



# Avaliando os Impactos dos Aspectos Não-Técnicos da Engenharia de Software em Ambientes de Desenvolvimento Global de Software: Um Caso Prático<sup>i</sup>

Leonardo Pilatti, Rafael Prikladnicki, Jorge Luis Nicolas Audy

Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS - FACIN  
Av. Ipiranga, 6681 – Prédio 32 – Porto Alegre, RS – CEP 90619-900  
leonardo.pilatti@gmail.com, {rafaelp, audy}@pucrs.br

**Abstract.** *Global software development has attracted a great number of researches in software engineering in the last years. Managers and executives face many challenges and difficulties in all levels, including technical and non-technical aspects. In particular the non-technical aspects include dimensions that go beyond the “right-wrong” method of acting. Very often these factors are under evaluated, showing a significant impact for the whole project. For this reason, the purpose of this paper is to present the results of a case study where we observed the main non-technical aspects of software engineering in global software development environments.*

**Resumo.** *O desenvolvimento global de software tem atraído um grande número de pesquisas na área de engenharia de software nos últimos anos. Gerentes e executivos têm enfrentado diversos desafios e dificuldades em diferentes níveis, desde os aspectos técnicos até os aspectos não-técnicos. Em particular, os aspectos não-técnicos englobam dimensões que vão além do método “certo-errado” de se agir. Frequentemente, o esforço empreendido nos aspectos não-técnicos é subestimado, mas são aspectos fundamentais neste contexto. O objetivo deste artigo é apresentar os resultados de um estudo de caso onde foram observados alguns dos principais aspectos não técnicos da engenharia de software em ambientes de desenvolvimento global.*

## 1. Introdução

Nos últimos anos, percebe-se um avanço em direção a globalização. Em particular, o setor de tecnologia da informação (TI) tornou-se um catalisador das organizações, onde é possível simplificar processos produtivos, utilizando soluções deste setor e aumentar a produtividade nas organizações. Assim, este setor acaba tornando-se um diferencial competitivo entre empresas que visam aumentar sua produção, bem como reduzir o *time-to-market* (Carmel et. al., 2005). Visando minimizar os custos do setor, bem como utilizar recursos geograficamente dispersos, muitas organizações começaram a investir em ambientes de desenvolvimento distribuído de software (DDS) (Herbsleb et. al., 2001). Na área de Engenharia de Software (ES), mercados nacionais têm se transformado em mercados globais, criando novas formas de competição e cooperação que vão além das fronteiras dos países. O desenvolvimento distribuído de software tem



sido caracterizado principalmente pela colaboração entre departamentos de organizações internacionais e pela criação de pequenos grupos de desenvolvedores que trabalham juntos, embora em cidades ou países diferentes. E quando a distância entre as equipes aumenta em escala global, surge o desenvolvimento global de software (Karolak, 1998).

Estas mudanças estão causando um grande impacto não apenas no mercado propriamente dito, mas na maneira como os produtos de software estão sendo criados, modelados, construídos, testados e entregues para os clientes. Neste sentido, o desenvolvimento global de software tem atraído um grande número de pesquisas na área de engenharia de software nos últimos anos (e.g. Herbsleb et. al., 2001; Damian et. al., 2006). São freqüentes os esforços que os pesquisadores têm feito no intuito de entender os fatores que permitem organizações multinacionais a obterem sucesso trabalhando através das fronteiras físicas e culturais dos países. Da mesma forma, diversos desafios e dificuldades são enfrentados, envolvendo aspectos técnicos e não-técnicos. Segundo Kiel (2003), os aspectos técnicos têm sido largamente abordados na área de ES. Mas no caso dos aspectos não-técnicos, são cada vez mais freqüentes as dificuldades que um ambiente global de software apresenta, pois além de esta ser uma prática recente, as pessoas têm diferenças na forma de ser, de agir e de aprender, o que pode tornar a relação à distância mais complexa.

Dentro deste contexto, este artigo tem por objetivo apresentar os resultados de um estudo de caso onde observou-se alguns dos principais aspectos não-técnicos da engenharia de software presentes em ambientes de desenvolvimento global. Estes aspectos foram identificados em estudos anteriores (Prikladnicki et. al., 2005), e agora discute-se seus impactos, as dificuldades surgidas e a proposta de soluções, dentro das análises do estudo de caso. O estudo descrito neste artigo visa contribuir com a crescente base de conhecimento empírica sobre DDS, e foi conduzido em uma unidade de desenvolvimento global de software, localizada na cidade de Porto Alegre.

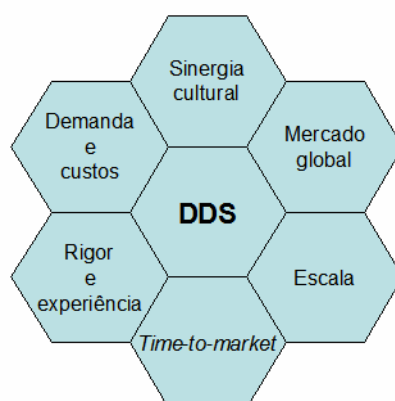
O artigo está dividido em 6 seções. A seção 2 apresenta a revisão teórica. A seção 3 descreve o estudo de caso, a metodologia de pesquisa e os resultados observados, identificando os principais aspectos não-técnicos encontrados. A seção 4 apresenta uma discussão sobre estes aspectos, suas dificuldades e propostas de soluções. Finalmente, a seção 5 apresenta as considerações finais e a seção 6 as referências.

## **2. Revisão Teórica**

### **2.1. Desenvolvimento Global de Software**

Antigamente, o desenvolvimento de software costumava ser realizado apenas por pessoas com altíssimo grau de especialização, trabalhando em centros de processamento de dados de países avançados (Sommerville, 2003). Hoje, entretanto, o desenvolvimento de software, cada vez mais, ocorre de forma distribuída.

Existem diversas razões que levam ao surgimento do desenvolvimento distribuído de software. Essas razões, ou um subconjunto delas, motivam um crescente número de organizações a desenvolverem software de forma distribuída. Entre elas, segundo Carmel (1999), destacam-se a sinergia cultural, o mercado global, o *time-to-market*, o rigor e a experiência, o aumento considerável da demanda e a redução de custos (figura 1).



**Figura 1. Algumas razões envolvidas no DDS**

Por este motivo, as equipes de projeto de software vêm se distribuindo geograficamente em escala mundial, inseridas no conceito de globalização que a sociedade tem vivenciado nos últimos anos. Do ponto de vista de alguns autores (Sengupta et. al., 2006; Sangwan et. al., 2007), a utilização de técnicas de desenvolvimento global representa uma revolução na maneira como se desenvolve software. Fuso horário, diversidades culturais, comunicação, confiança, dentre outros, devem estar alinhadas de forma que o foco de trabalho não seja perdido, prejudicando os resultados. Percebe-se que o desenvolvimento global é permeado por diversas forças que atuam sobre a estratégia, incidindo de maneira positiva ou negativa.

## 2.2. Aspectos Não-Técnicos da Engenharia de Software

Segundo Sommerville (2003), a engenharia de software pode ser definida pelo estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais. Mas como qualquer área do saber, a ES envolve conhecimentos técnicos (próprios da área) e não-técnicos, que agregam conhecimento de áreas complementares. Os conhecimentos não-técnicos envolvem fatores sociais, culturais, comportamentais, lingüísticos e políticos (Evaristo et. al., 2004; Herbsleb et. al., 1999; Herbsleb et al., 2001; Layzell et. al., 2000; Kiel, 2003).

Assim, os aspectos não-técnicos dizem respeito a entidades participantes da engenharia de software que não compõem o papel mestre de desenvolver software, mas que afetam a forma como este é desenvolvido e projetado. De acordo com Layzell et. al. (2000), muitas vezes a qualidade técnica de um projeto pode ser comprometida quando são desconsiderados os aspectos não-técnicos envolvidos, e segundo Kiel (2003), os projetos em ambientes de desenvolvimento distribuído de software normalmente falham devido a um conjunto de fatores sociais, políticos, lingüísticos e culturais.

Considerando-se que este estudo visa abordar os aspectos não-técnicos da engenharia de software em ambientes de desenvolvimento global de software, buscaram-se os principais aspectos identificados na literatura, que foram compilados em Prikladnicki et. al. (2005), e são apresentados na tabela 1. É importante destacar que esta não é uma lista exaustiva, mas sim uma lista com alguns dos importantes aspectos não-técnicos existentes.

**Tabela 1. Aspectos não-técnicos da engenharia de software**

| <b>Aspecto não-técnico</b> | <b>Autor</b>   |
|----------------------------|--|
| Confiança                  | Kiel, 2003<br>Burnell et. al., 2002  |
| Conhecimento               | Desouza et. al., 2004<br>Rus et. al., 2002   |
| Contexto                   | Kogut et. al., 2000  |
| Comunicação                | Herbsleb et. al., 1999<br>Herbsleb et. al., 2001<br>Burnell et. al., 2002<br>Layzell et. al., 2000 |
| Colaboração                | Layzell et. al., 2000<br>Brna, 1998  |
| Cooperação                 | Layzell et. al., 2000<br>Brna, 1998  |
| Coordenação                | Herbsleb et. al., 1999   |
| Cultura                    | Evaristo et. al., 2004<br>Hsieh, 2006<br>Burnell et. al., 2002                                     |
| Idioma                     | Favela et. al., 2001   |

**Confiança:** segundo Kiel (2003), o processo de aquisição de confiança é muito importante quando se trabalha em ambientes de DDS. A falta de confiança pode levar a relutância a compartilhar informações, ao medo de trabalhar de forma distribuída, desentendimentos e possivelmente pode causar impactos consideráveis e/ou irreversíveis nos projetos.

**Conhecimento:** um dos principais recursos de uma organização hoje em dia é o seu capital intelectual (Rus et. al., 2002) e em uma organização de desenvolvimento de software, isto não é diferente. Assim, o conhecimento é um aspecto crítico nas organizações como um todo, e se torna ainda mais crítico em ambientes de DDS, pois o compartilhamento de informação passa a ser bem mais complexo.

**Contexto:** as pessoas interpretam o mundo em diferentes contextos de acordo com o que elas vivem. Em DDS, o compartilhamento do contexto em que se vive é uma atividade bastante importante (Kogut, et. al., 2000), e em ambientes de desenvolvimento distribuído de software ele surge como um aspecto crítico.

**Colaboração:** segundo Brna (1998), o trabalho de forma colaborativa envolve o empenho mútuo dos participantes em um esforço coordenado para solucionar, juntos, um determinado problema. Em ambientes de DDS, a colaboração existe no momento em que uma atividade pode estar sendo realizada por duas equipes de forma integrada.



**Comunicação:** o desenvolvimento de software, particularmente nos estágios iniciais, requer muita comunicação. Quando as pessoas não estão no mesmo espaço físico, a comunicação pode não ser suficiente (Layzell et. al., 2000) e muitas vezes existem diversas barreiras para que a comunicação ocorra de forma produtiva.

**Cooperação:** segundo Brna (1998), o trabalho cooperativo é realizado através da divisão do trabalho entre os participantes, como uma atividade onde cada pessoa é responsável por uma porção da solução do problema. Em ambientes de DDS, a cooperação é visível no sentido em que as equipes de projeto podem cooperar entre si.

**Coordenação:** segundo Herbsleb et. al. (1999), a coordenação é uma parte fundamental do esforço envolvido no desenvolvimento de software. Quando a coordenação envolve a habilidade individual, as atividades humanas e o relacionamento entre as pessoas, os riscos passam a envolver não só os aspectos técnicos, mas também os não-técnicos. Assim, em ambientes de DDS a coordenação é uma atividade crítica.

**Cultura:** ambientes de DDS necessitam de uma grande cooperação de pessoas com diferentes bases culturais. A cultura difere em diversas dimensões consideradas críticas (multidimensional). As diferenças culturais surgem das diferenças culturais entre os membros das equipes e entre as diferentes cidades, países, entre outros. Por um lado as diferenças podem ser enriquecedoras, mas também pode ser negativo, principalmente quando elas mal se conhecem (Herbsleb et. al., 2001).

**Idioma:** em ambientes de DDS, muitas vezes o idioma falado pode ser diferente de um local para outro. E a comunicação através de um idioma que não é o seu idioma natural pode ser uma tarefa complicada se não for bem gerenciada (Favela et. al., 2001).

### 3. Estudo de Caso

#### 3.1. Método de Pesquisa

Esta pesquisa se caracteriza como um estudo predominantemente exploratório, sendo que o método de pesquisa principal foi o estudo de caso. O método de estudo de caso é adotado conforme proposto por Yin (2001). Por tratar-se de uma pesquisa qualitativa, devem-se ter claras as limitações deste tipo de pesquisa, principalmente no que se refere ao número de organizações estudadas (apenas uma), restringindo a generalização dos resultados obtidos. Os dados foram coletados através de entrevistas semi-estruturadas com questões abertas com os colaboradores da organização.

A aplicação do estudo de caso, bem como a análise de dados, contou com uma ampla revisão teórica sobre o tema, de maneira que a convergência entre a teoria e prática pudesse ser observada no estudo realizado. O instrumento de pesquisa foi aplicado em caráter de pré-teste com três colaboradores da organização, e os comentários feitos, utilizados para aprimorar o questionário. Foram analisados cinco projetos em três segmentos distintos da unidade de desenvolvimento, incluindo projetos de melhorias de sistemas nas áreas de produção, manufatura e vendas.

#### 3.2. Caracterização da Organização

A organização é uma unidade de desenvolvimento de software de uma grande empresa norte-americana cujo objetivo é o desenvolvimento de computadores e de periféricos. Localizada no Parque Tecnológico da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) – TECNOPUC, em Porto Alegre, Rio Grande do Sul, esta unidade possui mais de 300 colaboradores trabalhando em torno de 90 projetos que atendem as

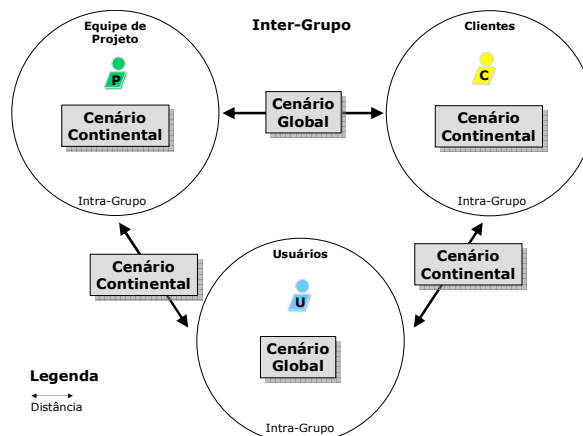


diversas fábricas da organização globalmente. Este centro foi criado em 2000 com os incentivos da Lei de Informática Brasileira que estimula as organizações localizadas no país a investir parte dos seus ganhos em pesquisa e desenvolvimento, fornecendo isenção ou redução de taxas (IPI) em produtos manufaturados de acordo com o Processo Produtivo Básico brasileiro (PPB). Atua em um ambiente de desenvolvimento global de software, uma vez que é uma das quatro unidades de desenvolvimento de software da empresa. A maior interação desta unidade é com a matriz, que está localizada nos Estados Unidos, e é responsável pela demanda dos projetos.

### 3.3. Classificação do Nível de Dispersão da Organização

Visando obter um entendimento de como ocorre o processo de desenvolvimento de software neste ambiente distribuído, é necessário entender como a empresa está situada globalmente, visualizando a sua relação com as outras unidades de desenvolvimento, bem como os clientes e usuários que interagem com ela. Por este motivo, é necessário identificar o nível de dispersão existente. Quando se fala em nível de dispersão, Evaristo et. al. (2004) o define como sendo a distância percebida entre os membros de grupo ou entre diversos grupos.

Como existem diversos projetos sendo desenvolvidos na organização e cada projeto possui o seu próprio nível de dispersão (a configuração e localização das equipes podem variar de um projeto para o outro), a classificação proposta a seguir considera o maior nível de dispersão para cada um dos atores (equipe de projeto, clientes e usuários). Entende-se que o nível de dispersão, ou a maior distância que poderá existir entre os atores na organização em geral é determinado pelo maior nível de dispersão existente. Neste caso, a figura 2 apresenta o nível de dispersão da organização, de acordo com a média observada nos cinco projetos estudados, e utilizando a representação proposta em Prikladnicki et. al. (2004).



**Figura 2 – Nível de Dispersão da Organização**

Esta representação permite a classificação do nível de dispersão entre os principais atores envolvidos em um ambiente distribuído. No cenário estudado, foram considerados a equipe de projeto, os clientes e os usuários. Em relação aos níveis de dispersão, o cenário continental indica que os atores estão distribuídos em países diferentes, mas em um mesmo continente. Já o cenário global, indica uma distância em continentes diferentes. Desta forma, nos projetos estudados, a equipe de projeto (P) estava distribuída em um cenário continental, localizada nos Estados Unidos e no Brasil.



Os clientes (C) estavam distribuídos também em um cenário continental, localizados em países da América Sul, América Central e América do Norte. Os usuários (U) estavam distribuídos globalmente em países como Brasil, Estados Unidos, Índia, entre outros. Apesar de haver esta dispersão, a comunicação entre equipe de projeto e os clientes era feita de forma global, tendo a equipe localizada no Brasil interagindo com clientes localizados nos Estados Unidos e Índia. A equipe de projeto localizada nos Estados Unidos apenas interagiu com a equipe de projeto localizada no Brasil. Por sua vez, a comunicação entre a equipe de projeto e os usuários era feita de forma continental, pois normalmente a equipe do Brasil se comunicava com os usuários dos Estados Unidos.

### 3.4. Resultados Observados

Após a coleta e análise dos dados, foi possível observar a presença e o comportamento dos aspectos não-técnicos identificados anteriormente. A tabela 2 recupera as informações da tabela 1, acrescentando os aspectos encontrados no estudo realizado.

**Tabela 2. Aspectos não-técnicos identificados**

| # | Aspectos não-técnicos | Fonte                    | Autor  |
|---|-----------------------|--------------------------|--|
| 1 | Confiança             | Teoria<br>Estudo de Caso | Kiel, 2003<br>Burnell et. al., 2002  |
| 2 | Conhecimento          | Teoria<br>Estudo de Caso | Desouza et. al., 2004<br>Rus et. al., 2002   |
| 3 | Contexto              | Teoria                   | Kogut et. al., 2000  |
| 4 | Comunicação           | Teoria<br>Estudo de Caso | Burnell et. al., 2002<br>Herbsleb et. al., 2001<br>Layzell et. al., 2000<br>Herbsleb et. al., 1999 |
| 5 | Colaboração           | Teoria<br>Estudo de Caso | Layzell et. al., 2000<br>Brna, 1998  |
| 6 | Cooperação            | Teoria<br>Estudo de Caso | Layzell et. al., 2000<br>Brna, 1998  |
| 7 | Coordenação           | Teoria<br>Estudo de Caso | Herbsleb et. al., 1999   |
| 8 | Cultura               | Teoria                   | Evaristo et. al., 2004<br>Burnell et. al., 2002<br>Hsieh, 2006                                     |
| 9 | Idioma                | Teoria<br>Estudo de Caso | Favela et. al., 2001   |



Observa-se, através da análise da tabela 2, que a grande maioria dos aspectos não-técnicos citados na literatura estão sim presentes em ambientes de desenvolvimento de software global. O contexto e a cultura, apesar de presentes nas análises, não foram observados como elementos singulares dentro do projeto, estando, no entanto diretamente relacionado com diversos outros fatores não-técnicos. Foi possível identificar, indiretamente, a influência da cultura em relação à coordenação, cooperação, colaboração e confiança. Acredita-se que isto foi fortemente influenciado pela diferença cultural entre os brasileiros e os americanos. O contexto também aparece como um elemento relacionado ao idioma, sendo que para os brasileiros a compreensão do que precisava ser executado acabava muitas vezes ficando comprometida.

Adicionalmente, outros elementos foram identificados na análise do estudo de caso, quais sejam: o poder – relacionado à tomada de decisão – e o relacionamento interpessoal. O primeiro diz respeito à imposição de soluções da matriz para a unidade de desenvolvimento estudada (ferramentas que deveriam ser utilizadas, processos e procedimentos a serem respeitados) devido a interesses políticos e a freqüente utilização dos mesmos recursos humanos em determinados projetos (devido ao relacionamento pessoal existente e a exigência de uma das equipes distribuídas em um caso específico). Além disso, alguns entrevistados mencionaram situações em que os clientes localizados nos Estados Unidos tinham a percepção de serem responsáveis por todas as decisões do projeto (envolvendo decisões sobre a arquitetura do sistema, padrões técnicos, entre outros). Muitas vezes a equipe do Brasil precisava respeitar todas estas decisões, sem ter um grande poder de tomada de decisão. A partir do estudo, pode ser observado que este elemento (poder) tem impacto direto na motivação dos participantes do projeto. De forma indireta, à medida que aumenta o poder de decisão, também aumenta a crença dos clientes de que a unidade distribuída está realizando um trabalho com qualidade.

Em relação ao relacionamento interpessoal, ele foi identificado devido à dificuldade de algumas pessoas em construir uma relação produtiva com integrantes de equipes distribuídas. Mesmo sendo um aspecto diretamente relacionado com os outros identificados, optou-se por separá-lo. Percebeu-se que, mesmo com um problema cultural, de comunicação, ou de outra natureza, não necessariamente estaria caracterizada a existência de um problema de relacionamento interpessoal.

O conhecimento foi um aspecto bastante citado, envolvendo o conhecimento das pessoas, do método de trabalho e de padrões existentes. Com relação ao contexto, identificaram-se alguns casos onde integrantes das equipes não percebiam que estavam interagindo com pessoas localizadas em outros países, e que era importante compartilhar o contexto em que estavam. A comunicação foi um dos aspectos mais citados, principalmente a forma de se comunicar, visto que a equipe brasileira muitas vezes tinha dificuldades relacionadas com o idioma. A coordenação e a cooperação também foram aspectos identificados no estudo de caso realizado. O primeiro foi identificado a partir de dificuldades que os gerentes de projeto disseram ter não apenas em gerenciar tecnicamente um determinado projeto, mas também em gerenciar a interação das equipes. Já a cooperação foi identificada na maioria das atividades, onde cada equipe possuía uma responsabilidade dentro de um mesmo projeto, e existia uma necessidade de cooperação.





#### 4. Discussão

A observação dos aspectos não-técnicos citados anteriormente em um caso real, e a identificação de outros possíveis aspectos não indica que eles sejam ou serão sinônimos de dificuldades dentro dos projetos. É importante deixar claro o relacionamento dos aspectos não-técnicos apresentados com o desenvolvimento global de software. De certa forma, os aspectos não-técnicos são muitas vezes tratados como riscos dentro de alguns projetos, sendo subestimados pelo gerente e pela própria equipe, ocasionando atrasos e problemas na comunicação dos envolvidos. Em outras situações, observa-se que os aspectos não-técnicos são apenas remediados quando existe um impacto no projeto, não havendo uma evolução construtiva referente ao seu impacto nos *stakeholders*. Desta forma, observa-se uma oportunidade significativa em relação à manutenção preventiva destes aspectos.

Considerando os aspectos técnicos existentes, é visível para uma organização ou uma equipe de projeto a necessidade de existir uma preocupação para com a correta utilização dos recursos, ferramentas, processos, metodologias e artefatos para desenvolver software de forma global. A equipe de uma forma geral deve estar preparada para atuar tecnicamente no projeto. Mas não foi visível a necessidade de se preocupar com a maneira de ser das pessoas, com a sua forma de aprender, de interagir, de se comunicar, enfim, com os aspectos não-técnicos existentes. E em ambientes globais de desenvolvimento isto se faz tanto necessário quanto aprender o processo de desenvolvimento que será utilizado, uma técnica ou uma ferramenta. Observou-se, portanto a existência reacionária para os aspectos não-técnicos, sem a existência significativa de trabalhos de prevenção e manutenção destes. É possível enumerar duas ocorrências onde foi possível observar a prática reacionária:

**Situação 1:** A comunicação entre um programador e seu líder de desenvolvimento ocorria em reuniões semanais de trinta minutos. O Líder de desenvolvimento muitas vezes estava localizado nos Estados Unidos e não dispunha de tempo para atender o programador. Por este motivo, a comunicação era amplamente prejudicada, sendo que ao final do projeto o software entregue não atendia a 75% do que havia sido demandado pelo cliente.

Solução tomada (reacionária): As pessoas tiveram que realizar horas extras a fim de atender toda a exigência do cliente.

Solução a ser proposta (preventiva): Aumentar o tempo de reunião, bem como a periodicidade de encontro entre o programador e seu líder. Trabalhar juntamente com o gerente do líder as prioridades entre os clientes, prestando desta forma, um monitoramento constante nas atividades do programador.

**Situação 2:** Devido a um problema com a interpretação do idioma, um gerente de projeto americano sentiu-se ofendido com a conversa realizada entre ele e dois analistas de negócio. Por este motivo, começou a criticá-los perante os seus superiores sem utilizar a argumentação correta (relativo ao problema do idioma). Ambos os analistas foram advertidos e se voltaram contra o gerente de projeto americano. Isto ocasionou no atraso do projeto.

Solução tomada (reacionária): Uma nova equipe foi estruturada, com analistas diferentes e um novo trabalho teve de ser realizado.



Solução a ser proposta (preventiva): Trabalhar o idioma em relação ao gerente de projeto e aos analistas e proporcionar reuniões informais visando trabalhar as desavenças ou possíveis conflitos entre os membros da equipe. Proporcionar treinamentos que procurem minimizar comportamentos exagerados por ambas às partes.

**Situação 3:** Devido a um feriado nacional no Brasil, a equipe de projeto localizada nos Estados Unidos não conseguiu encontrar parte da equipe no Brasil, e isto acabou criando uma situação bastante delicada entre as equipes distribuídas, o gerente do projeto (que estava nos Estados Unidos) e os gerentes superiores.

Solução tomada (reacionária): A partir desta dificuldade, diversos conflitos ocorreram, a confiança entre as equipes caiu significativamente, e o gerente precisou de diversas interações para resolver o problema, que foi causado por falta de compartilhamento de contexto entre as equipes.

Solução a ser proposta (preventiva): Além de compartilhar datas de feriados em cada contexto distribuído, os colaboradores podem enviar mensagens comunicando quando estarão ausentes e o devido motivo. Apesar de ser uma ação simples, muitas vezes é ignorada, e a sua falta é bastante sentida em um cenário global.

Neste sentido, e ilustrado pelas situações citadas, o que se propõe é a criação e utilização de recursos que possibilitem trabalhar de forma preventiva os problemas e dificuldades que surgem em decorrência dos aspectos não-técnicos. Estes recursos estão diretamente ligados aos objetivos e a importância que a organização dá para estes aspectos. Sugerem-se, inicialmente, a criação de treinamentos específicos para trabalhar os chamados *soft skills* (habilidades não-técnicas) e o incentivo ao compartilhamento de informações capturadas nos projetos. Isso possibilita o enriquecimento do conhecimento de cada indivíduo e da organização como um todo. O gerenciamento de expectativas também é tido como potencial recurso a ser utilizado, principalmente quando não se conhece a equipe fisicamente distante.

Em suma, em um projeto hipotético, quando a equipe está disposta a trabalhar a prevenção em detrimento a um problema ocorrido, existem boas chances de o projeto não ser atingido diretamente pelas conseqüências de uma má gestão de problemas e conflitos. Ela só será refletida ao término ou durante a reunião final da equipe. Em contrapartida, quando a mesma equipe subestima a importância da prevenção, chega um momento onde é necessário lidar com os problemas, podendo ocasionar atrasos, conflitos, e eventos inesperados.

Deste modo, ao invés de tratar os aspectos não-técnicos apenas como riscos subestimados em projetos de desenvolvimento, propõem-se a utilização de treinamentos e o compartilhamento do conhecimento prévio. É possível planejar e investir esforço em atividades de prevenção, que podem ser conduzidas de forma simultânea nos projetos da organização, contribuindo para minimizar os problemas decorrentes dos aspectos não-técnicos envolvidos. Mas sabe-se que problemas podem ocorrer mesmo investindo na prevenção. O que se busca não é a eliminação completa dos problemas, mas sim a oportunidade de diminuir a chance de ocorrência, dando a devida importância aos aspectos não-técnicos e reconhecendo-os como possíveis fontes de conflitos e problemas em ambientes de desenvolvimento global de software. Finalmente, é possível



estender a prevenção para todas as fases do desenvolvimento de software, passando pela fase de planejamento, desenvolvimento e pós-produção.

## 5. Considerações Finais

Este artigo apresentou a análise de resultados de um estudo de caso onde foi possível observar, em um caso prático, alguns dos principais aspectos não-técnicos da engenharia de software em ambientes de desenvolvimento global identificados na literatura. Procurou-se também propor uma solução ao gerenciamento destes aspectos do ponto de vista de ciclo de vida de projeto de desenvolvimento de software, partindo de uma abordagem reativa para uma abordagem preventiva.

Assim como o processo de desenvolvimento de software tem se tornado mais complexo, considerando-se o escopo das atividades e o número de artefatos envolvidos, a distribuição das equipes no tempo e no espaço tem tornado os projetos de DDS cada vez mais comuns (Meyer, 2006). Entretanto, o trabalho em ambientes de desenvolvimento distribuído de software é claramente mais problemático do que em ambientes centralizados. O valor da interação social não deve ser subestimado. A construção de confiança entre as equipes distribuídas deve ser facilitada e o trabalho na prevenção de dificuldades e problemas decorrentes dos aspectos não-técnicos devem ser valorizados. Os riscos técnicos e tecnológicos estão sempre presentes, e estudos relacionados com os aspectos técnicos têm sido amplamente divulgados.

Desta forma, o que pôde ser identificado neste estudo foi uma grande dificuldade em lidar com os aspectos não-técnicos presentes no processo de desenvolvimento de software em ambientes de desenvolvimento global. E isto muitas vezes pode comprometer as atividades técnicas que deveriam ser realizadas. Mesmo tendo o conhecimento de sua existência, os projetos ainda subestimam estes aspectos, tratando-os como meros coadjuvantes do processo de desenvolvimento, o que já foi também diagnosticado em fóruns internacionais e de reconhecida importância na área de engenharia de software (Herbsleb, 2005).

## 6. Referências Bibliográficas

- Brna, P. (1998), “Modelos de Colaboração”, In: IV WIE, Congresso da SBC, em inglês.
- Burnell, L. J., Priest, J. W., Durrett, J. R. (2001), “Teaching Distributed Multidisciplinary Software Development”, *IEEE Software*, 19(5), pp. 86-93.
- Carmel, E., Tija, P. (2005), “Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce”, UK: Cambridge.
- Carmel, E. (1999), “Global Software Teams – Collaborating Across Borders and Time-Zones”, EUA: Prentice Hall.
- Damian, D., Moitra, D. (2006), “Guest Editors' Introduction: Global Software Development”, *IEEE Software*. 23(5), pp. 17-19.
- Desouza, K. C., Evaristo, J. R. (2004), “Managing Knowledge in Distributed Projects”, *Communications of the ACM*, 47(4), pp. 87-91.
- Evaristo, R., Scudder, R. & Desouza, K. (2004), “A Dimensional Analysis of Geographically Distributed Project Teams: A Case Study”, *Journal of Engineering and Technology Management*. 21(3), pp. 75-189.



- Favela, J., Peña-Mora, F. (2001), “An Experience in Collaborative Software Engineering Education”, *IEEE Software*, 18(2), pp. 47-53.
- Herbsleb, J. D. (2005), “Beyond Computer Science”. In: 27th International Conference on Software Engineering (ICSE), 2005, St. Louis, Missouri, EUA, pp. 23-27.
- Herbsleb, J., Moitra, D. (2001), “Guest Editors' Introduction: Global Software Development”, *IEEE Software*. 18(2), pp. 16-20.
- Herbsleb, J.D., Grinter, R. (1999), “Splitting the organization and integrating the code: Conway's Law revisited”, In: 21st International Conference on Software Engineering (ICSE), pp. 85-95.
- Hsieh, Y. (2006), “Culture and Shared Understanding in Distributed Requirements Engineering”. In: *IEEE International Conference on Global Software Engineering*, Florianópolis, Brasil, pp. 101-105.
- Karolak, D. W. (1998), “Global Software Development – Managing Virtual Teams and Environments”, Los Alamitos: IEEE Computer Society.
- Kiel, L. (2003), “Experiences in Distributed Development: A Case Study”, In: *Workshop on Global Software Development at ICSE*, Portland, Oregon.
- Kogut, B., Meitu, A. (2000), “The Emergence of E-Innovation: Insights from Open Source Software Development”. Working Paper, The Wharton School, University of Pennsylvania.
- Layzell, P., Breneton, O. P., French, A. (2000), “Supporting Collaboration in Distributed Software Engineering Teams”, In: *Seventh Asia-Pacific Software Engineering Conference*.
- Meyer, B. (2006), “The Unspoken Revolution in Software Engineering”, *IEEE Computer*, 39(1), pp. 124, pp. 121-123.
- Prikladnicki, R., Audy, J. L. N. (2005), “Os Aspectos Não-Técnicos Intervenientes no Desenvolvimento Distribuído de Software”, In: *I WOSSES – Workshop Um Olhar Sociotécnico sobre a Engenharia de Software*, Rio de Janeiro, Brasil.
- Prikladnicki, R., Lopes, L., Audy, J. L. N., Evaristo, R., (2004), “Desenvolvimento Distribuído de Software: Um Modelo de Classificação dos Níveis de Dispersão dos *Stakeholders*”, In: *I Simpósio Brasileiro de Sistemas de Informação*, Porto Alegre, Brasil.
- Rus, I. Lindvall, M. (2002), “Knowledge Management in Software Engineering”, *IEEE Software*, 19(3), pp. 26-38.
- Sangwan R. et. al. (2007), “Global Software Development Handbook”, USA: Auerbach Publications.
- Sengupta, B., Chandra, S., Sinha, V. (2006), “A Research Agenda for Distributed Software Development”. In: 28<sup>th</sup> ICSE, Xangai, China.
- Sommerville, I. (2003), “Engenharia de software”, 6<sup>a</sup> ed., Addison Wesley, 2003.
- Yin, Robert. (2001), “Estudo de Caso: planejamento e métodos”. SP: Bookman.

---

<sup>i</sup> Estudo realizado pelo grupo de pesquisa em Desenvolvimento Distribuído de Software, do PDTI, financiado pela Dell Computadores do Brasil Ltda., com recursos da Lei Federal Brasileira nº 8.248/91.