

# A Vida Espiritual dos Projetos

Phillip G. Armour

(Publicado originalmente na Communications of the ACM, vol. 45, n. 1, Jan. 2002, p. 11-14)  
(Traduzido e adaptado em 18 de Abril de 2002 por Ricardo Caetano de Moraes)

O desenvolvimento de software é uma atividade inteiramente humana e fortemente intelectual. Existem poucos aspectos na criação de sistemas de computador que não tenham sua base em pensamento, raciocínio e lógica. Nós, desenvolvedores, somos excessivamente cerebrais e tendemos a viver conforme padrões racionais. Mas existem outros aspectos na construção de sistemas e no trabalho em grupo que não são tão literais e mecânicos. Há mais dimensões em um ser humano além da intelectual, e precisamos zelar por elas para sermos verdadeiramente efetivos nos negócios.

Um padre jesuíta, com pouco mais de 1 metro e meio de altura, explicou-me essas dimensões muitos anos atrás. Padre Jim me perguntou: “De onde vem a coragem? O que é compaixão?”. Ele afirmou que as pessoas evoluem em quatro dimensões que são um tanto independentes uma da outra. Elas são as dimensões física, intelectual, emocional e a que Padre Jim decidiu chamar “espiritual”. “Antes que você pense em algo distorcido”, disse ele, no seu legítimo sotaque irlandês, “deixe-me esclarecer o que quero dizer com ‘espiritual’”. Ele prosseguiu dando alguns exemplos das primeiras três dimensões. Correr uma maratona é uma atividade fortemente física, que requer uma dose intensa de treinamento, além de um certo tipo de capacidade corporal. Equilibrar o orçamento doméstico do mês é uma atividade intelectual que requer habilidades de raciocínio abstrato. Ficar irritado é um exemplo de exercer uma emoção. Mas existem muitos atributos que as pessoas possuem, disse Padre Jim, que não se encaixam em nenhuma dessas três categorias.

Coragem, tolerância, compaixão e honestidade, entre outras, são características que nós podemos ter ou não. Em alguns dias, somos mais honestos que em outros. Às vezes, somos mais solícitos com nossos colegas do que costume. A solicitude não é algo físico; ela pode ser manifestar através da execução de algum ato físico, mas ela não é uma capacidade motora, e podemos ter mais dessa capacidade em alguns dias do que em outros. A compaixão não é uma emoção, embora ela seja freqüentemente acompanhada da emoção da tristeza. Muitas vezes, os atributos emocionais e espirituais são fortemente associados, sendo que um dos exemplos mais conhecidos é o relacionamento entre a coragem e o medo. Ser corajoso não é não ter medo; é a capacidade de fazer algo apesar do medo. Correr uma maratona poderia requerer a qualidade espiritual da fortaleza, para superar a experiência física da dor intensa. Equilibrar o orçamento doméstico pode gerar uma emoção genuína.

O argumento de Padre Jim era simples: **Existem capacidades que nós temos como pessoas que não são físicas, intelectuais nem emocionais.** Padre Jim resumiu: “Tomemos cada habilidade humana que não se enquadre em uma das categorias anteriores e vamos chamá-la de ‘espiritual’”.

## Espírito em Projetos de Software

Mas o que isso tem a ver com desenvolvimento de software? Nessa atividade, costumamos empregar muito tempo trabalhando com as dimensões física e intelectual da construção de

sistemas. E o que fazemos com as outras? Especificamente, o que fazemos sobre o espírito dos projetos? Eu jamais vi qualquer pesquisa sobre o espírito dos projetos. Entretanto, minha experiência pessoal tem apontado que o espírito – presente ou ausente em um projeto – tem sido o fator decisivo. Eu trabalhei em projetos com abundância de intelecto e um mínimo de espírito, que foram relativamente mal sucedidos. Por outro lado, trabalhei em projetos que não tinham os melhores intelectos, mas nós tínhamos *alguma coisa* que nos fez bem sucedidos. Certamente, se você perguntar às pessoas que trabalharam nesses projetos, elas todas irão concordar que os projetos foram experiências gratificantes e que trouxeram um grande aprendizado. Somente por isso eles já valeram a pena.

Podemos intelectualmente construir metodologias e processos, podemos fisicamente nos instalar próximos uns aos outros, ou instalar poderosas estações de trabalho; podemos até mesmo tentar gerenciar intencionalmente os componentes emocionais dos projetos. Mas e o espírito? Podemos gerenciar e otimizar os componentes não-físicos, não-intelectuais e não-emocionais dos projetos? Eu creio que sim. De fato, acho que este pode ser um dos nossos maiores desafios como uma indústria, e aquele que verdadeiramente dá um significado ao que fazemos. Pessoalmente, somente fazer mais um sistema não me motiva muito mais. **O que importa para mim é se fazemos a diferença.** O sistema acrescentou valor às vidas das pessoas e às empresas? Aprendemos alguma coisa? Ele é importante? Outras pessoas acham que ele é importante? Nós crescemos como pessoas? Nós melhoramos depois dele?

Como fazer isso? Eis aqui algumas idéias:

**Um código de conduta.** Eu coloco isso em primeiro lugar, mesmo antes do propósito. Tenho observado, particularmente nos últimos anos, que o comportamento de uma equipe pode ser mais importante do que seus objetivos declarados. Isto é, **o código real de conduta associado ao projeto é mais importante e tem mais impacto do que seus papéis, responsabilidades, tarefas, metodologia, linguagem e processos.** A razão para isto é simples: Se o código de conduta *de facto* permite que as pessoas aceitem executar coisas que elas não têm intenção de fazer, na verdade não importa que papéis sejam atribuídos, nada será feito. E se as normas de comportamento do projeto permitem que as pessoas usem qualquer processo que elas desejem, não importa qual processo o projeto selecione para utilização. Os comportamentos que as pessoas demonstram são mais significativos. É importante notar que os projetos *sempre* possuem um conjunto de comportamentos aceitáveis, quer eles sejam claramente definidos ou não. O código de conduta pode ser explícito ou implícito, mas ele está sempre lá. A questão é: o código de conduta ajuda ou atrapalha a equipe?

**Um propósito maior.** Um propósito ou objetivo definido é fundamental para uma equipe. Esse objetivo pode ser mundano ou inspirador, tedioso ou eletrizante. **Para os projetos serem realmente bem sucedidos, o propósito tem que ser alguma coisa que eleve os espíritos das pessoas – alguma coisa em que elas possam acreditar.** Idealmente, o objetivo deveria ser tanto compartilhado quanto participativo. “Compartilhado” significa que a maioria dos membros da equipe tem uma chance de formular o objetivo, e “participativo” significa que ele requer a combinação dos esforços da maioria da equipe para ser alcançado. Há alguns anos, eu trabalhei com um grupo de pesquisadores de software que desejavam ser uma equipe. Mais corretamente, o chefe deles queria que eles fossem uma equipe. Depois de algum tempo, ficou evidente que suas tarefas individuais não tinham quase nada em comum. Eles eram trabalhadores independentes fazendo coisas que não

compartilhavam nenhum laço em comum. A “equipe” era simplesmente uma unidade administrativa. Nós raciocinamos que, a menos que eles desenvolvessem um objetivo *superior* em que todos eles pudessem acreditar, eles jamais seriam uma equipe. Bem, eles não desenvolveram isso, logo não se tornaram uma equipe. Sem um propósito comum e compartilhado, uma equipe não pode ser uma equipe.

**Contribuição.** Outro elemento do propósito que achamos ser necessário à formação de uma boa equipe é a idéia de contribuição. **O resultado final tem que acrescentar valor à vida de alguém, de alguma forma concreta e identificável, para que o objetivo seja inspirador.** Isto invariavelmente significa que o propósito está contribuindo para uma causa maior, ou pelo menos uma causa além do simples engrandecimento pessoal. Incidentalmente, a causa de meramente melhorar a lucratividade de uma empresa normalmente se encaixa nessa categoria. Vagos slogans tais como “foco no cliente” também não funcionam.

**Moral.** Não é só uma frase. Em muitas ferramentas de estimativa de software, existe um fator de produtividade separado e ajustável chamado “moral” ou “motivação”. Por exemplo, usando a ferramenta SLIM-Estimate da QSM para o desenvolvimento de um sistema de telecomunicações com 10 mil linhas de código, um aumento da motivação do seu valor mínimo para o seu valor máximo irá aumentar o Índice de Produtividade de 8,9 para 12,3. Tal projeto poderia ter uma economia de US\$ 604.000 (uma melhoria de 65% sobre US\$ 937.000) e 2,7 meses (uma melhoria de 27% sobre 10,1 meses) no ciclo de desenvolvimento [2]. Não é pouca coisa. **Na vida espiritual dos projetos efetivos, a moral é discutida, analisada e ajustada diariamente.**

**Apoio mútuo.** Equipes são organismos sociais. Nós desempenhamos um papel nas vidas uns dos outros. Em alguns casos, você pode gastar mais do seu tempo acordado com seu colega do que com seu(sua) companheiro(a). As equipes de projeto fornecem uma das maiores ocasiões para interação que podemos experimentar em nosso local de trabalho ou mesmo em nossas vidas. Por conseguinte, **temos uma responsabilidade de nos apoiar mutuamente.** Isso significa ser capaz de dar um ao outro feedback efetivo, tanto positivo quanto negativo, ouvir um ao outro, e, eu ousaria dizer, proporcionar um recurso espiritual no qual as pessoas possam se apoiar. No nível mais baixo, isto poderia significar simplesmente emprestar um ouvido quando necessário. Em níveis mais altos, pode significar validar o valor profissional e pessoal de um colega, e eles fazerem o mesmo com você.

**Liderança.** Em geral, os líderes não fazem equipes. Isto implica um grau de controle e manipulação que bons líderes não usam normalmente. Equipes fazem equipes. **Bons líderes tendem a guiar, orientar, apoiar e desafiar.** Uma das melhores definições de liderança foi dada por Jerry Weinberg quando ele disse “Liderança é o processo de criar um ambiente no qual as pessoas se desenvolvam” [3]. O que Weinberg está dizendo é que liderança (um conjunto de ações, não de palavras) atua no *ambiente* (não nas pessoas). Isto é, em essência, criar uma situação onde as pessoas irão fazer o que elas desejam fazer naturalmente, que é trabalhar juntas, serem cooperativas, ser produtivas, aprender e evoluir. Uma vez que começamos discutindo o código comportamental de conduta, devemos notar que *como* um líder se comporta é muito mais importante do que o que ele(a) *fala*.

**Cuidar de pessoas mais fracas.** É típico em muitas organizações pontuar as pessoas, para avaliar suas competências e habilidades. Em tempos de dificuldades econômicas, os pior

avaliados são aqueles que sairão primeiro. Eles são, afinal de contas, os que têm os piores desempenhos. Normalmente, usamos uma abordagem baseada na curva de distribuição normal: umas poucas pessoas são realmente boas, a maioria está no meio e existem umas poucas pessoas na parte baixa da curva.

Este modelo tem algum apelo, mas ele provavelmente está errado. A falácia é bastante simples, e pode ser facilmente explicada a partir da Estatística. A distribuição normal ou Gaussiana é baseada na suposição de valores aleatoriamente distribuídos de variáveis independentes. No entanto, os comportamentos e habilidades das pessoas não são nem distribuídos aleatoriamente nem independentes.

No mínimo as pessoas selecionaram seu trabalho por si próprias – elas fazem o que fazem porque elas gostam disso, são boas nisso, ou ambas as coisas. Qualquer recrutador que valha alguma coisa deveria estar preenchendo vagas na sua empresa com pessoas realmente boas. Bem, podemos assumir que a melhoria geral na população devida à auto-seleção é aleatoriamente distribuída, o que simplesmente faz com que a curva normal se desloque para a direita. Mas isso provavelmente não é verdade, uma vez que também tende a inclinar a curva para uma distribuição não-normal. Além disso, a suposição de independência é claramente errada. A própria razão pela qual existem equipes e projetos é que o trabalho das pessoas *não é* independente. Uma vez que perdemos a independência das variáveis, a curva normal não pode ser aplicada. Na verdade, **pessoas com baixo desempenho podem cumprir um papel valioso dentro das equipes e das organizações** [1]. Poderíamos considerar a possibilidade de nos livrarmos de pessoas com baixo desempenho crônico ou de pessoas cuja motivação ou ética de trabalho continuamente limita sua efetividade. Mas existem outras razões para baixa performance que são decorrentes de situações específicas. Descartar pessoas porque elas não têm habilidades ou conhecimentos não é uma boa receita para construir equipes ou organizações saudáveis.

Remover alguém que não está se concentrando no trabalho por causa da sua preocupação com sua esposa sofrendo de câncer não é uma boa coisa a fazer. Eu afirmaria que qualquer melhoria feita no desempenho devida à eliminação desta pessoa é mais que compensada pelo dano feito à ética da equipe e da organização. De fato, certa vez trabalhei em um projeto onde nós, como uma equipe, tomamos como objetivo comum a recuperação de um dessas pessoas com baixo desempenho crônico. Não apenas isso ajudou a consolidar a equipe, como nós (finalmente) conseguimos obter um bom desempenho dessa pessoa.

Em nossa fria busca pela eficiência e produtividade através do corte de custos, não deveríamos esquecer que software é feito *apenas* por gente. E as pessoas têm uma série de dimensões nas quais elas precisam evoluir, e uma variedade de capacidades que as fazem efetivas e produtivas.

Padre Jim morreu há alguns anos, mas certamente seu espírito vive naqueles que o conheceram. Para mim, ele realmente mostrou que o negócio de software também é o negócio de lidar com pessoas. Nós programamos máquinas, mas não somos máquinas. É o espírito que nos faz diferentes.

#### REFERÊNCIAS

1. Harvey, J. B. *How Come Every Time I Get Stabbed in the Back My Fingertips Are on the Knife?: And Other Meditations on Management*. Jossey-Bass, San Francisco, CA 1999.

2. QSM Inc. Slim-Estimate (using default and typical values for project costs and other factors). McLean, VA.
3. Weinberg, G. M. *Becoming a Technical Leader: An Organic Problem-Solving Approach*. Dorset House, New York, NY (1986), 12.

---

**PHILLIP G. ARMOUR** (armour@corvusintl.com) é vice-presidente e consultor sênior na Corvus International Inc., Deer Park, IL.

---

