

Transcrição das fitas de áudio gravadas durante o Workshop:

WOSES - 2005

**“UM OLHAR SOCIOTECNICO SOBRE A ENGENHARIA DO
SOFTWARE”**

Realizado em 21 de Novembro de 2005, na sede do
Banco Nacional de Desenvolvimento Economico Social - BNDES
Rio de Janeiro, Brasil

<http://www.cos.ufrj.br/woses/>

Promoção

BNDES – Banco Nacional de Desenvolvimento Econômico e Social
COPPE/UFRJ

Edição

Cássio Adriano Nunes Teixeira (BNDES; PESC/COPPE/UFRJ)
Nicolaas Ruberg (BNDES; PESC/COPPE/UFRJ)

Organização

Cássio Adriano Nunes Teixeira
Henrique Luiz Cukierman
PESC/COPPE/UFRJ – Linha de Pesquisa: Informática e Sociedade
Nicolaas Ruberg
PESC/COPPE/UFRJ – Linha de Pesquisa: Banco de Dados

Realização

PESC/COPPE/UFRJ – Linha de Pesquisa: Informática e Sociedade

Palestra 3: Os Aspectos Não-Técnicos Intervenientes no Desenvolvimento Distribuído de Software – FACIN/PUCRS

[Rafael Prikladnicki, PUC-RS:]

A idéia dessa rápida apresentação, antes de mais nada, deixa eu me apresentar rapidamente, sou professor da PUC Rio Grande do Sul, há 2 anos, mais ou menos, fiz minha graduação e meu mestrado lá e agora, atualmente desde março deste ano sou aluno de Doutorado também lá na PUC do Rio Grande do Sul, isso orientado pelo professor Jorge Audin. O que eu vou falar, vou tentar abordar para vocês, nesses 20 minutos, 25, meia hora, tentar dar um histórico do que é que a gente vem desenvolvendo lá na PUC do Rio Grande do Sul em termos de pesquisa numa área que é uma área que é bastante recente, que é uma área que a gente denomina de Desenvolvimento Distribuído de Software, vou procurar explicar para vocês o que é que significa Desenvolvimento Distribuído de Software, e a idéia é que eu vou explicar isso ao longo da apresentação, nós chamamos essa apresentação de, nós vamos tentar identificar alguns aspectos não técnicos presentes no desenvolvimento distribuído de software. Mas a idéia de porque é que nós chamamos de aspectos não técnicos, eu vou construir isso ao longo da minha fala.

Eu vou procurar caracterizar, antes de mais nada, essa área, porque não sei quantos de vocês sabem o que é que significa, quando nós falamos em Desenvolvimento Distribuído de Software, então, acho importante fazer essa contextualização, e depois vou entrar na discussão dos aspectos não técnicos, e um estudo que a gente tem trabalhado aí nos últimos 6 meses, que é a questão da fragmentação do conhecimento, que tem relação com o que o Professor Henrique falou na abertura, relação da Engenharia de Software com outras disciplinas, com outras áreas de conhecimento. Bem a primeira coisa é um slide que eu gosto muito de usar que é a motivação para trabalhar com Desenvolvimento distribuído. Desenvolvimento Distribuído, se eu pudesse resumir rapidamente, seria desenvolver software com equipes geograficamente dispersas.

Eu tenho equipes, eu tenho pessoas que estão localizadas em países diferentes, em cidades diferentes, em salas diferentes, e elas tem que se

organizar como uma equipe para desenvolver software. E aí a Motorola montou uma experiência onde eles tinham que desenvolver um projeto e eles tinham 20% da equipe nos Estados Unidos. E aí eles pegaram o resto da equipe de mais 6 países. E como eles mesmo relataram, eles disseram que diversos riscos, diversos desafios foram identificados e eles tiveram que, aos poucos, ir resolvendo isso. Então, essa seria a motivação principal dessa apresentação. Bom, desde 2001, nós trabalhamos na área de Desenvolvimento Distribuído de software na PUC desde 2001, quando foi feita a primeira dissertação de Mestrado. Hoje a gente já tem em torno de 4 dissertações concluídas, esse ano vão finalizar mais duas, e tem duas Teses de Doutorado que iniciaram nessa área.

A gente tem um grupo que é um grupo de pesquisa em Desenvolvimento Distribuído de software e existe um grupo global, que é liderado pela Universidade de Vitória no Canadá, que é o “Working Group in Distributed software Engineering”, que foi criado no ano passado. No final tem referências, para quem tiver mais interesse em acessar essas informações. Bom, isso aí é o site do grupo, depois vocês podem encontrar mais informações. Quais são os objetivos. Eu acho importante dar o contexto, quando eu for falar de aspectos não técnicos eu vou abordar a relação principalmente com essas 4 linhas de pesquisa.

Então, o objetivo do grupo é desenvolver pesquisa, na área de desenvolvimento distribuído de software, como a área de Desenvolvimento Distribuído ela surgiu, ela não nasceu no Brasil, ela tem muito mais ênfase fora do Brasil, muito motivado pelo desenvolvimento, pelo investimento de empresas multinacionais e centros de desenvolvimento na Índia, na China, no Brasil também, mas a área é muito mais a pesquisa nessas áreas é muito maior fora do Brasil. Então, a nossa idéia é desenvolver pesquisas nessas áreas, buscando parcerias internacionais, e utilizando métodos empíricos, métodos experimentais, entrando nas empresas, entendendo como é que elas funcionam e como é que a gente pode resolver alguns problemas dessas empresas. A gente tem 04 linhas básicas.

A primeira delas é bastante relacionada com o que a professora Regina falou, a questão do processo de desenvolvimento, só que aplicado ao

Desenvolvimento Distribuído software, a gerência de projetos, a gestão de conhecimentos, e a infraestrutura tecnológica, suporte tecnológico para Desenvolvimento Distribuído de software. Para você terem só uma idéia de números, existem 09 professores envolvidos nesse grupo, quatro alunos de Doutorado, 5 de Mestrado e um de Graduação, e alguns projetos tem parceria com a Universidade de Illinois, e alguns com a Universidade de Vitória. Nem todos,...

Dentro da linha de processo de desenvolvimento de software a gente tem trabalhado nessas 04 áreas principais. Que é Engenharia de Requisitos, Manutenção, Métricas e uma área que a gente denomina Modelos de Maturidade para Desenvolvimento Distribuído de software que a gente pode traduzir para, nas nossas observações e estudos, que a gente fez, nos percebemos que existia uma evolução, uma empresa que estabelecia, por exemplo, um centro de desenvolvimento nós temos lá na PUC um parque tecnológico, e nós temos dois casos bastante similares que é o caso de uma empresa subsidiária de uma multinacional de Portugal, que é a Atlantic do grupo, e as duas tem desenvolvimento off shore. A Dell tem seu desenvolvimento não parque, e o grupo Sonai criou um centro de desenvolvimento de software também no parque. As duas elas tiveram uma evolução praticamente idêntica, só que em tempos diferentes. Só que as duas tiveram que resolver os mesmos problemas em tempos diferentes. E aí a nossa idéia da questão da maturidade tentar estudar se existe, a gente não sabe; eu não sei responder, se existe se eu posso dizer que existe uma evolução na maturidade para Desenvolvimento Distribuído dentro de um cenário específico.

Então, é mais ou menos isso que a gente estuda dentro dessa, desse último projeto aí. Na gerência de projetos o foco principal é gerência de riscos, alocação e treinamento de recursos humanos. Na questão de gestão de conhecimentos, é a parte de processo de gestão de conhecimento e também a infraestrutura para gestão de conhecimento aqui a gente entende por ferramentas. E por fim, infraestrutura tecnológica para Desenvolvimento Distribuído de software. Qual a ênfase? Comunicação entre equipes distribuídas e licitação distribuída de requisitos, o seja, eu tenho dois analistas,

ou tenho um analista aqui e um cliente em outro lugar e eu preciso entender quais são os requisitos do software e eu preciso fazer isso de forma distribuída e a idéia é que se tenha alguma infraestrutura algum ferramental que suporte isso. E Planejamento e controle de projeto de forma remota. Bom, dado esse contexto, vou caracterizar um pouco o Desenvolvimento Distribuído software. Esse gráfico é só uma,..., eu estou tentando explicar aqui porque é que a gente começou da década de 90 para cá, a gente pode dizer, que começou a ter um grande investimento em criação de centros de desenvolvimento de software em países como Cingapura, Brasil, China, Índia, diversos outros países. E quais são os motivos? Redução de custos, busca de profissionais qualificados e esse gráfico tenta mostrar também um outro fator, que é a falta de profissionais disponíveis nos Estados Unidos.

O grande foco, um deles, pra investir em Desenvolvimento Distribuído sai das empresas multinacionais que tem a sua matriz nos Estados Unidos. Então, se a gente for pegar aqui desde a década de 60 até os dias de hoje, isso aqui é um dado de 98, essa linha azul mostra os profissionais disponíveis para desenvolvimento de software e a linha preta a demanda por serviços de software. então, existia muita demanda, e não existiam profissionais suficientes nos Estados Unidos. Um outro fator que fez com que as empresas tivessem que investir em opções fora dos Estados Unidos é a falta de profissionais disponíveis para isso. E aí, se eu fosse dizer quais são os motivos que levam,..., se eu fosse ter que decidir entre desenvolver software distribuído e desenvolver software com uma equipe numa mesma sala, eu sem dúvida, sem pensar muito, ia dizer que prefiro desenvolver software com uma equipe numa mesma sala.

Só que o cenário em que a gente vive no mundo atual, faz com que seja mais interessante desenvolver software distribuído e a gente tenha que aprender a lidar com isso. Então, a gente vê a questão da globalização, a gente vê a questão de vantagens competitivas, de estar perto de um mercado local, qualidade, a questão de eu desenvolver software com uma equipe que está na Índia, talvez eles tem muito mais experiência podem ter muito mais qualidade do que se eu tiver que desenvolver software na, com uma equipe que está na matriz da empresa, em qualquer lugar do mundo, e flexibilidade no

sentido de eu poder alocar recursos de baixo custo ou alocar recursos em outros países. Eu acrescentaria aí mais duas questões, além da globalização. um grande aumento da terceirização, e a importância que os sistemas, vamos chamar de sistemas de informação mas se quiser chamar de software, a importância que os sistemas de informação tem hoje para as empresas.

Eu não sou da época de 60, 70, mas naquela época a gente tinha, por exemplo, se eu pegar uma empresa e tentar identificar quais eram os sistemas que essas empresas tinham, elas não tinham tantos sistemas quanto a gente vê hoje. Tinha automatização de folha de pagamento, automatização de alguns outros sistemas, mas hoje isso é muito mais visível, as empresas dependem muito mais de sistemas e aí eu estava ouvindo uma palestra lá em Porto Alegre há um mês atrás, de uma pessoa falando, e achei interessante que ele falou o seguinte. Qualquer um de vocês, depois que acabar essa palestra, tentem acessar o site da Varig, da TAM, da Gol, qualquer companhia aérea, e tentem reservar uma passagem se não tiver o sistema no ar. E eu fiquei com esse exemplo na cabeça, porque também é, eu identifico como um terceiro fator para te ter essa grande ênfase na importância dos sistemas e isso faz com que, além da globalização, e além da terceirização de contratar empresas externas, contratar pessoas externas, eu tenha que investir em desenvolvimento de sistemas e aí as empresas tem optado por desenvolver isso com baixo custo, desenvolver de outras formas. Então, o que é que acontece?

A gente tem um cenário de mesma localização física e isso passa para um cenário distribuído. Então, eu não tenho mais a minha equipe num mesmo lugar. E aí eu tenho diversas questões que começam a surgir. E vocês você ver ao longo da apresentação que talvez muitas das questões que eu vou falar, elas existem no desenvolvimento vamos chamar de tradicional. A questão é que quando a gente está nesse cenário, essas questões são muito mais acentuadas. Bom, o que é que o Desenvolvimento Distribuído de software, é uma área recente, primeira coisa. Segundo; o que se busca, converter mercados nacionais em mercados globais. Por que? Porque te permite ter recursos globais, vantagens de estar perto do mercado local, poder investir, não estar só num país, a tua inserção no mercado global se torna mais

fácil, por ser recente, tem muitos conceitos envolvidos então, pegando aí só de um artigo a gente já consegue identificar uma série de conceitos diferentes.

Hoje em dia o pessoal fala muito em desenvolvimento “offshore”, fala muito em “offshore outsourcing”, e de uns tempos para cá tem surgido muito forte a questão do “offshore insourcing”, que a diferença do “offshore outsourcing” para o “offshore insourcing” é basicamente eu desenvolver software para fora ou desenvolver software para a demanda interna da minha organização. então, dei exemplos bem claros, de novo vou pegar o exemplo do parque tecnológico. Nós temos lá o centro de desenvolvimento da HP e da Dell.

O centro de desenvolvimento de software da HP ele é um centro de desenvolvimento que desenvolve software “offshore”, mas é ‘outsourcing,’ porque ele não desenvolve para a HP, ele desenvolve pra empresas externas. Já o centro da Dell, ele desenvolve software apenas para consumo interno da Dell global. Então, já é um cenário de “offshore insourcing”. São conceitos que tem sido muito abordados. E aí, questão de demandas e custos já falei, questão de sinergia cultural. Está presente no mercado global a questão do “time to market” que eu tenho cada vez menos tempo para colocar o produto no mercado, eu preciso criar estratégias para isso, e às vezes eu tenho um maior rigor, maior experiência com equipes que não estão no meu país de origem.

O país onde é a matriz da minha empresa. Então, a gente começou a perceber no Brasil, isto sendo muito mais recente no Brasil, eu diria que isso tem, se tiver 6 anos, talvez seja muito, um grande investimento em desenvolver software com equipes distribuídas. No nosso caso, o que foi que aconteceu? Por que é que a gente resolveu trabalhar com equipes distribuídas? Em 2001 foi assinado o primeiro convênio com empresa na PUC, que foi com a Dell. E naquela época o projeto era com uma equipe de desenvolvimento de software que era Dell Latin América, que eles chamavam. E a gente não conseguiu entregar o sistema, a gente tinha que fazer um sistema e simplesmente não conseguia listar requisitos. A gente tinha que listar requisitos de uma equipe americana, tinha que, então, primeiro, falar inglês, tinha que se comunicar por telefone, e tinha que tentar entender o que é que era para fazer, porque nós

iríamos desenvolver e eles eram os clientes. Então, nós tivemos esse primeiro impacto, esse primeiro desafio, e não conseguimos ter sucesso.

E a partir dali, na época eu era aluno de mestrado do convênio que a PUC tinha com a Dell e tem até hoje, e a gente parou, se olhou, e começou a pensar, bom, qual é a diferença que tem entre eu olhar para o Cássio, por exemplo, ele meu cliente e eu começar a entender o que é que o Cássio quer de mim, ou ligar para alguém que não fala a minha língua, não fala o meu idioma, que talvez culturalmente é diferente e tentar entender o que é que essa pessoa quer que eu desenvolva a gente começou a tentar entender quais eram as principais diferenças e como é que a gente poderia, o que é que se diferenciava e como é que a gente poderia resolver isso. E aí, surgiram todos esses aspectos. Porque é que a gente separou e aí é por isso começo a explicar porque é que a gente chamou o nome desse trabalho de Os aspectos não técnicos.

O meu Mestrado foi nessa área. e eu me propus a criar, como era uma área, na época eu defendi, eu comecei em 2002, fim de 2003, eu me propus a criar um modelo de referencia para Desenvolvimento Distribuído de Software, para identificar quais eram as principais dificuldades, quais eram as principais questões que nós deveríamos tentar priorizar. As questões importantes do Desenvolvimento Distribuído. E no meu, a gente chama de Seminário de Andamento, que o Mestrado tem dois anos, então, em um ano se faz um Seminário de Andamento com uma banca, onde se valida se verifica se o aluno está tendo um bom andamento, se a pesquisa está condizente com o que se espera da área, se é uma pesquisa relevante, e uma das professoras da banca do Seminário de Andamento disse o seguinte, que a gente tinha identificado, a gente tem problemas culturais, técnicos e a gente tem problemas não técnicos. Só que a gente não fez essa diferenciação, a gente simplesmente listou quais eram os problemas que se tinha. Que eram problemas de se desenvolver software de forma distribuída.

E essa professora disse o seguinte. Muito bom! Só que eu acho que vocês não têm “background” suficiente, para, por exemplo, pesquisar e propor alguma coisa para resolver o problema de diferenças culturais. Ou vocês não têm background, não tem conhecimento suficiente para resolver algum

problema de estudar Teoria de Comunicação e propor alguma coisa para resolver o problema de comunicação em desenvolvimento distribuído. E aí, na época, o Jorge Audi, que também foi meu orientador de Mestrado, ele disse o seguinte, na banca. A gente concorda, só que talvez a resposta para esses problemas não esteja na área de Engenharia de Software. Talvez elas estejam fora da área, talvez estejam em outras áreas. E se a gente tiver essa mentalidade de não ir porque ela não está nessa área, talvez a gente nunca vá resolver os problemas que a gente tem na nossa área. E ficou... Mas o que é que a gente fez? Nós resolvemos mudar e resolvemos identificar quais eram os aspectos e, óbvio, demos mais ênfase aos aspectos técnicos. Então, identificamos aspectos técnicos, comportamentais, lingüísticos, culturais, políticos, geográficos, financeiros, sendo que todos esses aqui para nós eram aspectos não técnicos. De lá para cá o que é que a gente observou?

Dos estudos feitos na área de Desenvolvimento Distribuído? Primeiro que muitas vezes os projeto em Desenvolvimento Distribuído eles são bem sucedidos, isso é similar a uma organização nível 1 de desenvolvimento de software, devido a competência dos integrantes da equipe. Então, se o cara tem uma orientação global, os caras sabem falar bem inglês, se a pessoa é sensível para entender que uma indiana é diferente culturalmente de um brasileiro, ele tem facilidade com(..)

(...) quais eram as dificuldades, o que é que a gente tinha na área, a gente ia para um Desenvolvimento Distribuído, as empresas não davam a devida importância até identificar o problema, então, diferença cultural não era um problema. Até eu ter o problema de diferença cultural. Até eu verificar que eu, brasileiro, falar com um americano, ou falar com um norueguês, ou eu gaúcho falar com alguém do nordeste, pode ser que tivesse algum problema cultural. Os processos, a gente notou nesses estudos que os processos ficavam mais formais, havia mais documentação e aumentava a pressão de que tudo está errado, no sentido de que eu não vejo a pessoa diariamente.

Então, se eu falei, tive uma reunião hoje, listei requisitos, e vou falar com a pessoa na outra semana, talvez para essa pessoa ela não vai ter a menor idéia do que eu estou fazendo. Ela não vê, ela não sabe o que eu estou fazendo, aumenta a impressão ou de que tudo está errado ou de que eu

não sei o que está acontecendo. Ou de que o projeto está parado. Tem pessoas que precisam estar toda a hora, todo o dia ligando, ir à mesa, batendo nas costas, se encontrando no café para saber como é que está indo. Mas em Desenvolvimento Distribuído isto não tem, a gente perde um pouco. Então, desde aquela época, quando essa professora falou, nesse seminário de Andamento, que nós não tínhamos background suficiente para abordar os aspectos não técnicos, que nós chamamos de não técnicos, acabei o meu Mestrado, identifiquei os aspectos não técnicos, mas não me aprofundei nos aspectos não técnicos, eu deixei eles identificados, mas coloquei explicito na minha Dissertação de Mestrado que a gente não ia abordar isso porque, primeiro porque não ia dar tempo e segundo porque o nosso viés, a gente ia se aprofundar mais nos aspectos técnicos.

Só que isso ficou na nossa cabeça. A gente não se deu por vencido. E aí depois do Mestrado nós começamos a estudar, se aprofundar um pouco mais em algumas questões. E aí surgiu uma lista, que não é uma lista total, tem muito mais coisa que a gente identificou como sendo, vamos dizer, alguns dos principais fatores aspectos não técnicos que a gente vê como grande importância em Desenvolvimento Distribuído de software que são a questão da confiança entre as equipes, a questão do conhecimento, a questão do contexto, da colaboração, comunicação, cooperação, coordenação, cultura, e idioma. Alguns deles, não são todos. Eu vou rapidamente passar por cada um só para vocês entenderem do que é que eu estou falando.

A questão da confiança é claro para todo mundo que a gente precisa ter confiança quando a gente não está no mesmo local físico e a gente está distribuído; eu preciso fazer alguma coisa, ou se eu espero que alguns de vocês façam alguma coisa e eu não vou ver vocês, eu não estou controlando vocês, eu tenho que confiar em vocês. Mas, acreditem, isso é difícil e é muito complicado em Desenvolvimento Distribuído, eu começar uma relação com alguém que eu nunca vi na vida, nunca vou ver, porque a minha relação é só por telefone, e adquirir confiança. Eu tive a experiência de definir um processo de gerência de risco, de forma distribuída com uma equipe indiana. E a primeira coisa que a gente fez, no caso, o coordenador da equipe do Brasil. Tinha 03 pessoas no Brasil e 03 pessoas na Índia. E nós tínhamos que

desenvolver um processo global para gerência de requisitos na empresa. Que tinha dois centros de desenvolvimento e eles iam usar o mesmo processo.

E a primeira coisa que a gente fez foi criar, fazer alguma atividade distribuída, para tentar criar um clima de confiança entre a equipe, para as pessoas se conhecerem. Nós fizemos uma coisa muito simples. Montamos um Power Point com as fotos e os papéis de cada um em cada um dos lados e trocamos as fotos. Na primeira “conference call” que a gente teve, cada um se apresentou, mas não se apresentou apenas tecnicamente. Disse o que gostava de fazer, para a gente conhecer um pouco mais a pessoa. E isso ajudou a gente a criar uma relação de confiança.

Então nós consideramos que esse é um aspecto muito importante porque a gente viu outras experiências de projetos em equipes que fracassaram simplesmente porque eu dizia que ia fazer, mas não fazia, ou eu ficava,..., eu nunca confiava completamente no trabalho que vinha, então eu tinha que testar algum projeto, eu tinha que testar algum pacote que vinha de uma equipe de fora, que eu não conheço, a gente viu muitos projetos fracassarem por essa falta de confiança. A questão do conhecimento é a questão de saber. Também é uma coisa que influi bastante é a questão de, eu tenho informação sobre o projeto, o que é que está acontecendo no projeto, saber onde buscar essa informação e se eu não souber onde buscar essa informação, eu saber a quem perguntar.

A gente sabe que no desenvolver software, existe uma ênfase muito grande na comunicação, em várias atividades, em desenvolvimento de software, em Desenvolvimento Distribuído ela é ainda maior, e isso impacta na questão,..., é muito fácil, eu ter a sensação de que eu estou desinformado, ou ter a sensação de que alguém sabe mais do que eu sei, ou que eu deveria saber alguma coisa que eu não sei. E quando está distribuído, tem gente que precisa estar toda hora mandando e-mail para saber alguma coisa do projeto. e a gente tem que gerenciar essa expectativa.

A questão do contexto é uma questão interessante, pois talvez essa questão do contexto não é tão, digamos, essa é uma coisa que a gente pode identificar como diferente de um projeto tradicional para um projeto distribuído. E eu vou resumir simplesmente em um exemplo muito breve. A questão de

feriados. Se eu estou trabalhando com uma equipe americana e eu não sei quais são os feriados nos Estados Unidos, ou seja, eu não sei qual é o contexto da minha equipe, pode ser que eu tenha um problema, pode ser que no planejamento do projeto eu tenha uma entrega justamente numa data que é feriado nos Estados Unidos ou que é feriado no Brasil. E isso pode comprometer o resultado final do projeto. Isso seria o contexto, está contextualizado do que, qual o sinal, onde é que aquela equipe está distribuída está inserida e contextualizar aquela equipe do que é que está acontecendo. Semana passada eu tive um exemplo também muito claro de contexto, eu estou escrevendo um artigo com um professor de Chicago, nosso parceiro, e o artigo era para o dia,..., é para o dia 30 de novembro, e a gente estava falando e ele me escreveu um e-mail, simplesmente assim. Meu contexto.

Estarei viajando do dia tal até o dia tal, fazer isso e depois fazer aquilo, então eu tenho esse, esse, esse e esse dia para trabalhar contigo. Pronto, era o que precisava. Eu posso contar com ele naqueles dias. A questão da colaboração e cooperação são muito similares, tem muita gente que confunde, mas a questão da colaboração seria, eu desempenhar um papel de, existir uma colaboração entre pessoas de uma equipe ou de equipes diferentes, de forma integrada, e essas pessoas estão colaborando de forma síncrona naquele problema. Ao contrário da cooperação, que isso geralmente é assíncrono, ou seja, a gente está cooperando mas eu faço aqui e você faz lá.

A questão da comunicação eu já falei, foi um aspecto que nós identificamos como muito presente, com maior ênfase na comunicação no Desenvolvimento Distribuído de software, a questão da coordenação, de coordenar as equipes não no sentido de gerenciar projeto, mas no sentido de coordenar a pessoa, qual o papel daquela pessoa, se ela entendeu o que está sendo delegado para ela, a questão da cultura. A cultura tem diversos exemplos que a gente pode comentar e realmente pode vir a ser um problema, principalmente quando a cultura é muito diferente. Um exemplo rápido que me vem à mente, é um exemplo muito simples, que não aconteceu num desenvolvimento de software, mas que ilustra como a cultura pode influenciar.

Duas pessoas, um brasileiro e um americano, isso aconteceu, estão nos Estados Unidos, o brasileiro foi jantar na casa do americano e os dois ficaram,

na hora de ir embora, o brasileiro foi embora, estava indo embora, os dois pararam na porta e ficaram meia hora conversando na porta. E nenhuma abriu a porta um para o outro. Depois de meia hora, um deles teve a brilhante idéia de perguntar porque é que um não abria a porta para o outro. E aí eles identificaram que na verdade no Brasil a gente abre a porta para o cara voltar, nos estados unidos, se o cara abre a porta é para o cara nunca mais voltar.

E aí eles ficaram meia hora, o brasileiro achando que o americano ia abrir a porta, e o americano achando que o brasileiro ia abrir a porta porque nos estados unidos, quando abre a porta, a visita tem que abrir a porta para dizer: Olha, eu estou abrindo porque eu quero voltar. E depois eles identificaram que, na verdade era um problema de diferença cultural. E aí, uma maneira muito simples de definir a diferença cultural é simplesmente dizendo que é um conjunto de expectativas, ou seja, qual a minha expectativa em relação a alguma coisa.

Em desenvolvimento de software o que a gente procura fazer em Desenvolvimento Distribuído é definir estas expectativas antes de alguma atividade acontecer. Então, o que é que se espera daquela equipe, qual é a expectativa daquela equipe em relação a mim? Qual é a cultura dela em relação ao que a gente está fazendo? E a questão do idioma. O idioma, nem tanto pela diferença de eu falar inglês e outro falar português ou espanhol, mas sim a questão da mensagem. Se eu falar em inglês com alguém, se um indiano for falar em inglês comigo e eu for falar inglês com o indiano existe também uma questão de, uma coisa é eu falar a e ele falar b em português comigo. Outra coisa é eu falar a em inglês de um brasileiro e ele falar b, inglês de um indiano.

Então existe também a questão da cultura envolvida, tem a cultura do indiano, existe a cultura do brasileiro, existe uma série de outras dificuldades que podem existir dessa questão do idioma. Então, em resumo, isso foi o que nós identificamos. Alguns dos principais aspectos não técnicos que nós não nos aprofundamos na minha Dissertação de Mestrado, mas nós reconhecemos como sendo de grande importância e que poderiam levar um projeto de Desenvolvimento Distribuído ao fracasso, como a gente viu que alguns realmente fracassaram. E aí, o que é que a gente percebeu? Como isso

poderia ser resolvido? A primeira coisa é um trabalho preventivo. Algumas das empresas começaram a dar treinamento em aspectos não técnicos de Desenvolvimento Distribuído.

E aí era muito engraçado ver as pessoas tendo treinamento com uma cara do tipo, nunca vou usar isso, para o que é que eu vou usar isso, isso é besteira, isso não influencia na minha atividade do dia a dia, e a gente dando treinamento, eu dei alguns treinamento, eu falava, olha, tudo o que eu estou falando aqui eu estou olhando para a cara de vocês e estou vendo que alguns não estão dando importância. Mas não adianta, na verdade é uma questão de vocês conviverem, terem o problema e verem realmente que é importante. O importante é vocês saberem que isso existe e, quando vocês tiverem esse problema, vocês vão saber a quem recorrer, onde recorrer para resolver. Também um outro fator é a importância que a empresa dá aos aspectos não técnicos. A empresa pode não estar nem aí e pode não dar a importância devida a esses fatores. E aí não tem o que fazer.

Se a empresa não reconhece que isso é importante, não somos nós que vamos reconhecer. E por último uma questão que nós viemos, começamos a trabalhar, muito devido ao comentário daquela professora, mas a gente fez uma coisinha um pouquinho diferente. Nós percebemos que, bom mas nós não temos background suficiente não temos habilidade suficiente, e aí eu começo a entrar naquele último, aquele parágrafo que o professor Henrique leu na abertura, da questão da abordagem interdisciplinar dos pesquisadores que atuam em Desenvolvimento Distribuído ou uma abordagem interdisciplinar das pessoas que atuam em Desenvolvimento Distribuído. A gente começou a ver que, a gente não vai conseguir encontrar todas as respostas, na Ciência da Computação, ou na Engenharia de Software, então, precisamos recorrer a outras disciplinas complementares. E aí entra o que a gente chamou de fragmentação do conhecimento. Que é reconhecer a existência da necessidade, de que, existe um mesmo problema ou uma mesma situação, existe um problema caracterizado e ele vai ser resolvido com conhecimentos fragmentados de várias disciplinas. E aí isso pode ser feito por uma mesma pessoa, pode ser feito por 2 pessoas diferentes, que são os tipos de

fragmentação do conhecimento que a gente tem. Bom, o que é que a gente entende.

A gente foi, para chamar de fragmentação do conhecimento, a gente foi buscar ajuda em especialistas, professores da área de educação. Primeira coisa, porque nós não somos da área e nós fomos até a área. E nós chamamos a primeira coisa que a gente fez, foi denominar disciplinaridade. A gente queria abordar a questão da disciplinaridade em Engenharia de Software. E aí quando a gente sentou na frente de um professor de educação e apresentou o conceito, ele disse o seguinte. Isso não é disciplinaridade, porque disciplinaridade é um tipo de fragmentação do conhecimento. Que pode ser também multi, interdisciplinar e sei lá quantos tipos tem. Na verdade o que vocês estão falando é dos vários tipos de organização do conhecimento, e um deles é o conhecimento disciplinar.

Então nós começamos a estudar isso mais a fundo para todos aqueles aspectos não técnicos que nós identificamos, tentar buscar respostas, não apenas na Ciência da Computação, mas em outras áreas que poderia nos responder de uma forma mais fiel. E aí, para a nossa surpresa e para a nossa alegria, nós achamos um artigo da Conferência internacional em Engenharia de Software desse ano, do James, que é professor; que ele disse o seguinte. Ele é um cara que atua na área de Desenvolvimento Distribuído e mais em um monte de coisas, mas o artigo dele foi focado em Desenvolvimento Distribuído e ele disse que, baseado na experiência que ele tem, baseado nas pesquisas que ele desenvolveu no Desenvolvimento Distribuído, a gente não ia conseguir encontrar todas as respostas na Ciência da Computação, a gente precisava recorrer a outras disciplinas. E aí uma questão nossa que é exatamente o que está naquele parágrafo. A gente acha que as respostas continuam dentro da Ciência da Computação e nem sempre elas estão dentro da Ciência da Computação. A gente tem que ir pra outras áreas. Por outro lado, não adianta, eu dizer que está em outra área, mas as pessoas não sabem o que é desenvolver um projeto com diferentes áreas do conhecimento, as pessoas são essencialmente tecnicistas e vão ficar simplesmente na Engenharia de Software, bom se esse problema envolve outra área então talvez não seja importante eu resolver. Mas a gente já pensa diferente, a gente já pensa que

se é importante porque todos esses aspectos não técnicos que a gente falou, muitos deles a gente não têm, a gente não consegue se aprofundar, conceitualmente na Ciência da Computação.

A gente precisa entrar em outras áreas. E se a gente não resolver, e isso eu estou falando tanto para Desenvolvimento Distribuído e alguns para desenvolvimento tradicional também, a gente pode não conseguir chegar ao final do projeto com sucesso. E aí tem a questão da falta de informação, eu não estou habilitado a trabalhar de forma integrada com outras disciplinas, eu me acomodo, ou não vou resolver, vou levando e descrença, não acredito que isso realmente funcione.

Então, o que é que a gente começou a fazer, especificamente para Desenvolvimento Distribuído? Essa aqui é a divisão que a Pós-graduação lá da PUC dá, até aqui, na verdade, para o programa de pós- graduação, tem a Ciência da Computação e tem uma linha chamada sistemas de informação que envolve essencialmente Engenharia de Software e banco de dados. Não vou entrar em discussão se isso é certo ou isso é errado. mas é como eles dividem, e já dividem há bastante tempo. E aí o Desenvolvimento Distribuído de software está aqui dentro. O que é que a gente está tentando fazer?

A gente está tentando abrir o Desenvolvimento Distribuído de software e tentar ir em outras áreas e verificar se essas outras áreas tem respostas para algumas das dificuldades que a gente tem no Desenvolvimento Distribuído, ou seja nas atividades técnicas de Desenvolvimento Distribuído. Porque por mais que elas sejam técnicas, quando a gente vai tentar resolver, a gente vai tentar executar elas, a gente precisa ter um viés técnico e não técnico e vamos chamar, de repente, de Sócio Técnico. E aí a gente se atreve também a conceituar os diferentes tipos de fragmentação do conhecimento. E aí foi buscar na educação, em autores como Nicolesco, francês, e esses dois são brasileiros.

E o que é que a gente achou? Nós achamos 04 tipos de fragmentação de conhecimento, ou seja, a gente pode atuar numa pesquisa de uma forma disciplinar. Que é simplesmente eu estou na minha disciplina, vou pesquisar ali dentro e acabou. Mas eu posso ter um viés multidisciplinar. Eu tenho varias disciplinas; reconheço a importância delas elas não tem ligação entre elas.

Pode ser que isso aqui, um exemplo claro seria, por exemplo, um currículo, currículo de um curso de graduação, por exemplo, eu tenho um currículo onde eu tenho conteúdo de várias disciplinas só que eles não se comunicam. Cada um faz a sua parte e ponto. A questão da interdisciplinaridade já é um pouco diferente. Eu começo a ter algum cruzamento entre as disciplinas. Eu começo a reconhecer que eu estou aqui mas para eu fazer isso aqui eu tenho alguma coisa de alguma outra disciplina.

É importante eu ter alguma outra coisa de alguma outra disciplina. E por último a que tem, digamos assim, um conceito que não é recente mas ele tem sido mais falado agora, a gente percebeu que o pessoal da área de educação, principalmente educação sociologia, tem abordado uma transdisciplinaridade. Que é, eu tenho diversas formas de relação entre as disciplinas. E aí se eu for fazer uma analogia, uma analogia com isso aqui, eu posso pensar nos trilhos de um trem, por exemplo. Um Multidisciplinar eu tenho 3 trilhos, cada trem faz o seu caminho, e ponto. Aqui, eles se cruzam, ou seja, esse caminho se cruza com esse, então, de alguma forma eu vou pegar um pouco desse caminho para fazer o meu. Mas cada um faz o seu, esse aqui vai fazer esse caminho, esse aqui vai fazer esse, mas eles vão pegar um pouquinho do outro caminho. Aqui eu não só faço o meu caminho, como posso ter uma possibilidade de repente de fazer um outro caminho. E aí vem o viés do que.

Qual seria a diferença que os pesquisadores da área definem a diferença entre intra disciplinaridade e trans disciplinaridade. Na interdisciplinaridade seria mais ou menos assim. Eu estou na minha área, reconheço que existem conceitos de outras áreas que são importantes, e eu vou recorrer a especialistas das outras áreas. Então, eu quero estudar diferenças culturais, vou recorrer a um cara que entende de diferenças culturais. Um cara de Engenharia de Software não é minha área, não faz parte da minha formação, mas eu reconheço que é importante. Então, eu vou recorrer a um profissional especialista nessa área. Por que? Porque eu sei que isso vai passar pelo meu caminho.

O transdisciplinar seria mais ou menos assim. Eu sou da Ciência da Computação, eu trabalho com Engenharia de Software, eu tenho problemas de comunicação e desenvolvimento de software, mas ao invés de recorrer a

especialistas nessa área eu mesmo vou procurar me formar em, tentar ser um especialista também da área de comunicação. Então, eu continuo na minha disciplina, mas eu vou buscar a formação em outras disciplinas. Seriam esses os tipos que a gente encontrou e essa seria uma rápida definição de cada um desses tipos.

Porque, de novo retomando, acho importante frisar, porque é que nós fizemos isso, porque nós queríamos identificar qual seria primeiro, embasar conceitualmente, o comentário daquela professora nos motivou a não só buscar mas embasar conceitualmente nosso trabalho para justificar porque é que nós achamos importante e ir atrás de conceitos que estão em outras disciplinas, em outras áreas. E também para verificar, bom, e agora? O que a gente esta fazendo aqui é disciplinar, é multi, é inter, ou é transdisciplinar.

Então a gente parou e eu não tenho resposta muito certa para dar para vocês, mas pensando rápido, a avaliação que eu faço, é uma opinião pessoal minha, eu acho que nós estamos numa posição interdisciplinar. A minha posição. Eu reconheço que, mas eu não tenho condições de responder, não tenho conhecimento suficiente para responder a certas questões. É obvio que o melhor, a evolução, seria, eu ir para transdisciplinar e aí o que eu coloco, o que a gente acha é que primeiro, talvez essa área de Desenvolvimento Distribuído é uma das áreas de Engenharia de Software onde a gente viu que é muito mais clara a necessidade de integração entre essas disciplinas. Pelo contexto.

Só que muitas vezes esses problemas que surgem não são resolvidos. E o mesmo autor diz o seguinte. Muitas pessoas dizem que isso não é resolvido porque está fora da área de Ciência da Computação e eu não preciso resolver e aí não é profundo e aí talvez eu continue com os meus problemas por anos, anos, anos e anos. Então existe uma necessidade de um maior esforço interdisciplinar, para buscar resposta para os problemas que a gente tem, e alterar os hábitos já estabelecidos. Reconhecer que o hábito que eu tenho atualmente pra interpretar, para buscar conhecimento pra responder os problemas que eu tenho, posso humildemente reconhecer que talvez eu tenha que evoluir um pouco mais. Só que isso não é um consenso dentro da nossa área. Realmente não é um consenso. A prova disso é o comentário que

essa professora fez, tenho certeza. Ano que vem eu vou fazer a minha qualificação, vou abordar isso de novo, porque eu estou fazendo Doutorado, com certeza vou para o lado técnico, mas eu vou abordar aspectos não técnicos porque eu estou nessa área de Desenvolvimento Distribuído e existe uma necessidade muito grande de abordar isso e nós vamos ouvir isso de novo e tem pessoas que não concordam, não pensam assim, então não é um consenso, a gente tem que procurar formas de mostrar que isso é importante.

E a forma que nós encontramos para mostrar é embasar isso de uma forma conceitual. Buscando conceitos de fragmentação de conhecimentos, buscando conceitos de outras áreas, mas não por nós, porque nós achamos que nos estamos num tipo de fragmentação interdisciplinar, ou seja, nós estamos recorrendo a outras pessoas para nos ajudar a definir e aí depois a gente pode tentar resolver os problemas dentro da nossa área.

Depois eu posso disponibilizar a apresentação, mas aí são só referencias, o link primeiro é do nosso grupo de pesquisas, e o segundo é do grupo global de pesquisas, onde o nosso grupo está inserido, e aí tem para quem quiser se aprofundar, quem quiser conhecer a área de Desenvolvimento Distribuído, tem artigos, tem eventos, tem livros... Ano que vem vai ter uma Conferência no Brasil, que é a primeira conferência na área, provavelmente vai ser aqui no Rio, e é bastante gratificante para nós ter esse reconhecimento, o reconhecimento de trazer uma Conferência pra cá, para discutir a questão do Desenvolvimento Distribuído. É isso. Espero ter conseguido pelo menos dar uma visão provocar pelo menos alguma discussão em relação a essa questão da fragmentação do conhecimento.

[Marcus Vinícius, DBA:]

Mais uma colocação, não é nem uma pergunta. Fragmentação do conhecimento, eu acho que tanto o mercado acadêmico, quanto o mercado de fato, discordo um pouco da questão do consenso. O mercado está pedindo isso, pessoas que tomem a decisão, não como você falou. No ciclo de vida hoje o nosso profissional não vai lá no final porque ele é um profissional que está fazedor de coisas. Eu acho que o consenso é mais na questão de ter que

tomar decisão, saber, eu acho que a gente está num passo adiante intradisciplinar como você colocou. E que é mais do que o interdisciplinar.

E mesmo os nossos currículos acadêmicos pedem isso do nosso pessoal de Ciência da Computação. Se você vir um médico, um advogado, ele sai sabendo fazer as coisas mas com a visão muito mais geral. A gente está formando pessoas para também ter essas características, de tomar decisão e saber formar. E nós, de Engenharia de Software, que vamos fazer de fato esse ciclo de vida completo, precisamos ter essa visão intradisciplinar das coisas.

[Rafael Prikladnicki, PUC-RS:]

Só um comentário, na verdade eu ficaria muito feliz se todo mundo fizesse esse tipo de colocação, mas o que a gente vê, é que existe uma relutância de algumas pessoas, tem muita gente que pensa que tem esse viés, e que realmente pensa que a questão da transdisciplinaridade é importante, simplesmente uma questão de ver e entender que a gente precisa ter um investimento em conceitos que não necessariamente são da nossa área de formação, só que não é consenso no sentido que existem pessoas que simplesmente não concordam e acham que a questão é técnica e ponto. E a questão, se tu quiser resolver, vai depender da tua própria formação e se tu quiser investir na tua formação, não técnica, ou Sócio Técnica, tu corre atrás. Exatamente.

[Arísio, A&S Consultoria:]

Sou um consultor independente, eu queria fazer um comentário, de repente, um pedido aqui, humilde, que o seu slide lá ressaltou, a parte de Desenvolvimento Distribuído de software e ficou a impressão de que todo esse enfoque, toda essa tentativa de priorizar esses problemas está restrito a esse segmento. Eu acho que não, porque ele é mais evidente no setor distribuído, mas ele é o calcanhar de Aquiles de todo o desenvolvimento de software. Acho que todo mundo conhece, eu não tenho tantas referências assim, das causas de falhas de projetos. O que a gente vê com mais frequência é aquele das causas. E mesmo nas versões mais atuais, quer dizer, depois de já ter tido aí uns 10 anos de avaliação dessas causas, quando a gente olha o que é que

está fazendo os projetos não darem certo, o que está encabeçando são coisas não técnicas. Sempre, todas as vezes.

Então, não dar importância para isso é enterrar a cabeça dentro da terra. É um tiro no pé. Acho que isso é uma coisa importante, é o assunto importante. Se a gente tivesse que apostar todas as fichas em alguma coisa, seria isso. Eu também humildemente gostaria que o currículo fosse diferenciado. A gente vem evoluindo daqueles que estão na área há muito tempo, fui um bebezinho do uso da informática aqui no Brasil, eu comecei em 65. Então, eu vi, participei dessa evolução, ouvi essas coisas ir mudando. Mas a impressão que eu tenho, eu acompanho o ambiente acadêmico, com mais profundidade, a minha formação é de Economia, então, eu não tive essa prática do software dentro da Faculdade. Mas fica-me a impressão de que a gente privilegia muito a Engenharia. Eu acho que a gente deveria privilegiar o arquiteto. Se fosse pra fazer um paralelo com o currículo da Faculdade deveria ser o de Arquitetura. Que tem a visão abrangente.

Que a técnica é uma coisa, é uma, não é nem a mais importante. Não deveria ser assim nessa, porque a técnica, a maioria dos projetos não precisa de muitas técnicas, não precisa de grandes coisas, a gente não precisa estar descobrindo nenhuma novidade enorme para fazer uma implementação daquele projeto. A maioria das vezes a gente está reutilizando uma técnica, a gente já sabe demais, e mesmo assim falha.

[Rafael Prikladnicki, PUC-RS:]

Só me permite um comentário, em relação à primeira colocação, pode ser que eu não tenha me feito entender, pode ser que não tenha sido claro na hora, mas os problemas que eu identifiquei são problemas que existem em desenvolvimento de software como um todo. A questão é que no distribuído, eu participei de um workshop em agosto e a pergunta foi o seguinte. A gente está, os problemas do Desenvolvimento Distribuído são diferentes do desenvolvimento tradicional ou nós estamos reinventando eles de uma forma mais acentuada? Eu acho que a resposta é, os problemas são os mesmo, de uma forma mais acentuada. Alguns podem até ser diferentes, mas a grande maioria são os mesmos.

O que acontece é que talvez a solução para esses problemas pode ser a estratégia para solucionar, às vezes são diferentes, e quando eu estou num desenvolvimento tradicional, eu já vi muitas vezes, bom tem problema de comunicação, toca a ficha, vamos levando, só que no desenvolvimento distribuído a diferença é que se eu não resolver aquele problema de comunicação, eu não ando. Essa é a grande diferença. São os mesmos, só que a maneira talvez de resolver e a forma que ele aparece, talvez seja um pouco diferente.

[Henrique Cukierman, COPPE:]

Rafael, duas perguntas assim rápidas. Uma, só se você pudesse esclarecer aquela extensão do DADEs em direção a outras disciplinas, como é que vocês estão tentando articular isso na PUC, você desse uma palhinha assim de 1, 2 minutos. A pergunta que eu tenho, eu fiquei com uma,..., dá uma impressão que o Desenvolvimento Distribuído,..., você fala da globalização, do “outsourcing”, do “insourcing”, e as vezes dá uma sensação que são iguais, é um diálogo entre iguais, e acho que há uma tendência a ter fortes assimetrias aí, no sentido de que você tem uma matriz, por exemplo, você citou o caso da Dell. A Dell tem uma matriz americana, então, há uma distribuição, mas uma distribuição que pode ser desigual, então o que eu acho que coloca uma questão muito, e aí sim, muito diferente, muito particular do DADES, é a questão do conhecimento, da produção de conhecimento. Produção de conhecimento local, produção de conhecimento global, apropriação de produção do conhecimento, questões que não apareceram com muita clareza na tua apresentação, também não no “paper”, e talvez você tivesse alguma coisa a comentar.

[Rafael Prikladnicki, PUC-RS:]

Em relação ao primeiro comentário...Essas tentativas de reunir várias disciplinas elas são muito complicadas, elas sempre fracassaram também. Vocês já tentaram reunir numa equipe de projetos, especialistas diferentes... claro, que esses problemas de tradução, eles vão continuar, mas aí seria um arranjo diferente de colocar um Antropólogo, um Psicólogo, um Sociólogo, num

projeto técnico de Engenharia. Então, a primeira coisa, a questão das disciplinas. O que é que a gente está fazendo? Estamos se aprofundando em conceitos que não são nossa área e estamos recorrendo a especialistas. Essa é a forma mais simples.

Escrevemos um texto, tanto que nós escrevemos a questão da disciplinaridade, fomos conversar com especialistas da área de Educação, participamos de eventos da área de Educação, para tentar entender como o pessoal de outra área enxerga aquele conceito, e uma questão, a forma mais simples de integração é simplesmente pedir feedback para essas pessoas. Uma forma mais evoluída seria o que aconteceu numa implantação de CMMI, numa empresa, que aí não é distribuído, é a questão da integração entre as disciplinas, o projeto CMMI nível 2 na Atlantic foi esse que a Ana Regina comentou, em um determinado momento, duas Psicólogas da equipe de Recursos Humanos da empresa começou a atuar diretamente com a equipe técnica. E funcionou muito bem. Eu diria que foi um dos grandes diferenciais. Tanto que a gente relatou isso no SBQS desse ano. Então, foi um dos grandes diferenciais do projeto.

A gente conseguiu juntar técnicos, pessoas da área de Ciência da Computação, Engenharia de Software mais precisamente, com duas Psicólogas que eram parte da equipe de RH, e essas pessoas fizeram todo um trabalho por fora, para garantir que as pessoas pudessem desempenhar, não só o papel técnico, mas o papel não técnico que era necessário para elas desenvolverem para ter sucesso no CMMI. Então, elas fizeram parte da equipe, da equipe de melhoria da empresa, que era como se denominava.

Então, tinha 8 pessoas técnicas e duas Psicólogas na equipe. Então, vai desde um simples feedback até a formar uma equipe de projeto para desenvolver software ou para implantação de processo, com pessoas de outras áreas. E é extremamente complexo, tanto que a gente tem um caso até hoje. O resto é sempre, eu vou recorrer, estou falando de motivação, vou chamar alguém lá da administração, a gente fez já diversas atividades em projetos CMMI, com uma Psicóloga. Da Puc, não dentro da empresa, a gente acompanhou 4 projetos até hoje e nesses 4 projetos existia um marco no projeto. O marco era um evento onde uma Psicóloga fazia uma atividade para

motivar, para apresentar, geralmente era para apresentar referenciais estratégicos da empresa, ou para apresentar políticas de software, só que tudo com Psicólogas, nunca com alguém técnico. São alguns exemplos. E a questão dois, que era das assimetrias... com certeza são diferentes, a questão é que a estratégia que a gente aqui foi de priorizar a questão da fragmentação e não tanto aprofundamento no Desenvolvimento Distribuído de software, só que, “insourcing” e ‘outsourcing’, são completamente diferentes. Uma diferença gritante é a questão de contrato. No primeiro, que é um desenvolvimento insourcing para demanda interna, tu não tem um contrato, entre a matriz com o centro de desenvolvimento de Porto Alegre.

Como eu tenho, por exemplo, se a Stefanini – uma empresa de desenvolvimento de software – vai desenvolver software para um cliente nos Estados Unidos. O contrato é muito mais rígido e a relação se torna diferente. Porque é uma relação cliente fornecedor. Aqui não, eu sou funcionário Dell, centro de desenvolvimento Brasil, eu sou funcionário Dell cliente departamento de Recursos Humanos. É diferente, a questão é que devido a foco mesmo do artigo e à questão de tempo, a gente não se deteve nas diferenças, mas até outros fatores, aspectos, se a gente fosse tentar definir alguns aspectos poderiam ser diferentes. Dos vários tipos, existem vários tipos de Desenvolvimento Distribuído e, com certeza, só a caracterização deles daria mais uma apresentação.

[Cássio, BNDES:]

Bom pessoal, se for mais uma perguntinha rapidinha, a gente mantém a nossa margem de 15 minutos de atraso, senão nós vamos para o almoço. Alguém tem mais alguma?

[Arísio, A&S Consultoria:]

Vocês já estão vendo que eu gosto de fazer comentários. Eu não guardei aqui o nome do Professor Francisco fez um comentário das multidisciplinaridades. Eu diria que do ponto de vista, vamos dizer, novamente eu tenho a visão prática de ter militado aí fazendo projetos ao longo de quase a minha vida toda, pelo menos nos projetos que eu participei e diria em muitos

que são semelhantes no dia a dia, as necessidades não são tão complexas. Eu acho que a minha percepção é que a falha acontece é porque não se dá importância para o problema. Tem um pouco a ver com o comentário; eu cheguei atrasado um pouquinho, mas ainda via parte inicial, da gente estar fortemente apegada a algum paradigma. Isso faz com que a gente deixe de enxergar coisas.

Deixe de avaliar opções. E ache que aquilo que estou fazendo é que está certo e pronto, acabou. Então, se a equipe não dá importância para essas coisas não técnicas, elas não serão vistas mesmo. Se desse a importância, na maioria das situações, eu diria que 80% dos problemas seriam solucionados, porque o simples fato da gente dar atenção ao problema, eu acredito muito na enorme capacidade de adaptabilidade do ser humano. Das pessoas. Então, quando a gente dá importância àquele assunto, a gente corre atrás. E o simples fato de correr atrás, alguma solução, algum paliativo, algum contorno do problema vai aparecer e aí, no mínimo, não vai ser tão drástico quanto ele seria se eu não tivesse nem visto...

Só para dizer que, de repente, eu sou um pouco mais otimista de achar que se a gente der importância ao fato ele tende automaticamente a ir caminhando para uma solução, que a gente vai percebendo, vai lidando com o assunto, vamos descobrindo alternativas, e o caminho vai sendo construído a partir daí.

[Cássio, BNDES:]

Aliás, este workshop aqui é a nossa tentativa de quer começar a construir um caminho diferente para a nossa Engenharia de Software, num país de terceiro mundo. Num país onde você tem uma série de sobredeterminações (...) também, mas eles lá andam mais. Bom, eu sugiro que a gente almoce aqui no Centro. Um lugar grande para quem quiser nos acompanhar, um restaurante chamado DC10, a gente pode ir juntos aqui e lá a gente decola nas nossas conversas também, mas depois tem que voltar duas horas, a gente retoma aqui...