis e vermelhos, e então os colocavam no fim de seu baralho de programação.

Escrevi meu programa em alguns formulários de código. Formulários de código eram folhas de papel grandes e retangulares, divididas em 25 linhas e 80 colunas. Cada linha representava um cartão. Você escrevia seu programa no formulário de código usando letras maiúsculas e lápis nº 2. Nas últimas 6 colunas de cada linha, escrevi a sequência numérica com aquele lápis. Como de costume, incrementei a sequência numérica em 10, de forma que pudesse inserir os cartões posteriormente.

Os formulários de código iam para as perfuradoras principais. Essa empresa tinha um grupo de mulheres que pegavam os formulários de códigos de dentro de uma enorme cesta e então, "digitavam" nas máquinas perfuradoras. Essas máquinas se pareciam bastante com máquinas de escrever, exceto que os caracteres eram perfurados em cartões ao invés de impressos em papel.

No dia seguinte, as perfuradoras enviaram meu programa de volta por meio de correio interno. Meu pequeno baralho de cartões perfurados veio embrulhado pelos meus formulários de código e um elástico. Eu procurei por erros nos cartões. Não havia nenhum. Então, coloquei o baralho da biblioteca de sub-rotina no final do meu baralho de programa e levei tudo para os operadores de computadores no andar de cima.

Os computadores ficavam atrás de portas trancadas em uma sala ambientalmente controlada com um piso elevado (para todo o cabeamento). Eu bati na porta e um operador pegou meu baralho austeramente e o colocou em outra cesta dentro da sala de computadores. Quando eles chegaram até ela, colocaram meu baralho para rodar.

No dia seguinte, peguei meu baralho de volta. Ele estava embrulhado em uma lista de resultados e atado com um elástico (usávamos muitos elásticos naqueles dias!).

Eu abri a lista e vi que minha compilação tinha falhado. As mensagens de erro na lista eram muito difíceis de entender, então as levei ao meu supervisor. Ele deu uma olhada, murmurou algo, fez algumas anotações rápidas na lista, apanhou meu baralho e pediu que o seguisse.

Levou-me até a sala de perfuração e sentou-se a uma máquina que estava vaga. Corrigiu um por um os cartões que tinham erro, e adicionou mais um ou dois cartões. Explicou rapidamente o que estava fazendo, mas tudo passou como um relâmpago.

Ele levou o novo baralho até a sala de computadores e bateu na porta. Disse algumas palavras mágicas a um dos operadores e então entrou na sala. Fez um sinal para que o seguisse. O operador preparou os drives para a fita e carregou o baralho, enquanto nós observávamos. As fitas rodaram, a impressora fez barulho e então estava terminado. O programa tinha funcionado.

No dia seguinte, meu supervisor agradeceu-me por minha ajuda e me despediu. Aparentemente, a ASC sentiu que não tinha tempo de nutrir um jovem de 17 anos.

Mas meu elo com a empresa não havia terminado. Alguns meses depois, fui contratado em tempo integral na ASC para operar impressoras offline. Essas máquinas imprimiam *junk mail* (lixo eletrônico) a partir de imagens impressas que haviam sido armazenadas em fitas. Meu trabalho era carregar as impressoras com papel, carregar as fitas dentro dos drives, resolver o atolamento de papéis e, se tudo corresse bem, apenas observar as máquinas funcionarem.

O ano era 1970. Faculdade não era opção para mim, nem atiçava qualquer sedução em particular. A Guerra do Vietnã ainda estava em andamento e os campus eram caóticos. Continuei a devorar livros sobre COBOL, Fortran, PL/1, PDP-8 e IBM 360 Assembler. Minha intenção era a de contornar os estudos e seguir direto, e o mais rápido possível, para a obtenção de um trabalho como programador.

Doze meses depois, atingi minha meta. Fui promovido a programador em tempo integral na ASC. Eu e dois bons amigos, Richard e Tim, também com 19 anos, trabalhamos com uma equipe de três programadores, escrevendo um sistema de contabilidade em tempo real para a Teamster[4]. A máquina era um Varian 620i. Era um simples minicomputador, similar em arquitetura ao PDP-8, exceto que tinha dois registros e um vocábulo de 16 bits. A linguagem era Assembler.

Escrevemos cada linha do código naquele sistema. E quero dizer *cada* linha. Escrevemos o sistema de operação, as cabeças de interruptor, os drivers IO, o arquivo de sistemas para os discos, o envolvimento do swapper e, até mesmo, o

4. Nota do tradutor: grupo de sindicatos de trabalhadores poderosíssimo nos EUA.

linkador relocável. Escrevemos tudo em 8 meses, trabalhando de 70 a 80 horas por semana para cumprir o prazo infernal. Meu salário era de U$ 7.200 por ano.

Entregamos aquele sistema e depois pedimos demissão.

Saímos repentinamente e com malícia. Veja, após todo aquele trabalho, e depois de ter entregado um sistema bem-sucedido, a empresa nos deu um aumento de 2%. Sentimos que estávamos sendo roubados e abusados. Vários programadores tinham trabalhos em outros lugares e simplesmente se demitiram.

Eu, entretanto, assumi uma abordagem diferente e bastante infeliz. Junto com um colega, invadimos o escritório do chefe e pedimos demissão juntos fazendo bastante barulho. Isso foi emocionalmente muito satisfatório – por um dia.

No dia seguinte, fui atingido pelo fato de estar desempregado. Tinha 19 anos, sem trabalho, sem graduação. Fiz entrevistas para alguns cargos de programador, porém sem sucesso. Então trabalhei na oficina de conserto de cortadores de grama de meu cunhado durante quatro meses. Infelizmente, eu era um péssimo reparador. No fim, ele teve que me dispensar. Caí em depressão.

Ficava toda noite acordado até às 3 da madrugada, comendo pizza e assistindo antigos filmes de terror na velha televisão preto e branco dos meus pais. Somente alguns dos fantasmas eram personagens dos filmes. Ficava na cama até à 1 da tarde porque não queria encarar meus dias melancólicos. Fiz um curso de cálculo em uma faculdade local e fui reprovado. Estava arrasado.

Minha mãe me chamou de lado e me disse que minha vida estava uma bagunça, e que eu tinha sido um idiota de pedir demissão sem ter um novo emprego, e por tê-lo feito de forma tão emocional e junto a um colega. Ela me disse que jamais se deve sair sem ter algo em vista e, que, devemos sempre agir com calma, tranquilamente e sozinho. Disse também, que eu deveria telefonar para meu antigo chefe e implorar pelo meu trabalho de volta. Ela disse, “Você precisa comer um pouco da torta da humildade”.

Jovens de 19 anos não são famosos por seu apetite por tortas de humildade, e eu não era exceção. Mas as circunstâncias haviam cobrado seu pedágio sobre meu orgulho. No final, telefonei para meu chefe e dei uma grande bocada na torta da humildade. E funcionou. Ele ficou feliz em me recontratar por U$ 6.800 por ano, e eu fiquei feliz em aceitar.

Fiquei mais dezoito meses trabalhando lá, observando minhas maneiras e tentando ser o funcionário mais valioso que podia. Fui recompensado com promoções e aumentos, e um contracheque regular. A vida era boa. Quando saí da empresa foi em termos amigáveis e com uma oferta para um emprego melhor debaixo do braço.

Você pode pensar que eu havia aprendido a lição; que agora eu era um profissional. Longe disso. Aquela foi apenas a primeira de muitas lições que precisei tomar. Nos anos que se seguiram, eu seria despedido de um emprego por perder negligentemente datas importantes, e quase despedido de outro por deixar vazar inadvertidamente informação confidencial para um cliente. Eu assumiria a liderança de um projeto condenado e o levaria até o fundo do poço, sem pedir ajuda a ninguém quando sabia que precisava. Defenderia agressivamente minhas decisões técnicas, mesmo que elas esvoaçassem diante das necessidades dos clientes. Contrataria uma pessoa completamente desqualificada, sobrecarregando meu empregador com uma enorme responsabilidade para se lidar. E, pior de tudo, faria com que outras duas pessoas fossem despedidas por conta de minha inabilidade em liderar.

1 PROFISSIONALISMO

"Ria, Curtin, velho garoto. Foi uma grande peça pregada em nós pelo Senhor, ou pelo destino, ou natureza, o que você preferir. Mas quem quer ou o que quer que a tenha pregado, certamente tinha senso de humor! Rá!".

– *Howard*, O Tesouro de Sierra Madre

Então você quer ser um profissional de desenvolvimento de software, certo? Quer erguer a cabeça e declarar para o mundo: "Eu sou um *profissional*!". Quer que as pessoas olhem para você com respeito e o tratem com consideração. Quer que as mães apontem para você e digam a seus filhos para eles serem como você. Você quer isso tudo. Certo?

Cuidado com o Que Você Pede

Profissionalismo é um termo carregado. Sem dúvida, é um distintivo de honra e orgulho, mas também é um marcador de incumbência e responsabilidade. Os dois andam lado a lado, claro. Você não pode ter honra e orgulho de algo pelo qual não pode ser responsável.

É bem mais fácil não ser profissional. Não profissionais não podem ser responsabilizados pelos trabalhos que fazem – eles deixam isso para seus empregadores. Se um deles comete um erro, o empregador limpa a bagunça. Mas quando um profissional comete um erro, *ele* limpa a bagunça.

O que aconteceria se você permitisse que um defeito passasse por um módulo e custasse à sua empresa R$ 10.000? O não profissional daria de ombros, diria que "essas coisas acontecem" e começaria a escrever o módulo seguinte. O profissional faria um cheque de R$ 10.000 para a empresa[1]!

Sim, a sensação é um pouco diferente quando é seu próprio dinheiro, não é? Mas essa sensação é a mesma que um profissional tem o tempo todo. Na verdade, isso é a essência do profissionalismo. Porque veja, profissionalismo tem tudo a ver com assumir responsabilidades.

Assumindo Responsabilidade

Você leu a introdução, certo? Se não, volte e leia; ela estabelece o contexto para tudo o que se segue neste livro.

Eu aprendi a assumir a responsabilidade ao sofrer as consequências por não fazê-lo.

1. A esperança é que ele tenha uma boa política de Erros e Omissões!

Em 1979, eu trabalhava para uma empresa chamada Teradyne. Era o "engenheiro responsável" pelo software que controlava o sistema base do mini e do microcomputador, que media a qualidade das linhas telefônicas. O minicomputador central era conectado via 300-baud ou conexão dial-up, usando linhas telefônicas para dezenas de microcomputadores satélites que controlavam o hardware de medição. O código era todo escrito em Assembler.

Nossos clientes eram os gerentes de serviços de grandes companhias telefônicas. Cada qual era responsável por 100.000 linhas telefônicas ou mais. Meu sistema ajudava esses gerentes de área a encontrarem e repararem defeitos e problemas nas linhas antes que os clientes percebessem. Isso reduzia a taxa de reclamação que as comissões de utilidade pública mediam e costumava regular as taxas que as companhias telefônicas podiam cobrar. Ou seja, esses sistemas eram incrivelmente importantes.

Toda noite, os sistemas passavam por uma "rotina noturna", na qual o minicomputador central dizia a cada um dos microcomputadores satélites para testar todas as linhas telefônicas sob seu controle. Toda manhã, o computador central puxava a lista de linhas com defeito, juntamente com as características da falha. Os gerentes de serviço da área usavam esse relatório para agendar reparadores a fim de consertar as falhas, antes que os clientes reclamassem.

Em uma ocasião eu enviei um novo lançamento para dezenas de clientes. "Enviar" é exatamente a palavra certa. Escrevi os softwares em fitas e as enviei aos clientes. Eles carregaram as fitas e então reiniciaram os sistemas.

O novo lançamento consertou defeitos menores e adicionou novos recursos que os clientes estavam exigindo. Havíamos lhes dito que proveríamos aquele novo recurso dentro de um determinado prazo. Eu passei a noite inteira trabalhando nas fitas para que elas chegassem no prazo prometido.

Dois dias depois, recebi uma chamada de nosso gerente de serviço de campo, Tom. Ele me disse que vários clientes reclamaram que a "rotina noturna" não se completava e, que, eles não receberam os relatórios. Meu coração disparou, pois a fim de enviar os softwares a tempo, havia negligenciado a rotina de testes. Eu havia testado a maior parte das outras funcionalidades do sistema, mas testar a rotina levava horas e eu precisava despachar o software. Nenhuma das correções de bugs estava no código da rotina, então me senti seguro.

Perder um relatório noturno era algo muito sério. Significava que o técnico que faria os reparos teria menos a fazer naquela data, mas teria trabalho em excesso depois. Significava que os clientes poderiam perceber um defeito e reclamar. Perder os dados de uma noite é o suficiente para fazer com que um gerente de serviços de área chamasse Tom e lhe desse uma bronca.

Liguei o sistema de nosso laboratório, carreguei o novo software e comecei a rotina. Levou várias horas, mas então ele abortou. A rotina tinha falhado. Se eu tivesse rodado esse teste antes do envio, as áreas de serviço não teriam perdido dados e seus gerentes não estariam tostando Tom naquele momento.

Telefonei para Tom para dizer-lhe que eu podia duplicar o problema. Ele me disse que outros clientes telefonaram com a mesma queixa. Então ele me perguntou quando eu poderia consertar o problema. Eu disse que não sabia, mas que estava trabalhando naquilo. Enquanto isso, disse que os clientes deveriam retornar ao antigo software. Ele ficou bravo comigo, afirmando que isso era um golpe duplo para os clientes, já que eles perderam uma noite inteira de dados e não poderiam utilizar o novo recurso conforme prometido.

Foi difícil de encontrar o defeito e testar levava várias horas. O primeiro reparo não funcionou. Nem o segundo. Foram necessárias várias tentativas e, portanto, vários dias, para perceber o que estava errado. O tempo todo, Tom ficava me telefonando com o intervalo de algumas horas para perguntar se eu havia consertado o problema. Ele também certificou-se de que eu estivesse sabendo da dor de cabeça que os gerentes de serviços estavam lhe dando, e o tanto que era embaraçoso para ele dizer que deveriam colocar as antigas fitas de volta.

No final, encontrei o defeito, enviei as novas fitas e tudo voltou ao normal. Tom, que não era meu chefe, se acalmou e deixou todo o episódio para trás. Meu chefe me procurou depois que tudo estava terminado e disse: “Aposto que você não vai fazer isso de novo”. Eu concordei.

Ao refletir, percebi que enviar as fitas sem testá-las tinha sido uma irresponsabilidade. A razão pela qual negligenciei o teste era para poder dizer que o envio fora feito no prazo. Tinha a ver comigo salvando minha pele. Não estava preocupado com o cliente, nem meu empregador. Estava concentrado somente em