

UNIVERSIDADE DE CAXIAS DO SUL
DEPARTAMENTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

SIDNEI DA SILVA

**Gestão de Pessoas no Contexto do
Trabalho Colaborativo à Distância**

Prof. Heitor Strogulski
Orientador

Caxias do Sul, Dezembro de 2007

*“Misericordioso e piedoso é o SENHOR; longânimo e grande em benignidade.
Não reprovará perpetuamente, nem para sempre reterá a sua ira.
Não nos tratou segundo os nossos pecados,
nem nos recompensou segundo as nossas iniquidades.
Pois assim como o céu está elevado acima da terra,
assim é grande a sua misericórdia para com os que o temem.
Assim como está longe o oriente do ocidente,
assim afasta de nós as nossas transgressões.
Assim como um pai se compadece de seus filhos,
assim o SENHOR se compadece daqueles que o temem.
Pois ele conhece a nossa estrutura; lembra-se de que somos pó.
Quanto ao homem, os seus dias são como a erva,
como a flor do campo assim floresce.
Passando por ela o vento, logo se vai,
e o seu lugar não será mais conhecido.
Mas a misericórdia do SENHOR é desde a eternidade
e até a eternidade sobre aqueles que o temem,
e a sua justiça sobre os filhos dos filhos;
Sobre aqueles que guardam a sua aliança,
e sobre os que se lembram dos seus mandamentos para os cumprir.”*

— SALMOS 103:8-18

AGRADECIMENTOS

A Deus pela sua misericórdia e amor pela minha vida; aos meus pais Paulo e Eligia; à minha esposa Daiane pelo seu apoio fundamental em momentos decisivos dessa caminhada; ao meu amigo do peito Alan Runyan pela motivação e pelos impagáveis momentos de inspiração; a todos os professores que compartilharam o seu conhecimento, em especial à professora Marília Villa que me encontrou na rua no último dia para a inscrição no vestibular e me incentivou a entrar na universidade, tenho certeza que foi Deus que colocou você naquele lugar, naquela hora.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
RESUMO	7
ABSTRACT	8
1 INTRODUÇÃO	9
1.1 Motivação	10
1.2 Objetivos	11
1.3 Metodologia e Organização	12
2 GESTÃO DE PESSOAS	13
2.1 Significado da Gestão de Pessoas	13
2.2 Atividades na Gestão de Pessoas	14
2.3 Considerações Finais sobre a Gestão de Pessoas	17
3 TIMES VIRTUAIS	19
3.1 Autonomia e “Empowerment”	19
3.2 Problemas e Dificuldades com Times Virtuais	20
3.2.1 Encontros Face a Face e Relações de Confiança	20
3.2.2 Comunicação	20
3.2.3 Produtividade e Criatividade	21
3.2.4 Fuso-horário	23
3.3 Times Virtuais e Software Livre	23
3.4 Considerações Finais sobre Times Virtuais	24
4 SOFTWARE LIVRE	26
4.1 Definição	26
4.2 Motivação dos Desenvolvedores de Software Livre	27
4.2.1 Bem Comum	28
4.2.2 Respeito ou Reconhecimento	29

4.3	Comunicação	30
4.4	Liderança e Hierarquia	31
4.5	Considerações Finais sobre Software Livre	32
5	ENTREVISTAS	34
5.1	Apresentando os Entrevistados	34
5.2	Comparação das Respostas	36
5.3	Considerações Finais sobre o Resultado das Entrevistas	41
6	CONCLUSÕES	44
APÊNDICE A	QUESTÕES ENTREVISTA	47
APÊNDICE B	ENTREVISTA: CHRISTIAN REIS	49
APÊNDICE C	ENTREVISTA: PAUL EVERITT	66
APÊNDICE D	ENTREVISTA: ALAN RUNYAN	77
REFERÊNCIAS	88

LISTA DE ABREVIATURAS E SIGLAS

BBS	Bulletin Board System
GIMP	GNU Image Manipulation Program
GNOME	GNU Object Model Environment
GNU	GNU's Not UNIX
GTK	The GIMP Toolkit
GUADEC	GNOME Users And Developers European Conference
IM	Instant Messaging
IRC	Internet Relay Chat
PyGTK	Python Wrappers for GTK
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
VoIP	Voice over IP

RESUMO

A dificuldade em se encontrar pessoal qualificado às necessidades dos projetos, aliada à busca por uma redução de custos e à popularização do acesso às redes de informação levaram à criação de uma nova modalidade de trabalho: o Trabalho Colaborativo à Distância. Esta nova forma de relacionamento profissional entre trabalhadores e seus gerentes requer novas maneiras de gerir equipes dentro de uma organização.

O Movimento do Software Livre, de forma indireta, levou ao aparecimento de comunidades de desenvolvimento de software que trabalham de forma descentralizada através das redes de comunicação. Essas comunidades precisaram desenvolver novas maneiras de se organizar e se comunicar para poder expandir-se além das fronteiras físicas.

O objetivo desse trabalho é documentar essas soluções, que muitas vezes não passam de convenções informais, suas aplicações com sucesso e suas falhas, e por fim fazer um comparativo com técnicas de gestão já estabelecidas.

Palavras-chave: Desenvolvimento Distribuído de Software, Gestão de Pessoas, Gestão do Conhecimento, Times Virtuais, Times de Trabalho Auto-Dirigidos, Comunicação, Open Source, Software Livre.

People Management in a Collaborative Remote Work Context

ABSTRACT

Difficulties finding qualified personnel for specific project needs, allied to the quest for cost reductions and the popularization of access to information networks, created a new model of collaboration: Remote Collaborative Work. This new relationship between workers and their managers requires new ways of managing teams within an organization.

The Free Software movement indirectly brought forth decentralized software development communities that communicate through information networks. Those communities had to create novel ways of organizing and communicating so that they could reach beyond physical frontiers.

This work documents those solutions, that more often than not are mere informal conventions, their successes and failures, and finally compare those solutions with established management practices.

Palavras-chave: People Management, Knowledge Management, Virtual Teams, Communication, Open Source, Free Software, Self-Directed Work Teams, Distributed Software Development.

1 INTRODUÇÃO

Cada vez mais a tecnologia se insere em nossas vidas, de forma irreversível. Tarefas que a apenas alguns anos atrás levariam semanas hoje podem ser concluídas em questão de minutos.

Muitos paradigmas foram e continuam sendo quebrados, principalmente no que diz respeito à comunicação entre as pessoas. Hoje é possível estabelecer uma conversa com qualquer pessoa em qualquer parte do mundo de forma instantânea e com custo praticamente inexistente, e isso tem impacto direto sobre a maneira como as pessoas trabalham e principalmente sobre as relações de confiança dentro do ambiente de trabalho (NANDHAKUMAR, 1999).

Num mercado global, impulsionado pelo alcance ilimitado das redes de informação, a busca por preços mais competitivos coloca um trabalhador de um país sub-desenvolvido em competição direta com outro em um país desenvolvido. As empresas buscam terceirizar tarefas de forma a minimizar os seus custos, muitas vezes empregando trabalhadores que estão a milhares de quilômetros (FRIEDMAN, 2005).

Toda uma nova área do conhecimento está sendo explorada: a era do trabalho colaborativo à distância está nascendo. Há muitas incertezas sobre como as relações no ambiente de trabalho devem ocorrer nesse contexto. Muito do que aprendemos ao longo das últimas décadas já não se aplica mais. Em um ambiente de trabalho onde gerente e subordinado podem estar separados por muitos fuso-horários de diferença a excelência na gestão de pessoas no contexto de trabalho à distância deve ser ponto fundamental na formação de bons administradores.

Hoje em dia, muitas mudanças significativas ocorrem simultaneamente ao redor do mundo, uma dessas em especial é de nosso interesse. É o chamado “Movimento do Software Livre”, algo bem diferente de tudo que já foi visto. Ao contrário das mudanças que aconteceram anteriormente, esta não tem um líder e não está ocorrendo em um local específico, ao contrário é composta por muitos anônimos e se propaga silenciosamente através das redes de informação. Um fato que possibilitou o surgimento do Movimento do Software Livre foi a perda do monopólio das co-

municações pelas grandes empresas, aumentando a concorrência e popularizando o acesso à Internet.

De certa forma, o Movimento do Software Livre é pioneiro em muitas das questões relativas à gestão de pessoas à distância. Os maiores avanços do nosso século aconteceram quando foi dado a um grupo de pessoas a oportunidade de exercer sua criatividade com liberdade (DERTOUZOS, 2001). É justamente isso o que o torna assunto interessante para nós. O Movimento do Software Livre possibilitou o surgimento de formas inovadoras de organização e interação entre as pessoas (REIS, 2003).

Fazendo uma análise do contexto atual das relações de trabalho em projetos relacionados à Software Livre, podemos aprender muitas lições importantes sobre como gerenciar pessoas. Devido à abertura das informações relacionadas aos projetos de Software Livre é possível fazer uma análise destas sem interferir no ambiente. Essa abertura pode mostrar tanto o melhor quanto o pior lado de uma pessoa. É extremamente educativo examinar como as relações interpessoais são construídas na falta de uma estrutura hierárquica, como as pessoas reagem quando há um conflito a resolver, e principalmente como ocorre a divisão e a delegação de tarefas em situações onde ninguém é obrigado a cumprir ordens.

Desta análise podemos absorver muitas lições, tanto com os sucessos como com os fracassos. Lições estas que serão cruciais para aumentar o nível de sucesso nesse cenário que se desenrola diante de nós. A gestão de pessoas à distância é relativamente nova, por isso muito do conhecimento está sendo desenvolvido no dia a dia das organizações modernas que precisam rapidamente se ajustar ao ritmo frenético da informação nos dias atuais.

1.1 Motivação

Há uma enorme quantidade de conhecimento sobre a gestão de pessoas à distância no mundo lá fora. Boa parte desse conhecimento não está em protocolos ou regras que possam ser seguidas mas em convenções bastante informais desenvolvidas ao longo do tempo conforme os problemas foram surgindo. Muitas dessas convenções foram criadas devido ao advento do Movimento do Software Livre, que leva milhares de pessoas ao redor do mundo a colaborar, muitas vezes de forma voluntária, com projetos de software que são distribuídos sem fins lucrativos.

O que motiva essas pessoas a agir dessa forma? De que forma elas interagem colaborativamente para alcançar os seus objetivos? Quais as soluções que elas adotam para se organizar e gerenciar essa mão-de-obra volátil e comumente não-

remunerada?¹ Principalmente, como essas soluções se aplicam ou poderiam ser aplicadas a projetos de desenvolvimento distribuído onde o objetivo principal não é a produção de Software Livre? Estas são perguntas cuja respostas podem contribuir imensamente para os futuros gerentes de pessoas. Há muito que aprender, tanto com os acertos quanto com os erros daqueles que lidam diariamente com esses problemas.

Nossa motivação aqui é buscar as respostas para algumas dessas perguntas, para sintetizar parte desse conhecimento acumulado de forma a torná-lo palpável, e também para trazer ao conhecimento dos gerentes de equipes a existência desse cenário tão amplo, real, e de suma importância nos dias atuais.

1.2 Objetivos

O campo da gestão de pessoas, assim como diversos outros relacionados às ciências não-exatas, tem uma abrangência enorme. Existe uma vasta gama de literatura que trata do assunto de forma acadêmica, formulando teorias e técnicas sobre como melhor gerenciar pessoas, especialmente no que se trata do modelo tradicional de empresa, onde o trabalhador tem horário fixo para entrar e sair do trabalho e o gestor tem pleno controle sobre o trabalhador e sobre o ambiente de trabalho, podendo monitorá-lo de perto.

Nosso objetivo aqui não é repetir o que já foi dito, mas sim descobrir e documentar novas maneiras de gerenciar pessoas num mundo onde cada vez as fronteiras físicas são menos significativas e o trabalho ocorre à distância, com grande flexibilidade de horários. Iremos buscar respostas a muitas das dúvidas que são comuns a gestores que subitamente se deparam com essa situação. Iremos documentar convenções desenvolvidas ao longo dos anos para resolver problemas enfrentados por pessoas envolvidas no movimento do software livre. Faremos comparações com técnicas já estabelecidas da gestão de pessoas, com o objetivo de contextualizá-las aos dias atuais e ao contexto de trabalho à distância. Mapearemos casos de sucesso e de fracasso ocorridos no movimento do software livre como possíveis sugestões para lidar com questões semelhantes nos ambientes corporativos. Abordaremos assuntos variados, como estratégias para comunicação e resolução de conflitos, formação de equipes, motivação e incentivo à criatividade, inteligência organizacional e faremos uma análise de alguns comportamentos padrão que podem ser encontrados num ambiente onde há muita liberdade e pouca hierarquia, para ajudar os gestores de pessoas a identificar cenários semelhantes quando estes se apresentarem, para que

¹Existem exemplos de pessoas que são remuneradas diretamente pelas suas contribuições a projetos de Software Livre, através de projetos como o *Google Summer of Code*, mas essa é a exceção e não a regra. Na maioria dos casos as contribuições a projetos de Software Livre são feitas ou no tempo livre do programador, ou como necessidade ou parte de sua ocupação principal, mas nesses casos a remuneração não pode ser relacionada diretamente à contribuição feita aos projetos de Software Livre.

possam estar prontos a lidar com eles.

1.3 Metodologia e Organização

De forma a alcançar esses objetivos, começaremos pela conceituação de alguns tópicos pertinentes ao assunto, explicitando relações entre os mesmos de forma a tornar clara a linha de pensamento seguida para chegar às conclusões finais. Essa conceituação constituirá a primeira parte do trabalho, encontrada logo após este capítulo introdutório.

Com o intuito de trazer outros pontos de vista pertinentes à análise do assunto, buscaremos a opinião de gerentes que tenham um perfil compatível com os objetivos propostos aqui, ou seja, gerentes que tenham conhecimento sobre o funcionamento das comunidades de software livre e suas formas de organização, que estejam envolvidos na gerência do trabalho colaborativo à distância e que possam contribuir com sua experiência, com casos de sucesso e fracasso e com sugestões para evitar potenciais problemas identificados na análise do trabalho colaborativo à distância.

Para colher as opiniões desses gerentes formularemos um questionário, o qual servirá de base para uma entrevista por telefone que será gravada e posteriormente transcrita na íntegra. Algumas entrevistas serão feitas na Língua Inglesa, mas a transcrição das mesmas será feita em Português. O questionário, assim como a transcrição das entrevistas, será anexado ao fim deste trabalho para referência. Os arquivos de áudio das entrevistas serão disponibilizados em formato MP3 junto com este trabalho. Para facilitar a comparação entre as respostas dos entrevistados, as respostas serão tabuladas e resumidas.

Optamos pelo uso da entrevista por telefone como forma de colher essas opiniões para aproveitar o aspecto dinâmico da conversa em tempo real e poder explorar e aprofundar melhor as opiniões dos entrevistados que possam ser pertinentes ao objetivo do trabalho.

Por fim, serão analisadas as respostas dos entrevistados em vista das questões e observações resultantes da conceituação dos tópicos deste trabalho. Baseado nessa análise, elaboraremos sugestões que possam servir de apoio à gestão do trabalho colaborativo à distância. Esta análise poderá ser encontrada no capítulo final deste trabalho.

2 GESTÃO DE PESSOAS

Apesar de toda a pesquisa já feita até os dias atuais, o trabalho de um gerente ainda é um assunto de considerável incerteza. Mintzberg mostrou através de estudos de atividades que o trabalho de um gerente é caracterizado por fragmentação de atividades breves e variadas, com frequentes interrupções. Um gerente prefere contatos verbais, mas ao mesmo tempo tem que lidar com quantidades substanciais de papel, e grande parte do seu tempo é gasto em reuniões, algumas não programadas. Pressão é criada pela velocidade com que essas atividades são conduzidas. Há muito pouco tempo para reflexão, e os gerentes tendem a operar focados no resultado imediato ao invés de objetivos a longo prazo (THOMSON, 2002).

2.1 Significado da Gestão de Pessoas

Christopher Bellavita analisou o significado da gestão, se fazendo perguntas como: Qual o significado da gestão em si? Será que ela envolve lidar com pessoas ou coisas como orçamentos, papéis, arquivos, ou talvez ambos? Ela envolve tomar decisões, e se envolve, por quem? É uma profissão ou uma combinação de muitas profissões? Ser um gerente efetivo é algo que se nasce sabendo ou algo que pode ser aprendido? Há alguma diferença entre gestão, administração e liderança ou estes são apenas nomes diferentes para a mesma coisa (BELLAVITA, 1990)?

Bellavita analisou muitos livros sobre gestão e encontrou alguns pontos de vista diferentes mas complementares:

Peter Drucker (DRUCKER, 1974) descarta a definição tradicional do gerente como alguém que é “responsável pelo trabalho de outras pessoas” e olha para a gestão como uma atividade que envolve “planejamento, organização, integração e medição”.

Gerald Caiden (CAIDEN, 1971) tem uma definição útil da gestão, e diferencia a gestão da administração. Caiden diz que “gestão é o termo preferido nos negócios, ... e administração é o termo preferido na administração pública. Ambos porém se referem ao processo, à maneira que as pessoas fazem as coisas acontecer através de

outras pessoas em um estrutura organizacional”.

William Oncken (ONCKEN, 1984) definiu a gestão de forma semelhante a Caiden, exceto que para Oncken a gestão envolve fazer as coisas acontecerem “através do suporte ativo de outros”.

Bellavita refletiu sobre essas e outras idéias e chegou à conclusão que a gestão envolve duas coisas, processos administrativos e trabalhar com pessoas. Para Bellavita, os processos administrativos são atividades que precisam ser cumpridas pelo gerente e pelas pessoas que trabalham com ele. Já a gestão é responsável em fazer com que as coisas aconteçam dentro de uma organização *com* o suporte ativo, livre e voluntário de outras pessoas para alcançar os objetivos da organização.

2.2 Atividades na Gestão de Pessoas

Um Engenheiro de Mineração chamado Henri Fayol passou grande parte de seu tempo trabalhando em um complexo metalúrgico e de mineração na França, primeiro como Engenheiro e depois mudou para Gerente, finalmente alcançando o posto de Gerente Diretor. Através de sua prolongada observação de outros gerentes, ele concluiu que todos os cargos de gestão contém os mesmos cinco elementos, apesar de que alguns gerentes isolados em indústrias distintas podem colocar mais ênfase em alguns elementos do que outros. (THOMSON, 2002) Os cinco elementos da gestão apontados por Fayol foram:

Previendo e Planejando olhando para o futuro; garantindo que os objetivos da organização estejam sendo atingidos; previsão a curto e longo prazo; estar pronto a adaptar planos à mudanças nas circunstâncias; tentando predizer o que irá acontecer.

Organizando garantindo que a estrutura da organização permita que as atividades básicas sejam cumpridas; direcionando; definindo responsabilidades; tomando decisões e apoiando-as através de um sistema eficiente de seleção e treinamento de pessoal.

Comandando talvez fosse melhor usar a palavra *liderando* aqui já que esse item se refere ao relacionamento entre um gerente e o seu/sua equipe em relação à tarefa sendo cumprida incluindo aconselhamento, reconhecimento, dando retorno, alocando trabalho e assim por diante.

Coordenando garantindo que indivíduos, equipes e departamentos trabalhem em

harmonia rumo aos objetivos compartilhados da organização.¹

Controlando garantindo que os outros quatro elementos estejam sendo cumpridos; operando sanções caso necessário (PUGH; HICKSON, 1996).

A pesquisa de Fayol e outros trabalhos mais recentes são baseados na observação do que os gerentes fazem. Enquanto isso é perfeitamente válido e nos dá uma indicação das qualidades necessárias para um bom gerente, não nos diz nada a respeito das qualidades necessárias para o futuro. A razão que torna essas mudanças inevitáveis é que o ambiente gerencial está em constante mudança, talvez mais rápido do que nunca na história humana, e as organizações e os gerentes precisam se adaptar e responder a essas mudanças. (THOMSON, 2002). Podem ser citadas como mudanças principais:

1. A globalização aumentando a pressão competitiva de forma que esta não está limitada a um país mas afeta organizações através do mundo.
2. Novas tecnologias, especialmente na área de TI, estão mudando a natureza do trabalho em si.
3. A gestão de performance irá constantemente buscar a integração de vários aspectos da Gestão de Recursos Humanos nos objetivos da organização como um todo.
4. Tendências demográficas como aumento no número de mulheres nos postos de trabalho, aumento da média da faixa etária e maior diversidade cultural.
5. Também haverá mais trabalhadores atípicos com maior flexibilidade de horários e através do trabalho à distância.
6. Dentro dos papéis de um gerente há uma crescente ênfase na liderança e na habilidade de concretizar mudanças ao invés da gestão no sentido de organizar uma situação estável e complexa (THOMSON, 2002)

Como podemos ver, Thomson identificou uma tendência crescente de que atividades da Gestão de Recursos Humanos sejam cada vez mais utilizadas pela Gestão de Pessoas, principalmente no que diz respeito à performance das pessoas gerenciadas. Mas quais exatamente são as atividades que compõem a Gestão de Recursos Humanos?

¹A diferença entre comandar e coordenar está em que o primeiro tem o poder dado, ou seja tem autoridade sobre o comandado. Coordenar, pode significar que quem coordena não tem a autoridade sob o ponto de vista hierárquico, mas tem o papel de gerir um trabalho através da negociação, convencimento e motivação. O primeiro dá ordens, o segundo sugere que algo deva ser feito de uma determinada maneira.

De acordo com (BRATTON; GOLD, 2003), a Gestão de Recursos Humanos é “aquela parte do processo de gestão que se especializa na gestão de pessoas dentro das organizações. A Gestão de Recursos Humanos enfatiza que os funcionários são cruciais para alcançar uma vantagem competitiva sustentável, que práticas de recursos humanos devem ser integradas na estratégia corporativa, e que especialistas em recursos humanos ajudam os controladores organizacionais a alcançar tanto objetivos de eficácia quanto de igualdade.”

A Gestão de Recursos Humanos é um campo de conhecimento e conjunto de práticas que definem a natureza do trabalho e regulam o relacionamento empregatício.

Segundo a definição de (BRATTON; GOLD, 2003), a Gestão de Recursos Humanos engloba as seguintes áreas funcionais:

Seleção a escolha de pessoas com as apropriadas habilidades, conhecimento e experiência para preencher postos de trabalho na organização. Práticas pertinentes englobam o planejamento de recursos humanos, análise de vagas, recrutamento e seleção.

Recompensa o design e administração de sistemas de recompensa. Práticas incluem avaliação do trabalho, a quantificação de performance e benefícios.

Desenvolvimento analisar requerimentos de treinamento para garantir que os funcionários possuam o conhecimento e habilidades para cumprir satisfatoriamente suas tarefas e progredir dentro da organização. A quantificação de performance pode identificar habilidades chave e competências dos funcionários.

Manutenção a administração e monitoramento da segurança no local de trabalho, políticas de saúde e bem-estar para manter uma força de trabalho competente e estar de acordo com as leis e estatutos reguladores.

Outra área de atuação que faz parte da Gestão de Recursos Humanos, e que está sendo aos poucos absorvida pela Gestão de Pessoas, é a resolução de conflitos entre membros de uma equipe. Dentro de uma equipe de trabalho, em particular quando essa equipe é formada por pessoas altamente especializadas, é inevitável o surgimento de disputas, algumas que irão ser resolvidas internamente pela equipe e outras que precisarão de intervenção do gerente. A controvérsia e oposição são atitudes esperadas dentro de um time (CLELAND, 1996).

O conflito pode surgir por inúmeros motivos, desde o comportamento dos membros do time até a discordância sobre o trabalho da equipe, e mesmo discordância sobre os pontos de vista profissionais. Uma regra essencial para lidar com os conflitos é confrontar o problema e não a pessoa. Um líder que constantemente trabalha para

desenvolver uma cultura de equipe onde a discordância é esperada e bem-vinda irá descobrir que o conflito é um instrumento facilitador para buscar melhores soluções e recomendações (CLELAND, 1996).

Há várias maneiras de resolver conflitos dentro de uma equipe. O modelo proposto por (CLELAND, 1996) é simples e composto de poucos mas importantes passos:

1. Compreender os fatos
2. Determinar os problemas relevantes
3. Entender o potencial impacto na organização
4. Considerar as alternativas
5. Gerar recomendações coletivas ou separadas

2.3 Considerações Finais sobre a Gestão de Pessoas

A Gestão de Pessoas ainda é uma área do conhecimento onde incertezas ainda existem apesar de toda a pesquisa existente. Além disso, diversas mudanças no ambiente gerencial, tanto local como globalmente, requerem que o perfil do gerente se modifique e se adapte para poder responder a essas mudanças. Um exemplo que pode ser citado é a necessidade de se adotar atividades da Gestão de Recursos Humanos como parte da Gestão de Pessoas de forma a monitorar melhor a performance das pessoas gerenciadas.

É possível identificar atividades que compõe a Gestão de Pessoas atualmente pela análise das atividades de gerentes em diversas áreas, conforme Fayol fez, mas isso não significa que essas são as atividades que um gerente deveria desempenhar. Concluímos, porém, que se essas são as atividades básicas que um gerente desempenha, elas devem ser atividades necessárias à Gestão de Pessoas. Podemos partir dessas atividades básicas para identificar quais as dificuldades encontradas pela Gestão de Pessoas ao desempenhar essas atividades, frente às mudanças citadas por Thomson. De forma mais objetiva, podemos relacionar diretamente as mudanças citadas por Thomson como conseqüências diretas do surgimento do Trabalho Colaborativo à Distância.

Mas de que forma podemos perceber o impacto dessas mudanças em um nível mais amplo? Podemos observar através de nossa experiência pessoal que os pontos citados se aplicam também à gestão de projetos de desenvolvimento de Software Livre. E talvez sejam ainda mais visíveis no contexto do desenvolvimento de Software Livre devido à natureza inerentemente diversa e distribuída dos indivíduos envolvidos nesses projetos.

No próximo capítulo iremos nos aprofundar um pouco mais na definição de Times Virtuais, dos quais o desenvolvimento de Software Livre é um dos mais visíveis exemplos (KIDANE; GLOOR, 2007). Iremos pesquisar principalmente as características que diferenciam essa forma de organização e os problemas que surgem devido ao aspecto distribuído desses times.

3 TIMES VIRTUAIS

Uma “organização virtual” pode representar uma tarefa, projeto ou organização permanente que é descentralizada e independente de conexão espacial ((FISHER; FISHER, 1998) e (HOEFLING, 2001), citados por (OKKONEN, 2002)). De acordo com Okkonen, a forma mais simples de uma organização virtual é o time virtual, que é um time distribuído que utiliza a tecnologia para garantir melhor conectividade, compartilhamento do conhecimento e baixos custos. A diferença entre um time virtual e um time comum é sua dispersão espacial, e muitas vezes cronológica. O time ainda é formado com o objetivo de concluir uma tarefa em comum, mas as pessoas não precisam estar no mesmo local físico.

É importante frisar a diferença entre um grupo de trabalho e um time. Muitos grupos de trabalho se definem como times, quando na verdade não o são. Um exemplo que ilustra bem a distinção é oferecido por Trina Hoefling (HOEFLING, 2001) “Galinhas em um galinheiro encontram-se num mesmo local e se espera que todas cumpram a mesma tarefa - pôr ovos. Seu trabalho de pôr ovos porém, não requer que elas trabalhem juntas para cumprir essa tarefa. Na realidade, construir um galinheiro com cooperação para pôr ovos não faria muito sentido e provavelmente só atrasaria a produção.”. Seguindo essa lógica podemos caracterizar um grupo de trabalho como uma equipe onde os membros da mesma trabalham independentemente para cumprir tarefas isoladas. Já um time é definido por uma equipe de pessoas que trabalha colaborativamente para um objetivo comum.

Duas características bastante marcantes que contribuem para o sucesso dos times virtuais são a Autonomia e o “Empowerment”.

3.1 Autonomia e “Empowerment”

O que é o “empowerment”? O empowerment pode ser representado pela seguinte expressão:

$$\text{Empowerment} = f(\text{Autoridade}, \text{Recursos}, \text{Informacao}, \text{Responsabilidade})$$

$$\text{Empowerment} = 0 \text{ se } (\text{Autoridade ou Recursos ou Informacao ou Responsabilidade} = 0)$$

Isso significa que somente quando as quatro variáveis se encontram presentes as pessoas se sentem responsáveis e agem responsabilmente.

Times Auto-Dirigidos são a forma mais avançada de empowerment. Uma definição pode ser dada a seguir:

Time Auto-Dirigido (pronome): Um grupo de pessoas que tem responsabilidade diária de gerenciar a si mesmo e ao seu trabalho com um mínimo de supervisão direta. Membros de um Time Auto-Dirigido tipicamente lidam com delegação de tarefas, planejamento e agendamento de trabalho, tomam decisões relacionadas a serviço e/ou produção e tomam ações para resolver problemas.

Enquanto grupos tradicionais tipicamente são organizados em tarefas especializadas e separadas com responsabilidades restritas, times auto-dirigidos são conjuntamente responsáveis por processos de trabalho completos, com cada indivíduo cumprindo diversas tarefas distintas. Enquanto uma organização tradicional pode ser dividida em grupos de especialistas funcionais, times auto-dirigidos são comumente responsáveis por produtos ou serviços completos ou podem também ser responsáveis por uma base geográfica ou de clientes. Isso é feito para criar, quando possível, pequenas entidades auto-sustentáveis que podem ser gerenciadas conjuntamente pela organização como um todo (FISHER, 2000).

3.2 Problemas e Dificuldades com Times Virtuais

Devido à natureza distribuída dos times virtuais, há diversos fatores que podem influenciar o bom funcionamento do time. A maioria desses fatores simplesmente não existe quando se trata de um time que está localizado num único local.

Iremos explorar a seguir alguns dos fatores citados como problemáticos e que são de grande importância para a gestão do trabalho colaborativo à distância.

3.2.1 Encontros Face a Face e Relações de Confiança

Muitos dos aspectos relacionados à criação de times virtuais são direta ou indiretamente relacionados à tecnologia. Porém há uma área na qual o trabalho virtual depende principalmente das pessoas, que é o estabelecimento das relações de confiança. Essa é uma área na qual os avanços em tecnologia não podem tornar as coisas mais rápidas. A barreira das relações de confiança é uma das barreiras mais citadas em relação aos times virtuais (HOEFLING, 2001).

3.2.2 Comunicação

A comunicação entre membros de um time é um dos principais fatores para se prever a efetividade de um time. Comunicação efetiva é sinal de time efetivo.

Além disso, a comunicação tem impacto direto sobre a satisfação e performance dos membros de um time. Em se tratando de times virtuais, especialmente em nível global, a comunicação é ainda mais complexa por ser feita através de computadores ou mesmo telefone, mas principalmente porque envolve uma interação intercultural (FERRIS; GODAR, 2006).

Múltiplos fatores podem contribuir para que hajam falhas de comunicação em times virtuais, reduzindo a efetividade do time. Dentre esses fatores, podemos citar:

- **Língua Falada e Escrita:** Na maioria dos casos a língua comum utilizada para a comunicação entre os membros do time localizados nos mais diversos países é a língua inglesa. Infelizmente a língua inglesa não é a língua oficial em grande parte dos países, por isso as diferenças em níveis de aptidão na comunicação falada e escrita podem causar dificuldades na assimilação de requisitos e na transferência de conhecimento entre membros do time devido ao fato de que uma parte da informação é perdida na tradução. Além disso, a língua falada possui mais um fator de complicação que é a diferença no sotaque (SHACHAF, 2005).
- **Diferenças Culturais:** Vários aspectos culturais podem interferir na comunicação. Grupos culturalmente diversos exibem níveis mais baixos de integração e coesão (WATSON; KUMAR, 1992). Membros de times com diversidade cultural não compartilham os mesmos modelos mentais, o que permitiria um melhor entendimento compartilhado (KLIMOSKI; MOHAMMED, 1994; CRAMTON, 2001). Isso acaba diminuindo a confiança entre membros do time, e criando falhas de comunicação, além de criar estereótipos, o que dificulta a validação de idéias, dificulta decisões e consenso, resultando em queda da produtividade (ADLER, 1991).

A comunicação não-verbal, em particular a comunicação escrita utilizando e-mail ou chat, pode ajudar a diminuir as falhas de comunicação, por possibilitar o uso de ferramentas como corretores ortográficos para corrigir eventuais problemas de digitação e dicionários de sinônimos para ajudar a evitar ambigüidades (SHACHAF, 2005).

3.2.3 Produtividade e Criatividade

A observação das diferenças em produtividade nos times de desenvolvimento de Software Livre despertaram o interesse de pesquisadores em estudar os padrões de comunicação desses times na tentativa de descobrir relações entre as características desses padrões de comunicação e medidas de produtividade como performance e criatividade (KIDANE; GLOOR, 2007).

A análise dos padrões de comunicação temporal em comunidades de desenvolvimento de Software Livre é uma das ferramentas que torna possível a visualização das oscilações dos níveis de criatividade e performance ao longo do tempo. Yared Kidane desenvolveu uma pesquisa nesse sentido, utilizando-se informações acessíveis publicamente na forma de listas de discussão de um projeto de software livre chamado “Eclipse”, e de ferramentas automatizadas de análise de emails, capazes de extrair informações sobre as redes sociais formadas pela interação entre os indivíduos. O objetivo de Kidane foi tentar definir quais as características dos padrões de comunicação entre os indivíduos eram responsáveis por definir os times de desenvolvimento de software livre mais criativos e eficientes. Seu trabalho se baseou no trabalho de Cross e Cummings (CROSS; CUMMINGS, 2004), que demonstraram a importância de um indivíduo que está no menor caminho que conecta outros dois indivíduos (centralidade de intermediação), os quais não estão conectados, pode ser associada com a habilidade de se obter e aplicar informação relevante para resolver problemas (WEBER, 2004), porém a pesquisa de Kidane se baseou nos dados sobre produtividade e comunicação no âmbito do grupo, além da medida de densidade (conectividade direta entre todos os indivíduos do grupo), extraídos das listas de discussão do projeto “Eclipse”.

Para tornar essas informações quantificáveis, Kidane definiu dois índices:

$$\text{Índice de Criatividade} = \frac{\text{melhoramentos implementados no período}}{\text{defeitos resolvidos no período}}$$

e

$$\text{Índice de Performance} = \frac{\text{defeitos resolvidos no período}}{\text{defeitos encontrados no período}}$$

Partindo desses dois índices, Kidane formulou algumas hipóteses e as validou. As seguintes hipóteses foram formuladas por Kidane:

1. A centralidade de intermediação de um grupo é negativamente relacionada à criatividade e produtividade do grupo.
2. A densidade do grupo é positivamente relacionada com a criatividade e produtividade do grupo.
3. A produtividade e a criatividade são negativamente relacionadas uma à outra.
4. A produtividade do grupo é negativamente relacionada com o número de mudanças (flutuações) na centralidade de intermediação do grupo ao longo do tempo.
5. A criatividade do grupo é positivamente relacionada com o número de alterações na centralidade de intermediação do grupo ao longo do tempo.

Através da análise dos dados, Kidane chegou às seguintes conclusões:

- Quando os grupos se tornam mais centralizados (aumento na centralidade de intermediação), a produtividade diminui.
- Grupos descentralizados são proporcionalmente mais produtivos (consertaram mais defeitos no mesmo período de tempo).
- A performance e a criatividade tem relação inversa, o que significa que a lei do “custo da oportunidade” se aplica ao desenvolvimento de software livre, ou seja, a implementação de melhoramentos ocorre em prejuízo ao conserto de defeitos.
- As oscilações entre estrutura de comunicação hierárquica e descentralizada são forte indicador de criatividade.
- Grupos com maior estabilidade na densidade de suas inter-relações são mais produtivos.

3.2.4 Fuso-horário

Uma das principais barreiras à comunicação e um grande causador de conflitos dentro de uma equipe geograficamente distribuída é a diferença de fuso-horário entre os membros da equipe. Grandes diferenças de fuso horário podem dificultar ou até impossibilitar o agendamento de reuniões entre membros da equipe. Considere por exemplo uma reunião onde um dos membros do time está no Brasil e outro na Austrália. Durante uma boa parte do ano, o horário mais razoável para se agendar uma reunião é 19hs no Brasil e 8hs (da manhã do dia seguinte!) na Austrália. Enquanto um dos membros da equipe está cansado, no final do seu dia de trabalho, o outro ainda está sonolento, talvez não tenha nem tomado seu café da manhã ainda. Nessa situação talvez os membros do time não estejam no ânimo mais propício para se envolver em uma discussão mais acalorada.

3.3 Times Virtuais e Software Livre

O conceito de times virtuais é bastante amplo, englobando os mais variados campos de atuação, como por exemplo campanhas políticas e ações de caridade online até o desenvolvimento distribuído de produtos e de software e a terceirização de serviços de tecnologia da informação (GLOOR, 2006). O advento da tecnologias de comunicação e informação modernas, ao possibilitar a comunicação instantânea a nível global, foi crucial para permitir a colaboração entre entidades geograficamente distribuídas, funcional e culturalmente diversas (KIDANE; GLOOR, 2007).

O desenvolvimento de Software Livre, que tem como uma de suas características mais marcantes o desenvolvimento distribuído de software, é uma das manifestações mais proeminentes do funcionamento de times virtuais colaborando através da Internet. O papel da Internet é central no sucesso do desenvolvimento de software livre (KIDANE; GLOOR, 2007). A Internet possibilitou aos times virtuais se interligarem através da distância, tempo, cultura, departamentos e organizações, dessa forma criando alternativas “qualquer um/qualquer hora/qualquer lugar” em contraposto a estruturas organizacionais mais tradicionais onde a norma é “mesmo horário/mesma hora/mesmo lugar” (LUEG; FISHER, 2003).

3.4 Considerações Finais sobre Times Virtuais

A natureza distribuída dos Times Virtuais traz novos desafios ao campo da Gestão de Pessoas. A comunicação entre pessoas em locais diferentes, mesmo que dentro de um curto raio de distância requer muito mais esforço por parte dos envolvidos. Os canais de comunicação comumente utilizados não possuem largura de banda suficiente para transmitir toda a riqueza de informação que é necessária para uma comunicação efetiva. Há grandes chances de que informações importantes sejam perdidas, gerando conflitos e desentendimentos. Esses conflitos e desentendimentos também podem ser causados por diferenças culturais.

O papel do gerente, nesse contexto, é principalmente ser um facilitador da comunicação. É preciso orientar as pessoas de forma que o melhor meio de comunicação seja utilizado para cada tipo de troca de informações. No caso de diferenças culturais por exemplo, o gerente deve orientar à utilização de mecanismos de comunicação que minimizem essas diferenças. Pesquisas citaram a comunicação por texto como sendo a forma mais recomendada de comunicação quando os envolvidos tem dificuldades de comunicação através de voz.

O gerente deve criar mecanismos para possibilitar o estabelecimento de relações de confiança. Uma das maneiras de se fazer isso são os encontros face a face.

Conforme visto nesse capítulo, o desenvolvimento distribuído de software que ocorre em grande parte dos projetos de Software Livre é um dos exemplos mais claros e destacados da forma de funcionamento dos Times Virtuais, criando e produzindo conhecimento colaborativamente através da Internet. Pela nossa própria experiência, podemos constatar a veracidade dessa afirmação. Mas estranhamente os problemas citados aqui não são tão comuns nos projetos de Software Livre. Talvez isso seja um reflexo da característica voluntária de participação nesses projetos, que levaria as pessoas envolvidas a se esforçarem mais para contornar esses obstáculos?

Podemos claramente ver que a definição de Times Auto-Dirigidos pode ser aplicada aos projetos de Software Livre. Pela nossa participação pessoal e conhecimento

do funcionamento interno desses projetos, podemos relacionar a autonomia e o “empowerment” à auto-motivação das pessoas envolvidas em projetos de Software Livre. Se essas pessoas não tivessem um certo nível de autonomia e poder de decisão, seria impensável que elas se sentissem motivadas a contribuir de forma voluntária com esses projetos.

Um líder de um projeto de Software Livre necessariamente precisa ter muitas das qualidades que são necessárias a alguém que irá gerenciar o trabalho colaborativo à distância, da mesma forma que as pessoas envolvidas em um projeto de Software Livre (desenvolvedores e outros membros do projeto) precisam ter determinadas características para que um projeto tenha sucesso.

No próximo capítulo iremos buscar mais informações sobre as características de liderança e organização dos projetos de Software Livre, assim como as qualidades e fatores motivacionais das pessoas envolvidas nesses projetos.

4 SOFTWARE LIVRE

4.1 Definição

O conceito de Software Livre (Free Software) é bastante antigo, apesar de não ser conhecido por esse nome antes da década de 80 (REIS, 2003). Do ponto de vista da Free Software Foundation o Software Livre pode ser definido da seguinte forma:

Software Livre é o software cuja licença garante ao usuário liberdades relacionadas ao uso, alteração e redistribuição. Seu aspecto fundamental é o fato do código-fonte estar livremente disponível para ser lido, estudado ou modificado por qualquer pessoa interessada.

A expressão “Open Source”, por sua vez, é comumente utilizada para designar software da mesma categoria, em contextos parecidos (REIS, 2003). Mas, qual a razão para utilizar dois termos para se referir a conceitos tão semelhantes?

Por volta de 1998, o termo “Open Source” foi criado com o objetivo de desenfaturar o teor filosófico associado à liberdade do termo “Software Livre” (GACEK; ARIEF, 2004). Existia na época uma motivação para quebrar barreiras de preconceito vinculadas à defesa do princípio da liberdade, de tal forma que a adoção de um termo um pouco mais moderado melhoraria em muito as chances de convencer empresas a adotar este tipo de software e sua forma de desenvolvimento. Além disso, o “Free” em “Free Software” é ambíguo na língua inglesa. Ele pode ser interpretado tanto no sentido de como “liberdade” como “preço”, com “preço” sendo a utilização mais comum no mundo dos computadores e do software.

No que diz respeito aos direitos e deveres do usuário de um aplicativo que é Software Livre ou Open Source as premissas básicas são quase idênticas. Na prática, uma licença considerada Open Source (e não de software livre) tende a ser mais permissiva em relação à geração de produtos derivados não-livres (REIS, 2003).

Hoje em dia as barreiras de aceitação dentro das empresas e entidades não são mais tão significativas. Os conceitos de Software Livre e Open Source estão bastante difundidos, apesar de serem comumente confundidos por suas diferenças sutis.

A Free Software Foundation, com o intuito de dirimir algumas das dúvidas mais freqüentes inclui em sua “Definição de Software Livre” a seguinte declaração:

“Software Livre” não significa “não-comercial”. Um programa livre deve estar disponível para uso comercial, desenvolvimento comercial e distribuição comercial. Desenvolvimento comercial de software livre não é mais tão incomum; o software comercial livre é muito importante (STALLMAN, 2002).

Apesar da ênfase estar na liberdade e não no preço do software, a grande maioria das aplicações classificadas como Software Livre ou Open Source é disponibilizada de forma gratuita através da Internet. Não apenas isso, mas uma considerável parte dessas aplicações também é criada por programadores sem que estes sejam remunerados diretamente pelo seu trabalho no desenvolvimento dessas aplicações, apesar deste cenário estar se modificando gradualmente.

4.2 Motivação dos Desenvolvedores de Software Livre

O que leva essas pessoas a criar software para ser distribuído sem custo algum? Mais do que isso, o que leva essas pessoas a desempenhar, muitas vezes sem remuneração, alguma tarefa que são consideradas extremamente tediosas pela maioria dos programadores?

A premissa básica das definições de Software Livre e Open Source é de que o código fonte esteja disponível livremente para ser lido e revisado por qualquer um que tenha interesse em fazê-lo. Porém o fato de que alguém se interesse em fazer isso sem receber em troca remuneração vai contra tudo o que se sabe sobre o campo do software (GLASS, 2000).

Glass fez uma retrospectiva através da curta história do software de onde extraiu alguns fatos básicos que são recorrentes:

A maioria dos programadores detesta fazer manutenção em código alheio.

O senso comum é que manter o código de outra pessoa é o oposto do que se pode chamar uma atividade intelectual criativa. E o código dessas aplicações, apesar de não se usar o termo “manutenção”, necessita de muita manutenção.

A leitura de código não é algo que se ensina. É inevitável a comparação com outras disciplinas. A programação é a única disciplina onde se aprende a escrever antes de aprender a ler.

Programadores detestam estudar código. De acordo com um artigo publicado há aproximadamente trinta anos por Richard DeMillo, Richard Lipton e Alan Perlis, o campo de desenvolvimento de software simplesmente não possui os

processos sociais que formam uma parte essencial da cultura onde as provas matemáticas são desenvolvidas (DE MILLO; LIPTON; PERLIS, 1979). Isto é, nenhum programador se dispõe a gastar tempo esmiuçando código da mesma forma que os matemáticos o fazem com os teoremas matemáticos.

A maioria dos programadores sempre gostaram de dinheiro. Não é nenhuma surpresa que o salário dos programadores estejam nas alturas, e que algumas das pessoas mais ricas do mundo estejam envolvidas no ramo do software. A maioria dos programadores está atrás do dinheiro que acompanha o trabalho. Contraste isso com o fato de que a maioria das contribuições para o Software Livre e Open Source não são remuneradas. É possível imaginar alguém se interessar realmente nisso?

Segundo a opinião de Glass, é completamente ilusório imaginar que o campo do software sofrerá tal mudança de forma a se voltar para a manutenção, leitura, e estudo do software sem qualquer remuneração. Apesar de algumas partes dessa mudança serem desejáveis, e boa parte delas são, pensar que elas irão realmente acontecer em larga escala e modificar as regras da produção de software a nível mundial é ser completamente inocente a respeito da natureza humana. Quaisquer mudanças que possam ocorrer seriam limitadas a alguns poucos “cultos” surgindo em nichos culturais dentro do campo do software.

Apesar de tudo, milhares de pessoas contribuem com milhares de projetos de Software Livre ou Open Source diariamente. Alguns dedicam-se quase que exclusivamente a produzir software sem nenhuma remuneração direta, enquanto que a maioria possui um emprego ou alguma fonte de renda e dedicam apenas uma parte de seu tempo livre à produção desse tipo de software. Segundo Karl Fogel em seu livro “Producing Open Source Software” (FOGEL, 2005), quando perguntados muitos [voluntários] dirão que eles o fazem [contribuir] por que eles querem produzir um bom software, ou por que eles querem estar pessoalmente envolvidos no conserto de problemas que importam para eles. Mas estas não são as únicas razões envolvidas. Seria muito difícil imaginar que alguém possa permanecer envolvido em um projeto quando nenhuma palavra é dita em apreciação pelo seu trabalho, ou que ninguém responda às suas questões ou ouça a sua opinião em discussões. Fogel sugere que um desses motivos é o desejo nato do ser humano de trabalhar junto com outros, e de uma troca mútua de respeito através de atividades colaborativas. Vamos analisar alguns outros motivos freqüentemente citados por estudiosos do assunto.

4.2.1 Bem Comum

Quando pessoas não ambientadas com a realidade da produção de Software Livre e Open Source são confrontadas com esses conceitos a primeira reação e a primeira

justificativa que surge à mente é de que isso é uma forma de trabalho voluntário, feito com o objetivo principal de contribuir para o “Bem Comum da sociedade”. De certa forma, isso não deixa de ser verdade, especialmente em se tratando dos benefícios relacionados às entidades sem fins lucrativos.

Alguns dos princípios básicos do Software Livre e Open Source são compartilhados pelas entidades sem fins lucrativos, como por exemplo o compartilhamento dos mais variados recursos - sejam eles financeiros ou culturais - voluntarismo e colaboração. Quando chega a hora de investir em infraestrutura as entidades sem fins lucrativos enfrentam um dilema: será que os recursos financeiros são bem investidos quando aplicados em software proprietário, onde o único beneficiado direto é a empresa que produziu o software? Não seriam melhor investidos esses recursos se aplicados na causa em que a entidade está engajada? Além disso, grande parte dessas entidades sem fins lucrativos possui necessidades bastante específicas que nem sempre podem ser atendidas por software produzido em larga escala. Sem dúvida se o desenvolvimento dessas aplicações seguir um modelo de Software Livre ou Open Source, e essas aplicações forem disponibilizadas praticamente sem ônus a outras entidades com necessidades semelhantes, de certa forma elas estarão contribuindo socialmente e conscientemente ao evitar que essas outras entidades desperdicem dinheiro de forma desnecessária, além de diminuir ou até evitar depender de multinacionais estabelecidas em países desenvolvidos (YEE, 1999).

4.2.2 Respeito ou Reconhecimento

No livro “Hackers: Heroes of the Computer Revolution” (LEVY, 1994), S. Levy comenta: “Julgue as pessoas apenas pelo valor do que elas criam, não por quem elas são ou pelas credenciais que elas apresentam”. Essa é a essência de uma meritocracia pura - qualquer um pode participar e ser reconhecido pela qualidade do seu trabalho, independente de idade, *status*, ou qualquer outro aspecto externo. Certamente há custos envolvidos pois as credenciais ajudam a encurtar caminhos, economizando o tempo e energia que são gastos avaliando as capacidades de um indivíduo. Mas também há benefícios: mais energia é empregada em criação e menos em auto-promoção pois o seu trabalho fala por você (WEBER, 2004).

Grupos engajados em atividades colaborativas devem criar normas comportamentais de tal forma a possibilitar a obtenção e manutenção do *status* através de ações que contribuam para os objetivos do grupo. Essas normas não surgem do nada. Muitos desenvolvedores de Software Livre experientes afirmam que o respeito, e conseqüentemente o status, são adquiridos quando um indivíduo do grupo envia mensagens freqüentes e bastante explicativas, e essa conclusão não surgiu acidentalmente, pelo contrário, ela surge pois esses mesmos desenvolvedores são recompensados com respeito e status dentro de um projeto ao se envolver em longos

e intrincados argumentos, não importando se esses argumentos ajudam ou não ao projeto diretamente (FOGEL, 2005).

4.3 Comunicação

A habilidade de se comunicar clara e concisamente talvez seja a mais importante habilidade que alguém pode ter num ambiente de desenvolvimento de Software Livre. Ao longo do tempo essa habilidade se torna mais importante do que o simples talento para desenvolver programas. Um grande programador que se comunica de forma pobre consegue fazer apenas uma coisa de cada vez, e ainda assim terá grandes problemas em convencer os outros a prestar atenção. Já um péssimo programador com grande habilidade para se comunicar pode coordenar e persuadir muitas pessoas a fazer diversas coisas, e dessa forma ter maior influência na direção de um projeto (FOGEL, 2005).

Nos projetos de desenvolvimento de Software Livre, talvez mais proeminentemente do que em outros tipos de projetos, a forma com que alguém se comunica é de extrema importância para que essa pessoa seja ouvida e respeitada. “Você é o que você escreve” é um ditado bastante utilizado, e para exemplificar a importância dessa frase Karl Fogel cita uma história contada por Jim Blandy, onde uma pessoa estava contribuindo de forma significativa para o projeto GNU/Emacs e ganhou considerável respeito entre os desenvolvedores. Em certo ponto se tornou necessário que essa pessoa passasse por um processo burocrático para que suas contribuições fossem aceitas, e quando Jim pediu que os papéis fossem assinados pelo empregador ou por um responsável na instituição educacional, o indivíduo respondeu “Bem, eu tenho apenas treze anos e moro com meus pais”. O fato de que esse menino de treze anos não escrevia da forma que se espera de alguém com sua idade não revelou a ninguém quem ele era.

Para que a comunicação seja efetiva é recomendável observar algumas sugestões. A atenção à estrutura e formatação de uma mensagem eletrônica é provavelmente um dos fatores mais importantes. Recomenda-se que as mensagens sejam enviadas em texto puro ao invés outros formatos, que podem ser opacos a aplicativos leitores de mensagens que não suportem formatação. Isso é ainda mais crítico hoje em dia pois muitas pessoas passaram a usar aplicações baseadas na Web para ler seus emails, e estas em sua maioria não tem a mesma capacidade que uma aplicação mais poderosa em seu computador tem para exibir a mensagem (FOGEL, 2005).

Outra sugestão é focar no conteúdo da mensagem. Quando uma mensagem é enviada para um fórum público ou lista de discussão não se pode esperar que todos os leitores tenham o mesmo nível de conhecimento sobre o assunto, e nem que eles estejam a par do contexto da discussão relativa à sua mensagem. Incluir informação

que contextualize melhor o assunto sendo discutido de forma conveniente para os leitores é muito importante, especialmente quando se leva em consideração o fato de que sua mensagem poderá ficar disponível na Web por muitos e muitos anos e que pessoas certamente irão encontrá-la quando estiverem buscando informações sobre assuntos semelhantes. É importante frisar que, principalmente quando se espera que outros concordem com sua opinião ou que ajudem você a solucionar um problema, exagerar e ser rude são atitudes que não levam a lugar algum e somente contribuirão para diminuir o respeito que outros tem por você (FOGEL, 2005).

A identidade é de grande importância na comunicação online. Na falta de uma presença física, de um rosto ao qual associar uma personalidade, o seu nome ou o apelido que você utiliza para a comunicação nos diversos meios é o que será utilizado por outras pessoas para identificar que se trata de você.

4.4 Liderança e Hierarquia

Comumente o movimento do software livre é caracterizado como tendo por sua característica principal a estrutura não-hierárquica (RAYMOND, 2001). Grande parte das análises teóricas sobre o sucesso do movimento do software livre são baseadas no ponto de vista de economistas, ao quais não é surpreendente que um sistema de inovação distribuído e desagregado tenha sucesso maior do que uma alternativa hierarquicamente organizada, porém uma análise mais prática de uma ampla gama de projetos de software livre revela que na prática apenas os projetos que possuem algum tipo de estrutura hierárquica, seja por escolha democrática ou por mérito, são os que obtêm sucesso (HEALY; SCHUSSMAN, 2003).

Um aspecto interessante sobre a liderança nos projetos de software livre é o fato de que um líder de sucesso em um projeto de software livre não é necessariamente um líder por natureza ou por escolha, mas alguém em quem os programadores que contribuem para o projeto podem confiar, alguém cujos objetivos são suficiente semelhantes aos seus próprios, e que não sejam poluídos por interesses comerciais, políticos ou egocêntricos (LERNER; TIROLE, 2002).

Healy e Schussman chegaram à conclusão de que, contrariamente à sabedoria convencional, a hierarquia tem sim uma importância vital para o sucesso de projetos, se não em todos os níveis pelo menos no que diz respeito às relações entre os desenvolvedores principais e os demais contribuidores de um projeto (HEALY; SCHUSSMAN, 2003).

4.5 Considerações Finais sobre Software Livre

Ao longo desse capítulo observamos algumas características marcantes que definem uma pessoa que contribui com o desenvolvimento de um projeto de Software Livre.

Percebemos que essas pessoas possuem um alto nível de motivação, sendo que elas desempenham voluntariamente atividades que são consideradas o oposto de uma atividade criativa, atividades que são repudiadas pela maioria dos programadores envolvidos em projetos de desenvolvimento de software.

Notamos também que esses indivíduos são auto-motivados e que a qualidade técnica de um indivíduo em um projeto distribuído não é tão importante quanto a sua capacidade de se expressar e se comunicar, de buscar, produzir e disseminar a informação dentro do seu grupo.

Quanto à liderança dentro dos projetos de Software Livre, identificamos que o papel principal de alguém que gerencia esse tipo de projeto é ser um líder com objetivos alinhados aos indivíduos participantes do projeto. Essa pessoa também precisa ser capaz de se comunicar clara e concisamente, de expressar os seus objetivos de forma transparente para que seja construída uma relação de confiança entre o líder e os membros do projeto.

Ao contrário do que é comumente dito a respeito dos projetos de Software Livre, foi identificado que a hierarquia é sim um aspecto importante na organização desses projetos, a tal ponto que a esmagadora maioria dos projetos de Software Livre que obtém sucesso possui uma estrutura hierárquica.

Identificamos que a comunicação é um fator essencial nesse contexto, e que as relações de confiança são construídas a partir da interação entre os diferentes indivíduos e a forma com que eles se expressam. De fato, conforme o exemplo citado a identidade de um indivíduo é construída a partir da forma que ele se apresenta, independente de idade ou de qualquer outra característica física.

Munidos dessas informações, temos agora um conhecimento sobre quais pontos podem ser problemáticos nos Times Virtuais, e que alguns desses problemas aparentemente não ocorrem em projetos de Software Livre. Com o objetivo de explorar melhor esses fatores apontados como problemáticos em relação a Times Virtuais, Software Livre e Gerência de Pessoas, foi elaborado um questionário base, aplicado em forma de entrevista telefônica a alguns gerentes de pessoas envolvidos em projetos que possuem por característica principal a utilização de Times Virtuais para a produção de Software, tanto Livre quanto Proprietário, e que tem como membros de seus times desenvolvedores com experiência em desenvolvimento de Software Livre.

No próximo capítulo iremos fazer uma análise comparativa das respostas dos

entrevistados e da contribuição das mesmas para o melhor entendimento dessas questões.

5 ENTREVISTAS

Quando falamos de Gerenciamento de Times Geograficamente Distribuídos é essencial citar exemplos relacionados direta ou indiretamente ao Movimento do Software Livre, por motivos citados anteriormente: a natureza intrinsecamente distribuída desse modelo de desenvolvimento de software as características organizacionais desses projetos. A adoção de características organizacionais e de modelos de desenvolvimento inspirados pela análise do relativo sucesso obtido por estes projetos tem sido recomendada por muitos defensores do Movimento do Software Livre como uma alternativa viável aos modelos tradicionais de organização e desenvolvimento de software. Algumas empresas tem levado essa recomendação a sério e colocado essas idéias em prática, criando uma nova categoria de empresas que caminha na fina linha que separa o modelo tradicional de organização e desenvolvimento do modelo empregado por um projeto de Software Livre. Podemos citar vários exemplos, tanto no Brasil quanto no Exterior, de empresas que se encaixam nessa descrição.

5.1 Apresentando os Entrevistados

Entrevistamos Christian Reis, sócio-proprietário da “Async Open Source”, empresa brasileira que produz aplicações comerciais utilizando um modelo baseado em Software Livre, e co-gerente de desenvolvimento do projeto “Launchpad” na “Canonical Ltd.”, empresa fundada pelo empreendedor Mark Shuttleworth que é a principal patrocinadora da distribuição GNU/Linux “Ubuntu”. Como parte do seu envolvimento no projeto Launchpad, Christian Reis é responsável por definir prioridades, calendários, desenvolvimento de funcionalidades, controle de qualidade entre outras atribuições. O time envolvido no desenvolvimento do projeto Launchpad é geograficamente distribuído com desenvolvedores em pelo menos 8 países distintos, em fuso-horários diferentes e com linguagens nativas diferentes.

Outro entrevistado foi Paul Everitt, sócio-proprietário da “Agendaless Consulting” e fundador da “Zope Europe”, entidade sem fins lucrativos focada na divulgação e no crescimento do servidor de aplicações Zope no continente europeu.

Paul Everitt foi responsável pela gestão de um projeto ambicioso entre membros da Zope Europe que juntou sobre uma única cobertura desenvolvedores de empresas associadas à Zope Europe em cinco países diferentes da Europa para atender à Oxfam GB, baseada na Grã-Bretanha. Além desse projeto, Paul se envolveu em outros dois projetos em conjunto com a empresa Enfold Systems, Inc., baseada em Houston, TX, EUA, onde gerenciou funcionários da Enfold Systems, consultores externos e desenvolvedores do cliente sendo atendido.

O terceiro entrevistado foi Alan Runyan, sócio-proprietário da Enfold Systems, Inc. e fundador do projeto Plone, uma ferramenta de gerenciamento de conteúdo online e Open Source. Alan gerenciou o desenvolvimento no projeto Plone por vários anos, até iniciar sua própria empresa de consultoria, a Enfold Systems, que tem desenvolvedores no Brasil, Austrália e Estados Unidos. Alguns dos projetos de consultoria da Enfold Systems terceirizam serviços para empresas em outros continentes e também desenvolvedores “freelance” de vários países como Alemanha, Ucrânia, França e Áustria.

Apresentamos a seguir uma comparação lado a lado das respostas dos três entrevistados, organizadas por tópicos.

5.2 Comparação das Respostas

Christian Reis	Paul Everitt	Alan Runyan
Processo de Seleção		
Inicialmente era baseado apenas no aspecto técnico. Foi modificado para levar em conta a habilidade de comunicação.	Em projetos anteriores a gama de escolhas foi limitada, baseado na experiência. Considera que a característica essencial é a confiança.	Boa comunicação escrita e auto-motivação são essenciais. Capacidade de criar relações de confiança à distância.
Encontros Face a Face (Relações de Confiança)		
Não são importantes. Na comunidade de Software Livre isso não era comum até pouco tempo atrás.	Essencial, principalmente no início do projeto. É fácil estereotipar alguém que você não conhece pessoalmente.	Essencial para os gerentes conhecer pessoalmente os desenvolvedores. Talvez não seja tão importante entre os desenvolvedores. Alguns desenvolvedores tiveram melhores experiências após encontrar pessoalmente com clientes.
Fuso-horário		
Causa muitos problemas. Divisão de equipes entre Ocidente e Oriente. Não poder encontrar a pessoa no IRC ou telefone é muito frustrante. Reduzir comunicação entre pessoas que não estão no mesmo fuso-horário.	Pode ser aproveitado como uma oportunidade, mas requer um gerente pró-ativo, que esteja disposto a acordar mais cedo ou ficar no trabalho até mais tarde para estabelecer a comunicação.	Horários muito distintos são um problema sério sem um processo sofisticado de gerência. Funciona melhor se as pessoas forem auto-motivadas e se a gerência tiver qualidade técnica e sofisticação.
(continua na próxima página)		

Tabela 5.1 – continuação da página anterior

Christian Reis	Paul Everitt	Alan Runyan
Comunicação		
<p>Comunicação por voz não é um problema se não há pessoas de países do oriente como membros do time. Porém a comunicação por voz é desaconselhada quando requer que os indivíduos envolvidos na conversa tomem decisões arquiteturais que necessitam amadurecimento de uma idéia. Nesses casos o email é mais recomendado pois o indivíduo tem tempo de pesquisar, analisar fatos e somente após isso responder à questão.</p>	<p>Os meios de comunicação baseados em voz favorecem os indivíduos cuja língua nativa é o inglês pois o tempo de resposta deles é muito menor. Os indivíduos que não tem o inglês como língua nativa precisam de mais tempo para articular uma resposta.</p>	<p>A diferença da velocidade de compreensão e de formular uma resposta é um forte causador de falhas de comunicação. Os indivíduos envolvidos na equipe devem ser orientados que não percam muito tempo elaborando uma resposta e simplesmente sejam francos e peçam que a pessoa diminua o ritmo ou repita o que foi dito de forma a tornar mais claro.</p>
(continua na próxima página)		

Tabela 5.1 – continuação da página anterior

Christian Reis	Paul Everitt	Alan Runyan
Performance e Métrica		
<p>Medir a performance é talvez o problema mais sério em um projeto distribuído. Antes de termos um processo de release ao qual associar tarefas estávamos perdidos. Montar grupos de trabalhos pequenos com três pessoas, uma sendo o líder do grupo e responsável em entender o que está sendo feito e discutir os detalhes também ajudou. Esse líder é o termômetro dentro da equipe. A um nível macro, o progresso é medido em quantidade de tarefas. O líder do grupo faz as estimativas.</p>	<p>A chave é saber “gerenciar a pilha”. Não há porque falar sobre o que será feito daqui a três meses. É necessário reduzir o radar dessas pessoas a um horizonte sem distrações, onde fica mais fácil medir a velocidade delas. De forma incremental você calibra e ajusta as estimativas das tarefas.</p>	<p>Usamos basicamente o número de tarefas cumpridas e de horas produtivas trabalhadas. Concordamos como um time no tempo estimado para cada tarefa. Idealmente deveríamos comparar o tempo real que uma tarefa levou com a sua estimativa.</p>
(continua na próxima página)		

Tabela 5.1 – continuação da página anterior

Christian Reis	Paul Everitt	Alan Runyan
Criatividade e Troca de Conhecimento		
<p>A criatividade é centralizada em algumas poucas pessoas. Há abertura para que outros dêem idéias, mas é muito difícil que essas idéias sejam aceitas. Se existe hierarquia, quem está no topo decide se a idéia é boa ou não. Se a estrutura é horizontal, é mais fácil que a idéia seja aceita. Encorajamos muito o compartilhamento de conhecimento entre a equipe. Existe uma cultura de que o conhecimento deve ser difundido entre a equipe.</p>	<p>É importante encorajar o compartilhamento de informações e a solicitação de ajuda e de “feedback” de outros membros da equipe, mesmo que isso tome tempo e torne um indivíduo mais lento. Se as pessoas não forem incentivadas a colaborar e a serem criativas então elas farão isso. As pessoas irão agir de acordo com o seu interesse próprio ao invés de colaborar para o progresso do grupo.</p>	<p>Acredito que há formas de se facilitar o processo criativo. Temos feito reuniões semanais onde uma pessoa é responsável por falar sobre um tópico que ela tenha pesquisado, e possibilitar a criação de novas idéias. Já a troca de conhecimento é muito mais custosa quando as pessoas não estão no mesmo local físico. Simplesmente não existe a situação em que alguém faz uma pergunta a outro, um terceiro ouve e intervém. Na realidade não há muito compartilhamento de conhecimento dentro de nossa empresa hoje.</p>
(continua na próxima página)		

Tabela 5.1 – continuação da página anterior

Christian Reis	Paul Everitt	Alan Runyan
Conflitos		
<p>É muito mais fácil se desentender com alguém à distância do que localmente. Um desentendimento à distância é normalmente uma falha de comunicação. Um desentendimento local, normalmente é uma incompatibilidade séria. Em vários casos a gente teve desentendimentos sérios. Na maior parte das vezes se as pessoas se desentendem seriamente, mudam as equipes. Demora para chegar nesse ponto, mas normalmente é a solução correta.</p>	<p>Quase todos os problemas com pessoas que não são responsáveis e causam problemas, são os que em geral causam conflitos e tomam a maior parte do tempo do gerente de projeto. Mas há outros tipos de conflitos, como por exemplo a respeito da especificação. Quando terminamos um dos projetos que mencionei, tivemos um comentário irônico de alguém do lado do cliente. E essa pessoa disse: “Paul nos deu o que nós pedimos, agora nós precisamos descobrir se era isso que queríamos mesmo”.</p>	<p>Diferenças de fuso-horário podem causar conflitos se não houver um processo. Os maiores conflitos que tivemos foram causadas por falhas de comunicação, problemas de agendamento e transferência de conhecimento. É bastante comum as pessoas não enxergarem as mesmas prioridades quando elas estão em um escritório remoto quando comparado às pessoas que estão no dia a dia no mesmo escritório.</p>

5.3 Considerações Finais sobre o Resultado das Entrevistas

Nas entrevistas foi solicitado aos entrevistados que fizessem uma análise das suas equipes para saber se eles funcionavam como um grupo de trabalho ou como um time. Isso levou a uma introspecção mais profunda da forma como suas equipes se organizavam. Por fim, os entrevistados acabaram concordando que na verdade suas equipes deveriam estar funcionando como um time, mas devido à falta de uma melhor comunicação da visão do projetos em que essas equipes estavam envolvidos, em grande parte do tempo na verdade essas equipes estavam funcionando como grupos de trabalho. Os indivíduos envolvidos não conseguiam ver o projeto como um todo portanto trabalhavam independentemente, apenas visando seus objetivos diretos e não o objetivo do projeto como um todo.

Os entrevistados em geral foram unânimes quanto à sua opinião sobre o impacto do fuso-horário sobre a equipe. Em sua opinião, é impraticável ter membros da equipe em fuso-horários extremamente distintos a não ser que uma de duas condições seja satisfeita: a total independência entre as tarefas dos membros que estão em fuso-horários extremamente distintos ou a presença de um sofisticado processo de gestão que ajudasse na comunicação entre os membros do time. É também necessário que um gerente nessa situação seja pró-ativo, que esteja disposto a acordar às seis da manhã se necessário para poder ter uma comunicação síncrona com membros do time que não estão disponíveis no seu horário normal de trabalho.

Quando questionados a respeito dos diferentes tipos de comunicação que utilizavam para gerenciar os membros da equipe e também utilizados para a comunicação entre os membros da equipe, todos afirmaram alternar sua comunicação entre falada e escrita sem muito critério.

De acordo com os entrevistados, eles nunca tiveram problemas com a comunicação falada pois não tinham como membros de seus projetos indivíduos de culturas orientais, que são citados por Schachaf como sendo os mais problemáticos quando se trata desse tipo de comunicação.

Foram citados como mecanismos de comunicação de uso freqüente o IRC, Skype¹, IM, email e telefone. De acordo com os entrevistados, esses mecanismos podem ser classificados em pelo menos duas categorias: mecanismos de comunicação síncrona e assíncrona, onde os meios baseados em texto são assíncronos (não há expectativa que uma resposta ocorra dentro de um tempo determinado) e os meios baseados em voz são síncronos. Um dos entrevistados afirmou que gostaria de estar utilizando video-conferência mas não teve uma boa oportunidade ainda.

Uma observação importante foi levantada sobre a comunicação baseada em voz,

¹Aplicação VoIP multiplataforma que permite ligações gratuitas de computador para computador, desde que ambos usem a mesma aplicação, ou ligações para telefones convencionais a um custo bastante reduzido

e aparentemente todos os entrevistados estavam cientes da questão. Uma comunicação baseada em voz tem um fator que pode a tornar indesejável mesmo quando os indivíduos não tem problema nenhum em se comunicar dessa forma: tipicamente essa comunicação não é gravada o que torna difícil referenciá-la. A solução recomendada pelos três entrevistados foi que sejam feitas anotações durante a discussão e que essas anotações resumam o conteúdo das conversas e as ações que resultaram delas e que os indivíduos envolvidos na discussão revisem e concordem com o resumo. Dessa forma é possível referenciar o resumo para em uma discussão posterior como registro das ações acordadas. Sem esse tipo de registro a comunicação baseada em voz seria praticamente inútil pois os indivíduos esqueceriam imediatamente o que foi discutido.

As características de auto-motivação e autonomia foram citadas como essenciais para o funcionamento de uma equipe geograficamente distribuída pelos entrevistados. Todos relacionaram essas características como sendo inerentes da maioria dos desenvolvedores de Software Livre. Porém a auto-motivação está diretamente ligada à questão da voluntariedade, da liberdade em escolher qual tarefa será cumprida ao invés da imposição no cumprimento de determinada tarefa.

A autonomia foi citada por Alan Runyan como um aspecto que pode definir o sucesso de um projeto. Em suas próprias palavras:

...se essas pessoas são inteligentes e auto-motivadas, mesmo se o seu gerente for incompetente essas pessoas conseguem ir além disso com sua motivação e mesmo assim ter sucesso, executar o que é necessário...

A necessidade de encontros face a face como forma de estabelecer identidade e relações de confiança foi questionada por Christian Reis. Seu ponto de vista foi de que os encontros face a face não eram economicamente viáveis até pouco tempo e que na comunidade de Software Livre isso nunca foi necessário, porém ele concordou que algumas pessoas poderiam se beneficiar de um encontro face a face mas nem todas necessariamente precisam disso. Paul Everitt porém defendeu os encontros face a face com o seguinte argumento:

...é muito fácil criar um estereótipo a respeito do cliente ser estúpido, ou do cliente estereotipar os desenvolvedores como sendo preguiçosos. É fácil estereotipar alguém com quem você nunca tomou uma cerveja. Se você tem um projeto que nunca entrará em crise, isso não é tão importante. Mas realmente não existe esse tipo de projeto. Então quando as coisas ficam difíceis, ao invés de apontar dedos, você precisa ter uma equipe que quer que os outros tenham sucesso porque eles se conheceram face a face.

Christian defendeu os encontros face a face como algo desejável para comunicar

idéias complexas mais facilmente a um grupo grande de pessoas, pois nesse caso a largura de banda da comunicação é essencial para uma boa transferência da mensagem e dos objetivos desejados.

Segundo Christian Reis, fica difícil definir se a hierarquia é criada para gerenciar o sucesso do projeto (ou seja, é algo adicionado após um certo tempo do início do projeto) ou se ela é sempre necessária, sendo um fator determinante do sucesso do projeto. Alan Runyan sugere que a hierarquia é sim necessária, principalmente para servir como ponto de referência para a resolução de questões quando as mesmas não estão sofrendo nenhuma ação.

A utilização de métricas para a medição da performance de pessoas num contexto de trabalho geograficamente distribuído foi um assunto bastante unânime. Nenhum dos entrevistados utiliza processos rígidos de métrica. Os três entrevistados deste trabalho citaram que suas métricas são vagamente definidas em relação ao número de tarefas cumpridas, em alguns dos casos levando-se em conta a estimativa da equipe sobre o relativo esforço envolvido no cumprimento da tarefa. Também foram unânimes ao dizer que confiam nas estimativas fornecidas pela equipe sobre o tempo necessário para cumprir cada tarefa.

Em se tratando de conflitos, foi unânime a afirmação que o principal causador de conflitos em uma equipe geograficamente distribuída é a falta de confiabilidade de um indivíduo do time. Essa falta de confiabilidade comumente se manifesta em atrasos na entrega de tarefas. Os conflitos emergem quando um membro da equipe depende de outro e esse membro não comparece ou não entrega o resultado prometido em tempo hábil. Paul Everitt considera imperdoável que um indivíduo atrase uma tarefa e não deixe o seu gerente saber que a tarefa irá atrasar até o último minuto. No seu ponto de vista um bom gerente deve se preocupar em minimizar as catástrofes, diminuindo o tamanho das unidades de trabalho e comunicando possíveis problemas ao cliente o quanto antes possível. Segundo ele, as más notícias não envelhecem muito bem. Além do mais, isso afeta a moral da equipe como um todo, levando à “síndrome da janela quebrada”.²

²Em um artigo da Atlantic Monthly de 1982 intitulado “Broken Windows”, James Q. Wilson e George Kelling argumentaram que a desordem em uma comunidade, se não corrigida, torna inútil os esforços dos residentes daquela comunidade em manter suas casas e vizinhanças livres de comportamentos descontrolados. “Se uma janela em um prédio for quebrada e não for consertada, todas as outras janelas logo serão quebradas também... Uma janela não consertada é um sinal de que ninguém se importa, então quebrar mais janelas não custa nada... Uma propriedade sem atenção se torna um alvo fácil para pessoas que estão interessadas em se divertir ou saquear.”

6 CONCLUSÕES

A Gestão de Pessoas está evoluindo e se adaptando a mudanças no contexto em que se insere. Essas mudanças, citadas no capítulo sobre Gestão de Pessoas, são em grande parte resultado do aparecimento do Trabalho Colaborativo à Distância.

O conceito que melhor define essa forma de trabalho é o de Times Virtuais. Times Virtuais são times distribuídos que utilizam a tecnologia para garantir melhor conectividade, compartilhamento do conhecimento e baixos custos (OKKONEN, 2002).

Como exemplo mais destacado de Times Virtuais, temos os projetos de Software Livre, que se utilizam da tecnologia para o desenvolvimento geograficamente distribuído de software (KIDANE; GLOOR, 2007).

Devido a essa característica de que os indivíduos estão geograficamente distribuídos, surgem obstáculos ao funcionamento dos Times Virtuais. As áreas que foram identificadas como mais problemáticas nesse contexto foram a comunicação, relações de confiança, criatividade e produtividade, porém as duas primeiras são menos problemáticas quando se trata de projetos de Software Livre.

O fator motivacional tem grande influência sobre essa observação. Devido à participação voluntária e à auto-motivação dos indivíduos participantes dos projetos de Software Livre, há uma pré-disposição dos mesmos em facilitar o estabelecimento de relações de confiança e também de serem claros e concisos na comunicação individual ou em grupo. O fato de que a forma de se expressar é responsável por definir a identidade do indivíduo frente ao grupo é um incentivo a dispender esforço extra nesse sentido.

Podemos afirmar que a gestão de equipes geograficamente distribuídas possui características que a tornam bastante diferente da gestão de equipes mais tradicionais, localizadas em um mesmo escritório. Uma atenção especial deve ser dada à escolha dos indivíduos que irão trabalhar nesse ambiente. Algumas características são marcantes e essenciais para o sucesso desse tipo de equipe. As principais características identificadas foram:

- Bom inglês técnico, escrito e falado

- Habilidade de superar as barreiras de comunicação num ambiente geograficamente distribuído
- Capacidade de estabelecer relações sociais e de confiança sem conhecer pessoalmente um indivíduo
- Auto-motivação e autonomia para superar possíveis deficiências de um gerente tradicional

Todas essas características se encontram presentes em indivíduos que participam ativamente de comunidades de Software Livre. Segundo Paul Everitt, são características tão comuns a esses indivíduos quanto a água é comum para o peixe. Dessa forma, o processo de recrutamento de indivíduos para uma equipe geograficamente distribuída deve levar em conta esse aspecto e procurar ativamente por indivíduos que se encaixem nesse perfil.

O gerente deve estar atento às falhas de comunicação que possam ocorrer devido às características pessoais de cada indivíduo. Além disso, o gerente deve ser um facilitador da comunicação, criando canais de grande largura de banda para a troca de idéias e conhecimento, incentivando os indivíduos a buscar ajuda quando se sentirem travados e a comunicar com antecedência possíveis atrasos decorrentes de problemas inesperados.

Os indivíduos precisam ser orientados quanto à melhor forma de comunicação a ser utilizada em cada caso. A boa utilização dos recursos existentes e a atenção em se utilizar a ferramenta adequada para se comunicar em cada caso é de vital importância para evitar falhas de comunicação e conflitos.

Quando da utilização de conversas telefônicas para resolução de problemas, é essencial que sejam feitas anotações sobre os assuntos discutidos e que essas anotações sejam aprovadas pelo grupo envolvido na conversa. Uma conversa telefônica sem um registro por escrito ou gravação não possui valor prático pois não pode ser referenciada.

Nossa conclusão principal nesse trabalho é que a Gestão de Pessoas no Contexto do Trabalho Colaborativo à Distância é mais relacionada com a orientação e liderança de indivíduos auto-motivados do que às atividades gerenciais tradicionais.

Identificamos que há características que tornam um indivíduo mais indicado para participar desse tipo de trabalho. O processo de seleção é bastante importante.

Uma sugestão para trabalhos futuros seria estabelecer critérios para um processo de seleção que possibilite identificar indivíduos mais aptos a trabalhar em times virtuais.

Outra sugestão seria elaborar um processo de formação que possa desenvolver essas características identificadas aqui como essenciais para um indivíduo que irá fazer parte de um time virtual.

Por fim, poderíamos sugerir uma exploração mais aprofundada e definição de um mecanismo que possibilite identificar qual a forma de comunicação mais adequada para determinadas situações. Seria muito interessante pesquisar qual o impacto que a vídeo-conferência por exemplo poderia causar sobre as relações de confiança como substituição dos encontros face a face.

APÊNDICE A QUESTÕES ENTREVISTA

1. Fale um pouco do seu histórico. Como você se envolveu inicialmente com projetos de Software Livre? Como foi a transição da posição de desenvolvedor ou contribuidor com projetos de Software Livre para a posição de gerência?
2. Fale um pouco sobre os projetos em que você se envolveu, especialmente aqueles onde os indivíduos envolvidos na equipe se encontravam geograficamente distribuídos.
3. Nestes tipos de projetos onde os membros da equipe se encontram geograficamente distribuídos, é mais comum os indivíduos interagirem como um grupo de trabalho (onde cada um trabalha independentemente dos outros membros do grupo) ou como uma equipe (onde todos trabalham de forma colaborativa com o mesmo objetivo comum)?
4. Em geral, você se envolve na escolha dos indivíduos que farão parte da equipe desses projetos?
5. Quais são as características e aptidões que você considera como mais importantes ao selecionar um indivíduo para que irá fazer parte de uma equipe geograficamente distribuída?
6. Um dos aspectos citados como essencial para a performance e conseqüentemente o sucesso de um projeto onde a equipe envolvida no projeto é geograficamente distribuída é o desenvolvimento de laços de confiança entre os indivíduos. Nos projetos que você gerenciou, você organizou alguma atividade social onde os membros da equipe se encontraram face a face de forma a estabelecer laços de confiança?
7. É comum nos seus projetos que ocorram modificações na equipe durante o tempo de vida do projeto? Como os indivíduos reagem a essas mudanças?
8. Diferenças de fuso-horário podem ao mesmo tempo ser uma vantagem (com indivíduos trabalhando independentemente em horários distintos) ou uma

desvantagem (quando acontecem bloqueios devido à dependência de resultados entre pessoas em fuso-horários diferentes). Nos projetos que você gerenciou, a diferenças de fuso-horário foram uma vantagem ou uma desvantagem, e por quais razões?

9. Outro fator essencial para o sucesso de um projeto geograficamente distribuído é a comunicação. Diferenças culturais e de linguagem (sotaque, gírias) tem grande impacto sobre a comunicação. Alguns estudos sugerem que diferentes formas de comunicação podem ajudar a minimizar essas diferenças. Por exemplo, a utilização de email ou outras formas de comunicação textual podem ajudar a minimizar as diferenças culturais. Por outro lado, na comunicação textual as emoções dificilmente são transmitidas com a mesma fidelidade. Quais são os diferentes mecanismos utilizados para a comunicação entre os membros da equipe?
10. As falhas de comunicação, sejam elas causadas pela diversidade cultural ou pelo tipo de canal de comunicação utilizado são um forte causador de conflito. Que tipos de problemas de comunicação você experimentou nos projetos em que esteve envolvido?
11. Avaliar a performance e a qualidade do trabalho de um indivíduo é bem mais simples quando esse indivíduo se encontra no mesmo ambiente físico, você pode simplesmente caminhar até a pessoa e verificar pessoalmente se ela esta sendo produtiva ou não. Avaliar alguém que se encontra num local remoto é muito mais difícil devido ao estreitamento dos canais de comunicação. Quais as métricas que você utiliza para a avaliação da performance de um indivíduo (por exemplo, quantidade de testes, performance da aplicação criada, número de problemas consertados)?
12. Falando sobre a criatividade e os processos de troca de conhecimento e de resolução de problemas. Como esses aspectos são tratados na sua organização? A informação e o conhecimento são centralizados em um pequeno número de pessoas ou fluem livremente entre todos os membros da equipe? Os membros da equipe são encorajados a buscar e a compartilhar novas idéias e soluções junto ao coletivo?
13. Finalmente, sintá-se à vontade para expressar sua opinião sobre os desafios e as decepções ao gerenciar uma equipe geograficamente distribuída em comparação à gerência de uma equipe que se encontra no mesmo local físico.

APÊNDICE B ENTREVISTA: CHRISTIAN REIS

S: Então Kiko, comece falando um pouco da sua história, como é que você entrou nesse mundo do Software Livre, de programação. Sua formação foi em Engenharia de Software não é?

C: É, eu fiz Engenharia de Computação na graduação.

Durante a graduação, eu fiquei meio desiludido, porque eu achava que era muito trivial a nossa profissão. É engraçado porque durante a faculdade em si, eu não sinto que os professores dão um parecer honesto do que a gente realmente faz enquanto engenheiros. Então foi essa a impressão que eu tive, de que na graduação ninguém fez um esforço especial em explicar o que era realmente interessante sobre a nossa profissão. E para mim o interessante é muito menos o lado técnico... que é interessante, ou seja, lidar com tecnologia, com solução de problemas é interessante, mas de uma forma geral eu acho muito mais interessante as dificuldades, os desafios, as coisas que a gente não faz bem na nossa profissão. Aí eu sinto que é uma coisa que vale a pena a gente se dedicar.

E a outra coisa que me incomodava muito durante a graduação era o fato de que tudo tinha que ser feito no (Microsoft) Windows. No começo, quando a gente usava o (Microsoft) DOS eu achava até interessante, mas quando passou a usar o (Microsoft) Windows, tudo passou a ser muito... muito certinho, muito maquiado, muito... simplificado. Eu fiquei com a impressão de que eu não estava... que aquilo ali não era... eu não sei explicar direito. Eu suponho que é porque eu achava que não era “underground” o suficiente. Eu achava que o (Microsoft) Windows era um negócio muito trivial, que qualquer pessoa, qualquer usuário sentava e usava então não era um negócio que eu estava muito interessado. E quando alguém me falou sobre o (GNU/)Linux eu pensei: “Esse negócio é interessante, cara”.

Eu tinha muito contato com o trabalho “online” porque quando eu estava no meu segundo grau eu fiquei muito envolvido com modem, com BBS, então eu já estava meio acostumado com esse negócio de usar uma rede para (me) comunicar, então eu estava meio acostumado com isso aí. Quando apareceram o (GNU/)Linux e o Software Livre, com essa mesma idéia de que online as pessoas colaborariam (para o

desenvolvimento do GNU/Linux), isso foi uma coisa que me chamou muita atenção. Eu falei “Cara, isso aqui é o futuro, as pessoas vão usar a Internet pra trabalhar juntas. As pessoas vão trabalhar à distância e construir as coisas colaborativamente, baseado no que elas acham melhor e não porque tenha uma organização por trás ditando como deve ser”. Eu achei muito interessante essa mudança, eu falei “Software Livre é uma coisa muito especial, não tem nada parecido no mundo hoje em que valha a pena eu investir o meu tempo, investir a minha energia em aprender mais a respeito e a participar”. Essa foi a minha motivação inicial. Realmente foi algo bem idealista. Eu falei “Software Livre é o futuro, isso aqui é como a gente deve trabalhar”, então por isso eu me envolvi (com o Software Livre).

S: Mas até essa época você nunca havia ouvido falar de alguém que colaborasse à distância sem ser relacionado ao Software Livre?

C: Cara, no mundo... então, no mundo “online” anterior, onde (o acesso à Internet) era discado existia algum trabalho colaborativo. Desenvolvimento a gente fazia... eu tinha uns amigos e a gente fazia junto umas atividades de pirataria, de “hacking”, de “cracking” sabe? Então tinha um cara na Itália, um outro na Suécia, uns americanos com quem eu trocava idéias. E esse grupo aí trabalhava assim. Um cara fazia o software que a gente usava, era um tipo de “scanner dial-up”, a gente olhava, pegava o código-fonte, dava uma fuçada. Então tinha uma colaboração, por isso quando apareceu o Software Livre eu me senti muito confortável, porque quando eu participava dessas rede eu achava muito bacana. Achava “Pô, isso aqui é muito maneiro cara”. Ter amigos em qualquer lugar do mundo, trocar idéias, produzir algo que é útil para todos.

Então depois disso aí, eu fui viajar. Eu viajei em 98, trabalhei na Espanha, trabalhei nos Estados Unidos, trabalhei em Brasília. Vários trabalhos pequenos, todos eles envolvendo algo de Software Livre.

Quando eu voltei para o Brasil e comecei a trabalhar em Brasília, eu realmente pensei “Eu preciso sentar e aprender como funciona esse negócio de Software Livre. Preciso começar a trabalhar ativamente, como um participante. Preciso mesmo começar agora a me envolver mais como um usuário interessado como eu era, e mais como alguém que contribuía para os projetos. Eu preciso mudar isso.”, então passei um ano em Brasília fazendo bico mas participando de projetos (de Software Livre), lendo bastante, começando a escrever e contribuir código para os projetos. E aprendi muito sobre UNIX, sobre (GNU/)Linux nessa época, porque realmente eu sentei e dediquei quase que todo o tempo livre que eu tinha só para isso.

S: Nessa época você já tinha interesse em saber como os projetos se organizavam, como a comunicação entre as pessoas acontecia, como funcionavam as hierarquias dentro dos projetos de Software Livre ou você estava mais focado no desenvolvimento?

C: Não, eu não conseguia ver esse nível ainda. Você só percebe a questão da organização, de como funciona, quando você meio que transcende o nível básico. Você tem que ter algum tipo de experiência com o trivial, por que senão você só se preocupa com “Caramba, meu software não roda, minha compilação está falhando, tem uma biblioteca que está faltando...”, então quando você está envolvido com essas coisas no dia-a-dia você nem nota, mas assim que você dá um passo adiante e essas questões triviais passam a não ser mais as que mais te preocupam, aí você começa a perceber que existem os padrões, que os projetos se organizam de uma certa forma, de que existem certas pessoas que ajudam, certas pessoas que não ajudam, de que cada projeto tem um universo-zinho e é uma sociedadezinha especial. Então eu só fui reparar isso depois que eu voltei pra São Carlos, depois que eu comecei... mais ou menos em 2000-2001 que eu comecei a reparar isso aí, de que existia um outro nível, que não era simplesmente ter bons desenvolvedores. Não era simplesmente sentar e escrever código. De que existia uma questão social por trás que para mim era muito interessante, porque eu queria trabalhar em algo em que a questão social fosse muito significativa. E que ao mesmo tempo fazia muita diferença para o que você estava construindo. A questão social delimita o que você está construindo. É incrível isso aí, mas é assim que funciona.

S: E quando você começou a colocar em prática esse aprendizado que você teve durante esse período? Você abriu a sua própria empresa? Você se baseou nesse modelo de organização para a sua própria empresa ou você optou por um modelo mais tradicional?

C: Quando eu abri a empresa eu tinha a intenção de trabalhar com Software Livre, criar um lugar onde a gente pudesse sustentar o desenvolvimento de ferramentas que fossem ajudar o Software Livre. Eu queria contribuir para a idéia geral do software livre. Essa era a minha motivação. Eu não tinha ainda a idéia de usar desenvolvimento distribuído. Eu sabia que ia ser distribuído porque os projetos iam ser tocados por pessoas que não estavam no escritório. Mas o escritório em si eu nunca pensei em não ter um escritório. Eu acho que ter um escritório é uma coisa muito valiosa. Até hoje eu acho isso. Minha experiência com projetos de desenvolvimento distribuídos é de que há muitos problemas em se fazer desenvolvimento distribuído.

A minha empresa, quando eu criei ela, ela era totalmente centralizada. A gente trabalhava no escritório, nem tinha computador em casa. Realmente tinha um outro enfoque.

S: Você já tinha essa consciência de que fazer uma empresa com desenvolvimento distribuído seria mais difícil?

C: A gente não pensava nessa possibilidade. Eu era muito cético em relação a isso. E quando eu fiz o mestrado, no primeiro ano, eu li muita publicação, muitos

artigos. Eu li centenas de artigos. Eu tenho fotografias da minha semana de conclusão onde pela minha casa inteira tem artigo espalhado. Tipo, 160 artigos, algo assim. Um desses artigos é o artigo do Conway¹, que talvez você já tenha lido. O título é “Conway’s Law Revisited”². E ele fala sobre as dificuldades em desenvolvimento distribuído. Então nesse artigo é que me chamou a atenção o fato de que ele estava apontando que era possível fazer desenvolvimento distribuído, e que era muito problemático. Eu achei isso fantástico. Esse foi o primeiro artigo que de fato apontou que as pessoas estavam interessadas em fazer desenvolvimento distribuído. Até então pra mim, eu nem reparava que era interessante, que era importante.

Então essa idéia de desenvolvimento distribuído eu não tinha inicialmente. Mas assim que eu li esse artigo eu fui começar a reparar que realmente haviam muitos aspectos em Software Livre que tratavam com o desenvolvimento distribuído, e que com o Software Livre era bem menos um problema que com o Software Não-Livre. Eu pensei: “Cara, como é possível que esse cara está falando sobre todos esses problemas: problema de cultura, problema de comunicação, fuso-horário.” Eu acho que esse artigo é de alguém que trabalhava na Bell Labs, um estudo sobre a Bell Labs. Acho que ele estava falando sobre equipes em três lugares diferentes, tipo Estados Unidos, Alemanha e Holanda, algo assim. E no artigo ele fala sobre as diferenças culturais. Eu pensei: “Como esses caras, na Bell, onde tem tudo que precisam para comunicação, teleconferência, dinheiro, recursos para viagens. Como esses caras tem todos esses problemas, e ainda assim, em Software Livre onde se tem bem menos recursos, bem menos infraestrutura, as pessoas conseguem entregar software, conseguem produzir software.” Eu achava isso muito fantástico. E no meu mestrado uma das coisas que eu exploro é isso, eu falo sobre o que tem de especial no Software Livre que faz com que seja possível superar todas essas dificuldades.

S: Até esse momento você tinha apenas desenvolvimento local, dentro do seu escritório, aí passou a se envolver com o projeto do Ubuntu (distribuição GNU/Linux) onde há desenvolvimento tanto de Software Livre como Software Proprietário, correto?

C: Correto.

S: Então, como foi essa transição?

C: Quem a gente tem que agradecer por essa transição é você! Porque foi você que me ligou quando o Mark (Shuttleworth, patrocinador do projeto Ubuntu) estava

¹Melvin Conway publicou um artigo na revista “Datamation” em Abril de 1968 após esse mesmo artigo ser rejeitado pela “Harvard Business Review”. A tese apresentada por Conway nesse artigo foi a seguinte “Qualquer organização que crie um sistema (definido abertamente) produzirá um design cuja estrutura é uma cópia da estrutura de comunicação da própria organização.”

²O artigo “Conway’s Law Revisited” analisa as implicações da tese de Conway através de um estudo de caso sobre a parte mais difícil de um projeto de software distribuído: a integração dos diferentes módulos desenvolvidos por equipes diferentes em locais diferentes (HERBSLEB; GRINTER, 1999).

aqui (em Porto Alegre, no Brasil) e falou: “O Mark está aqui e acho que você deveria conversar com ele”, e eu falei “Tá bom”. Tinha acabado de chegar do Marrocos, no mesmo dia, estava entrando no avião aí você falou: “Encontra com o Mark em São Paulo”. Você foi o cara que me... na verdade eu não tinha menor noção do que se tratava. O que era o Ubuntu... na verdade não existia o Ubuntu ainda. Não tinha a menor idéia sobre nada. Eu tinha ouvido falar do Mark Shuttleworth só porque ele tinha oferecido uns “bounties”³ para o (projeto) Mozilla. Então ele tinha feito alguns comentários em alguns “bugs” mais ou menos um ano antes, em 2003. E eu lembro que as pessoas falaram assim: “Esse bilionário aqui está querendo patrocinar algumas coisas no desenvolvimento do projeto Mozilla, muito interessante”, e essas eram coisas (os “bugs”) que incomodavam o Mark pessoalmente. Não era nada de infraestrutura, nada disso. Mas você foi quem me chamou atenção para o fato de que o Mark estava fazendo algo interessante e me falou para ir lá conversar com ele. E eu conversei com o Mark e achei ele meio doido. Falei: “Caraca, esse cara ta muito doido cara”, e eu lembro que olhei assim a equipe e pensei: “Cara, esses caras vão trabalhar totalmente distribuídos, não sei como isso vai funcionar. Eles são malucos.”.

S: Eu não tinha noção do que eles iriam fazer na verdade. Não sabia que iria ser um projeto distribuído.

C: Eu pensei: “Esses caras são muito doidos, isso nunca vai dar certo.” Mas eu sabia que o Mark tinha dinheiro, e estava com vontade de fazer isso aí. Então nos começamos a conversar por email. A gente fez um contrato com a Async, mas eu não trabalhava no projeto, apenas alguns engenheiros que trabalhavam para a gente foram trabalhar lá. O Celso, o Salgado e o De Bonzi. Essas três pessoas foram trabalhar lá mas eu não trabalhava. Eu ia lá só para ajudar, para ver como as coisas estavam indo. Eu estava interessado, eu queria que desse certo, mas tinha muita coisa pra fazer aqui no escritório.

S: Quando você disse: “Eles foram trabalhar lá”, você quis dizer fisicamente ou eles trabalhavam do escritório da Async?

C: Bom, é interessante essa pergunta. A primeira coisa que a gente fez quando assinou o contrato foi ir para uma conferência em Oxford. O trabalho era para ser distribuído, mas eles tinham uma noção de que era importante trazer essas pessoas e reunir elas em um único lugar para conversar a respeito do que se estava fazendo. Mesmo de início, e mesmo eles tendo uma operação já completamente distribuída... por exemplo o Lalo era uma pessoa que estava trabalhando com eles na época, mas estava morando em Porto Alegre. Eles rodavam uma operação inteiramente

³A palavra “bounties” (plural de “bounty”) significa recompensa. Alguns projetos de Software Livre passaram a se valer de recompensas (monetárias) oferecidas para fazer com que problemas fossem consertados mais rapidamente.

distribuída. Tinha gente na Austrália, nos Estados Unidos, na Europa Continental, e no Brasil. E talvez por causa disso eles tinham uma noção de que era importante juntar as pessoas, trazer elas para perto.

S: Interessante... então eles já tinham uma idéia da importância da interação entre as pessoas, do encontro face a face para criar uma relação de confiança. De criar um vínculo interpessoal para que quando a conversa estiver acontecendo via texto a pessoa lembre: “Ah, eu conheço essa pessoa, a gente sentou juntos numa mesa para jantar e conversou”.

C: O Mark é muito esperto nesse sentido. Ele repara essas coisas rapidamente. Quando ele montou a empresa ele já sabia que iria ser importante reunir as pessoas. Ele sabia que é uma coisa interessante, que as pessoas que estão trabalhando distribuídas... muitas acham que é um benefício... que é uma coisa interessante o fato de que você viaja, conhece outros lugares, outras pessoas. Outra coisa que o Mark queria muito era que... ele sabia que explicar as coisas à distância era muito mais complicado. Era muito mais rápido colocar todos em uma mesma sala e falar sobre o que ele queria fazer.

S: Exatamente. Alguns dos artigos que eu li mencionam a importância do encontro face a face.

C: Mas é muito interessante você pensar que no Software Livre não existe esse problema.

S: Você não acha que há uma tendência a interpretar a pessoa de forma diferente uma vez que você encontrou ela face a face?

C: Nos projetos de Software Livre em que eu participei isso nunca fez uma diferença. “Online” eu conseguia perfeitamente resolver minhas diferenças. E naquela época, em 2002-2003 quase não existia “sprint” (encontro de desenvolvedores para uma sessão conjunta de desenvolvimento no mesmo local físico). Ninguém fazia isso, reunião, encontro, não existiam conferências, nem a GUADEC, nada. Então o Software Livre andou por muito tempo sem nenhum contato pessoal, isso é muito recente. A minha opinião é meio radical, mas eu concordo que há alguma vantagem em conhecer as pessoas, em explicar determinados assuntos pessoalmente, mas é bem menos significativo do que quando se desenvolve Software Proprietário. Na faixa de uma ordem de magnitude mais importante ter contato pessoal num projeto de Software Proprietário Distribuído do que em um projeto de Software Livre.

S: Então de início você disse não se envolver com esse projeto (Launchpad) pessoalmente. Quando você começou a sentir a necessidade de se envolver com o projeto?

C: Olha... sendo sincero eu achei que o projeto do Launchpad estava muito enrolado, em Janeiro-Fevereiro de 2005. Eu pensei: “Esses caras não vão conseguir entregar esse projeto. Eles estão sofrendo pra caramba, o Mark está ficando estres-

sado. Está todo mundo infeliz. O projeto está sendo mal-visto na empresa. Vou conversar com o Mark, eu acho que a gente pode mudar isso. Vai ser difícil, mas a gente pode fazer isso mudar e dar a volta por cima.”. E desde então estou tentando descobrir uma forma de resolver... de melhorar o processo de desenvolvimento. Basicamente o que eu fui fazer foi melhorar o processo.

S: Você já tinha uma noção do por quê de esse projeto estar nessa situação, sendo mal-visto e cheio de problemas?

C: É um projeto de software clássico, onde o projeto é mal ou não gerenciado. Nem o Mark nem o Steve (Alexander) nessa época tinham experiência com desenvolvimento em larga escala de sistemas, e eu já tinha. Então eu tinha uma idéia melhor de quais as dificuldades que as pessoas teriam em interagir, qual seriam os problemas do sistema a longo prazo, como fazer para manter as pessoas motivadas, como distribuir o conhecimento. Eu sabia que a gente tinha todos esses problemas, e eu acho que o Mark e o Steve tinham muito pouca experiência nisso, então eles foram aprendendo à medida que foram fazendo. Eles não tinham uma idéia clara de como gerenciar, eles foram amadurecendo a idéia. Era uma coisa fazer o Ubuntu, eles conseguiram fazer, mas para isso eles reuniram pessoas que sabiam o que estavam fazendo, e queriam fazer o que estavam fazendo. Tinha muito pouca interferência do Mark em como as coisas iam ser feitas. Basicamente era um Debian (distribuição GNU/Linux) modificado. Então não existia liberdade para o Mark falar: “Vamos fazer um sistema de pacotes dessa forma”. Não tinha engenharia pesada, era integração, garantia de qualidade e algumas melhorias. Era bem limitada a interação. No Launchpad, estava acontecendo uma coisa totalmente nova, então as chances de você errar, ou confundir, ou não explicar as coisas direito é muito maior.

S: Talvez as pessoas envolvidas no projeto (Launchpad) não entendiam o projeto como um todo?

C: É totalmente verdade. Demorou muito tempo para as pessoas entenderem o que era que a gente estava construindo. Primeiro porque não existia aplicação, não existia “site”, não existia nada, só uma idéia e algum código. A gente perdeu muito tempo para definir as coisas fundamentais, acho que ali a gente perdeu um tempo bastante significativo para definir as coisas básicas. Demorou muito tempo para decidir que as coisas seriam hospedadas num “site” central “launchpad.net”, a gente demorou muito tempo para tomar essas decisões fundamentais, essas decisões que iriam dar aos desenvolvedores uma idéia mais real, mais concreta do que a gente estava construindo. E eu vou dizer uma coisa: Isso é uma diferença cultura importantíssima entre o Software Livre e o Software Proprietário. Em Software Livre tem alguém que sabe o que está fazendo, que é uma idéia concreta e está caminhando naquela direção. Projetos bons de Software Livre tem alguém que tem uma idéia boa do que quer fazer e as pessoas que se integram ao projeto elas

absorvem aquela cultura, elas são auto-selecionadas e elas querem fazer aquilo ali. Então quando você participa de um projeto de Software Livre, uma das coisas que existe automaticamente é você escutar o cara que organiza o projeto.

S: Acho que essa questão que você citou é muito importante. A auto-seleção, a pessoa ser voluntária para aquele projeto. Mas no caso do projeto Launchpad não foi assim, vocês tiveram que escolher pessoas que iriam trabalhar nesse projeto não é?

C: É, a gente contratou pessoas que tinham um interesse geral nessa área, mas assim que você oferece um salário, “pessoas que tem interesse geral em nessa área” se torna um conceito muito mais amplo. Você pode falar para mim: “Você está interessado em Inteligência Artificial?” e eu iria falar assim para você, Sidnei: “Ah, sou interessado sim.”. “Participaria de um projeto de Software Livre em Inteligência Artificial?”, “Provavelmente não.”. Mas se você me oferecesse um salário significativo eu ia falar assim: “Eu acho que é uma coisa interessante para se trabalhar.”. Então a motivação... não estou dizendo que é uma questão mercenária, só estou dizendo que sutilmente o fato de que existe um trabalho remunerado altera a forma que você escolhe o projeto no qual você quer trabalhar. Então existe essa questão sim. Mas voltando ao ponto inicial, acho que grande parte das dificuldades foi que a gente não sabia direito o que a gente queria fazer no começo. Se nem o Mark, se nem o gerente sabiam, para os desenvolvedores era ainda mais obscuro.

S: Você acha que essas pessoas envolvidas no projeto tinham um perfil mais autodidata, mais autônomo, ou elas tinham um perfil onde era mais necessário ser gerenciadas?

C: Variava, Sidnei. O problema é, para mim transparecia que todo mundo precisava ser gerenciado, que não era possível fazer um projeto onde todo mundo se auto-gerenciasse. Mas eu acho que o principal motivo é porque não era claro o que tinha que ser feito. Se fosse claro o objetivo geral, se existisse uma certa liberdade pra você construir algo que você quisesse, algumas pessoas no projeto teriam se auto-gerenciado bem. Algumas precisariam de um gerente de qualquer forma, outras iriam se auto-gerenciar bem caso houvesse uma visão mais clara e um pouco mais de latitude no sentido de você poder escolher o que você acha que é a solução mais correta para um problema.

S: Então você considera importante também a liberdade de poder escolher tarefas dentro dos projetos?

C: Sim. Eu acho que no Software Livre existe uma... quando você está construindo alguma coisa, o gerente do projeto (de Software Livre) se sente motivado a aceitar a sua contribuição. Existe uma forte tendência a esse cara ser legal com você. Porque ele sabe que se ele te tratar mal você vai embora. Então se o cara sabe que se você mandar um “patch” pra ele e ele disser: “Esta totalmente errado

isso que você fez aqui, detesto a forma como isso foi feito”, você vai embora. Em projetos maiores, tipo o (GNU/)Linux, o “kernel” em si, acontece sem dúvida, mas o tamanho da comunidade é absurdo.

S: O tamanho é tão grande que não faz uma diferença.

C: Acho que até faz diferença, acho que muita gente se alienou do projeto do “kernel” porque fez muito trabalho e foi rejeitado. Posso dar exemplos concretos como o Richard Gooch, o cara que fez o “devfs” e mesmo o Eric Raymond que escreveu o “cml2”. Essas pessoas que tiveram seu trabalho rechaçado nunca mais voltaram a contribuir. Mas de forma geral a comunidade é tão grande que o projeto consegue se manter de forma saudável mesmo com essa característica um pouco atípica. E no fundo acho que para um projeto grande, daquele tamanho, você tem que ter muito mais cuidado naquilo que você aceita. O impacto sobre você aceitar alguma coisa no (GNU/)Linux (kernel) é muito maior do que se você aceitar num projeto pequeno. Agora de forma geral, eu penso no Johan (Dahlin, atual mantenedor do projeto “PyGTK” e contribuidor no projeto “GTK”) que trabalhou com a gente aqui. Ele cria uma solução para o “PyGTK” ou para o “GTK” que ele acha que é a correta, e não fica perguntando muito para ninguém, ele simplesmente faz. E depois “faz acontecer”. Se os caras não gostarem, ele dá um jeito de eles gostarem. Então existe uma diferença muito grande nesse sentido. Na Canonical (empresa por trás do Ubuntu e do Launchpad), existe muito mais uma pressão em se implementar a solução considerada correta pelos outros, tem muito mais detalhes. E isso é uma coisa que dificulta bastante num projeto distribuído. O único motivo pelo qual a gente pode fazer isso na Canonical é porque a gente paga os salários, porque se fossem voluntários e você rejeitasse o trabalho dos voluntários eles iriam embora.

S: Mudando um pouco de assunto, que outras questões assim você acha que influenciam a questão do trabalho distribuído? Que problemas de comunicação vocês tiveram? Pessoas em locais diferentes, com culturas diferentes, isso influenciou no projeto?

C: Todos os problemas possíveis num projeto distribuído a gente teve. Fuso-horário, desmotivação, falhas de comunicação, desentendimentos, erro de especificação... todos os problemas possíveis em desenvolvimento distribuído a gente teve. Então você pode me perguntar sobre coisas específicas e eu posso falar.

S: Bom, então comece falando sobre os problemas com o fuso-horário. Que problemas o fuso-horário causou para vocês?

C: Cara, vou dizer o seguinte. Eu não aceito ninguém trabalhando nos meus projetos que não esteja em um fuso-horário compatível com o meu. Eu não cogito contratar australianos, neo-zelandeses para trabalhar no Launchpad. Não cogito. Isso para mim está fora de cogitação. Porque o fato de você nunca poder encontrar

a pessoa no IRC ou no telefone é muito frustrante.

S: Então você teve pessoas que estavam nessa situação no projeto?

C: A gente teve e a gente precisou basicamente dividir a equipe para resolver esse problema. Então hoje existe uma parte da equipe que é australiana, que trabalha em partes do projeto com pessoas que são do leste, do oriente, e tem a equipe de pessoas que são do ocidente. Eu sou gerente da equipe do ocidente.

S: E outra pessoa gerencia a outra parte da equipe?

C: É, o Steve (Alexander) gerencia a parte do oriente. E é uma divisão engraçada porque é uma divisão tanto espacial quanto funcional. As pessoas do oriente trabalham em determinadas coisas, e as pessoas do ocidente trabalham em outras completamente diferentes.

S: O que eu acho que seria o correto para otimizar essas equipes e trabalhar em paralelo...

C: O fato é o seguinte, você quer reduzir comunicação intensa entre pessoas que não estão no mesmo fuso(-horário).

S: Tem outros fatores que você citou, fatores culturais...

C: Muitos problemas interessantes com relação a isso aí. Menos do tipo de problemas típicos, de que uma pessoa fala uma coisa que nessa cultura significa isso e em outra cultura significa algo diferente. Muito menos concreto e pedante. Muito mais sutil, sabe? Bem mais variação pessoal do que cultural.

S: Não houveram problemas com linguagem, por alguém não falar inglês direito ou algo assim?

C: Teve um pouco de problema sim, um pouco com os brasileiros e um pouco com os espanhóis. Mas de forma geral o inglês é muito bom na nossa equipe, muito bom.

S: Vocês usam mais comunicação falada ou é mais por email, mais escrita?

C: Usa muito IRC, muito email também. E bastante telefone. As três formas. E usa elas de forma um pouco diferente, em situações diferentes.

S: Em qual situação seria melhor usar o telefone do que o email por exemplo?

C: Quando tem muito para se falar, e são assuntos diferentes. Digamos que eu tenho dez assuntos diferentes para conversar com você, eu te ligo e resolvo esses dez. Mas o telefone não é tão bom para discutir algo técnico que precisa de paciência. Algo que precisa que você amadureça a idéia. Se você precisa amadurecer a idéia, email funciona melhor. O telefone é algo que você pode perguntar para a pessoa e ela vai te responder logo. Se é algo que a pessoa precisa pensar, que vocês vão pensar juntos, é raro usar o telefone.

S: E como você lida com o fato de que o que você fala no telefone gera um conhecimento que não é referenciável? Por exemplo você não pode dizer: "Ah, foi na ligação telefônica X, no minuto Y que assunto Z foi decidido.". Como você lida

com esse problema?

C: Não tem jeito. Lidamos mal. A gente tenta consolidar o que a gente conversou no telefone em alguma forma de documentação, ou num “bug”, ou num documento de especificação, ou num email para a lista (de discussão). Mas é algo que não funciona muito bem, então eu hoje tento restringir soluções telefônicas para as diferentes equipes. Eu tento fazer com que as pessoas dentro dos grupos de trabalho do Launchpad comuniquem-se mais pelo telefone, porque eu acho que dessa forma o conhecimento se dissociou pelo grupo no qual ele seria mais valioso.

S: Vocês fazem conferências telefônicas com várias pessoas, ou um a um?

C: Normalmente um a um, mas periodicamente faz-se uma teleconferência.

S: Voltando às diferenças culturais, alguns artigos citam a utilização do email como uma forma de minimizar os problemas de comunicação quando as pessoas tem sotaques e estilos de comunicação diferentes para minimizar essas diferenças.

C: Isso nunca aconteceu com a gente. Eu acho que de fato é um problema para o Oriente, para se trabalhar com chineses por exemplo. Eu acho que nesse caso a comunicação escrita funciona muito melhor.

S: O artigo que falava sobre isso citou chineses, japoneses e indianos.

C: É, eu concordo que para esses realmente a comunicação por email é melhor. É que quase todos os países com os quais a gente lida são países onde as pessoas em geral falam inglês bem. A gente escolheu padronizar a comunicação em inglês. A gente faz um esforço grande para que as pessoas falem inglês muito bem.

Você tem razão. Tem dois caras, acho que são escoceses na Canonical e eu lembro de o Gustavo Niemeyer me falou: “Cara, não entendo nada do que eles falam!”.

S: É, eu já tive problemas em entender australianos, o sotaque deles é totalmente diferente do britânico.

C: Eu pessoalmente nunca tive esse tipo de problema, sempre consegui me dar bem. Eu nunca observei esse problema na prática, de as pessoas preferirem usar email ao invés de telefone por causa de sotaque ou gíria. Na maior parte das vezes, quando você usa o telefone é porque tem que usar o telefone, por que tem muita coisa pra se resolver, ou alguma coisa crítica que você precisa fazer.

Existe uma resistência a usar o telefone. A gente tende a preferir usar o IRC e usar email, mas de tempo em tempo a gente usa o telefone. Às vezes é porque você precisa de um descanso, você digitou o dia inteiro e quer descansar as mãos. Ou às vezes é só porque você quer resolver alguma coisa urgente ou é porque você precisa discutir um monte de coisas diferentes ao mesmo tempo.

S: Como funciona a questão da criatividade dentro do time, quais os meios de comunicação quando se está criando uma idéia ou solução nova? Vocês usam o telefone, email, IRC, encontros face a face?

C: Varia... a gente na verdade dificilmente pensa em coisas totalmente novas.

S: Mas existe alguém que centraliza a criação, ou é uma discussão coletiva onde é aberto a todos dar idéias?

C: Depende do que é. Algumas coisas eu, o Steve ou o Mark temos uma idéia bastante clara do que a gente quer que seja feito, já em outras situações não, então varia.

S: E quando vocês chegam com essas idéias, é aberto a todos para dar palpites ou é restrito?

C: Todos podem dar palpite, mas a questão é que tem algumas coisas que na verdade são estratégicas ou que o Mark realmente quer que sejam feitas de uma determinada forma, e aí é controverso. Para essas coisas aí é bem mais difícil alguém conseguir dar uma idéia e que essa idéia seja aceita.

S: Então qual o primeiro passo quando surge uma nova idéia, é começar uma especificação, ou um grupo de discussão?

C: Depende muito da funcionalidade. Às vezes pode ser algo que surge a partir de um email e alguém vai no dia seguinte, implementa e integra. Existe esse cenário aí. Em coisas grandes, a gente viaja, senta junto e conversa a respeito. Algumas coisas a gente faz experiências, outras coisas a gente senta e escreve uma especificação completa e discute essa especificação antes de fazer.

S: Uma das últimas questões que eu tenho aqui é sobre a avaliação dessas pessoas que se encontram remotas. Pois quando você está no mesmo escritório é fácil ver se a pessoa está sendo produtiva ou se está tendo problemas apenas sentando do lado dela e conversando. Quando você está remoto como você faz para detectar que a pessoa está bloqueada e não consegue se liberar, e não está comunicando isso a ninguém?

C: Esse é um problema sério, sabia. Esse é “o” problema sério numa equipe de um projeto distribuído e a gente nunca conseguiu resolver isso de uma forma muito boa até esse ano. Até esse ano sempre foi muito difícil isso. A solução que a gente encontrou foi a seguinte: primeiro, é muito importante ter um ritmo periódico de trabalho, sabe. Ter um processo de “release” foi muito importante. Sem o processo de “release” a gente estava perdido, por que a gente batia muito a cabeça, não conseguia decidir, não conseguia nem saber se tinha atrasado ou não. Tendo um processo de “release” você pode dizer: “Vou fazer isso nesse release”, e se não acontecer você sabe que atrasou. Então sem dúvida isso foi muito importante. E a segunda coisa que ajudou muito a gente foi montar grupos de trabalho pequenos, de três pessoas, e o líder do grupo não desenvolve. Ele só desenvolve coisas que não são importantes. As coisas importantes quem faz são os outros engenheiros e o líder do projeto se envolve a baixo nível para saber como as coisas estão indo. Ele liga pra pessoa e conversa: “E aí, quando a gente vai fazer isso? Vamos mudar essa API, vamos consertar isso de tal jeito.”. Então essa foi a mudança, a gente tem

uma pessoa na equipe que é responsável em entender tudo o que está sendo feito, discute em detalhe o que está sendo feito. E dessa forma, por que essa pessoa tem uma comunicação muito mais freqüente com os outros engenheiros você consegue contornar esse problema.

S: Então essa pessoa é o termômetro dentro do time, sentindo quando algo que deveria ser feito não está sendo feito e cutucar a pessoa.

C: Exato. Tem duas maneiras de monitorar, o jeito macro onde você olha o “release” inteiro. Se o seu release é mensal, chegou na segunda semana e você não fez mais da metade das tarefas, você está mal. Então macro você consegue que as coisas não estão indo bem dessa forma aí. E no micro, o cara comunica direto com a equipe dele então ele sabe se o cara está com dificuldades, se está travado ou não.

S: Você citou “chegar na metade de um projeto e ver se a metade das tarefas foram cumpridas”. Como você quantifica isso? Você tem as tarefas registradas, mas você avalia o esforço relativo de cada tarefa ou simplesmente o número de tarefas?

C: É muito difícil. Eu basicamente confio que o líder da equipe individual vai saber dizer se as tarefas são grandes ou pequenas, se as tarefas são factíveis naquele tempo ou não. Na verdade a gente faz uma análise mais qualitativa do que quantitativa, analisa tarefa por tarefa para estimar quanto tempo vai levar.

S: E vocês avaliam a qualidade final do trabalho, em número de testes, performance? Como é feita essa avaliação? E se a pessoa cumpriu as tarefas mas não escreveu nenhum teste por exemplo?

C: Sem teste não entra, não tem jeito. Ela é subjetivamente avaliada. Como parte do processo a gente tem revisão de código. Se não tem testes, se a cobertura de testes é ruim ou se o código é inaceitável, não entra, é rejeitado. Mas isso é para garantir que o código está bom. Mas para avaliar se a pessoa está indo bem ou não, isso é algo que só o gerente pode fazer.

S: Então vocês dão prioridade a qualidade do código sobre a finalização no tempo correto?

C: Não, é um “trade off”. A gente decide. As vezes tem situações em que você tem que falar: “Bom, isso aqui não está perfeito, mas está bom o suficiente e tem que entrar agora.”. Tem outras situações em que a mudança é mais significativa então a gente diz: “Se isso não ficar muito bom, vai ficar muito ruim”. Varia. Depende muito do que é a mudança.

S: Uma questão que você não comentou muito foram os conflitos na questão do desenvolvimento distribuído. Quer fazer algum comentário?

C: Certamente é muito mais fácil se desentender de alguém à distância do que localmente. Um desentendimento à distância é normalmente uma falha de comunicação. Um desentendimento local, normalmente é uma incompatibilidade séria. É muito complicado. Em vários casos a gente teve desentendimentos sérios. E aí

a questão é os gerentes se envolverem, conversarem... é muito complicado. Na maior parte das vezes se as pessoas se desentendem seriamente, mudam as equipes. Demora para chegar nesse ponto, mas normalmente é a solução correta.

S: E como essas falhas de comunicação surgem? Você consegue apontar se foram problemas no meio de comunicação utilizado, falta de comunicação?

C: Varia muito. Tem situações por exemplo, em que uma pessoa não foi tão legal com a outra no telefone, não foi tão gentil. Tem situações em que uma pessoa dependia de alguém mais e essa pessoa não apareceu ou não fez o que tinha que ser feito. Não tem um caso clássico que eu possa ilustrar.

S: Mais algum comentário relacionado aos assuntos discutidos?

C: A gente encoraja muito o compartilhamento do conhecimento entre a equipe. A gente pede que as pessoas escrevam email, postem “bugs”, discutam os problemas encontrados, que não se mantenham restritos a um grupo específico. Quando eu não falo para o Steve que algo importante aconteceu ele fica muito p. Existe uma cultura de que a gente deve difundir a situação e o conhecimento entre a equipe. Existe muita abertura para as pessoas fazerem o que elas quiserem. Existem poucas coisas sobre as quais a gente se preocupa muito com como é feito. Mas em geral existe bastante abertura, e isso é algo que para mim realmente é a chave da eficiência, deixar que o desenvolvedor possa decidir o máximo possível.

Conflitos que eu já observei são: depender de alguém mais e essa pessoa não atender isso é um problema que em geral gera conflito; falta de gentileza entre as pessoas ou simplesmente dificuldade de comunicar para a pessoa o que você está querendo dizer exatamente, essas três coisas geram frustração.

Sobre a questão da remuneração, a gente não divulga. A questão é que sendo uma equipe distribuída, cada país é diferente nesse aspecto. Se você ganha X no Brasil, vai morar na Inglaterra o valor é totalmente irreal.

S: O Gustavo (Niemeyer) está na equipe de vocês?

C: Não.

S: Ele comentou comigo que na opinião dele a revisão mútua de código foi um dos fatores que diminuiu muito os conflitos, por que você sabe que o seu código também vai ser revisado pela pessoa cujo código você está revisando. Você tem algum comentário sobre isso?

C: Todo o nosso código é revisado, desde que eu entrei na Canonical. Foi a primeira coisa que eu fiz.

S: Mas está todo mundo no mesmo nível no que diz respeito à revisão de código?

C: No Launchpad não está todo mundo no mesmo nível. A maior parte das pessoas são revisoras, mas algumas não. Mas para mim isso é um problema e a gente está tentando mudar esse ano, para que todas as pessoas passem a revisar código. A gente não tinha isso antigamente, mas é uma mudança que a gente tá

tentando fazer.

S: Eu havia entendido que no caso do Gustavo não há hierarquia, não são só os experientes que revisam mas todos.

C: No nosso caso não é bem isso, os revisores foram escolhidos primeiro porque eram pessoas que tinham um bom conhecimento então parte disso de fato existe, em parte você tem razão, mas por outro lado os revisores foram escolhidos porque eram gente que queria fazer revisão. Muita gente fala: “Revisão vai ser muito trabalho, vou ter que revisar muita coisa que não tem nada a ver com o meu trabalho.”. Entende o que eu quero dizer? A pessoa realmente tem que estar interessada em investir tempo e revisar código que não tem nada a ver com o que ela tem que fazer. No fundo revisar código para os outros é fazer um favor para a pessoa. Porque aquela revisão não vai melhorar a sua vida necessariamente em alguma coisa. Do ponto de vista do indivíduo faz pouca diferença mas do ponto de vista da equipe faz muita diferença.

S: Outro argumento do Gustavo foi de que a pessoa que revisa acaba aprendendo mais sobre o código em si.

C: Isso é totalmente verdade. Mas no final do dia, revisar o código dos outros não contribui para você terminar as tarefas que você tem que fazer. Se você tem três bugs pra consertar, e passa o dia revisando código dos outros... entendeu?

S: Se quiser fazer mais algum comentário...

C: A gente não tem nenhuma métrica concreta que a gente usa para avaliar a contribuição de um indivíduo. Eu sou meio contra métrica de forma geral. Acho que a avaliação deve ser de forma informal. Acho que se deve usar métricas para orientar, mas a avaliação tem que ser subjetiva, porque não tem métrica que possa te ajudar. Por exemplo, quantidade de testes. Se o cara escreveu só um teste, mas era o teste que fez a diferença, que testou a parte que era crítica. Digamos que o código que o cara ta fazendo não dá para testar muito bem. Digamos que o código tenha que se comunicar com um sistema remoto e ele tenha que melhorar a confiabilidade do código quando o serviço remoto não está funcionando de forma confiável. Como você testa isso direito? É muito mais difícil.

S: Vocês tem algum tipo de avaliação tipo Avaliação 360⁴ que hoje em dia se utiliza bastante?

C: Sim, a gente faz 360 a cada seis meses.

S: E o que mais?

⁴Uma Avaliação 360 possibilita ao indivíduo avaliado um retorno sobre como seu comportamento relacionado ao trabalho é visto por outros colegas. É usado primariamente com o objetivo de desenvolvimento pessoal, mas pode ser usado para outros objetivos. Avaliações são coletadas dos colegas de trabalho, que são convidados a preencher um questionário anônimo, comumente através de email ou pela Internet. A informação é combinada em gráficos e tabelas e entregue ao indivíduo avaliado por um gerente treinado que irá ajudá-lo a desenvolver um plano de ação baseado no resultado da avaliação.

C: Eu não concordo que utilizar email ao invés de teleconferência reduza as falhas de comunicação. Eu acho que são ferramentas diferentes usadas para finalidades diferentes. O fato de que o telefone não é registrado é um grande problema, concordo totalmente. Mas por outro lado é a forma mais rápida de se discutir algo que você já tem um a idéia de como resolver.

Sobre fuso-horário e horários de reuniões, eu acho que a gente tende a escolher um horário que seja bom ou adiar a resolução do problema. Raramente a gente decide que a teleconferência tenha que ser em algum horário específico senão não será possível resolver o problema. Você sempre tenta encontrar um horário bom. A gente prefere atrasar a fazer com que a pessoa tenha que usar um horário que para ela não vai ser razoável.

Raramente eu acho que as diferenças de fuso-horário contribuam para a performance da equipe. É muito melhor ter uma equipe distribuída em fusos-horários similares. Eu não acredito de forma alguma em “cobertura 24h”. Acho que o custo é muito maior que o benefício.

Eu estive envolvido na escolha e formação da equipe. Entrevistei quase todo mundo que trabalha hoje no projeto. Houveram mudanças na equipe desde então, mas na maior parte contratações. Tiveram pouquíssimas demissões. Não sei de forma geral como isso impactou os outros membros da equipe. As pessoas que foram embora estavam realmente infelizes no trabalho. Então de certa forma, alivia o engenheiro se você tira alguém que está desmotivado. Por outro lado se você tira alguém que faz uma e sobrecarrega outros engenheiros, cria frustração também.

S: Nessa questão sobre a formação da equipe, minha pergunta era sobre o perfil desse indivíduo. Se para trabalhar numa equipe geograficamente distribuída você procura uma pessoa que seja um bom comunicador, que seja autodidata, autônoma, ou simplesmente possuir experiência em uma área específica é suficiente.

C: No começo a gente contratava simplesmente baseado na experiência. Basicamente procurando pessoas que tinha competências técnicas. Mas nos últimos dois anos a gente se esforçou em procurar pessoas que fossem trabalhar bem de forma distribuída, porque faz muita diferença.

S: Em relação a estruturas hierárquicas, um artigo aponta que a maioria dos projetos de sucesso (em Software Livre) na verdade possuem estruturas fortemente hierárquicas, ao contrário do senso comum de que os projetos de Software Livre são caóticos e não possuem estrutura hierárquica. Como isso afeta o Launchpad, como vocês se organizam?

C: Existe uma hierarquia sim, a gente introduziu esse ano. A gente não tinha isso, era bem horizontal, hoje em dia a gente tem essas equipes de trabalho que são geralmente três pessoas, acima das equipes tem o Steve e eu, e acima de nós é o Mark. Olha, eu não sei dizer o que leva ao que. Se o fato de ter uma hierarquia leva

o projeto a ter sucesso, ou se o fato de ter sucesso faz com que uma hierarquia seja necessária para gerenciar o número de contribuições e o risco dessas contribuições. A minha impressão é a seguinte, sem hierarquia para a gente, a comunicação tem que ser mais intensa. É muito difícil não ter hierarquia, essa é que é a verdade. Trabalhar sem uma hierarquia é muito complicado no nosso caso específico. Eu não consigo ver tudo que está sendo feito por todos os engenheiros. Para uma equipe grande o suficiente você tem que ter uma hierarquia, não tem como não ter, senão os gerentes ficam muito estressados. Então a impressão que eu tenho ao falar isso pra você é de que é mais provável que a hierarquia apareça porque ela é necessária porque sem hierarquia o volume de contribuições e a qualidade do projeto sofrem, mas não acho que a hierarquia em si define o sucesso do projeto.

S: Outro artigo que eu li indica que a hierarquia influencia a criatividade e a produtividade do projeto, que se a estrutura é fortemente hierárquica o fluxo de idéias criativas fica muito centralizado. Acho que no caso de vocês isso não é um problema porque as idéias surgem de um grupo pequeno de pessoas no topo da cadeia hierárquica.

C: Acho que o que cara quis dizer nesse artigo é que se existe hierarquia, quem está no topo decide se uma idéia é boa ou não. Já se a estrutura é horizontal, é mais provável que sua idéia seja aceita sem resistência. Se você tem uma hierarquia, existe uma forma mais clara de negar algo que está sendo feito, de dizer que aquilo não é uma boa idéia.

S: Então a hierarquia restringe o número de contribuições por criar um certo processo necessário para a aceitação de uma idéia.

C: Exatamente. E paradoxalmente, eu acho que o “throughput” é melhor quando existe uma hierarquia, por que os líderes ou gerentes tem mais tempo de olhar as contribuições individuais. Se o Linus (Torvalds) tivesse que olhar todos os patches individuais do kernel do (GNU/)Linux, ele estaria perdido. Mas como tem pessoas que se interessam pelos subsistemas, essas pessoas tem tempo para conversar com os contribuidores para tentar discutir, para tentar conseguir alguma melhoria.

APÊNDICE C ENTREVISTA: PAUL EVERITT

S: Estamos entrevistando Paul Everitt hoje, sobre o assunto “Gerenciando Equipes Globalmente Distribuídas”. Paul, fale um pouco sobre o seu histórico, especialmente sobre os diferentes projetos que você gerenciou, especialmente aqueles onde parte ou toda a equipe envolvida no projeto se encontravam em geograficamente em locais diferentes.

P: Certo. Voltando atrás um tempão, antes de me envolver em consultoria com Internet, eu era um oficial na Marinha dos EUA, e isso me ensinou bastante sobre liderança e gerenciamento de recursos e coisas desse tipo. Aprendi muito sobre coisas que não conhecia. Então entrei no ramo de consultoria em Internet, abrindo a Digital Creations, que acabou por criar o Servidor de Aplicações hoje conhecido como Zope.

Após isso, o Zope se tornou um projeto Open Source, e eu aprendi sobre como o Open Source e as Comunidades Open Source em geral abordam o desenvolvimento distribuído através de técnicas não-coercivas, você não pode despedir alguém. Essa foi uma experiência de aprendizado muito valiosa. Tomei essas experiências e apliquei-as a três projetos de consultoria em que eu trabalhei, acredito que em todos os três você estava envolvido.

Um deles foi um projeto, quando ainda estava morando na Europa, com a Oxfam, uma entidade de caridade na Inglaterra. Nesse projeto, cinco dos sub-contratados estavam em cinco diferentes países, eu estava em um sexto país, e o cliente em um sétimo país. Foi mais do que certamente muito e muito cedo. E nós tivemos muitos erros no gerenciamento desse projeto, aprendemos muito sobre o que fazer e o que não fazer, e eu mantinha uma lista uma lista de “coisas a fazer diferente da próxima vez”, e quando essa lista passou de 40 itens então eu parei de contar.

No último ano e meio eu tive a chance de trabalhar com a Enfold (Systems, Inc.) em dois projetos, o segundo dos três (clientes) foi o Open Society Institute. Nesse projeto uma pessoa estava na Ucrânia, uma em Londres, duas no Brasil, duas em Houston, um na Louisiana, eu estava na Virgínia, e o cliente em New York. O custo do projeto era alto, as expectativas eram altas, a complexidade era alta.

Mas nós nos demos razoavelmente bem. Foi muito sofrido na primeiro terço do projeto, recuperamos no terço do meio, e no último terço realmente nos sentimos que o projeto era uma máquina bem lubrificada que podia funcionar por si só.

Tomei mais essas lições e as levei para o terceiro e atual projeto que eu estou gerenciando nesse momento, outro projeto com a Enfold. Com não mais tantos indivíduos em locais diferentes, mas dessa vez o cliente está organizado em vários locais diferentes.

Tudo isso são passos incrementais. Você espera dar cinco passos para a frente, você pode dar dois passos para trás, mas se você escrever as lições que você aprendeu e aplicá-las no próximo projeto fosse estará tendo progresso constante.

S: Voltando à sua época na Zope Corporation (ex-Digital Creations), lá vocês tinham um escritório remoto não é? Vocês não estavam todos no mesmo escritório. Você também gerenciava essas pessoas nesses escritórios remotos?

P: Realmente, nós tínhamos dois escritórios remotos: um em New York e outro no Oregon. E na realidade não estávamos prontos para isso. Um dos escritórios tinha uma cultura muito distinta da nossa, com seis pessoas se não me engano, mas nós nunca absorvemos esse pessoal completamente. Nós éramos muito inexperientes para entender como ter uma operação remota com sucesso. O outro escritório realisticamente era semelhante a apenas três desenvolvedores Open Source que poderiam muito bem estar trabalhando de suas próprias casas. E ironicamente, nós absorvemos esses melhor do que os desenvolvedores mais tradicionais que estavam no outro escritório.

S: Talvez isso se devesse ao fato de que essas três pessoas estavam realmente trabalhando como se fossem desenvolvedores Open Source, não é?

P: É verdade. E eles estavam envolvidos com a comunidade, já estavam trabalhando com o nosso software. Então eles já conheciam a cultura, o software e as ferramentas. Mas mais importante do que isso, eles já conheciam a cultura de como trabalhar com pessoas que não estavam na mesa ao lado deles.

S: Nesses projetos que você participou, as pessoas envolvidas em geral trabalhavam como um time, com todos perseguindo o mesmo objetivo, ou como um grupo de trabalho, onde cada um tinha um objetivo diferente?

P: Eu diria que muito mais como um time, pois havia um planejamento conjunto, visibilidade nas coisas dos outros envolvidos, e dependências nos outros envolvidos, a habilidade de mover tarefas de uma pessoa para outra. Dito isso, essa na verdade é a resposta científica, mas a resposta artística é: essas pessoas não estavam no mesmo local, então nos sentíamos como um grupo de trabalho, mas na prática era um time.

S: Nesses projetos, em geral, você esteve envolvido na escolha dos indivíduos que formaram o time, ou esses eram simplesmente as pessoas que estava disponíveis

então você simplesmente teve que formar um time com o que tinha à mão?

P: No primeiro projeto eu tinha controle dos recursos, mas dentro de um grupo de pessoas que eu não conhecia muito bem.

S: Então você realmente não tinha como escolher os indivíduos mais acostumados a trabalhar num ambiente distribuído, certo?

P: Isso mesmo, eu podia escolher baseado na experiência deles, mas não no seu, se puder usar essa palavra, “profissionalismo”, sua habilidade de trabalhar bem num contexto de consultoria. Eu irei usar uma expressão comum nos Estados Unidos: “Eu estava voando às cegas”.

S: Então se você tivesse a oportunidade de escolher essas pessoas baseado nas suas habilidades, quais as habilidades que você consideraria mais importantes para alguém que vai trabalhar numa equipe distribuída?

P: Absolutamente a habilidade número um seria a confiança. É uma lição que eu aprendo e esqueço, re-aprendo e re-esqueço, num ciclo infundável. Eu vim aprendendo essa lição desde os tempos da Zope Corporation. A qualquer momento eu trocava um programador super-gênio que é um consultor medíocre por um programador medíocre que fosse um consultor genial. Quando você está num ambiente distribuído, você precisa acreditar que a pessoa vai realmente fazer o que ela disse que vai fazer. Eu sou bastante flexível e eu escuto tão bem quanto possível as pessoas mas eu tenho uma regra imutável nos meus projetos: se você me disser que vai terminar algo até amanhã de meio-dia, não venha me dizer às onze e cinquenta e nove que você teve um problema. Tudo bem em dizer que algo vai demorar mais do que eu esperada, mas se eu sou surpreendido após o fato e você nem mesmo me disse que não conseguiu terminar, isso é algo que eu não posso tolerar.

S: Então você diria que a confiabilidade, que a comunicação é um fator que torna um fator que torna uma pessoa mais confiável, se ela comunicar tão cedo quanto possível que ela irá se atrasar?

P: Sim, mas isso é o resultado de ter um bom caráter. É algo que você simplesmente não pode ensinar. Mas há algumas pessoas que sabem: “Eu tenho que agir profissionalmente, eu tenho que ser responsável pelo que eu disse que eu iria fazer.”. E parte de ser responsável por isso é deixar alguém saber quando você irá se atrasar e não criar surpresas.

S: Alguns dos artigos que eu pesquisei apontam que a confiança entre os desenvolvedores é um fator importante relacionado à performance de uma equipe. Poderíamos dizer que confiança e confiabilidade são aspectos bastante semelhantes.

P: Exatamente o que eu iria dizer.

S: Mas não apenas a gerência, mas também os parceiros numa mesma equipe precisam ter uma relação de confiança.

P: Completamente verdade. Infelizmente organizações grandes e projetos

grandes são otimizados para o elo mais fraco. Elas criam toda uma sobrecarga, cheia de métricas e gerenciamento para procurar o indivíduo que tem a pior performance. É muito frustrante para mim se eu sou uma pessoa confiável e outra pessoa no projeto está sempre em todas as reuniões dizendo: “Ah, esqueci de falar que eu vou me atrasar. Encontrei alguns problemas. O cachorro comeu meu tema de casa.”. Então as pessoas talentosas vão começar a ficar desgostosas e a moral vai baixar. É a síndrome da janela quebrada.

S: E isso afeta a performance do projeto, certo?

P: Completamente. Pois as pessoas começam a pensar que realmente as coisas não importam.

S: Nesses projetos, acredito que você algumas vezes se deparou com pessoas em fuso-horários bastante diferentes. Isso foi algo que você considerou uma vantagem, devido ao fato que alguém poderia estar trabalhando enquanto os outros dormiam, ou uma desvantagem, devido ao fato que alguém pode ficar bloqueado esperando pelo resultado de alguém que já terminou seu dia de trabalho.

P: Isso é o que realmente me faz gostar desse campo. É tudo uma arte. Cada situação pode ter um resultado bom ou um resultado ruim. A forma que você aborda a situação define o resultado você terá. Você e eu viemos de um histórico de Open Source. Muitas das questões que são relevantes a essa questão são intrínsecas a nós. Nós entendemos a comunicação assíncrona. É como perguntar a um peixe para descrever a água. Um peixe não pode descrever a água, é algo natural para ele, faz parte do ambiente onde ele vive. Um gerente de projeto ficaria horrorizado com a cultura Open Source porque simplesmente não há controle, não é tão mensurável, coisas desse tipo. Mas se você apenas aceitar que você está nadando na água e não pensa sobre a água, você pode aproveitar a oportunidade de ter alguém na Ucrânia ao invés de ser espremido pelas dificuldades, e você menciona algumas das oportunidades que se as tarefas forem gerenciadas corretamente essa pessoa (na Ucrânia) pode começar antes do resto do time que só irá acordar quase 10 horas mais tarde. Você ainda precisa ter alguém que irá gerenciar a ligação síncrona, como um gerente de projeto que irá acordar às seis da manhã se necessário e começar a conversar com essa pessoa, e essa pessoa tem que estar disposta a estar trabalhando às seis da tarde.

S: Você mencionou que nesse projeto havia uma pessoa na Ucrânia. Como foi a comunicação com essa pessoa? Você encontrou alguma barreira de linguagem ou cultural que impactou a comunicação com essa pessoa?

P: É similar à questão do peixe e da água. Eu entendo perfeitamente de onde vem essa pergunta, e eu sei que vários gerentes de projetos clássicos cairiam nesse problema. Mas pessoas que vem da comunidade Open Source sabem que se eles não falarem inglês muito bem elas simplesmente não conseguirão se integrar à comu-

nidade Open Source. Quando eu encontro uma pessoa na Ucrânia, essa pessoa só foi encontrada porque ela tem um inglês funcional. Ao mesmo tempo, eu como consumidor do trabalho dessa pessoa, entendo que você tem que aceitar um pouco de bagunça na comunicação e nem sempre levar tudo ao pé da letra e ter cuidado com as coisas que você diz. Você não pode fazer uma piada sobre o episódio de “Seinfeld” da noite passada porque eles muito provavelmente não tem isso lá. Então se você tem interesse de ter essas duas partes co-existindo num local comum da comunidade Open Source onde se fala predominantemente inglês mas não completamente, então as barreiras de linguagem e cultura não serão tão importantes. Sempre haverão as exceções, especialmente quando uma crise acontece e quando as intenções estão escondidas, e as pessoas sempre tem que ter respeito suficiente pelas outras de forma que se elas receberem um email que parecer insultante que elas devem entender que é um problema de linguagem ou cultural e dar à pessoa o benefício da dúvida.

S: Nesses projetos, que formas de comunicação, que mecanismos de comunicação você utilizou para se comunicar com os membros do projeto? Principalmente texto, ou voz? Em que casos você preferiria usar um ou o outro?

P: Definitivamente voz, mas em diferentes circunstâncias co-oferecer diferentes modos de comunicação. Atualmente se tornou natural para mim saber, e para a outra pessoa também, qual dos diferentes modos usar para determinadas tarefas e contextos. Todos nós parecemos saber bem quando usar mensagens de texto, chamadas via Skype (VoIP) ou outros meios. Parte disso é saber quando a comunicação precisa ser síncrona ou assíncrona.

S: Nessa manhã eu entrevistei o Kiko (Christian Reis), acho que você conhece ele. Ele tem uma empresa baseada em Open Source no Brasil, e também trabalha para a Canonical, e ele mencionou que por exemplo a comunicação baseada em voz não é muito efetiva quando a pessoa tem que pensar a respeito da resposta, porque é necessário haver um tempo entre o momento que você recebe a pergunta e o tempo que você formula uma resposta.

P: Deliberação.

S: E nesses casos é mais recomendável usar algum tipo de comunicação assíncrona, baseada em texto.

P: Completamente, e isso faz parte da cultura. Em uma conversa telefônica, aqueles que cuja linguagem nativa é o inglês irão dominar. Eu e você conhecemos algumas pessoas, e eu sou uma delas, que simplesmente adoram dominar uma conversa. É exatamente o mesmo em uma mídia baseada em texto favorece os “geeks” ao invés dos humanos normais, porque os “geeks” podem escrever muito rápido. Onde pessoas que estão uma geração adiante simplesmente não podem acompanhar.

Outro ponto porém, e essa é uma lição que eu continuo a aprender e é muito importante, há alguns poucos diferentes fatores que determinam qual o tipo de comu-

nicação a ser usado. Um deles é “síncrono/assíncrono”, outro é “grupo grande/um a um” e outro que você citou é a questão da linguagem. Mas outro ainda que eu tenho que mencionar que os clientes não necessariamente apreciam o valor disso, é você tem que saber se você precisa que a comunicação seja registrada e parte da história permanente do projeto ou se ela é transitória. Pois se você manda um email para alguém e essa pessoa responde, quase todo mundo salva as mensagens enviadas e você sempre pode voltar atrás e encontrar aquela conversa após se você precisar.

Eu vejo que infelizmente muitos dos consultores clássicos adoram ligações telefônicas e são terríveis ao manter notas das discussões, e na minha opinião esse tempo é total e completamente desperdiçado. Ninguém irá lembrar o que aconteceu naquela ligação. Nenhuma ação é gerada a partir daquela ligação, a menos que você seja bom ao tomar notas.

Mesmo emails, a maioria de nós recebe milhares de emails todos os dias, mesmo emails não são muito bons nesses dias. Nem sempre é claro qual é o resultado de uma interação por email. Uma conversa por email pode simplesmente parar. Eventualmente ela para. E a menos que as pessoas sejam disciplinadas em suas conversações por emails e sempre terminem suas discussões com resumos que digam: “Tivemos questão X, e a resposta foi Y, e a ação a ser tomada é Paul vai fazer Z.”. Poucas pessoas fazem isso, dessa forma ferramentas de gerência de problemas e semelhantes, especialmente se elas forem levem suficientemente para competir com email são ótimas para gerenciar a comunicação.

S: Muito interessante.

P: Eu aprendi muito a respeito disso no primeiro projeto, e no segundo projeto se tornou muito melhor durante a duração do projeto, e quando eu disse que o projeto praticamente se auto-gerenciava é porque as pessoas adotaram uma mentalidade voltada ao gerenciamento de problemas então elas sabiam que quando elas diziam algo que tinha alguma ação associada, elas deveriam dizer isso através da ferramenta de gerência de problemas ou não dizer. Pois é pior dizer algo, é melhor não dizer nada do que dizer algo que precisa de uma ação numa chamada telefônica e não registrar.

S: Isso é algo que o Kiko falou também, que o email é melhor quando o assunto é pequeno e auto-contido e uma chamada telefônica é melhor quando há inúmeros problemas a serem resolvidos em um curto espaço de tempo. Você concorda nisso?

P: Sim. Pois há um pouco de ambigüidade com a gerência de problemas. É mais ou menos como a diferença entre navegar e fazer uma busca na Internet. A navegação é quando você sabe o que está procurando mas não sabe onde está, e a busca é quando você não sabe exatamente o que está procurando.

S: Acredito que as pessoas precisam ser cuidadosas também para dividir tarefas em pequenas ações. Pois se você registrar uma tarefa que necessita de dez ações

diferentes como um único problema a ser resolvido algumas dessas ações serão esquecidas.

P: É verdade, o mesmo ocorre quando há uma descrição muito vaga com algo como duas semanas de trabalho associadas com ela. É quase garantido que isso será um problema mais tarde.

S: Você teve muitos conflitos nos projetos nos quais tomou parte? Eles foram relacionados à falhas de comunicação, ou talvez falhas de especificação? Que tipos de problemas você teve?

P: Engraçado, esses são os dois problemas que eu iria citar. Eu adicionaria a esses dois o problema da confiabilidade. Quase todos os problemas com pessoas que não são responsáveis e causam problemas, são os que em geral causam conflitos e tomam a maior parte do tempo do gerente de projeto, e você tem que trabalhar de forma altamente focada, por exemplo: “Nos próximos sete minutos você irá fazer isso, e me dizer qual foi o resultados, e nos outros sete minutos você fará aquilo” e isso é terrível. E isso não tem muita relação com projetos distribuídos ou não, é simplesmente da natureza de algumas pessoas.

Mas há outros tipos de conflitos, como por exemplo a respeito da especificação. Quando terminamos um dos projetos que mencionei, tivemos um comentário irônico de alguém do lado do cliente. E essa pessoa disse: “Paul nos deu o que nós pedimos, agora nós precisamos descobrir se era isso que queríamos mesmo”.

S: E isso foi após a finalização do projeto?

P: Exato. Então isso é basicamente como a democracia e a sociedade. Há um ditado, se não me engando do Churchill: “A democracia é a melhor das piores escolhas para formas de governo”. É uma terrível forma de se governar, mas é melhor do que as alternativas. E assim, tudo termina sendo uma bagunça, e o conflito é gerado no meio disso. E coisas como o “Agile” tentam lidar com isso. “Agile”, no seu cerne se não me engano, diz que o mundo é bagunçado mas não vamos tentar prevenir a bagunça, só iremos tentar limpar frequentemente. Então se você adota um modo de pensar como esse e tem paciência e quebra as coisas em pequenos passos de forma que quando o conflito aparece o tamanho do conflito é minimizado.

S: Você diria que projetos distribuídos tem mais tendência a gerar conflitos ou menos tendência?

P: Provavelmente... vou dar uma não-resposta: eu diria que menos. Eu diria que se a equipe for deficiente em sua composição ele irá falhar antes mesmo de chegar a um conflito. Se você tem uma equipe distribuída que está trabalhando efetivamente pois foi montada a partir dos tipos certos de pessoas, então provavelmente... terá menos tendência a gerar conflito. Provavelmente a melhor resposta é que as tendências são iguais, mas os tipos de conflitos são diferentes. Você tem questões políticas dentro de um escritório, e isso é um tipo de conflito que você não tem numa

equipe distribuída. Você tem várias técnicas coercivas em uma empresa tradicional, do tipo: “Faça isso ou irei lhe demitir”. Ou: “Você quer o seu bônus no fim do ano?”. As formas tradicionais de gerenciamento para forçar, coagir, as pessoas a resultados positivos, você não pode fazer esse tipo de coisas num ambiente distribuído. Muitas das relações simplesmente não estão presentes. As vezes não há um vínculo empregatício, apenas um “freelance”, então as pessoas apenas tem que manter o respeito mútuo para manter as coisas funcionando.

S: Falando um pouco sobre a performance, como você avalia alguém que está trabalhando remotamente. Se essa pessoa está tendo uma boa performance, ou se ela está travada. Que tipo de métrica você se utiliza para medir o trabalho dessa pessoa, por exemplo número de tarefas ou testes? Que mecanismos você utiliza para analisar a performance de alguém que está trabalhando remotamente?

P: Eu irei confessar que enquanto eu tenho muito interesse sobre esse tópico, eu tenho pouca experiência. E ao mesmo tempo, estou bastante convencido que mesmo pessoas que dizem saber como medir a efetividade pessoal, elas realmente não sabem. Elas inventam sistemas que podem ser facilmente subvertidos como por exemplo: algo que aprendi recentemente foi como “gerenciar a pilha”. É um conceito de se originou do “Agile”. Não há razão para se falar sobre o que será feito daqui a três meses. Faz mais sentido ir ao gerenciador de problemas e dizer: “Para esse horário amanhã, essa é a sua única tarefa de alta prioridade. Eu a tornarei altamente mensurável. Você não pode chegar para mim e dizer que três outras pessoas pediram para você fazer uma outra tarefa. Se isso acontecer, você virá e me dirá que terá um atraso. Se eu não ouvir de você, essa única tarefa irá ser finalizada nesse horário amanhã.”. Se você consegue reduzir o radar dessas pessoas a um horizonte sem distrações será bem mais fácil mensurar a velocidade delas. E conforme você calibra e descobre qual a velocidade delas pode ser, você pode ajustar as estimativas das tarefas e dizer: “Ok, você pode facilmente concluir essas duas tarefas até amanhã.”. Mas manter um horizonte sem distrações e um radar pequeno onde você realmente pode mensurar a velocidade dessas pessoas é a melhor esperança.

S: Há muitas dificuldades em se detectar que um projeto está atrasado. E o que você acabou de dizer para mim parece que é ter um processo iterativo e incremental de estimativa e ajuste. É dessa forma que você enxerga isso?

P: Exato. Eu era um grande proponente de projetos com custo fixo e isso simplesmente gera muitos conflitos entre o vendedor e o cliente, pois o cliente muda a sua idéia a respeito do projeto e há um conflito. Você tem que re-negociar o preço do projeto, e o vendedor tem que constantemente dizer: “Não, eu não irei fazer isso que você está me pedindo agora porque o preço já foi definido.”. Da mesma forma acontece entre o gerente de projeto e alguns dos recursos do projeto. Se você tentar dizer: “Esse pilha moderadamente especificada de coisas irá ser feita em 7.13

semanas-equipe” você irá apenas gerar conflito, pois muitas vezes o cliente não sabe muito bem o que quer. O desenvolvedor não sabe muito bem onde os 10% dos grandes problemas estão, e o gerente de projeto apenas está tentando colocar você numa estimativa razoável e deixar o especialista tomar conta.

S: Fale um pouco sobre o compartilhamento do conhecimento. Você disse que tenta manter os desenvolvedores livres de distrações, certo? Você encoraja eles a serem pró-ativos e buscar ajuda dentro da equipe, buscar novas idéias e sugestões sobre como resolver um problema, ou você espera que a equipe restrinja a comunicação pois ela pode torná-los mais lentos?

P: Esse é um ponto importante, pois a velocidade de uma pessoa pode ser grandemente impactada por reuniões, e isso é terrível, mas também por ajudar outra pessoa. Você está aumentando a velocidade da outra pessoa mas diminuindo a sua própria. As pessoas são como animais que respondem a incentivos. Se você tem incentivos negativos a respeito de ajudar outros, eles não irão fazê-lo. Se você focar muito em: “Em três semanas você tem que terminar isso, ou você está despedido” então elas farão o melhor para sua performance em detrimento da equipe como um todo. Então tendo um horizonte de tarefa de um dia para terminar uma tarefa sem nenhuma distração funciona muito bem, pois se você passa por um período onde as pessoas precisam falar sobre a arquitetura, você pode incluir isso. Você pode dizer, eu sei que vocês estão fazendo isso agora, ou, aí vai a maior: Sidnei passa uma tarefa para Paul e Paul precisa de três quartos de um dia para terminar a tarefa, Paul pode chegar no fim do dia e dizer: “Ei, eu passei meu tempo ajudando o Bob por que ele precisava de ajuda”, e o gerente de projeto pode dizer, “Ok, eu posso te dar mais um dia”.

S: Você já se envolveu em algum projeto onde os indivíduos envolvidos foram consultados durante a especificação do projeto ou discussões arquiteturais. Como você gerencia a criatividade distribuída?

P: Eu ainda estou refinando esse processo. Eu acho que eu errei algumas vezes, acho que vou acertar mais cedo ou mais tarde. Nos projetos em que participei isso foi muito único, pois só trabalhei em projetos em que sou bom. Pois os projetos me encontram, não sou eu que encontro os projetos. E os projetos que me encontram, me encontram pois estão no tipo de assunto que eu sou bom. Isso significa que como um gerente de projetos, e isso é algo que me incomoda no gerenciamento clássico, e isso é algo que provavelmente será bastante familiar para você: organizações odeiam a idéia que gerentes de projetos sejam técnicos. É sempre visto como uma falha fatal quando um gerente de projeto tenta se envolver na tecnologia, o que parece ser algo retardado para mim. Por que isso seria uma coisa ruim? Eu entendo que você não queira que o gerente de projeto tenha muitas tarefas, e esse tipo de coisa. Eles devem estar aumentando a velocidade dos outros ao invés de contribuir diretamente para o

trabalho. Ao mesmo tempo o gerente de projeto ele é aquele que senta na frente do cliente e interpreta o que o cliente quer e precisa em ações. Se eles forem estúpidos e não tiverem nenhum conhecimento sobre a tecnologia eles irão cometer tantos erros! Eles irão criar tarefas que são impossíveis ou proibitivamente caras. Eles não irão direcionar o cliente para o caminho de menor resistência. Eles simplesmente dirão: “O cliente está sempre correto, vamos construir essa nave especial que eles nos pediram.”. Então se você pode ter um gerente de projeto que consegue colocar as coisas numa estimativa bem razoável antes de envolver o time técnico, isso é metade da batalha. A outra parte é, e isso é algo que eu estou aprendendo, digamos que você quebra seu projeto em iterações ou marcos, e digamos que a iteração dura umas três ou quatro semanas. Quando o gerente de projetos acha que está próximo da estimativa, gaste meio dia, junte o time técnico e não faça mais nada além de analisar as estimativas. Que o time técnico diga: “Oh, não, não é assim que iremos implementar isso.”. Isso é uma das sugestões do Alan (Runyan). E eu acho que isso funciona muito bem. Eu não acredito que iria funcionar ter todo o time técnico participando da conversa com o cliente, porque terão muitas vozes e será muito difícil chegar a algo que seja uma ação concreta.

S: Gostaria de falar algo sobre as falhas e problemas que você teve ao gerenciar times remotos?

P: Eu diria que a pior de todas é entrar numa marcha fúnebre e fazer algo que você sabe que é impossível, seja por tempo, complexidade ou custo. Ninguém gosta de ir dar más notícias ao cliente. Você quer ver o cliente feliz não zangado, certo? Mas más notícias são melhor entregues cedo. Elas não envelhecem muito bem. Então numa perspectiva de falhas, essa é uma das coisas que eu estou tentando melhorar. É dizer ao cliente: “Eu sei que você quer isso, mas se disser para nós fazermos isso, você irá ficar louco da vida com a gente depois, pois é muito arriscado ou irá atrair problemas ou terá uma performance terrível.”

Tem outra coisa que eu odeio que é depender de heróis, e criar um plano que pode demorar seis semanas, mas se espremermos bastante a gente consegue fazer em três semanas. Se um projeto adota essa cultura, estará sempre em crise. Tudo bem trabalhar vinte e quatro horas de vez em quando, mas se você fizer isso o tempo todo é terrível.

Aprendendo com as falhas, uma das coisas que eu aprendi foi a falhar cedo ao invés de falhar tarde. Nós dois participamos de um projeto onde a data final nos estávamos em apenas cinquenta por cento, e durante a pausa do Natal, eu e você enquanto os outros estavam de folga, recalibramos o projeto todo então voltamos para o time e dissemos: “Escutem, sabemos que isso é ruim, mas aqui está um plano realístico”.

S: Você acha que é importante que toda a equipe, quando a equipe é distribuída,

que toda a equipe se encontre no início do projeto e tenha uma certa interação social para criar relações de confiança?

P: Isso é uma afirmação maravilhosa. Por que é muito fácil criar um estereótipo a respeito do cliente ser estúpido, ou do cliente estereotipar os desenvolvedores como sendo preguiçosos. É fácil estereotipar alguém com quem você nunca tomou uma cerveja. Se você tem um projeto que nunca entrará em crise, isso não é tão importante. Mas realmente não existe esse tipo de projeto. Então quando as coisas ficam difíceis, ao invés de apontar dedos, você precisa ter uma equipe que quer que os outros tenham sucesso porque eles se conheceram face a face.

APÊNDICE D ENTREVISTA: ALAN RUNYAN

S: Fale um pouco sobre o seu histórico, sobre os diferentes projetos que você gerenciou, especialmente aqueles em que parte ou toda a equipe envolvida nos projetos era geograficamente distribuída.

A: Vou iniciar falando sobre o projeto Plone. No início, havia um projeto Open Source que eu ajudei a iniciar. Quase todo mundo que estava nesse projeto estava em locais geograficamente distintos e os indivíduos estavam em vários fuso-horários diferentes. A única maneira desse projeto funcionar com o mínimo processo na época foi por que todos eram altamente motivados. Não havia nenhum tipo de estrutura centralizada de tarefas. Havia uma pessoa com idéias na cabeça. Tínhamos uma ferramenta de gerencia de problemas simples e listas de discussão, IRC e telefone.

A maior parte do gerenciamento era feita de forma voluntária. Não haviam datas a ser cumpridas, mas algumas funcionalidades que deviam ser implementadas num tempo razoável, e de forma a aproveitar as habilidades desses indivíduos envolvidos no projeto.

Na época, foi uma grande experiência de aprendizado como gerente, mas foi certamente algo que seria quase impossível se fosse apenas um gerente não-técnico, ou alguém que não tivesse as relações com as mais ou menos cinqüenta pessoas ao redor do mundo que eu tinha.

Isso foi mais ou menos o que aconteceu a nível do projeto Plone.

Agora, falando da Enfold Systems, que foi a entidade comercial que surgiu a partir dessa experiência Open Source. Na Enfold Systems, nós temos pessoas que estão geralmente geograficamente distribuídas, pois são pessoas da comunidade Open Source que nós contratamos. Alguns deles estão no Brasil, vários nos Estados Unidos mas muitas delas saíram de Houston para ficar mais perto da família.

A gerência dessas pessoas na verdade é mais desafiante porque tem duas fases: uma é o gerenciamento voltado a datas específicas dos projetos comerciais e a orçamentos, gerenciando expectativas dos clientes, que é bem diferente do lado Open Source. Então vou falar disso, e depois vou falar da criação do processo para ter outras pessoas gerenciando, que eu acho que é bem mais interessante, de um ponto

de vista de negócios.

Esses projetos iniciaram internamente. Tínhamos muito pouco processo, programação estilo “cowboy”. Tínhamos reuniões diárias, listas de discussão e o que fizemos naquele ponto foi tentar segmentar o tipo de tecnologias que estavam sendo usadas. Um processo foi colocado em prática com o objetivo de separar diferentes componentes de forma a que eles fossem desenvolvidos independentemente. As pessoas que estavam no Brasil trabalhariam em um grupo de componentes, e as pessoas em Houston em um outro grupo de componentes, e quem trabalharia na interface do usuário seria o cliente.

Isso funcionou muito bem até um certo ponto. Nós atrasamos algumas datas de entrega, principalmente pois não tínhamos uma clara visibilidade, dando tarefas aos desenvolvedores sem ter um bom senso para motivar os desenvolvedores. Eles não sabiam qual sua medida para sucesso era. Isso era muito difícil para nós gerenciarmos. Isto é, se as pessoas não tem idéia do que considerar como sucesso, eles vão hesitar.

Tivemos um projeto de sucesso. O maior problema que tínhamos era extrair informações da especificação para quebrar os componentes em partes que poderiam ser desenvolvidas independentemente por desenvolvedores distintos e a integração dos diferentes componentes. As partes eram desenvolvidas de forma que independentemente elas funcionavam muito bem, mas quando integradas haviam muitos problemas. E realmente não havia muito processo formal além do que estava na minha cabeça, e eu precisava falar com as pessoas diariamente. E novamente, isso seria muito difícil de replicar sem ter o tipo de relação pessoal e experiência em projetos que eu tinha com esses desenvolvedores, e também a habilidade de manter todas as especificações vindas dos clientes na minha mente e regurgitar essas idéias em algum formato de tarefas diárias ou semanais que eu gerenciaria pessoalmente.

S: Então você diria que nesses projetos haviam muitas pessoas envolvidas, mas elas não estavam cooperando como um time (com todos trabalhando para o mesmo objetivo), mas ao invés disso elas tinha tarefas separadas e cada um era apenas responsável por sua própria tarefa sem ter uma visão geral do escopo do projeto, correto?

A: Bem, eu acho que isso é correto até um certo ponto. Eu acredito que todos sabiam mais ou menos como os componentes iriam se encaixar no final das contas. Se essa visão foi claramente articulada e poderia ser repetida de volta para mim pelos desenvolvedores, isso é questionável. Bom ponto. Não havia confirmação a partir dos desenvolvedores se a visão era completamente entendida por todos.

De forma mais direta. Pessoas estavam trabalhando em tarefas independentes e não tinham um entendimento claro de como tudo se encaixaria em pontos do projeto. E era bem questionável se esses desenvolvedores tinham uma idéia concreta

de como o projeto iria funcionar a nível geral. É uma boa questão, não tenho certeza se eles realmente compreendiam.

S: Quando você seleciona as pessoas para trabalhar em um projeto, que tipo de habilidades você busca? Habilidades de comunicação, pessoas que conseguem se auto-gerenciar e trabalhar de forma mais autônoma. Que tipo de habilidades você busca?

A: As pessoas que a gente procura são pessoas que conseguem se comunicar bem, que são dedicadas ao seu trabalho, pessoas que tem boa comunicação escrita, e pessoas que possam provar que são auto-motivadas. Quando procuramos alguém para trabalhar remotamente, há algumas questões que eu procuro esclarecer, tento garantir que eles possam se auto-agendar ou pelo menos entregar tarefas em dia. Se alguém não consegue se comunicar muito bem, nem entregar tarefas em dia, e essa pessoa está remota, e eu não tenho uma relação com essa pessoa para entender como ela trabalha, em geral não é uma boa pessoa para trabalhar com a gente.

S: Nesses projetos é comum que pessoas entrem e saiam da equipe, ou a equipe se mantém fixa do início ao fim do projeto?

A: Claro, em alguns desses projetos tivemos pessoas saindo e entrando temos poucos atritos, mas isso acontece de vez em quando. Talvez isso aconteça em três de cada dez projetos.

S: Você mencionou que você busca pessoas com quem possa criar uma relação pessoal e entender a forma que eles trabalham. Você poderia descrever isso como uma relação de confiança. Há alguns artigos que apontam as relações de confiança como um fator chave de uma equipe eficiente, especialmente uma equipe distribuída. Você concorda com isso?

A: Eu concordo plenamente com isso, que a confiança é a moeda de troca em uma equipe distribuída. Confiança e mérito são basicamente as únicas coisas que você deve buscar enquanto gerente. Essas coisas são muito importantes, mas especialmente se a equipe é distribuída são as únicas ferramentas que você tem para trabalhar. Alguém pode ter muitas habilidades, com certeza, mas você tem que confiar que eles podem realmente cumprir com o prometido, que eles irão se comunicar em tempo correto. A confiança é a chave essencial sem a qual você provavelmente não faria progresso algum com projetos geograficamente distribuídos.

S: Em projetos geograficamente distribuídos, as diferenças de fuso-horário podem realmente determinar a falha completa de um projeto se houverem muitas dependências entre pessoas em fusos-horários muito diferentes. Você concorda com isso?

A: Eu concordo com isso. De fato eu já vi casos onde o fuso-horário trabalhou tanto a favor e contra um projeto. Eu sinto que inerentemente ter múltiplos fusos-horários é muito difícil. Não seria tão difícil se fossem horários próximos. Mas

horários muito distantes são um problema sério sem ter um processo sofisticado de gerência. Na maior parte dos nossos projetos nós tentamos manter uma visão baseada no “Agile”, ou seja, menos processo e mais indivíduos auto-dirigidos. As pessoas que são tipicamente desenvolvedores Open Source que entendem como desenvolver e são auto-motivados. Isso realmente não funcionaria muito bem para muitos projetos em muitas empresas pois requer certas habilidades de gerência, com gerentes muito técnicos e muito sofisticados. Nós buscamos pessoas que são auto-motivadas, com boas habilidades para comunicação, pessoas que estão em fuso-horários diferentes podem ser um grande problema e muito frequentemente o são. A única vez que isso funcionou bem (fuso-horários diferentes) foi quando as tarefas eram completamente independentes e o gerenciamento era altamente envolvido. Nossa gerência de projetos dentro da Enfold ainda é bastante imatura. Somos bastante centrados no desenvolvedor e isso nos causa problemas quando os desenvolvedores tem pontos de vista diferentes e a gerência não se envolve rapidamente para resolver o assunto. E nesse caso ter fuso-horários muito diferentes realmente pode acabar com um projeto, pois enquanto o gerente está dormindo você tem pessoas sendo improdutivas, ou mesmo não produtivas nas coisas que você necessita para o próximo dia.

S: Então de certa forma as diferenças de fuso-horário podem ser uma fonte de conflito, certo?

A: Certamente, sem os processos corretos.

S: Você mencionou o aspecto da comunicação. Que mecanismos de comunicação diferentes você utiliza em seus projetos. Você usou principalmente comunicação baseada em texto, IRC, IM, ligações telefônicas, VoIP?

A: Nós pedimos para que os membros da equipe tenham reuniões diárias. Em projetos nós temos reuniões Scrum¹ virtuais diárias (via telefone). Todos devem estar disponíveis em pelo menos duas formas de comunicação simultaneamente. Nós usamos o IRC ou IM e Skype. Por termos um grupo grande de pessoas temos uma comunicação a nível de grupo com o IRC e comunicação pessoa a pessoa com o Skype. E às vezes às pessoas ligam para um número para uma teleconferência quando há reuniões diárias em que os desenvolvedores do lado do cliente estão envolvidos. De certa forma é uma combinação dos vários meios. A comunicação é muito custosa, se você não está fisicamente presente e não há “conversa de bebedouro” se torna muito problemático manter as pessoas informadas e motivadas, é um problema bastante difícil de ser resolvido. Listas de discussão e ferramentas de gerência de problemas

¹Scrum é um método de gestão de projetos que está se tornando cada vez mais comum na indústria de software. Pequenos times com um máximo de seis a oito pessoas dividem o seu trabalho em “mini projetos” que tem uma duração de até um mes durante o qual um número limitado de tarefas detalhadas é resolvido. Enquanto métodos tradicionais tem por foco principal manter a linha, o Scrum assim como outros métodos ágeis tem por objetivo o resultado final do projeto

que são fortemente integradas com email são bastante efetivas.

S: Você acha que alguns meios de comunicação são mais efetivos do que os outros? Em que situações você usaria por exemplo comunicação por voz ao invés de comunicação via IRC ou IM?

A: Quando há confrontação entre os desenvolvedores, uma discussão em que há troca de argumentos, a voz tem muito mais largura de banda. A maneira que uma pessoa enuncia, como ela fala como ela projeta suas emoções através da voz, um toque pessoal. Realmente não há como ter uma noção de como as pessoas estão se comunicando quando elas estão utilizando texto. A comunicação é muito estreita. Então tudo que requer uma troca de argumentos, uma significativa largura de banda. Mas o melhor de todos que nós ainda não estamos utilizando seria a videoconferência. Nossa empresa ainda não tem uma plataforma para aproveitar isso. Eu acredito que alguns dos nossos desenvolvedores em algumas discussões mais ferrenhas teriam grande vantagem ao usar videoconferência, ao ver a expressão facial das pessoas envolvidas em resolver problemas difíceis.

S: Algumas das pessoas que entrevistei mencionaram o fato de que a conversa de voz não é muito eficiente quando há questões que necessitam de um processo mais longo de elaboração de uma resposta que a comunicação via voz não é um meio muito eficiente para isso.

A: A comunicação via voz é boa para resolver problemas, para o aspecto mais social, para o lado mais qualitativo. Quando é puramente quantitativo, quando é algo que é muito profundo e não pode ser respondido imediatamente, algo que requer que a pessoa internalize a questão para poder responder ela não funciona muito bem. Pensamentos e questões mais elaboradas deveriam ser transmitidos através de emails ou algum outro tipo de mensagem de texto e mesmo se forem expressados através de voz deveriam ser resumidos em uma forma de texto para que se gere um registro histórico, uma confirmação do que foi falado. Mas eu acredito que a voz é uma boa forma de calibrar e adicionar pontos de dados para chegar a um resultado final. Mas concordo totalmente que quando se requer que a pergunta seja internalizada, algo que requer uma resposta que provoque um pensamento introspectivo, que o texto é mais eficiente. Na minha opinião a voz é muito boa para iniciar e finalizar uma questão, ou para resolver problemas.

S: Outro problema com a comunicação via voz é que as pessoas que não tem o inglês como língua primária podem necessitar de mais tempo para articular uma resposta, e as pessoas que tem o inglês como língua primária tem uma vantagem.

A: Eu concordo com isso. Nós passamos por esse problema no início, quando você não falava inglês muito bem, e muitos dos desenvolvedores na X3ng eram muito tímidos para falar no telefone, preferindo usar texto. O problema é que não importa o quanto você pense, no fundo as pessoas são indivíduos sociais e os

gerentes e tomadores de decisão irão sempre preferir falar com um humano, e muitas empresas de consultorias grandes tem estruturas que lidam com essa sobrecarga de comunicação dos indivíduos. Mas isso não é algo que pode ser feito facilmente numa empresa de consultoria pequena, que tem faturamento abaixo de dois a cinco milhões de dólares por ano. Somente empresas de maior porte podem ter uma estrutura que suporte tradução e tudo que envolve a terceirização para países onde a língua inglesa não é predominante.

S: Voltando ao assunto da confiança, você acha que a utilização da voz para comunicação, como aspecto social, pode ser algo que aumenta a confiança entre os indivíduos?

A: Certamente, é mais fácil transmitir a mensagem e mais fácil estabelecer confiança, sendo mais social, estando mais próximo da esfera social de uma ação.

S: Na sua opinião, qual a importância de um encontro face a face no início de um projeto para estabelecer relações de confiança.

A: Ótimo ponto. Eu acho que isso é uma ótima idéia. Nós nunca fizemos nada assim. Seria realmente uma ótima idéia. Eu acredito que é crítico que pelo menos a gerência tenha uma boa visão de quem é quem no projeto. Não sei se isso teria tanta importância para os desenvolvedores. Mas certamente os tomadores de decisão e os gerentes necessitam disso. Na nossa organização isso faria muito bem na minha opinião. E eu já ouvi dos desenvolvedores que eles tiveram melhores experiências após conhecer pessoalmente outros desenvolvedores ou mesmo os clientes.

S: Como você mede a performance de membros da equipe que não estão no mesmo local físico. Quando você está no mesmo escritório é fácil caminhar até a pessoa e ver se ela está indo bem, se ela está tendo progresso ou não. Como você faz isso quando não está no mesmo escritório?

A: Nós tentamos fazer com que os membros da equipe concordem com marcos e métricas em que eles serão medidos, e é praticamente isso. Nós temos medidas menos estritas com os membros que estão remotos do que os que estão no escritório.

S: E que tipos de métricas vocês usam em geral? Problemas resolvidos, número de testes...

A: O número de tarefas cumpridas, número de horas produtivas trabalhadas.

S: Você leva em conta a dificuldade relativa das tarefas? Por exemplo alguém pode terminar cinco tarefas difíceis e outra pessoa pessoa pode terminar vinte tarefas simples no mesmo período de tempo. Como você mede isso?

A: Nós concordamos como um time a respeito de quanto tempo uma tarefa pode levar. Se fossemos uma melhor empresa, iríamos comparar nossas métricas mas não fazemos isso. Idealmente, compararíamos as tarefas cumpridas e o tempo estimado com o número de horas reportadas como trabalhadas. Não temos realmente uma boa maneira de medir a eficiência. Somos pequenos o suficiente, mas temos esse

problema de definir padrões e verificar se as pessoas estão alcançando esses padrões. Nós simplesmente sabemos pois falamos com os desenvolvedores diariamente. É muito mais qualitativo pois somos uma empresa pequena. Mas ainda é um problema, não temos uma forma objetiva de identificar a eficiência.

S: Há três questões inter-relacionadas que foram abordadas superficialmente. Quando há uma discussão via voz, é necessário um resumo do que foi discutido para que seja possível referenciar aquela discussão mais tarde.

A: E para provar que as decisões foram tomadas naquela conversa, sim.

S: Outra questão é que quando há comunicação via email pode acontecer que as pessoas coloquem muitas questões não relacionadas em um mesmo email, muitos itens que necessitam ação e quando esse email é respondido algumas das questões podem ser deixadas de lado e acabar por ser esquecidas. Seria a solução para esse problema a utilização de ferramentas de gerência de problemas de forma a poder manter um estado nessas ações ou tarefas para marcar quais foram cumpridas ou não?

A: Isso é algo que estamos tentando fazer. Mas o que acontece é que itens informacionais são tão importantes quanto essas tarefas. Mas suas atividades são diferentes em natureza. E esses são tão importantes quanto diferentes. O que estamos tentando fazer e criar tarefas a partir de problemas. Quando alguém cria um item que necessita ação tentamos criar uma tarefa associada. O problema é criar uma cultura onde os membros da equipe identifiquem e solicitem outros membros da equipe a criar tarefas a partir de itens que não foram respondidos. É um problema cultural, que nenhum gerente pode resolver sozinho. Especialmente se for uma equipe de curta duração que eles entendam e “comprem” a idéia. É mais fácil não fazer algo do que fazer algo. Quando conseguirmos que nossas equipes entendam que essas são as medidas em que ele são monitorados, possivelmente o número de tarefas criadas. O fato de que eles identificaram tarefas. Essa é provavelmente uma boa maneira de garantir a consistência. Isso é o que acontece hoje, se uma tarefa não for registrada, ela certamente será perdida, ou você fica dependente do heroísmo de uma determinada pessoa em garantir que algo seja implementado. E não há como manter uma equipe grande baseado em algo tão subjetivo.

S: Falando um pouco sobre a criatividade e como novas idéias são criadas e como a solução de problemas acontece. Você costuma receber “feedback” de membros remotos da equipe quando um projeto inicia ou quando um problema precisa ser consertado ou você depende da equipe local para isso?

A: A gente não diferencia muito entre a equipe local e remota. Nós solicitamos o “feedback” de todos. O problema é realmente a visibilidade sobre todas as coisas diferentes que estão acontecendo ao mesmo tempo já que temos muitos projetos acontecendo ao mesmo tempo. Pelo lado criativo, nós solicitamos o “feedback”

dos membros do time, mas não temos muito “feedback” de pessoas que não estão na equipe dando sugestões, e essa é a forma que a gente se organiza. Mas eu acredito que há formas de se extrair a criatividade, possivelmente com exercícios, um dos quais a gente tem feito que são as reuniões semanais onde uma pessoa tem a liberdade de falar sobre um tópico que ela tenha pesquisado e que a empresa tenha alocado algum tempo para ela fazer essa pesquisa, para essa pessoa falar a respeito do assunto de um nível de pesquisador e de como o item em foco poderia ser incorporado, podendo gerar uma nova idéia. O desenvolvimento de software é um processo criativo, ou pelo menos os melhores desenvolvedores são criativos. Se você puder aproveitar essa criatividade e motivação e puder juntar uma boa quantidade de pessoas, e tiver um processo não existe problema que não possa ser resolvido. É realmente uma questão de cultivar a criatividade e a motivação da equipe como um todo, e isso é um problema difícil de se resolver. Nós geralmente lidamos com isso tratando todos da mesma forma e ouvindo o “feedback” de todos.

S: Quando as pessoas da equipe estão no mesmo espaço físico, na mesma sala, o fato de eles estarem no mesmo local significa que eles podem compartilhar mais facilmente qualquer conhecimento que eles possam adquirir ou soluções para novos problemas que possam aparecer. Há menos barreiras para buscar ajuda junto a outros membros da equipe. Como isso mapeia para uma situação onde a equipe é geograficamente distribuída?

A: Você realmente não tem esse tipo de comunicação “fora de banda”. Você não tem a situação onde alguém faz uma pergunta a outra pessoa e um terceiro ouve a questão e intervém. Na verdade, a gente não tem isso nem mesmo no nosso escritório local no momento e eu realmente espero que você ajude com isso, e para que a gente tenha uma lista de discussão ou outro meio para que as pessoas possam fazer perguntas fora de tópico entre todos os membros da empresa. Um dos problemas na realidade é que a próxima coisa que pode acontecer ali é que as pessoas acabem gastando boa parte do seu tempo em discussões fora de tópico quando você realmente quer que eles trabalhem. Há uma tendência natural para alguém sair do tópico num bebedouro por exemplo, por que ali elas já estão fazendo uma pausa. E você quer que mesmo a pausa delas possa criar algum valor para a empresa. Então você quer que de alguma forma exista algum tipo de conversação formal mas fora de tópico. É difícil lidar com as expectativas que você terá conversações fora de tópico enquanto você está trabalhando. Quando você está trabalhando você não precisa ser importunado ou... A grande diferença entre um email, que requer pensamento para formular uma resposta comparado a simplesmente ouvir uma conversa de terceiros e falar significa que a quantidade de esforço é ordens de magnitude menor quando se trata de comunicação face a face. Na realidade isso não mapeia muito bem. Na realidade a única forma de fazer isso mapear bem seria ter grande largura de banda

ou grande interatividade entre membros remotos, o que seria muito custoso.

S: Você acredita que centralizar o fluxo de informação, como por exemplo, centralizar em uma pessoa que escuta a todos os outros membros da equipe e direciona-os para a pessoa certa poderia ajudar ou seria mesmo melhor ter um canal amplo onde qualquer um pode despejar seus pensamentos e idéias e qualquer um pode contribuir? Qual seria a forma mais eficiente de compartilhar conhecimento e criatividade?

A: Eu não acredito que exista uma resposta correta para isso. Eu acredito que depende de quem a empresa é, de como a cultura da empresa é. Para nós, eu acredito que ter um canal mais amplo onde as pessoas pudessem dar e receber “feedback” funcionaria melhor do que uma pessoa centralizadora de conhecimento. Porque a pessoa centralizadora conhece todas as habilidades, mas seria muito limitada. Não temos muitas pessoas na nossa empresa que poderiam entender e rotear as outras pessoas. Certamente um modelo um pouco mais descentralizado, semelhante ao modelo Open Source seria uma alternativa melhor. Não praticamos esse compartilhamento de conhecimento no momento, somos bastante centralizados e organizados ao redor de projetos.

S: Você acredita que uma estrutura hierárquica dentro de uma empresa ajuda na performance da equipe, ou a equipe seria mais produtiva se a estrutura hierárquica não existisse? Por exemplo, foi feita uma análise de projetos Open Source e descoberto que os projetos de sucessos são aqueles que tem uma estrutura fortemente hierárquica onde os problemas são escalados entre os níveis diferentes, contrário ao senso comum sobre os projetos Open Source que prega que os projetos se organizam de forma caótica e não estruturada. Você concorda que ter uma estrutura hierárquica é melhor para um projeto do que ter uma estrutura horizontal?

A: Absolutamente. Novamente, isso é um fenômeno social. Todos estão constantemente testando todos e buscando respostas. As pessoas precisam saber quando há silêncio na sala quem é o responsável por responder. E isso só pode existir se houver hierarquia. E eu concordo que não há como manter todos no mesmo nível. É essencial ter pessoas que tem papéis e responsabilidades. E isso por si só pode implicar algum senso de hierarquia.

S: Você acha que essa hierarquia funcionaria se fosse imposta, ou você acha que só funcionaria se fosse como um tipo de democracia, onde o líder emerge da multidão e é escolhido, não apontado?

A: Eu acredito que é uma meritocracia, com certeza. Isso precisa necessariamente ser um processo evolutivo. Quando tentamos apontar uma pessoa... já tentamos fazer isso no passado e falhamos. Você precisa dar a todos os mesmos pontos iniciais. E mesmo que alguém entre com um salário melhor ou habilidades melhores, essa pessoa realmente deve começar em um nível mais baixo e deve ter a oportunidade

de galgar a níveis mais altos rapidamente. É assim que eu penso.

S: Você já teve algum tipo de conflito relacionado a salários por exemplo? Que tipos de conflitos você já teve em projetos distribuídos?

A: Os maiores conflitos que tivemos foram causados por falhas de comunicação, problemas de agendamento (por que pessoas tem diferentes feriados e férias), transferência de conhecimento... esses foram os maiores. É bastante comum as pessoas não enxergarem as mesmas prioridades quando elas estão em um escritório remoto quando comparado às pessoas que estão no dia a dia no mesmo escritório.

S: Você acredita que os problemas de comunicação que você teve foram devido ao meio de comunicação utilizado ou à forma que as coisas foram comunicadas?

A: Provavelmente ambos. É sempre uma mistura de ambos. O que as pessoas não entendem é que a primeira coisa que deveria acontecer é serem francas e dizer: “Eu não entendo o que você esta tentando transmitir. Por favor, diminua o ritmo e me explique o que você quer dizer.”. E esse tipo de comunicação é muito difícil de acontecer remotamente. Eu acho que por causa da grande barreira de entrada para a comunicação remota, e isso é realmente algo que deveria ser levado em conta quando se esta comunicando remotamente. Meu sentimento é que desenvolvimento distribuído é muito bom quando se tem um processo muito bom e você realmente pega junto ou se você tem pessoas altamente motivadas e inteligentes. E mesmo assim é necessário ter uma pessoa bastante inteligente gerenciando esses desenvolvedores, e que tem tempo para gerenciá-los. Mas ao mesmo tempo isso ainda é diferente do que ter pessoas num mesmo local, é muito mais simples porque você tem uma banda de comunicação muito mais larga.

S: Você já teve problemas de comunicação relacionados à diferenças culturais ou de linguagem?

A: Sim, mas não tão frequentemente quanto esperado.

S: Você nunca teve membros em uma equipe que eram da Índia, Japão ou China?

A: Não, nós não contratamos pessoas que são culturalmente “esquisitas” trabalhando com americanos. Quer dizer, os chineses e indianos seriam mais culturalmente esquisitos, principalmente por que eles tipicamente escutam e fazem, ao invés de debater e nossas equipes são focadas em torno do debate e meritocracia e discussão, não execução de ordens que vem de cima, que é obviamente a melhor forma, a hierarquia mais rígida, mas não somos assim e nossos funcionários não querem que seja assim. Eles querem hierarquia suficiente para entender onde eles podem ir e buscar respostas mas ao mesmo tempo ter flexibilidade suficiente para executar e ser criativos.

S: Quando entrevistei o Kiko, ele falou algo que achei realmente interessante, que a maior parte das habilidades e forma de pensar que se espera de alguém que está trabalhando remotamente são inerentes num indivíduo que tem uma larga experiên-

cia em trabalhar com projetos Open Source. Por exemplo, essas pessoas geralmente tem um bom inglês técnico, boas habilidades para comunicação e auto-motivação. Você concorda?

A: Sim, concordo.

S: Então isso quer dizer que se você estivesse procurando alguém para um cargo numa equipe de desenvolvimento distribuído, você buscaria alguém que tem um histórico de envolvimento em projetos Open Source.

A: Correto.

S: Você gostaria de fazer mais algum comentário sobre os desafios de gerenciar uma equipe geograficamente distribuída?

A: Eu acredito que as coisas que eu gostaria de adicionar são: em nossa empresa nós realmente nos referimos às pessoas como tendo “cultura Open Source”. Nós conhecemos que Open Source é um traço cultural o que significa que existem pessoas que são um tanto quanto acadêmicas, questionadoras e independentes, pessoas que são capazes de se relacionar socialmente a outras pessoas que não estão na mesma sala, que são capazes de ter empatia por outras pessoas que estão em locais distantes. E eu acho que esse é um aspecto muito importante e que requer que as pessoas cresçam dentro do nosso ambiente. Kiko e eu temos um histórico bastante similar. E você é tão jovem que cresceu em meio à tecnologia. Seria muito difícil para uma pessoa que fosse mais velha, e eu já vi isso dentro da nossa empresa, que essa pessoa entenda e seja muito efetiva. E quando eu digo muito efetiva, esse é um ponto chave para se entender isso: sim, você pode ter pessoas realmente inteligentes trabalhando para você e não é difícil fazer algo funcionar. Por que se essas pessoas são inteligentes e auto-motivadas, mesmo se o seu gerente for incompetente essas pessoas conseguem ir além disso com sua motivação e mesmo assim ter sucesso, executar o que é necessário. Mas o que é muito mais difícil é pegar alguém que seja medíocre e colocar para desenvolver remotamente e ter uma gerência efetiva que possa gerenciar essas pessoas à distância. Não tenho visto muitos desses. E é daí que surgem os resultados negativos em organizações que se organizam de forma distribuída.

REFERÊNCIAS

ADLER, N. **International dimensions of organizational behavior**. [S.l.]: PWS-Kent, 1991.

BELLAVITA, C. **How public organizations work: learning from experience**. [S.l.]: Praeger Publishers, 1990.

BRATTON, J.; GOLD, J. **Human resource management: theory and practice**. [S.l.]: Palgrave Macmillan, 2003.

CAIDEN, G. **The dynamics of public administration: guidelines to current transformations in theory and practice**. [S.l.]: Dryden Press, 1971.

CLELAND, D. **Strategic management of teams**. [S.l.]: Wiley-IEEE, 1996.

CRAMTON, C. The mutual knowledge problem and its consequences for dispersed collaboration. **Organization Science**, v.12, n.3, p.346–371, 2001.

CROSS, R.; CUMMINGS, J. Tie and network correlates of individual performance in knowledge-intensive work. **Academy of Management Journal**, v.47, n.6, p.928–937, 2004.

DE MILLO, R.; LIPTON, R.; PERLIS, A. Social processes and proofs of theorems and programs. **Communications of the ACM**, v.22, n.5, p.271–280, 1979.

DERTOUZOS, M. L. **The unfinished revolution: human-centered computers and what they can do for us**. [S.l.]: HarperInformation, 2001.

DRUCKER, P. **Management: tasks, responsibilities, practices**. [S.l.]: Harpers & Row Publishers, 1974.

FERRIS, S. P.; GODAR, S. H. **Teaching and learning with virtual teams**. [S.l.]: Information Science Publishing, 2006.

FISHER, K. **Leading self-directed work teams: a guide to developing new team leadership skills**. [S.l.]: McGraw-Hill Professional, 2000.

- FISHER, K.; FISHER, M. **The distributed mind**: achieving high performance through the collective intelligence of knowledge work teams. [S.l.]: AMACOM, American Management Association, 1998.
- FOGEL, K. **Producing open source software**. [S.l.]: O'Reilly, 2005.
- FRIEDMAN, T. **The world is flat**. [S.l.]: Farrar, Straus and Giroux, 2005.
- GACEK, C.; ARIEF, B. The many meanings of open source. **Software, IEEE**, v.21, n.1, p.34–40, 2004.
- GLASS, R. The sociology of open source: of cults and cultures. **IEEE Software**, v.17, n.3, p.104–105, 2000.
- GLOOR, P. **Swarm creativity**: competitive advantage through collaborative innovation networks. [S.l.]: Oxford Univ Pr, 2006.
- HEALY, K.; SCHUSSMAN, A. **The ecology of open-source software development**. [S.l.: s.n.], 2003.
- HERBSLEB, J.; GRINTER, R. Splitting the organization and integrating the code: conway's law revisited. **Proceedings of the 21st international conference on Software engineering**, p.85–95, 1999.
- HOEFLING, T. **Working virtually**: managing people for successful virtual teams and organizations. [S.l.]: Stylus Publishing, LLC., 2001.
- KIDANE, Y.; GLOOR, P. Correlating temporal communication patterns of the eclipse open source community with performance and creativity. **Computational & Mathematical Organization Theory**, v.13, n.1, p.17–27, 2007.
- KLIMOSKI, R.; MOHAMMED, S. Team mental model: construct or metaphor? **Journal of Management**, v.20, n.2, p.403, 1994.
- LERNER, J.; TIROLE, J. Some simple economics of open source. **Journal of Industrial Economics**, v.50, n.2, p.197–234, 2002.
- LEVY, S. **Hackers**: heroes of the computer revolution. [S.l.]: Penguin, 1994.
- LUEG, C.; FISHER, D. **From usenet to co webs**: interacting with social information spaces. [S.l.]: Springer, 2003.
- NANDHAKUMAR, J. Virtual teams and lost proximity: consequences on trust relationships. **Virtual Working: Social and Organizational Dynamics**, p.45–56, 1999.

OKKONEN, J. Performance in virtual organisations. **Frontiers of e-Business Research**, v.1, n.S 267, p.279, 2002.

ONCKEN, W. **Managing management time: who's got the monkey**. [S.l.]: Prentice-Hall, 1984.

PUGH, D.; HICKSON, D. **Writers on organizations**. [S.l.]: Penguin Books, 1996.

RAYMOND, E. **The cathedral and the bazaar: musings on linux and open source by an accidental revolutionary**. [S.l.]: O'Reilly, 2001.

REIS, C. Caracterização de um processo de software para projetos de software livre. **ICMC. Brazil**, 2003.

SHACHAF, P. Bridging cultural diversity through email. **Journal of Global Information Technology Management**, v.8, n.2, p.46–60, 2005.

STALLMAN, R. The free software definition. **Free Software, Free Society: Selected Essays of Richard M. Stallman, Boston: GNU**, 2002.

THOMSON, R. **Managing people**. [S.l.]: Butterworth-Heinemann, 2002.

WATSON, W.; KUMAR, K. Differences in decision making regarding risk taking: a comparison of culturally diverse and culturally homogeneous task groups. **International journal of intercultural relations**, v.16, n.1, p.53–65, 1992.

WEBER, S. **The success of open source**. [S.l.]: Harvard University Press, 2004.

YEE, D. **Development, ethical trading and free software**. http://firstmonday.org/issues/issue4_12/yee/index.html, acessado em Novembro de 2007.