

MICROSOFT CORPORATION

(2ª parte)

A primeira parte do caso sobre a Microsoft Corporation apresenta a situação estratégica que a empresa enfrentava em 1994. O presente caso trata da estrutura organizacional e das práticas administrativas correntes na empresa.

O Antigo Estilo Administrativo

Em meados dos anos 80, o ambiente de trabalho dos programadores na Microsoft era “deliberadamente caótico”. Quase não existia hierarquia. As equipes individuais de desenvolvimento de produto eram pequenas. Geralmente não mais que três pessoas. A criação de software tinha o controle diário direto do presidente fundador Bill Gates. A filosofia de Gates resava que, com menos estrutura, as pessoas poderiam ser mais inovadoras. Ferramentas do software que depois deveriam funcionar juntas eram construídas em unidades totalmente independentes, com um mínimo de troca de informações.

Dentro da Microsoft, as promoções aconteciam porque as pessoas conseguiam descobertas técnicas e não por terem capacidade administrativa. E, para completar as coisas, o estilo mão-na-massa de Gates muitas vezes significava ele passar por cima da hierarquia e realizar mudanças importantes nos objetivos e mesmo em programas, sem lembrar-se de avisar os envolvidos. Como seria de se esperar, existia conflito perene entre o desejo de cada grupo de melhorar o produto cada vez mais e a necessidade de pôr esse mesmo produto na rua.

Dentro da organização adotada e desprovida de estrutura da Microsoft, as crises de desenvolvimento de produto, constantes, eram resolvidas em confrontações agressivas e com muita gritaria de parte a parte. Gates, sempre a par de todo o processo e claramente no controle da empresa, não dava a maior importância a esses encontros que classificava como “comunicações em banda larga superior”. Seu evidente e claro objetivo de “ser o produtor líder de softwares para computadores pessoais” suplantava tudo o mais. Era voz corrente que ele demonstrava “paranóia competitiva”, temendo que pudessem tomar a Microsoft de assalto, assenhorear-se dela ou mesmo a destruí-la.

Gates jamais cessava de querer “vencer” e mesmo destruir tais ameaças.

O Mundo dos PCs Muda

Este estilo funcionou bem para os softwares aplicativos e para os primeiros computadores de mesa, mas perdeu eficiência quando a Microsoft começou a desenvolver sistemas maiores, como o Windows. Mesmo então, Gates se mostrou notavelmente flexível em matéria de organização. Entre 1983 e 1994, reconhecendo as próprias limitações, Gates passou por três diretores (responsável por setor) – James Towne, Jon Shirley e Michael Hallmann, cada um deles contribuindo com uma nova disciplina para a empresa – enquanto tratava de reorganizar seu próprio trabalho.

Numa jogada, mais ambiciosa, enquanto o projeto do Windows, de US\$ 100 milhões, estava em andamento, Gates reorganizou a Microsoft em duas divisões independentes: a System Software e a Business Applications. Cada uma delas era dirigida por um vice-presidente. A idéia visada era a de que – com o MS-DOS assim dividido internamente – os grupos externos de programadores de aplicativos, cujos projetos giravam em torno do MS-DOS, poderiam ter comunicação direta com o grupo da System Software sem revelar informações potencialmente valiosas para a Microsoft Applications.

Mas, ironicamente, o resultado foi que a Microsoft teve muito mais conhecimento sobre todas as atividades em aplicativos de seus concorrentes. Gates procurou aplacar as reclamações do pessoal de fora da empresa, insistindo em criar uma “Muralha da China” entre as duas operações. Os cépticos comentaram que essa medida estava mais para peneira do que para muralha.

Os Melhores e Mais Trabalhadores

O estilo da Microsoft sempre fora o de contratar programadores melhores e de maior capacidade de trabalho, onde quer que estivessem, e colocá-los num ambiente com grande liberdade de ação. Para uma colocação, às vezes entrevistavam-se centenas de pessoas. Não existia orçamento para contratações. O pessoal de RH tinha toda a liberdade para contratar aquele talento especial, na hora em que fosse encontrado.

Ao contratar alguém, a Microsoft não dava atenção à educação formal ou experiência do candidato. Afinal de contas, nenhum de seus fundadores, Bill Gates ou Paul Allen, havia terminado a faculdade. Não importava quantos títulos aparecessem nas credenciais de um candidato. Ele só seria contratado após passar pelo crivo da empresa, respondendo sobre seus conhecimentos em programação e outras qualidades e capacidades.

O processo da entrevista era feroz: de quatro a seis programadores e diretores passavam de uma a duas horas examinando o candidato. Os entrevistadores faziam picadinho do novato; propunham-lhe perguntas técnicas muito difíceis e inesperadamente lhe estendiam lápis e papel dizendo: "resolva este problema". A ênfase era em como pensa o candidato e como ele trabalha, quando sob pressão. Durante muito tempo Gates insistiu em entrevistar pessoalmente todo programador que se candidatasse. E até 1990, Gates era capaz de viajar não importa para onde para descobrir um talento especial.

Dentre as características de organização mais notáveis da Microsoft estavam os "arquitetos", os sete samurais do software que tinham sido conselheiros de Gates, explorando novas tecnologias, e eram autores da estrutura de sistema mais importante. Abaixo deles, cada programador era classificado em 1 de 6 níveis, indo de 10 a 15. se algum programador conseguisse 15 de classificação e se tornasse um arquiteto, era provável que um escritório de advocacia o tornasse um sócio sênior.

Os pedidos empilhavam-se. As equipes de desenvolvimento eram, deliberadamente, conservadas com poucas pessoas. Mesmo com os projetos tornando-se cada vez mais complexos. Por exemplo, a Microsoft tinha apenas 18 desenvolvedores a cargo de todo o projeto de planilhas eletrônicas, enquanto a Lótus, no início dos anos 90, tinha cerca de 120. Conforme Gates: *"Só as equipes reduzidas trabalham direito. Quando começamos o Excel, tínhamos apenas dez pessoas trabalhando nele, eu inclusive. Hoje, sete pessoas trabalham nele."* Cada grupo ocupado com um novo código trabalhava num regime darwiniano, isto é, a cada seis meses os desenvolvedores eram reexaminados e os 5% inferiores, eliminados.

O Processo de Desenvolvimento do Software

Já em 1994, o formato do desenvolvimento de um software na Microsoft havia evoluído para um sistema bem menos caótico. Segundo

o Sr. Robert Muglia, diretor do Windows NT, *“uma das coisas que a Microsoft aprendeu muito bem foi criar produtos que satisfaçam a necessidade dos clientes, concentram-se naquilo que o cliente quer e, ao mesmo tempo, realizam tudo isso com a bossa de quem entende de negócios.”* A Microsoft criou duas funções básicas: gerentes de produtos e gerentes de programas.

Os *Gerentes de Produtos* controlavam o relacionamento com segmentos específicos de clientes. Eram responsáveis pela compreensão continuada das necessidades do cliente do ponto de vista descritivo e pela maioria das apresentações ao cliente, assuntos de venda, propaganda, composição do preço, formação da força de vendas, assuntos relativos ao gerenciamento dos canais, etc.

Os *Gerentes de Programação* trabalhavam com o pessoal do produto, para compreender a fundo as necessidades do cliente em nível técnico, e então detalhar essa compreensão em especificações para o design. Durante todo o processo, os gerentes de programa trabalhavam com grupos de desenvolvimento, grupos de teste e grupos de educação do usuário para garantir que o produto satisfizesse as necessidades previamente definidas.

O Sr. Peter Neupert, diretor Sênior, do Desenvolvimento de Produto Internacional destacou:

As especificações originais para a funcionalidade do programa - no que diz respeito ao “timing” e aos tipos de desempenho que são críticos - são estabelecidas pela alta direção da Microsoft. O segundo nível de especificações são as da interface programática, que tornam os sistemas operacionais compatíveis. O nível seguinte são as interfaces dos aplicativos ou APIs. Nesta altura já temos um maço de documentos internos que descrevem a interação entre alguns dos componentes, os sistemas de arquivo de administração de memória e coisas do gênero. Outra interface crítica é a interface programática que chega às mãos do usuário final. À medida que desenvolvemos o produto, precisamos nos cert de que essa interface está presente; de outra maneira o produto falhará comercialmente. O grande desafio é manter cada pessoa da equipe de desenvolvimento ligada a esse fato, pela duração de todo o processo de programação. Essas interfaces são tão importantes que as pequenas equipes que fazem o programa geralmente são organizadas em função delas e das ferramentas específicas,

necessárias para desenvolver o subsistema, exatamente como um sistema de arquivo.

E o Sr. Muglia continua:

Os Gerentes de Programação têm as especificações e são os responsáveis da garantia que o produto dá ao consumidor de cumprir o que ele, consumidor, deseja, em todos os detalhes. Descem ao nível de dizer com precisão como será a aparência do produto para o cliente externo. Não descem até o nível das estruturas de dados que os implementam. Porém, as equipes de desenvolvedores são as reais donas dos códigos. Isto estabelece um bom relacionamento porque a maioria dos desenvolvedores não quer ser quem toma as decisões acerca do que fazer, mas deseja, isso sim, ficar com o controle absoluto do código e dos algoritmos.

O Processo das Especificações

No nível da equipe de projeto, delineamos os assuntos e, para elementos mais importantes, usamos especificações detalhadas quanto ao que os produtos devem fazer - mas tudo isso tende a ser documentação verbal. Usamos quadros brancos para discutir as implementações tecnológicas (alvos) as quais, às vezes, acabam sendo escritas como especificações. É provável que nove entre dez vezes isso não aconteça. Não somos bons nessa história de manter as especificações de nossos produtos. Dentro de um sistema com 8 milhões de linhas como o NT há tanta coisa que, tentar documentá-las todas provavelmente dobraria o tamanho de nossa equipe e inibiria nossa capacidade de realizações no futuro. De modo geral, nosso código acaba sendo a especificação. Porém, as pessoas que o escreveram entendem os objetivos detalhados e as alternâncias que foram feitas. No caso de perdermos toda uma divisão, teríamos uma tarefa imensa para recriar o código e seu correspondente raciocínio. Temos, porém, pessoas que já trabalharam em "processamento de palavras" por, digamos, oito anos, e somos capazes de manter a continuidade nesse setor de modo que um núcleo de pessoas consegue manter os níveis de conhecimento necessários. (Operamos desse modo porque) nossa tecnologia muda com muita rapidez.

Ainda o Sr. Muglia:

Olhamos as espec como um ponto de partida muito importante para documentar quaisquer acordos que venham a existir. Geralmente o desenvolvimento de uma especificação nada mais é que o processo de uma equipe. Apenas 10 a 20% do tempo é gasto com a escrita das especificações. Porém, cada pessoa-chave interage com outras cinco ou seis, para chegarmos ao acordo de que isto é o que devemos fazer. Mas, quando as especificações já estão escritas, elas já (é comum) se transformaram em lixo porque nessa altura já aprendemos cinquenta fatos ou mais, desde quando estabelecemos o primeiro acordo sobre as especificações. Mas nós não partimos para atualizar nossas espec Em vez disso, a cada quatro meses, juntamos todos os "consertos" que os clientes nos pediram para fazer e os consolidamos em pacotes de manutenção (que se transformam na documentação atualizada do programa). Os mesmos desenvolvedores acertam o código que, originalmente, serviu para escrever o programa.

Alvos da Competição

Eis o que o Sr. Neupert comentou sobre este processo:

Nossa concentração na concorrência é incrível. Se é que existe um ponto mais fixado num programa, esse é vencer a concorrência. No negócio das redes, esse ponto é ser mais rápido que a Novell; no negócios de planilhas eletrônicas, é ser melhor que a Lotus. Desse ponto de vista, você pode começar a ver o que é necessário para alcançar o objetivo. Claro que esses não são os únicos alvos, e geralmente eles nem são quantificáveis. É comum precisarmos tomar decisões que implicam apenas em julgamentos. Por exemplo, um dos assuntos mais difíceis quando se lida com Sistemas Operacionais é decidir que tamanho de memória você quer instalar e quais são as compatibilidades que você precisa. Tais decisões são primeiramente abordadas em nível bem geral por Bill (Gates) e Paul (Maritz). Mas as especificações mudam constantemente à medida que descobrimos o que podemos e o que não podemos conseguir dentro de um objetivo e como isso afeta os outros. Por exemplo, na

primeira versão do NT queríamos que o sistema trabalhasse em oito megabytes. Quando a coisa não funcionou, doze passou a ser nosso objetivo e, por fim, visamos um sistema de dezesseis megabytes. Essas decisões mais abrangentes afetam seriamente nosso posicionamento no mercado. Outro exemplo pode ser visto em nossa próxima versão do Windows onde precisaremos capacitar grande parte da base de máquinas já instaladas de modo a transformar o programa num fenômeno de massa. Isso significa estabelecermos critérios de programas tendo como base a composição do hardware que, acreditamos, será importante no futuro. A questão é que as especificações são um processo dinâmico. Os pontos básicos importantes estão onde pensamos que a concorrência estará e no que achamos que o mercado espera.

Administração do Projeto

O Sr. Muglia continua:

Iniciado o projeto, passamos a projetar o cronograma. Uma de nossas ferramentas essenciais é o Microsoft Project. Os próprios desenvolvedores estabelecem o cronograma e os planos de integração. São eles que sabem como todas as peças se juntam no nível de maior detalhe; sabem qual é a ordem das dependências e projetam cronogramas para disciplinar as seqüências e as dependências. Marcam revisões do projeto mensais, no mínimo, para tudo isso. As equipes de desenvolvimento dão a partida e apresentam qual a exata situação de cada componente do grupo e em que função está trabalhando. Todos procuram antever com antecedência de um mês os problemas em potencial que podem surgir nas seqüências e nas funções. À medida que os projetos entram nos estágios mais abrangentes, aumentamos a freqüência das reuniões do núcleo das equipes de modo a descobrir e resolver os problemas. Todos compreendem exatamente o que significa contar problemas, a necessidades de consertá-los no futuro e quais são as prioridades.

Paul Maritz, Vice-Presidente para Sistemas Operacionais, prossegue com o relato:

Algumas empresas utilizam um design e realizam um ciclo de implementação. Nós não. A menos que você pense no modo como vai implementar o código na hora em que está fazendo o design dele, você não conseguirá fazer a implementação. Você deve pensar em termos de custos à medida que executa o design. Os níveis de abstração estão muito altos? E por aí afora. O primeiro passo é fixar a taxinomia. A seguir, dividimos o processo em áreas e funções diferentes. Então definimos as interfaces entre elas. Escrevemos as intenções dessas interfaces. Depois tornamos a dividir o processo em funções ainda menores. Os usuários destas últimas podem criticá-las segundo os termos de seus objetivos e interfaces. Grande parte deste trabalho é frito de modo informal entre as equipes que interagein extensivamente. Procuramos cuidadosamente estabelecer, em primeiro lugar, os critérios para cada função e subfunção. Todos lêem os códigos uns dos outros de modo a terem um vocabulário em comum. Todo esse processo é inerentemente cheio de alternâncias. Cada grupo tem uma grande preocupação com sua própria funcionalidade, com seus propósitos e com a seqüência do tempo.

Neste processo, nota o Sr, Neupert,

Nós jamais pomos as equipes em competição umas com as outras dentro da empresa. Claro que pode haver opiniões frontalmente divergentes dentro da equipe quanto ao que seja a abordagem correta ou a arquitetura acertada. Incentivamos essa atitude o tempo todo. O desafio é ter o assunto resolvido de um modo ou de outro. Mesmo após termos uma decisão inicial, pode acontecer de que, quem perdeu ir atrás da própria opinião só para provar que estava certo. Por exemplo, aconteceu um conflito entre o Windows e o OS/2. Internamente a maioria acreditava que o Windows não conseguiria ser melhor do que a nova arquitetura da Intel para explorar memória em modo protegido e coisas do gênero. Entretanto, um dos elementos do grupo continuou pesquisando o assunto apesar do fato de que noventa por cento de nossos recursos estarem, àquela altura, voltados para o OS/2. Ninguém imaginava que ele conseguiria: mas quando, finalmente, ele descobriu um modo de fazer o Windows funcionar, a solução encontrada tinha muitas características melhores e parecia servir ao mercado melhor do que o OS/2. Foi assim que mudamos nossa aposta no futuro do OS/2 para o apoio ao Windows.

Nada de Fórmulas Mágicas

Paul Maritz continua:

Com 1500 pessoas trabalhando no Sistema Operacional e nos programas que costumam implicar em 2 milhões de linhas de código, o processo apresenta enorme complexidade. Não temos nenhuma mágica para escrevermos nosso código. Nossos programas devem rodar em múltiplas plataformas de hardware e devem se adaptar à medida que essas plataformas são modificadas. Não existe um corpo de procedimentos que possamos seguir. Pessoas inteligentes são a base de tudo, mas as equipes podem suplantar a experiência e capacidade desses indivíduos. O maior problema é lidar com a interação espacial entre os subsistemas. Centenas de coisas estão acontecendo em paralelo. O que excede em muito a complexidade de qualquer ferramenta CASE; por isso não usamos essa ferramenta. Também não usamos programas de "macrocontrole" como o Método 1 ou o Método 2 de Andersen. Preferimos, em vez disso, que os elementos do grupo documentem o programa à medida que avançam com o trabalho.

Desenvolvedores veteranos ficam responsáveis pelo design geral. Entretanto, a maioria dos problemas acontece quando da implementação. Não se pode tolerar a perda do controle sobre o código de base. É preciso feedback constante de todos os elementos. Nos primeiros estágios, o costume é "deixar os portões relativamente abertos" de modo que todos possam ver tudo o que acontece. Entretanto, mais tarde, com mais de 200 desenvolvedores trabalhando num programa como o NT, se você permitir que os desenvolvedores verifiquem os detalhes dentro do sistema, você não conseguirá nem dar a partida. Tudo falharia.

Para manter a coordenação, o grupo dos Sistemas Operacionais Microsoft usou uma técnica para levar o programa a um "teste de construção". Pelo menos uma vez por semana, no mais das vezes, duas a três vezes por semana, cada grupo recompila seu software de modo que toda a equipe possa construir um programa consistente e coerente, com todas as novas funções e características em seus devidos lugares.

É freqüente que a tentativa de unificar todo o programa ou subsistema falhe. Entretanto, essas “construções” forçam as pessoas a se esforçarem até encontrar e consertar o que houve de errado desde a última “construção”. Se os erros não forem consertados neste estágio, as interações tornam-se rapidamente tão complexas que seria impossível fazê-las coerentes, mesmo que cada subsistema possa operar efetivamente cada um por sua conta.

Nas palavras de Paul Maritz:

Mais tarde fica difícil melhorar o desempenho de qualquer programa. O que significa que, além das “construções”, dividimos o processo em marcos, cada um dos quais vale por uma base de um dado cliente. Embora nossos “designers de teste” executem seqüências formais de testes para disciplinar cada etapa deste processo, não podemos adivinhar o que cada pessoa faz ou fará com o programa. Conseqüentemente, a cada marco fazemos os arranjos para que o programa possa ser usado para um dado propósito no mundo real. É assim que conseguimos feedback dos usuários do programa e trabalhamos a partir dessa base. A cada etapa sucessiva aumentamos a abrangência do programa. A cada marco zeramos a lista dos defeitos para essa clientela e para essa série de testes.

Para cada subsistema existe uma equipe composta de uma a dez pessoas. Impossível ter equipes maiores do que isto. As pessoas precisam ter a capacidade de entender toda a complexidade do subsistema e suas interfaces. Devem conhecer-se muito bem mutuamente e serem capazes de confiar e de julgar as mútuas necessidades e soluções. Conseqüentemente desenvolvemos uma série de rituais que forçam a todos se reunirem em torno do que estão construindo.

O Sr. Peter Neupert ressalta “Cada um de nossos grupos de aplicativos tinha, geralmente, suas próprias ferramentas. Agora passamos a ter convenções e estamos fabricando algumas ferramentas; estas, porém, não são usadas com consistência pelos vários grupos. De preferência a forças ferramentas e interações, nossa atitude predominante é:” Quando as pessoas precisam saber de alguma coisa, devem ir e procurar as informações de que precisam.”Geralmente isso

acontece com o indivíduo. A permanência da pessoa dentro da empresa realmente é produtiva em termos do conhecimento de como se deslocar dentro do sistema, a quem fazer as perguntas e com quem falar.”

Um Negócio Movido a Pessoas

O Sr. Neupert continuou,

Há uma coisa de que todos, de Bill para baixo, têm certeza. Este é um negócio que é movido a pessoas. Você pode ter os sonhos que quiser, mas não poderá realizá-los se não tiver as pessoas certas às suas ordens. Embora saibamos que nosso pessoal já realizou coisas extraordinárias, ainda pensamos que temos muitas fraquezas. Jamais fomos capazes de conseguir o número certo de pessoas que precisamos porque os desafios que assumimos sempre se expandem mais depressa do que nossa infraestrutura. Por exemplo, as atuais programações da TV a cabo e multimídia estão crescendo com muita rapidez. Conhecemos as dimensões e a capacidade que temos e as que precisaremos. Esforçamo-nos ao máximo para encontrar e conservar o excelente pessoal de que necessitamos. Mas, uma vez que eles estão dentro, é extremamente difícil fazer com que intercambiem com outros grupos de produto. E a compreensão do nosso pessoal que está num dado grupo de produto sobre os desafios que as pessoas de outro enfrentam é (muito limitado).

Nós não temos nenhum mecanismo-forma pelo qual deslocamos pessoal por meio de uma sucessão de desafios em grupos. Como a demanda por gente capacitada é muito grande, todas as divisões querem arrebanhar os melhores. E, apesar de nossa política, não somos tão bons quanto gostaríamos de ser em nos livrarmos dos piores. Devido ao aumento contínuo dos desafios, alguns grupos achatam as pessoas. Existem aqueles que permaneceram no mesmo nível, trabalhando os mesmos problemas por mais de dez anos. Tudo bem, eles são competentes, não existem problemas pressionando e o resultado do trabalho deles ainda acrescenta valor à empresa. Porém, isso não vai resolver nosso problema de falta de mão-de-obra.

Desenvolvimento do Pessoal

O Sr. Michael Murray, Vice-Presidente de Recursos Humanos, descreveu as práticas da Microsoft neste terreno:

Depois de contratadas, as pessoas são colocadas diretamente numa equipe pequena, com um conjunto de produtos acabados dos quais se espera que escrevam ou testem os códigos do software em períodos de tempo muito apertados. Não se trata de aprendizagem. Eles estão logo de início ombro a ombro com pessoal que já está trabalhando aqui há um, dois, cinco ou dez anos e que possuem vasto conhecimento. Não organizamos muitos seminários sobre como trabalhar em equipe de desenvolvimento de software. Um novo desenvolvedor ou novos empregados de nossos grupos de produção passarão apenas por duas semanas de treinamento formal, ou introdução às metodologias que usamos no desenvolvimento.

Sempre demos destaque a três atributos: pessoas inteligentes, pessoas que trabalham duro e pessoas que sabem como conseguir que as coisas aconteçam. Se você tem essas três qualidades, sua primeira promoção será para um posto de liderança técnica num pequeno projeto, depois será guindado a administrador de grupo, administrando vários outros líderes. Onde quer que você esteja, será difícil encontrar um executivo sequer (mesmo no nível da vice-presidência) que não execute algum trabalho individual dentro dos programas que estejam sendo executados.

Nossa única recomendação a nossos empregados é avisá-los de que o progresso de suas carreiras é responsabilidade deles mesmos. Só agora estamos desenvolvendo nosso primeiro programa formal onde poderemos criar algumas posições de desenvolvimento - deslocando alguém, propositadamente, para um posto por seis a oito meses de modo que adquira experiência dentro da empresa. Evidentemente conversaremos com as pessoas e discutiremos várias oportunidades de trabalho. Também mantemos um banco de dados para nossas estrelas ascendentes. Estamos dando início a alguns programas de liderança para treinar essas pessoas em conceitos de estratégia mais amplos. Mas nos escalões mais baixos,

deixamos que os meios informais como e-mail se encarreguem de identificar as oportunidades. No passado nós diríamos que, se a empresa crescesse, todos cresceríamos com ela. Hoje ainda contratamos os mesmos super empregados de então, porém passamos de uma empresa de 3000 a 4000 empregados a uma com 17000 pessoas e as oportunidades de subir ou de interagir com o topo não são tão grandes como costumavam ser. Hoje costumo dizer que se uma pessoa quer avançar mais depressa, deve olhar para os lados. E, se ela for nessa direção, vai encontrar uma empresa muito diferente em outras áreas. Esta coisa que todos achamos que é a Microsoft, na verdade, é constituída por muitas culturas em diversos lugares. E essas culturas diferentes têm modos diferentes de se comunicar, de recompensar, etc.

A Cultura Microsoft

O Sr. Murray descreveu deste modo à cultura Microsoft:

Parte de nossa cultura é questionar tudo. Numa apresentação não é nosso estilo ficar lá sentados e nos maravilharmos com a apresentação. Em vez disso o trabalho é o de calcular o que está errado na apresentação, onde está a falha, onde eles falharam quando desenvolvendo um pensamento analítico. Acontece de você sair de uma reunião pensando, puxa, esta empresa é cínica, amarga ou se irrita com qualquer coisa. Mas eu digo que isso simplesmente é um modo muito cauteloso de fazer negócio. Um outro aspecto bastante comum de nossa cultura é uma grande intolerância: intolerância com imprecisão ou com incorreções na análise, na descrição dos problemas ou na compreensão de questões fundamentais. Tudo isso é impulsionado por Bill Gates mais do que ninguém. É um grande aprendizado ser interrompido por ele ou vê-lo interromper alguém no meio de uma apresentação. Isso pode ser muito d para uma pessoa menos articulada, mas brilhante.

Um outro aspecto de nossa cultura é nossa "semana de 60 horas". Acho esse aspecto muito parecido com minha experiência na Apple. Essas empresas foram fundadas por pessoas que eram jovens, entusiastas e incrivelmente apaixonadas pelo que estavam fazendo.

Cada vez que você aparecia com uma idéia nova, você podia ver um novo leque de possibilidades. Por isso as coisas sempre se encaminhavam para uma infinita quantidade de trabalho e uma quantidade finita de tempo. Todos estavam sempre impacientes porque tudo andava tão depressa. As pessoas que fundaram e que se juntaram a essas empresas eram únicas, altamente competitivas, extremamente inteligentes e haviam decidido pautar suas vidas pelo seu trabalho. E de repente você descobre que criou uma cultura de trabalho de alta velocidade. E não foi inconscientemente. As próprias pessoas reforçaram essa idéia: o lugar de estacionamento do seu carro correspondia à hora em que você chegava à empresa; se seu carro era o último a sair, isso era indicativo de quanto tempo você havia trabalhado. Mesmo quando se era transferido para um cargo mais administrativo, você descobria que a ética permanecia a mesma. Nenhum de nós jamais achava que tinha feito tudo o que podia num dia.

Todas essas características culturais se desenvolviam por si sós enquanto todos trabalhavam ao lado de todos. Não tenho certeza de que algum de nós já realizou o que ter 17000 empregados significa em termos de manter alguns desses fatores culturais. Você vai descobrir que empresas como a Hewlett Packard ou a Intel gastam bastante tempo conversando com seus empregados a respeito de que tipo de empresa são e querem ser. Penso que Bill Gates dá muito valor aos nossos valores culturais - ou àquilo a que chamamos de nossos "fatores de sucesso" - porém ele não é o tipo de Presidente que quer falar muito a respeito. Ele prefere muito mais falar a respeito da nova tecnologia , do novo produto que estamos desenvolvendo ou da nova estratégia de negócio que temos.

Bill tem uma visão bem definida de onde a empresa deve estar e confia muito nessa visão. Mas é uma visão técnica de onde a empresa deve ir. Será que ele partilha o tipo de empresa que deseja que nós sejamos? Penso que não. Isso não faz parte do roteiro. Mas acho que todos os empregados diriam que bom que Bill é nosso presidente. Adoramos ouvi-lo falar. Adoramos quando ele fala a respeito do futuro. Nós acreditamos. Nós honramos nossa bandeira. Mas esta empresa não é introspectiva. Bem ao contrário, temos um enorme pára-brisa que nos permite enxergar o grande panorama à nossa frente, e temos um acelerador bem largo para não errarmos o pé e temos um pequeno, mas eficiente breque.

Mas, se você olha para os lados para ver onde esteve, não vai achar nada interessante.

A Motivação

Os empregados da Microsoft ganham salários relativamente modestos se comparados ao restante da indústria. Mas os melhores empregados recebiam bônus generosos sob a forma de opções de ações. Até Bill Gates - embora com patrimônio bilionário - jamais ganhou acima de 195 mil dólares em salários por ano. Mesmo assim a rotatividade de empregados sempre ficou abaixo da média da indústria.

O que atraía as pessoas para a Microsoft com seus salários medianos e as enormes exigências pessoais? Paul Mantz, Vice-Presidente para Sistemas Operacionais, responde: *"Eles vêm para cá porque a administração entende de software e é mesmo apaixonada pelo assunto. Todos eles já trabalharam com software e gostam de interagir com o pessoal dos projetos. ... Nosso pessoal tende a motivar a si mesmos com pura pressão. Lima das melhores motivações é 'saber que está bem feito'. Somos conduzidos pelos clientes, pela concorrência e pelo sucesso. Não podemos sobrevalorizar tais fatos. Entretanto, uma das grandes contribuições foi o fato de termos nos tornado uma grande empresa muito depressa; este fato alavancou fortemente nossas ações."*

Bob Muglia vai mais além,

É provável que nos tempos modernos não tenha havido uma empresa com impacto financeiro tão forte sobre tantas pessoas como teve a Microsoft. Trabalhar na Microsoft por algum tempo deu às pessoas um sucesso financeiro que não teriam alcançado nem se tivessem um negócio próprio. Muito de nossa atração provém da fama dos produtos da empresa, de sua posição dentro da indústria e do fato de que, dentro da Microsoft, você pode encontrar um grande número de pessoas inteligentes. Um grande número dos que vêm trabalhar aqui, saídos diretamente das faculdades, não têm o dinheiro como motivação maior e nem mesmo entendem o valor das ações. Quando as recebem, eles dizem apenas, "legal", mas não pensam a respeito. Existem muitas histórias a respeito de pessoas que venderam essas ações assim que subiram uns cinco

dólares e que poderiam ter feito muito mais dinheiro se esperassem um ano, quando essas mesmas ações passaram a valer dez ou vinte vezes mais. No passar dos anos, as pessoas repetidamente disseram que a Microsoft não poderia continuar a crescer exponencialmente para sempre. Isto é bem verdade, mas onde iremos parar ninguém pode ainda definir.

Um Foco do Desempenho

Nós não fazemos o oba-oba que outras empresas às vezes fazem. Entretanto, de certo modo as coisas são bem facilitadas para nossos programadores. Você quer um novo computador com algumas características especiais? Está bem, você vai ter o que quer. As pessoas gostam de saber como vai ser a próxima geração dos sistemas operacionais e gostam de utilizá-los antes que outros o façam. Mas essas coisas não acontecem à toa. 'Todos são submetidos a um exame anual de desempenho com uma "verificação de objetivos" no meio do ano. A avaliação é feita em cima de objetivos determinados pelo pessoal e pelos seus gerentes. Todos têm entre cinco a dez objetivos que devem alcançar no prazo de seis meses. A cada seis meses cada empregado recebe uma folha de revisão em branco. Tomam seus objetivos de seis meses atrás e fazem eles mesmos uma avaliação do que conseguiram. Quando se trata de avaliação anual, eles também analisam suas próprias deficiências e dizem o que pretendem fazer a respeito. Os gerentes responsáveis tomam essas avaliações e escrevem nelas suas próprias avaliações sobre o desempenho do empregado, comparando o resultado presente com os objetivos de seis meses atrás. Escrevem também suas opiniões sobre os pontos fortes e os pontos fracos do empregado. Então, junto com o empregado, estabelecem quais serão os objetivos para os próximos seis meses. Pode ser divertido trabalhar em cima de outras coisas, porém estas são as coisas que se espera que você consiga.

Peter Neupert disse,

Tentamos estabelecer uma métrica que medisse a produtividade, experimentando-a nos projetos menores. O que é mais eficaz para a produtividade: melhorar os procedimentos na ponta do processo

de design ou definir melhor a funcionalidade e a qualidade? Quantas linhas de código devem ser desenvolvidas por uma única pessoa? Quantos defeitos podem ser tolerados por unidade de tempo? Enquanto temos alguns padrões em nossas cabeças, fica difícil escrevê-los ou aplicá-los nos projetos. Também é difícil olhar as pessoas e dizer o que estão fazendo naquele momento. Damos aos programadores, propositadamente, escritórios privativos de modo que possam ter o tempo de que necessitam para trabalhar sem interrupções. Queremos que eles tirem o fone do gancho e tranquem a porta sempre que acharem necessário. Mas também esperamos que eles se entrossem nas equipes e trabalhem a noite toda se a equipe assim precisar.

O fato interessante é que esses caras eram perfeccionistas obsessivos. O mais d é conseguir que parem de trabalhar e ponham o produto à venda. Ficam muito perturbados quando pensam que você está vendendo algo que é menos que perfeito, porque consideram que seus nomes estão associados ao produto e ao seu funcionamento. Outro problema aparece quando você se aprofunda bastante num produto e vem um inovador da programação e diz "Ei, poderíamos fazer X. Não é uma grande idéia?" e sem ligar para o que você diz começa a trabalhar na coisa. O que eles nem imaginam é que ao fazer X vão em direção diferente de outras pessoas que achavam que aquela função deveria ser Y. Os programadores sempre acham que as coisas serão mais fáceis do que realmente são. De modo geral elas são mais difíceis por um fator 4. À medida que você acrescenta complexidade, aquilo que você acha que vai levar uma semana para ser feito acaba levando seis e pode sair muito d de seu objetivo original. O perfeccionista obsessivo acha que você deveria ter muito mais funcionalidade e tenta enfiar um desempenho $2T$ no tempo T original. E, claro, você jamais saberá o quanto $2T$ é complexo até que um usuário finalmente o experimente. O maior problema é: quanto mais subsistemas são acrescentados, maior é a quantidade de pessoal necessário e mais especializado esse pessoal deve ser. Obviamente os problemas de coordenação também aumentam.

"Porém no futuro, diz Bob Muglia, as pessoas saberão que precisam alcançar seus objetivos. Temos uma escala de cinco pontos, com incrementos de 0.5. Entretanto, na prática, só usamos a escala entre as medidas 2.5 e 4.5. Acima dos 4 estão as superestrelas e os 2.5 têm algum problema. Todos têm também um nível de compensação. Os níveis para profissionais vão de 10 a 15 onde estão os arquitetos. Os aumentos de níveis é que são as

verdadeiras promoções. Os títulos não têm grande significado. Quando alguém está pronto para o próximo nível, nós o promovemos formalmente, damos-lhe um ótimo bônus e ele recebe também um aumento que é associado à promoção. Mas não concedemos bônus por desempenhos excepcionais nos projetos. No passado já demos recompensas para todas as pessoas de uma divisão. Pois sempre foi complicado escolher dez ou vinte pessoas entre um grupo de cem e dizer-lhes 'vocês fizeram um trabalho excepcional. Existe um limite muito fino entre a vigésima e a vigésima primeira pessoa'. dentro de um projeto, e você pode fazer com que apenas poucas pessoas se sintam bem enquanto cinquenta ou cem pessoas sintam-se pior ao conceder esse bônus seletivo."

O Sr. Murrey repara,

Também existe feedback administrativo. Os empregados têm a oportunidade de escrever sobre o desempenho de seus dirigentes e o que acham que eles fizeram de certo ou errado. Também realizamos um exame anual para os empregados que é anônimo, para compreender quais são as tendências gerais, quais as questões, o que as pessoas gostam ou não, e assim por diante. Procuramos aprender com nossa experiência por meio de um processo chamado de post mortem. Depois que você termina uma tarefa, reúne o pessoal responsável, conversa com os clientes, e com qualquer outra pessoa que possa lhe dar informações sobre o que funcionou ou o que não funcionou, o que precisamos melhorar da próxima vez e o que aprendemos com tudo isso. Então procuramos condensar tudo num relatório. O problema é, será que os outros grupos lerão o relatório? E, se o lerem, vão levá-lo a sério?

Outro ponto que salta aos olhos é o imenso poder de desenvolvimento pessoal que existe numa empresa que opera a si mesma através de e-mails. O conhecimento é partilhado de modo tão rápido e tão abertamente que se transformam processo educacional constante. Pessoas como Natham Myhrold, Vice-Presidente Sênior para Tecnologia Avançada, costumam despachar e-mails de dez páginas sobre as últimas cogitações de seus pares. Muitos de nós somos incluídos nesse grupo. Pode acontecer de não darmos nenhuma contribuição para aquele memorando em particular. Porém, ele planta idéias em nossas mentes. Nathan pode nem chegar a saber o efeito que causou ou em quem. Mas muitas

As pessoas vão para casa no fim do dia pensando nas novas idéias e na possibilidade de implementá-las de algum modo novo no projeto em que estão trabalhando. Tudo isso acontece organicamente. Não existe nenhuma estrutura, nenhuma regra. Não há modo de alguém poder criar um sistema de controle para e-mail para tentar explorar o correio como meio educacional. Também não existe recompensa alguma por alguém se juntar a essa partilha de conhecimentos via e-mail. Do jeito como agimos na empresa, é tudo orgânico.

Olhando para O Futuro

Os métodos da Microsoft, olhando para trás, foram, evidentemente, muito bem sucedidos. E, no correr do tempo, a organização desses métodos foi mudando. Entretanto, em 1994, houve muita preocupação acerca de como o tamanho cada vez maior da empresa, a natureza de mudança constante do mercado e as enormes e crescentes complexidade e interatividade de seus programas iriam afetar suas abordagens administrativas. Estes fatores já haviam sido motivo para os grupos de serviço e suporte ao produto da Microsoft, que atendem chamados dos clientes, excederem sua renda em 5% em 1994.

Como a Microsoft era muito importante para seus clientes corporativos, em 1990, foi criada a Microsoft Consulting Services. Em 1990, Mike Hollmann, um dos diretores da Microsoft, destacava: *"Não somos muito conhecidos por nosso cliente que busca suporte. Em 1995, quero a Microsoft que fornece suporte e serviços tão conhecidos por nossos clientes quando a Microsoft fornecedora de produtos. Quero que a Microsoft seja campeã de suporte a software."*

E assim surgiu a necessidade cada vez maior de controlar a qualidade do produto antes que fosse posto no mercado, um problema que, devido à sua liderança, a Microsoft sempre enfrentara. Como resolver esta questão, dentro da estrutura altamente descentralizada, era um grande problema para a Microsoft. À medida que os clientes, a complexidade e os canais de comunicação aumentavam em número, aumentava também a preocupação de que o direcionamento da empresa mudasse de desenvolvimento de software para atividades de marketing e serviços.

Antecipando-se à natureza de tais mudanças, a Microsoft havia começado um programa de pesquisas de US\$200 milhões sob a direção de Nathan Myhrvold, um físico que havia trabalhado com o professor Stephen Hawking. Myhrvold era uma pessoa muito articulada, criativa e carismática; era apaixonado por possibilidades imaginativas e filosóficas. Nas palavras de Myhrvold:

Precisamos aprender a pensar sobre rodovia da informação como um fenômeno que se assenhora do computador pessoal como uma via que está entrelaçada com o tecido mesmo de nossas vidas - em nossa sala de estar, nossos bolsos, nossas cozinhas - onde quer que haja pessoas e informações para os computadores e softwares. Até o momento, nosso mercado primeiro, a computação pessoal, teve natureza local. Isso está mudando. E o mesmo tipo de mudanças na formação de preços que os microprocessadores trouxeram para o mundo dos computadores está vindo novamente com o aparecimento das fibras ópticas, comutadores ATM (modo de transferência auto-sincrônico) e das comunicações via satélite.

O verdadeiro drama da revolução nas comunicações e nos computadores vai começar nos próximos dez anos. Um dos motivos pelos quais a Microsoft já teve tanto sucesso foi nossa capacidade de administrar a manutenção do "ciclo de participação". À medida que crescia a capacidade do computador, mantivemos nossa parte dos "ciclos-por-segundo" consumidos no mercado. Dado o crescimento exponencial do mercado, estivemos surfando numa onda. Fundamentalmente, nosso negócio cresceu na mesma razão do crescimento da formação do preço das memórias. Para manter essa situação, precisamos continuar a nos expandir exponencialmente. Hoje existe no mundo o mesmo número de computadores pessoais e de VCRs e cerca de número igual de computadores e de televisores. Mas, enquanto as pessoas não substituem seus aparelhos de TV por vinte anos, elas trocam de PC a cada poucos anos. Isto vai comandar o ciclo de reposição de todos os softwares.

Até agora, a computação sempre disse respeito a criar ferramentas destinadas a análises, criação de documentos, design. Em muitas áreas, os computadores simplesmente tomaram o papel da caneta e

da tinta, dos modelos físicos e das calculadoras. O próximo e evidente estágio será ter de obter informações com as pessoas, ser a janela do mundo das informações e das comunicações entre as pessoas. O e-mail trará a verdadeira comunicação com rapidez. Uma vez que você tenha a capacidade fundamental de ler, interpretar, comunicar e distribuir informação, novos mundos abrem-se para você. Para nós, o desafio técnico dos próximos cinco anos será o de satisfazer todas as necessidades para que esse mundo novo torne-se realidade, O sistema atual não vai permitir, graciosamente, que você realize todas as coisas maravilhosas que imagina. E o software que suporta essas novas redes de comunicação deverá ser totalmente redesenhado.

Muitos argumentaram que, à medida que os equipamentos tornam-se mais baratos, alguém pode chegar e tornar obsoleto todo o enorme investimento realizado pelas RBOCs (Regional Bell Operating Companies - operadores de telefonia). O valor das tecnologias existentes está, quase por definição, evaporando-se numa razão de 50% ao ano. Como aconteceu com os velhos mainframes, os comutadores e PBXs do passado sofrerão pressões tremendas. Os ATMs podem vir a ser os novos "microprocessadores" do mundo das comunicações. Certamente permitirão que novos conceitos, empresas e soluções apareçam.

Quando os sistemas de voz tornaram-se essencialmente gratuitos? Hoje, grande parte dos serviços locais e domésticos nem são medidos. Grande parte custa apenas cinco centavos de dólar o minuto, O que acontecerá quando for possível 60 megabytes por segundo nos sistemas instalados, os quais, antes, apenas suportavam 10 kilobytes por segundo? A formação de preço de tudo isso vai ter que mudar.

Sabemos muito bem o que é um computador pessoal. E eles serão um dos nós principais das rodovias da informação - mas jamais os únicos. Num futuro próximo, os PCs constituirão a plataforma mais disponível para fazer qualquer coisa. A base de PCs já instalada é impressionante. Nos EUA, ano passado, foram vendidas tantas máquinas 486 Windows quantos Sega Genesis games. Mas ninguém ainda domina ou pelo menos é especializado nos novos sistemas que estão para surgir. Nesse caminho ainda haverá inúmeros erros e mortes entre as empresas e os hardwares. Já estão dizendo que toda essa concorrência vai transformar a indústria da computação

em mero commodities. Não acredito. Reestruturar o mundo das comunicações vai fazer surgir muitas e enormes empresas, exatamente como o microprocessador fez. Uma das coisas mais burras que já ouvi falarem foi o argumento de que a IBM cometeu um grande erro em não manter os direitos sobre seu sistema operacional para PC. Olhe só para todos os que tentaram manter um sistema proprietário. Da DEC e Xerox até a HP e a Wang, quase todos morreram ou agonizam. Enquanto a IBM criou toda uma indústria e continua a maior dentro dela.

A questão principal é como conseguir dinheiro nesse cambiante mundo orientado a software. A resposta é administrar bem essas pessoas inteligentes que criam os softwares.

Perguntas

1. Quais as razões organizacionais e estruturais mais importantes para o sucesso da Microsoft? Quais os problemas que você prevê para o futuro?
2. Como deve ser a futura macroestrutura da Microsoft? Por quê? Qual é sua proposta para resolver questões específicas e/ou falhas nos procedimentos da Microsoft? Como ela deverá se organizar para explorar as emergentes oportunidades do software?