

MuNDDoS - Um Modelo de Referência para Desenvolvimento Distribuído de Softwareⁱ

Rafael Prikladnicki¹, Jorge Luis N. Audy¹

¹Faculdade de Informática (FACIN)

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

90.619-900 – Porto Alegre – RS – Brasil

e-mail: [rafael, audy]@inf.pucrs.br

Resumo

Atualmente, é cada vez mais significativo o número de empresas que estão distribuindo seus processos de desenvolvimento de software ao redor do mundo, visando ganhos de produtividade, redução de custos e melhorias na qualidade. Por isso, o desenvolvimento distribuído tem atraído um grande número de pesquisas na área de engenharia de software nos últimos anos. Os engenheiros de software têm reconhecido a grande influência desta nova forma de trabalho e estão em busca de modelos que facilitem o desenvolvimento de software com equipes geograficamente dispersas. Além dos engenheiros, os gerentes e os executivos têm enfrentado desafios e dificuldades em diferentes níveis, envolvendo fatores técnicos e não-técnicos. Neste sentido, este artigo tem como objetivo propor um modelo de referência para desenvolvimento distribuído de software. O principal método de pesquisa utilizado é o estudo de caso e a base empírica da pesquisa envolveu duas unidades de desenvolvimento de software de duas empresas multinacionais de grande porte localizadas no Brasil. A pesquisa, resultado de uma dissertação de mestrado, contribui no sentido de apresentar dados empíricos e propor um modelo de referência para a área.

Abstract

More than a decade ago, seeking lower costs and access to skilled resources, many organizations began to experiment with remotely located software development facilities. Economic forces are relentlessly turning national markets into global markets and spawning new forms of competition and cooperation that reach across national boundaries. This change is having a profound impact not only on marketing and distribution but also on the way products are conceived, designed, constructed, tested, and delivered to customers. The number of organizations distributing their software development processes worldwide aiming at heightened profit and productivity as well as cost reduction and quality improvements keeps increasing. Engineers, managers, and executives face formidable challenges on many levels, from the technical to the social and cultural. This way, the purpose of this paper is to propose a reference model for distributed software development. The research method is case study and the empirical base involves two software development centers from two multinational organizations located in Brazil. The research, resulted from a master dissertation, contributes presenting empirical data and proposing a reference model for this area.

1. Introdução

Nos últimos anos pode-se perceber um grande avanço em direção à globalização dos negócios. Na área de desenvolvimento de software isto não é diferente. O software tem se tornado um componente estratégico para diversas áreas de negócio. Neste sentido, para as organizações que buscam sucesso, é clara a necessidade do uso da Tecnologia da Informação (TI) como diferencial competitivo [1].

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. Tem se tornado cada vez mais custoso e menos competitivo desenvolver software no mesmo espaço físico, na mesma organização ou até mesmo no mesmo país. Por outro lado, o avanço da economia, a sofisticação dos meios de comunicação e a pressão por custos têm incentivado o investimento maciço no desenvolvimento distribuído de software (DDS). Mesmo que a ES ainda não seja uma disciplina madura [2], as melhorias nas ferramentas e métodos nas últimas décadas têm permitido que grupos de diferentes localidades e culturas, com diferentes expectativas possam formar uma equipe para trabalhar em projetos distribuídos. Por isso, visando a redução de custos, maior qualidade no processo de desenvolvimento e a possibilidade de obter recursos em âmbito global [3], muitas organizações começaram a investir em DDS. 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 DDS tem atraído um grande número de pesquisas na área de engenharia de software [1, 2, 4, 5]. Os engenheiros de software têm reconhecido a grande influência desta nova forma de trabalho no seu dia-a-dia e estão em busca de modelos que facilitem o desenvolvimento de software com equipes geograficamente distantes. São freqüentes os esforços que os pesquisadores têm feito no intuito de entender os fatores que permitem organizações multinacionais ou virtuais a obterem sucesso trabalhando através das fronteiras físicas e culturais dos países. Também são freqüentes os diversos desafios e dificuldades que os engenheiros, gerentes e executivos têm enfrentado em diferentes níveis de complexidade [1].

Alguns estudos [1, 4, 6, 7] observam uma dificuldade de implantar, executar e monitorar projetos em ambientes de DDS devido a fatores não-técnicos tais como fatores sociais, culturais, comportamentais, psicológicos, lingüísticos e políticos (confiança, diferenças culturais, idioma, entre outros). Por outro lado, outros estudos [3, 5, 8, 9, 10] observam estas mesmas dificuldades devido a fatores técnicos (gerência de projeto, processo de desenvolvimento de software, complexidade e tamanho de projetos, tecnologia e infraestrutura de comunicação disponível, entre outros).

Esta área de pesquisa apresenta grandes desafios e muitas lacunas a serem preenchidas. Esta pesquisa procurou estudar as características do DDS, sua operacionalização e os fatores críticos de sucesso envolvidos. Com isto, apresenta-se um modelo de referência para a área de forma a contemplar também novas características descobertas e analisadas nos estudos de caso desenvolvidos neste estudo. Desta forma, a questão de pesquisa que norteou este estudo foi: *Como os fatores envolvidos no DDS podem ser relacionados em um modelo de referência específico para esta área?*

O artigo está organizado em 7 seções. A seção 2 apresenta o referencial teórico. Na seção 3 descreve-se o método de pesquisa utilizado e a seção 4 apresenta o estudo de caso e os resultados encontrados. A seção 5 apresenta as lições aprendidas. A seção 6 apresenta o modelo de referência proposto. A seção 7 apresenta as considerações finais.

2. Referencial Teórico

2.1. Desenvolvimento Distribuído de Software

Na última década observou-se um grande investimento na conversão de mercados nacionais em mercados globais, criando novas formas de competição e colaboração [1]. Entretanto, o mercado global de software vinha passando por diversas crises. Por um lado, um grande número de falhas em projetos. De outro, uma crescente demanda, atingida pela escassez de recursos capacitados. Nesse ambiente, muitas organizações encontraram no DDS uma alternativa, experimentando o desenvolvimento em locais remotos.

Considerando sua recente aplicação e grande abrangência, o DDS possui uma diversidade de conceitos que auxiliam sua caracterização. Dentro do contexto de desenvolvimento distribuído podem ser aplicados conceitos de equipes globais [11] e organizações virtuais [2], por exemplo, ampliando a extensão do conhecimento envolvido. Segundo Marquardt [11], uma equipe global refere-se a um grupo de pessoas de diferentes nacionalidades que trabalham unidas em um projeto comum, através de culturas e fusos-horários distintos, por um extenso período de tempo. Com relação às organizações virtuais, Karolak [2] as define como entidades caracterizadas por realizar partes de um projeto em locais distintos, comportando-se como se estivesse no mesmo local.

Quando a distância física entre os atores em um ambiente de DDS envolve mais de um país, Karolak [2] define uma instância do DDS chamada de desenvolvimento global de software (*Global Software Development – GSD*). O GSD é instanciado através de equipes globais de desenvolvimento de software (*Global Software Teams*), que Carmel [6] define como sendo um grupo de pessoas distribuídas em dois ou mais países, colaborando ativamente em um projeto comum. Neste sentido, o DDS tem sido caracterizado pela colaboração e cooperação entre departamentos de organizações e pela criação de grupos de desenvolvedores que trabalham em conjunto, localizados em cidades ou países diferentes, distantes temporal e fisicamente. As figuras 1 e 2 ilustram a mudança para o novo cenário.

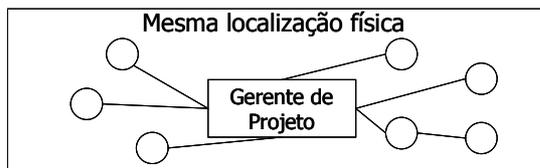


Figura 1. Equipe centralizada [4]

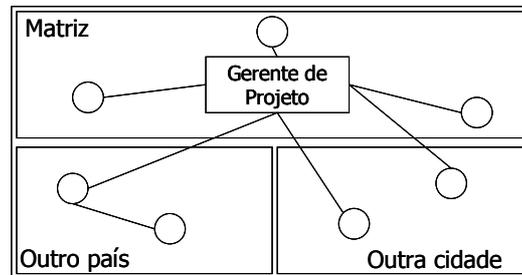


Figura 2. Equipe distribuída [4]

Apesar de muitas vezes este processo ocorrer em um mesmo país, em regiões com incentivos fiscais ou de concentração de massa crítica em determinadas áreas, algumas empresas, visando maiores vantagens competitivas, buscam soluções globais. Neste contexto, surgem os conceitos de *offshore outsourcing* (contratação de uma organização externa, localizada em um outro país) e *offshore insourcing* (contratação de uma subsidiária da própria organização, também localizada em um outro país), o que potencializa os problemas e os desafios existentes no DDS. Diversos fatores têm contribuído para o crescimento do DDS, entre eles:

- A necessidade de recursos globais para serem utilizados a qualquer hora;
- As vantagens de estar perto do mercado local, incluindo o conhecimento dos clientes e as condições locais para explorar as oportunidades de mercado;
- A grande pressão para o desenvolvimento *time-to-market*, utilizando as vantagens proporcionadas pelo fuso horário diferente, no desenvolvimento conhecido como *follow-the-sun* (24 horas contínuas, contando com as equipes fisicamente distantes).

Segundo Carmel [6], não existem dúvidas para qualquer profissional que trabalha na área de ES que tanto o desenvolvimento de software tradicional quanto o distribuído possuem diversas dificuldades. As principais características que os diferenciam são: **dispersão geográfica** (distância entre equipe de projeto, clientes e usuários, por exemplo); **dispersão temporal** (diferenças de fuso-horário); e **diferenças culturais** (incluindo idioma, tradições, costumes, normas e comportamento). Estas diferenças refletem-se em diversos fatores [1]. Entre eles, destacam-se **questões estratégicas** (decisão de desenvolver ou não um projeto de forma distribuída, tendo por base análises de risco e custo-benefício); **questões culturais** (valores, princípios, etc. entre as equipes distribuídas); **questões**

técnicas (fatores relativos à infra-estrutura tecnológica - redes de comunicação de dados, plataformas de hardware, ambiente de software - e ao conhecimento técnico necessário - processo de desenvolvimento, etc. - para o desenvolvimento dos projetos distribuídos); e **questões de gestão do conhecimento** (fatores relativos à criação, armazenamento, processamento e compartilhamento de informações nos projetos distribuídos).

Várias ferramentas e ambientes têm sido desenvolvidos ao longo dos últimos anos para ajudar no controle e coordenação das equipes de desenvolvimento que estão inseridas neste tipo de ambiente [8, 9]. Muitas destas ferramentas estão focadas no suporte aos procedimentos de comunicação formal tais como elaboração de documentos, processos automatizados e canais de comunicação não interativos.

2.2. Trabalhos Relacionados

O DDS apresenta um grande impacto na forma como os produtos são concebidos, desenvolvidos, testados e entregues aos clientes [1]. Assim, a estrutura necessária para o suporte desse tipo de desenvolvimento se diferencia da utilizada em ambientes centralizados. Diferentes características e tecnologias se fazem necessárias, crescendo a importância de alguns detalhes antes não percebidos. Atualmente encontram-se disponíveis na literatura da área alguns estudos que propõem modelos de referência para o DDS. A seguir apresentam-se três abordagens existentes [2, 6, 12].

2.2.1. Karolak, 1998. O trabalho de Karolak [2] aborda o DDS seguindo, segundo o autor, o ciclo de vida tradicional de um projeto de desenvolvimento de software (Visão e Escopo, Requisitos, Modelagem, Implementação, Teste, Entrega e Manutenção). O autor propõe um modelo para desenvolver projetos de DDS abrangendo um conjunto de atividades ao longo do ciclo de vida. O modelo considera atividades desde a estratégia a ser adotada para atuar em DDS (empresas globais, parcerias estratégicas, entre outros) até a manutenção do software desenvolvido, mas não se aprofunda na análise de como implementar cada uma.

2.2.2. Carmel, 1999. Em seu trabalho, Carmel [6] aborda a formação de equipes globais de desenvolvimento de software e os principais fatores que devem ser considerados ao montar uma equipe para um projeto distribuído (Figura 3). O autor sugere a existência de cinco fatores que podem levar uma equipe distribuída ao fracasso: comunicação ineficiente, falta de coordenação, dispersão geográfica, perda do espírito de equipe e diferenças culturais, chamadas de forças centrífugas. Além disso, sugere a existência de seis fatores que podem levar a equipe ao sucesso: infra-estrutura de comunicação, arquitetura do produto, construção de equipe, técnicas de gerência e tecnologia de colaboração, chamados de forças centrípetas.

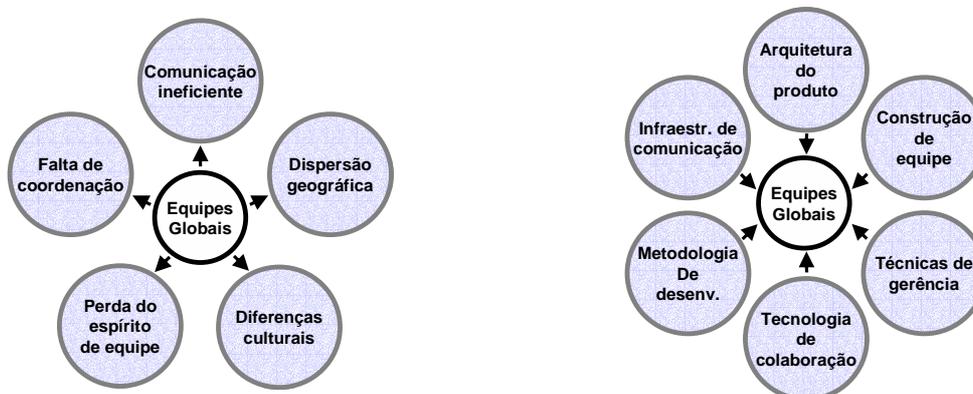


Figura 3. Forças centrífugas e centrípetas de equipes globais [6].

2.2.3. Evaristo, 2003. A pesquisa realizada por Evaristo [12] propõe a existência de algumas dimensões no contexto de equipes de projetos distribuídos. Estas dimensões (Figura 4) auxiliam no entendimento dos problemas, vantagens e desvantagens deste tipo de ambiente e afetam diretamente a performance destes projetos.

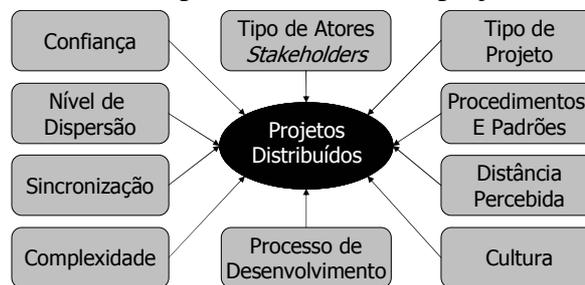


Figura 4. As dimensões de Evaristo [12]

2.2.4. Análise Crítica. As abordagens descritas anteriormente destacam os projetos distribuídos sob três perspectivas diferentes. Enquanto Karolak [2] propõe um modelo específico para projetos de DDS, tendo como guia um processo tradicional de desenvolvimento de software, Carmel [6] aborda a questão da formação de equipes globais de software, suas características e os principais desafios inerentes a formação de equipes de sucesso. Por sua vez, Evaristo [12] também aborda a questão da formação de equipes, mas relacionando com equipes distribuídas para qualquer tipo de projeto, não necessariamente projetos de DDS. Todas as abordagens apresentam características essenciais do DDS.

Outros autores também abordam algumas destas características [4, 5, 7, 8, 11, 13], ampliando os conceitos relacionados e aprofundando o estudo em pesquisas mais específicas. Modelos e métodos têm sido desenvolvidos para a área de DDS, tendo por objetivo identificar as principais dificuldades e propor soluções para minimizá-las. Diferentes modelos buscam formular estratégias de atuação na área. Mas nem todos compreendem os diversos fatores envolvidos ou atuam apenas como suportes conceituais às atividades executadas nos projetos com equipes geograficamente distribuídas. Neste sentido, tem sido necessária a criação de processos específicos para atuar neste cenário, procurando considerar os fatores que interferem no sucesso dos projetos de DDS.

Apesar de diversos estudos estarem sendo desenvolvidos na área de DDS nos últimos anos, a carência de uma base teórica mais estável e consistente na área direcionou o estudo de modo a obter dados empíricos relacionados ao tema de estudo. A ausência de modelos de referência específicos para DDS, em conjunto com as demandas existentes nas organizações, foi a principal motivação para o desenvolvimento deste estudo. Sendo assim, nas seções a seguir serão explorados os problemas e desafios encontrados na literatura, relacionando-os à classe de problemas que surge com o DDS.

3. Método de Pesquisa

Muito embora a ampla revisão teórica desenvolvida, não se tem conhecimento de que o problema apresentado tenha sido abordado sob a mesma perspectiva. Assim, esta pesquisa se caracteriza como um estudo predominantemente exploratório, sendo que o método de pesquisa principal foi o estudo de caso. Neste estudo, pode-se justificar o uso de métodos qualitativos pelo fato de ele envolver o estudo de DDS no seu contexto real, com a compreensão do estado da arte em situações onde a prática se antecipa à teoria.

O método de estudo de caso foi adotado conforme proposto por Yin [14]. Neste estudo de caso foram identificados como elementos de análise dificuldades, soluções, e fatores críticos de sucesso do DDS. Utilizou-se como instrumento de pesquisa um roteiro para uma entrevista semi-estruturada, com questões abertas. Por tratar-se de uma pesquisa

qualitativa, deve-se ter claras as limitações deste tipo de pesquisa, principalmente quanto ao número de empresas estudadas, restringindo a generalização dos resultados obtidos.

O estudo foi dividido em quatro fases:

Fase 1: estudo da base teórica da área.

Fase 2: desenvolvimento de um estudo de caso em duas organizações atuando em DDS, visando levantar dados sobre os fatores presentes em cada organização.

Fase 3: realização de análises críticas sobre os dados coletados nas fases anteriores (teórica e empírica). Nesta fase gerou-se uma descrição da área de DDS, a avaliação da importância dos fatores encontrados e as práticas identificadas nos estudos de caso.

Fase 4: definição de um modelo de referência para DDS, contendo as informações obtidas da base teórica e a consolidação dos dados extraídos do estudo de caso.

A consolidação dos resultados envolveu três atividades essenciais para a riqueza dos dados coletados: descrição da área de DDS, fatores críticos de sucesso e práticas relatadas. Estas atividades culminaram nas lições aprendidas (apresentadas na seção 5).

4. Estudo de Caso

O estudo de caso foi desenvolvido em duas unidades de desenvolvimento de software, cada uma pertencendo a uma organização multinacional com filiais no Brasil e no exterior. A unidade de análise do estudo foi projetos de DDS, escolhidos de acordo com a caracterização distribuída dos mesmos. Ambas as organizações disponibilizaram acesso irrestrito aos procedimentos deste estudo, tanto no acompanhamento do processo, quanto no que se refere à documentação, tendo um forte apoio diretivo. A coleta de dados foi constituída por fontes primárias e secundárias. As fontes primárias foram constituídas de 22 entrevistas semi-estruturadas individuais em profundidade (11 em cada organização). Como complemento, foram utilizadas fontes secundárias, tais como a consulta e a coleta de documentos das organizações (a missão, os referenciais estratégicos, os processos de negócios, as atas de reuniões e a descrição de processos de desenvolvimento de software).

O critério inicial para a definição dos entrevistados (Tabela 1) centrou-se na unidade de análise e nos objetivos do estudo. Neste sentido, a população envolvida direta ou indiretamente constituía-se por integrantes das equipes dos projetos, gerentes de desenvolvimento, responsáveis pela área de qualidade e implantação dos processos de desenvolvimento de software na organização, além de um gestor de nível estratégico na organização. A amostra foi não probabilística, por conveniência, e procurou-se, em conjunto com a organização, uma representatividade dos diversos grupos envolvidos.

A técnica de análise de dados utilizada foi a de análise de conteúdos segundo Krippendorff [15]. Associada a ela foi utilizada a análise documental, para compreender as informações fornecidas pelos entrevistados. O instrumento de pesquisa foi desenvolvido a partir da teoria estudada e representada pelo protocolo de pesquisa desenvolvido para o estudo de caso. Foram desenvolvidos dois questionários, cada um explorando uma dimensão de informações que deveriam ser capturadas: dimensão organizacional, contendo informações da organização como um todo e dimensão de projetos, contendo informações relacionadas aos projetos selecionados para participar do estudo.

Tabela 1. Distribuição dos entrevistados

| Função | Quantidade | Dimensão | Organização |
|----------------------------|------------|-------------------------|-------------|
| Gerente de Desenvolvimento | 2 | Organizacional, Projeto | 1, 2 |
| Gerente de Projeto | 4 | Projeto | 1, 2 |
| Líder Técnico | 4 | Projeto | 1, 2 |
| Desenvolvedor | 4 | Projeto | 1, 2 |
| Analista de Qualidade | 1 | Projeto | 1 |
| Analista de processo | 1 | Projeto | 2 |

4.1. Organização 1

O primeiro estudo de caso (EC1) foi desenvolvido em uma unidade de desenvolvimento de software de uma organização de grande porte, chamada aqui de *Organização 1*, com sede em São Paulo. O estudo de caso foi desenvolvido na principal unidade de desenvolvimento de software. A unidade possui certificação ISO 9001 desde 1996 e nível dois de maturidade de desenvolvimento de software reconhecida pelo padrão SW-CMM desde 2002, com 80 colaboradores atuando no desenvolvimento de software. Com relação aos clientes, a unidade trabalha apenas com clientes externos à organização.

O processo de desenvolvimento de software é bem definido, seguindo as normas e padrões da ISO 9001 e do modelo de maturidade SW-CMM (nível dois). O processo é baseado em parte de outros processos conhecidos na comunidade acadêmica e empresarial, tais como RUP (*Rational Unified Process*) para o desenvolvimento de software e a metodologia do PMI (*Project Management Institute*) para o gerenciamento de projetos. Estes processos não são aplicados na sua totalidade, sendo adaptados à realidade da organização. Foram entrevistados integrantes dos seguintes projetos:

Projeto 1: O objetivo era desenvolver uma aplicação para uma grande empresa cuja matriz estava localizada nos Estados Unidos. Apesar de o projeto ter sido requisitado pela matriz, era também gerenciado pela filial localizada no Brasil. A equipe de projeto estava localizada no Brasil, em dois locais diferentes, enquanto que os clientes estavam localizados tanto no Brasil quanto nos Estados Unidos. Além disso, os usuários eram todos os funcionários da empresa, mas no projeto eles estavam sendo representados por pessoas tanto dos Estados Unidos quanto do Brasil.

Projeto 2: O objetivo era desenvolver um sistema para um banco de grande porte do Estado de São Paulo. Neste sentido, o banco contratou uma empresa que terceirizou o serviço, contratando a unidade de desenvolvimento estudada, caracterizando assim o banco como sendo o usuário, a empresa contratante como sendo o cliente e atuando também como parte da equipe de projeto e a empresa contratada como sendo o restante da equipe de projeto. Tanto a equipe de projeto, quanto os clientes e usuários estavam localizados no Brasil. No entanto, a equipe de projeto estava distribuída em duas cidades, enquanto que o cliente e os usuários atuavam na mesma localização física, distantes de parte da equipe de projeto. A outra parte da equipe de projeto atuava dentro do banco.

4.2. Organização 2

O segundo estudo de caso (EC2) foi desenvolvido em uma unidade de desenvolvimento de software de uma organização de grande porte, chamada aqui de *Organização 2*, com sede nos Estados Unidos. A unidade onde o estudo foi aplicado está localizada em uma cidade do Estado do Rio Grande do Sul. Ela possui mais de 180 colaboradores trabalhando em projetos que atendem as necessidades da área de TI da empresa (demanda interna). A unidade possui nível dois de maturidade de desenvolvimento de software reconhecida pelo padrão SW-CMM desde 2003.

O processo de desenvolvimento de software é bem definido, seguindo as normas e padrões do modelo de maturidade SW-CMM (nível dois). O processo tem como base o MSF (*Microsoft Solution Framework*) para o desenvolvimento de software e a metodologia do PMI para o gerenciamento de projetos. Além disso, o RUP também é utilizado, dependendo das características dos projetos. Estes processos não são aplicados na sua totalidade, sendo adaptados à realidade da organização, desenvolvidos pelos próprios funcionários. Foram entrevistados integrantes dos seguintes projetos:

Projeto 1: O objetivo era desenvolver uma aplicação para gerenciar talentos dentro da empresa, e seria utilizado pela área de recursos humanos na matriz nos Estados Unidos. A equipe de projeto estava localizada no Brasil e nos Estados Unidos, enquanto que o cliente (departamento de recursos humanos) estava localizado nos Estados Unidos no mesmo espaço físico. Além disso, os usuários também estavam localizados nos Estados Unidos no mesmo espaço físico, mas distantes dos clientes.

Projeto 2: O objetivo era desenvolver uma aplicação para a área de manufatura da organização. A equipe de projeto estava fisicamente distante, localizada no Brasil. Os clientes e os usuários estavam localizados nos Estados Unidos, no mesmo espaço físico, mas distantes fisicamente entre si.

4.3. Resultados do Estudo de Caso

O estudo de caso realizado evidenciou diversos fatores do DDS nas duas organizações, atuando em projetos distribuídos em nível internacional. Neste sentido, esta seção identifica os principais fatores encontrados, resultado do esforço de consolidação para os elementos de análise estudados.

4.3.1. Dificuldades do DDS. De acordo com os fatores identificados na análise preliminar, pode-se dizer que as dificuldades do DDS, nas organizações estudadas, estão centradas na falta de uma padronização das atividades entre as equipes distribuídas, na dificuldade de compartilhar informações e na falta de um processo de desenvolvimento bem definido (refletindo nas atividades de engenharia de requisitos). Além disso, também se verificaram dificuldades em relação a barreiras de idioma e comunicação, diferenças culturais, contexto e aquisição de confiança entre as equipes distribuídas (Tabela 2).

Tabela 2. Dificuldades encontradas

| Dificuldades do DDS | Fonte |
|---|----------|
| Engenharia de requisitos | EC1, EC2 |
| Processo de desenvolvimento de software | EC1, EC2 |
| Gerência de configuração | EC1 |
| Gestão do conhecimento | EC1 |
| Barreiras de comunicação e idioma | EC1, EC2 |
| Diferenças culturais e contexto | EC2 |
| Confiança | EC1, EC2 |

4.3.2. Soluções encontradas para as dificuldades do DDS. Apesar de existirem diversas possibilidades de soluções aplicáveis visando minimizar os problemas do DDS, nas organizações estudadas as soluções se concentraram principalmente na necessidade de definir padrões de trabalho, investimento em planejamento e constante gerência de riscos, integração e aquisição de confiança entre as equipes, treinamento constante, um processo de desenvolvimento único e uma efetiva elicitação de requisitos (Tabela 3). Pode ser que, com as características atuais, não seria necessária uma melhoria na elicitação de requisitos, pois os requisitos já poderiam estar de acordo com as necessidades do usuário.

Tabela 3. Soluções Implementadas

| Soluções Encontradas | Fonte |
|--|----------|
| Planejamento e <i>engagement</i> | EC1, EC2 |
| Treinamento | EC1, EC2 |
| Padronização | EC1, EC2 |
| Investimento na gerência de riscos | EC1 |
| Processo de desenvolvimento de software comum | EC1, EC2 |
| Melhoria no processo de elicitação de requisitos | EC1, EC2 |
| Aquisição de confiança e integração | EC1, EC2 |

4.3.3. Fatores Críticos de Sucesso (FCS) para DDS. A identificação dos fatores críticos de sucesso está diretamente ligada com a forma de trabalho de uma determinada organização. Para uma mesma atividade podem existir diferentes fatores em cada organização, cada um ligado à estratégia adotada. Visando a consolidação dos resultados, identificou-se como FCS o gerenciamento das expectativas das equipes (definição clara de papéis e responsabilidades, documentos existentes, entre outros), integração das equipes, comunicação eficaz e *feedback* constante, definição e formalização de um processo único, investimento em treinamento, planejamento e clara definição do *engagement*, infraestrutura para suportar a interação distribuída e o acompanhamento do projeto por todos os membros das equipes, na medida do possível (Tabela 4).

Tabela 4. Fatores Críticos de Sucesso

| Fatores Críticos de Sucesso | Fonte |
|---|----------|
| Processo de desenvolvimento de software | EC1, EC2 |
| Treinamento | EC1, EC2 |
| Planejamento e <i>engagement</i> | EC1, EC2 |
| Infra-estrutura | EC2 |
| Gerenciamento de expectativas | EC2 |
| Integração das equipes | EC1, EC2 |
| Comunicação e <i>feedback</i> | EC1, EC2 |

5. Lições Aprendidas

O estudo de caso realizado nas organizações 1 e 2 ilustrou diversos fatores presentes no DDS (seção 4.3). Estes fatores, baseado no confronto entre a teoria e as descobertas empíricas, estão apresentados na forma de lições aprendidas, que são uma das bases de sustentação do modelo de referência proposto neste estudo. A seguir apresentam-se as lições identificadas, que estão detalhadas em [16].

Lição 1: A gerência de projetos e a gerência de riscos requerem um esforço e passos adicionais

A grande maioria dos gerentes de desenvolvimento e gerentes de projeto entrevistados citou que em projetos distribuídos as atividades envolvendo a gerência de projetos e principalmente a gerência de riscos tomam proporções maiores que em projetos tradicionais (centralizados), requerendo um maior esforço e alguns passos adicionais aos modelos tradicionais existentes. Entre estes passos, pode ser destacada a necessidade de integrar a gerência de risco nos níveis estratégico e tático com a gerência de risco em nível de projeto (decidir se um projeto pode ser desenvolvido de forma distribuída).

Lição 2: A existência de um processo de desenvolvimento de software único e bem definido responde por grande parte dos resultados obtidos em um projeto de desenvolvimento distribuído

Nos projetos onde não havia uma clara definição do processo de desenvolvimento, surgiram muitas dificuldades, relacionadas com o próprio processo (requisitos, gerência de configuração, teste, etc.) e outras relacionadas com dificuldades derivadas (comunicação, sincronização, confiança entre as equipes). Percebeu-se que um processo único e bem definido de acordo com o ambiente em que o projeto está sendo desenvolvido pode ser a solução para muitas dificuldades no DDS. Neste caso, geralmente as equipes de projeto devem optar por: forçar a padronização; criar um novo processo baseado nas melhores práticas encontradas em cada local distribuído; ou impor algumas regras em alto nível.

Lição 3: A gestão do conhecimento incentiva o compartilhamento de informações e estimula a aprendizagem por experiência

As entrevistas realizadas indicaram que um grande diferencial do DDS está ligado ao fato de que um investimento na gestão do conhecimento (seja através de ferramentas ou

atividades de incentivo ao compartilhamento de informação) estimula o aprendizado por experiência e minimiza certas dificuldades. Isto se deve ao fato de que muitas pessoas vivenciam certas situações e não compartilham, pois não são estimuladas para tal.

Um dos gerentes de projeto entrevistados disse que “apesar da existência de uma base histórica dos projetos na organização, não existe a cultura de utilizar esta informação de forma eficiente, por que as pessoas não estão aptas a utilizá-la, não querem utilizar ou simplesmente não foram treinadas para utilizar os benefícios de um repositório de dados”.

Lição 4: A engenharia de requisitos é o maior desafio do ponto de vista do processo de desenvolvimento de software

Todas as entrevistas realizadas com os gerentes de projetos e líderes técnicos apontaram para dificuldades nas atividades que envolvem a engenharia de requisitos. Em um dos projetos a instabilidade dos requisitos foi o maior problema, principalmente devido à distância entre as equipes, o que dificultava o entendimento e a convergência de idéias. Todos os projetos identificaram os requisitos como um grande desafio, envolvendo atividades desde a realização de reuniões até a formalização (documentação) dos requisitos que eram definidos, a rastreabilidade e controle dos mesmos.

Lição 5: A fase de planejamento é importante para a organização e a gerência de projetos de desenvolvimento distribuído de software

Identificou-se o planejamento como uma etapa fundamental para decidir e planejar o desenvolvimento de um projeto de forma distribuída. Sendo assim, acredita-se que a visão do processo de planejamento como sendo uma etapa preliminar a um conjunto de projetos de desenvolvimento de software é um fator decisivo também em projetos distribuídos. Um gerente de projeto disse que “quando você está trabalhando com projetos distribuídos, deve existir um planejamento formal antes de o projeto começar. Sem isto, existe uma grande possibilidade de descobrir riscos não considerados quando a decisão de desenvolver o projeto de forma distribuída foi tomada. E estes riscos podem variar desde restrições de segurança e disponibilidade de ambiente físico até complexidade dos requisitos e a experiência em projetos distribuídos”.

Lição 6: O investimento em gestão de equipes distribuídas minimiza as dificuldades relacionadas aos fatores não-técnicos

Uma das organizações estudadas investiu em treinamentos para a equipe localizada na unidade de desenvolvimento de software. Como resultado, as interações com parte da equipe distribuída (incluindo equipe de projeto, clientes e usuários) fluíram mais facilmente. Problemas identificados antes dos treinamentos passaram a ocorrer com menos frequência, indicando que a gestão das equipes distribuídas não pode ser menosprezada.

A importância do treinamento é reforçada por um gerente de projeto, que disse que “sempre haverá problemas no trabalho com pessoas distribuídas. Os problemas existirão até mesmo com pessoas em um mesmo espaço físico. Mas em um cenário global isto é mais crítico devido à dispersão geográfica e o fuso-horário. Por isso, o investimento em treinamento é essencial para minimizar o impacto das diferenças e ser mais efetivo. E este treinamento é realizado assim que a pessoa entra na empresa”.

Lição 7: Ferramentas de apoio atuam como facilitador

Ambas as organizações possuem estratégias para trabalhar com ferramentas globais, visando uma gestão de conhecimento e interação globais. Além disso, ferramentas de suporte à comunicação, tais como e-mail, vídeo conferência, tele conferência e bate-papo são frequentemente utilizadas. Um dos gerentes de projeto entrevistados disse que “existe uma ferramenta na organização para suportar a gestão de conhecimento de forma global, e a idéia de desenvolver esta ferramenta surgiu em uma reunião geral de toda a empresa. Por isso, a ferramenta foi desenvolvida como um projeto interno e agora se espera que ela auxilie na troca de informações entre as equipes distribuídas”.

Lição 8: Desenvolver software de forma distribuída é um processo que amadurece com o tempo

Segundo Paulk [17], maturidade significa o estado ou condição de pleno desenvolvimento, estado ou qualidade de maduro. Quando o modelo CMM foi proposto, ele foi estruturado e desenvolvido de uma forma que houvesse a representação da maturidade dos processos de uma organização através de níveis, identificando assim o quão maduros e desenvolvidos estariam estes processos.

A partir dos dados extraídos do estudo de caso, percebeu-se que existia uma diferença de maturidade entre as organizações com relação ao DDS. Por estar a mais tempo atuando em projetos distribuídos (4 anos), a *Organização 1* estava em um estágio mais avançado, sendo identificados problemas em diferentes níveis de dificuldade em relação à *Organização 2* (1 ano atuando em DDS). Neste sentido, com base nos dados coletados e das entrevistas realizadas, percebeu-se que o DDS também é um processo que amadurece com o passar do tempo. Por este motivo, a identificação da maturidade de projetos distribuídos complementa a literatura da área.

6. Modelo de Referência Proposto

Tendo em vista os trabalhos relacionados descritos na seção 2.2 e o estudo de caso desenvolvido e descrito na seção 4, esta pesquisa propõe um modelo de referência para DDS. Este modelo tem por objetivo identificar as características dos projetos em ambientes de DDS, e servir de apoio ao desenvolvimento de software realizado por equipes de trabalho heterogêneas e geograficamente dispersas.

O modelo (Figura 5) é composto por três ciclos (planejamento estratégico, planejamento tático/operacional e aprendizado), identificando o relacionamento entre eles. Além disso, oferece uma base para a condução de projetos de DDS, visando (1) facilitar a identificação de fraquezas, oportunidades e planejar melhorias para minimizar possíveis problemas, (2) garantir que projetos de DDS sejam viáveis com grupos de diferentes níveis de capacidade, e (3) aprimorar a capacidade da organização em projetos distribuídos. A seguir descreve-se o modelo proposto, denominado MuNDDoS – **Maturidade No Desenvolvimento Distribuído de Software** [18].

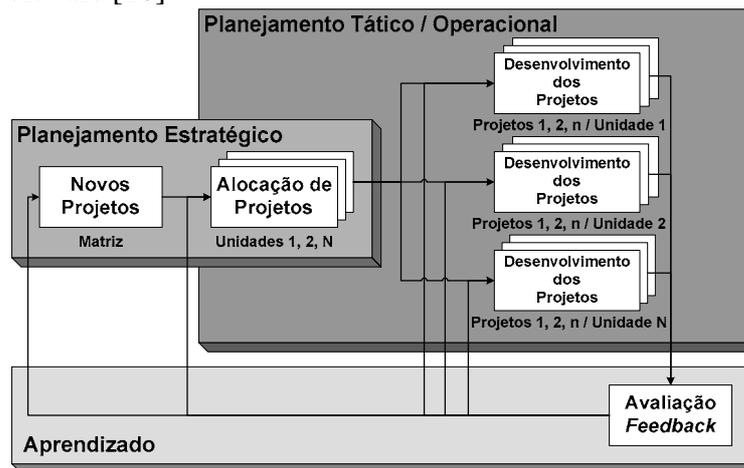


Figura 5. Modelo de Referência Proposto

6.1. MuNDDoS

O modelo de referência (MuNDDoS) foi elaborado para atuar como facilitador nos projetos de DDS [18]. Além disso, a forma como o modelo foi concebido permite a

identificação de fraquezas e de oportunidades de melhorias nos projetos. Para isso, o modelo sugere a existência de duas dimensões: organizacional e de projetos.

Ampliando a visão relativa ao processo de desenvolvimento de software, buscando adotar uma visão estratégica com relação ao processo, pode-se identificar a etapa de planejamento como sendo a primeira a ocorrer. Esta etapa envolve a definição das estratégias que deverão conduzir o processo de desenvolvimento como um todo, ao longo do tempo (dimensão organizacional). Pode-se identificar a etapa de planejamento como sendo preliminar a um conjunto de ciclos de projetos de desenvolvimento de software (dimensão de projetos) derivados do processo de planejamento.

Identificam-se dois ciclos de planejamento necessários para a gestão de projetos de DDS. O primeiro envolve o planejamento estratégico. Este ciclo é conduzido pela matriz e diz respeito à identificação e priorização de novos projetos a serem desenvolvidos, sejam eles projetos internos (de departamentos internos à organização) ou externos (projetos requisitados por clientes externos). Cabe aos participantes deste nível de planejamento buscar o alinhamento estratégico entre os objetivos de cada unidade distribuída e da matriz. O segundo ciclo envolve o planejamento tático-operacional, no âmbito da unidade de desenvolvimento de software distribuída. A transição entre os dois ciclos de planejamento ocorre exatamente na alocação dos projetos, envolvendo atividades de planejamento e definição dos projetos que serão desenvolvidos em cada unidade distribuída, de acordo com políticas de alocação previamente definidas, análise de risco e custo-benefício. O planejamento tático é de responsabilidade final dos responsáveis por cada unidade de desenvolvimento, enquanto que o planejamento operacional envolve a gestão do projeto (dimensão de projetos), sob responsabilidade do gerente de projeto.

A dimensão de projetos envolve especificamente a gerência do projeto de desenvolvimento de software, centrada na coordenação geral do trabalho entre os colaboradores, interfaces entre as equipes, comunicação, contato com os clientes e resolução de conflitos. Apesar de ser caracterizada pelo desenvolvimento dos projetos, esta dimensão não estava focada na definição de um processo, mas sim na identificação dos fatores e a relação entre eles. Assim, mapas conceituais (Figura 6) podem prover uma representação gráfica simples dos principais elementos identificados, tornando claro quais os conceitos envolvidos e os fatores relacionados que podem ser diretamente afetados. Pode-se ainda avaliar a necessidade de ajustes no desenvolvimento dos projetos, em decorrência da existência ou da inexistência de algum fator.

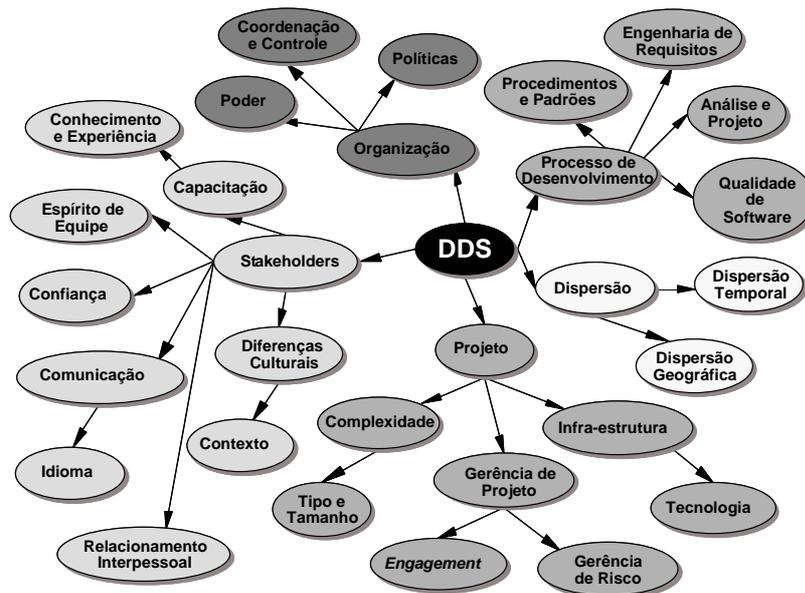


Figura 6. Variáveis identificadas na dimensão de projetos

O último ciclo proposto no modelo de referência é o de aprendizado, relativo à avaliação das atividades realizadas, e estratégias adotadas. O modelo sugere a existência de um processo para suportar a coleta de dados, envolvendo a avaliação dos trabalhos realizados, lições aprendidas, etc. Desta forma, o modelo realimenta os ciclos de planejamento.

6.2. Estágios de Capacidade

Ao recuperar os dados identificados nos estudos de caso desta pesquisa (Organizações 1 e 2), percebe-se que a capacidade da organização em DDS se desenvolve ao longo do tempo, com o ganho constante de experiência. Nas entrevistas ficaram evidentes algumas diferenças no nível de maturidade de atuação em ambientes de DDS, tais como as características do processo de desenvolvimento, a carga de treinamento existente, entre outros. Neste sentido, analisando o modelo proposto sob o ponto de vista de capacidade em projetos distribuídos, identificam-se quatro estágios, de acordo com a experiência em projetos de DDS existente nas organizações. Os estágios identificados fazem parte de uma proposta inicial de um modelo de capacidade para projetos distribuídos, baseada nos modelos de maturidade para processos de software [17] e para equipes dispersas [6] existentes na literatura e no modelo de referência (seção 6.1). A seguir descreve-se cada estágio e suas características:

6.2.1. Estágio 1: Inicial. Neste estágio os projetos distribuídos são desenvolvidos de uma forma desorganizada. Os projetos são considerados únicos e não existe uma maior preocupação em buscar informações sobre projetos similares ou das equipes envolvidas. Existe apenas o processo de captação de novos projetos para serem desenvolvidos (Novos Projetos). A decisão de desenvolver o projeto de forma distribuída é por conveniência. Não existe um planejamento e processo formais, nem uma avaliação das vantagens do desenvolvimento dos projetos em ambientes de DDS. As ações são caracterizadas como sendo reativas.

6.2.2. Estágio 2: Básico. Neste estágio os projetos ainda são considerados únicos e apresentam um nível básico de organização. Os fatores envolvidos na dimensão de projetos (Desenvolvimento dos Projetos) passam a ser analisados para minimizar dificuldades. Existe uma tendência para a prevenção de problemas, mas sem consulta a experiências anteriores. A decisão de desenvolver o projeto de forma distribuída continua sendo por conveniência. O planejamento e processo formais são realizados somente em nível de projeto. Não existe um processo estabelecido de avaliação e *feedback* (Figura 7).

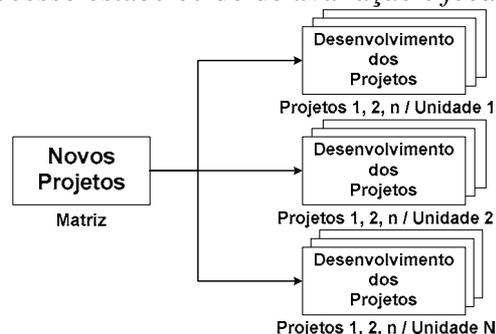


Figura 7. Estágio Básico

6.2.3. Estágio 3: Planejado. Neste estágio inserem-se os ciclos de planejamento estratégico (dimensão organizacional) e tático-operacional (dimensão de projetos). Um projeto não é mais considerado único. Além dos fatores envolvidos na dimensão de projetos, existe um processo formal para analisar e decidir se existem vantagens de se

desenvolver o projeto de forma distribuída (Alocação de Projetos). Isto faz com que a abordagem preventiva não seja apenas uma tendência, mas uma realidade. A consulta a experiências anteriores ocorre apenas internamente à cada processo. Neste sentido, não existe um processo estabelecido de avaliação e *feedback* baseado em conhecimento organizacional. (Figura 8).

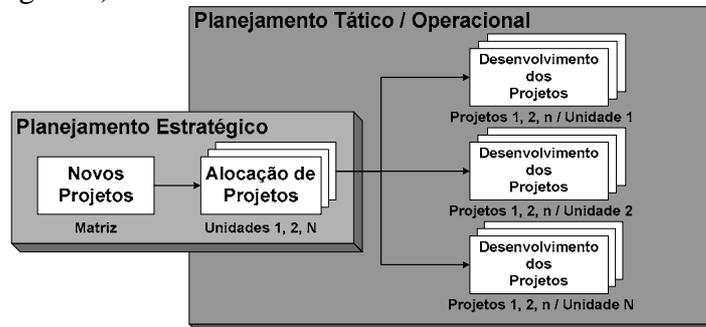


Figura 8. Estágio Planejado

6.2.4. Estágio 4: Otimizado. Neste estágio, insere-se o processo de Avaliação e *Feedback*. Além do processo de Alocação de Projetos e dos fatores envolvidos na dimensão de projetos, todos os projetos já desenvolvidos geram uma base de conhecimento que será utilizada como subsídio para o desenvolvimento de novos projetos, realimentando os ciclos de planejamento estratégico e tático-operacional (Figura 9).

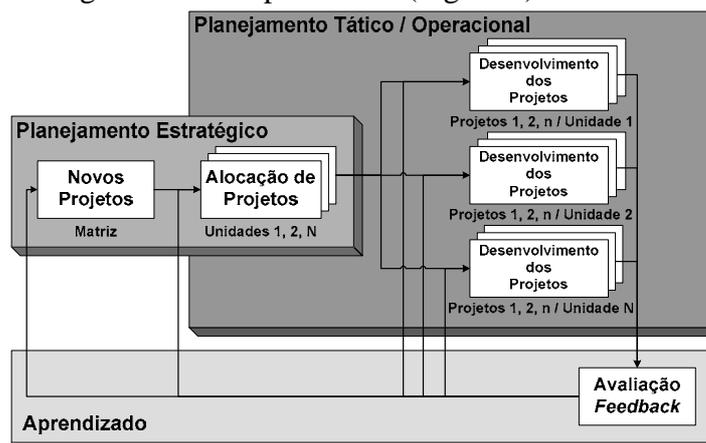


Figura 9. Estágio Otimizado

Como a capacidade diz respeito à maturidade da organização em projetos distribuídos, e tendo por base a forma como o estudo foi concebido, o reconhecimento de uma determinada organização em um estágio específico deve ser realizado através dos processos existentes dentro da sua realidade. Considerando que esta é uma proposta ainda incipiente, para uma organização ser reconhecida em um estágio de capacidade ela deve satisfazer as atividades envolvidas no respectivo estágio.

6.3. O modelo de referência na prática

Parte deste modelo de referência (Novos Projetos e Alocação de Projetos) está sendo implementada na prática, em uma das organizações estudadas. Como foi proposto um modelo de referência, a organização está atualmente criando seus próprios processos e implementando uma ferramenta, tendo o MuNDDoS como suporte ao processo que está sendo definido. Além disso, um programa de treinamento foi desenvolvido para tratar de alguns conceitos envolvidos no desenvolvimento dos projetos (dimensão de projetos), considerando inicialmente aquisição de confiança, diferenças culturais, e comunicação entre equipes distribuídas.

Além disso, a organização, baseada no modelo de referência proposto, tem reforçado a importância do papel da dispersão no processo de desenvolvimento de software e as consequências que ela pode trazer. Como consequência, alguns “processos distribuídos” têm sido criados, tais como especificação de requisitos e gerência de risco [19, 20, 21]. Em um futuro próximo, pretende-se analisar qualitativa e quantitativamente os resultados da implementação deste modelo, buscando melhorias no processo, e a sua implantação em outras organizações. Por fim, este projeto deu origem ao grupo de pesquisa em DDS, no programa de Pós-Graduação em Ciência da Computação da PUCRS.

7. Considerações Finais

O desenvolvimento de software sempre se apresentou de forma complexa. Existem diversos problemas e desafios inerentes ao processo [18]. Assim como o processo de desenvolvimento de software tem se tornado cada mais complexo, a distribuição das equipes no tempo e no espaço tem tornado os projetos distribuídos cada vez mais comuns. O software é cada vez mais indispensável para a sociedade moderna [2], onde a globalização é uma característica fundamental. Atualmente diversas empresas estão distribuindo seus processos de desenvolvimento de software ao redor do mundo, visando ganhos de produtividade, redução de custos e melhorias na qualidade. Neste contexto, o ambiente de DDS surge como um grande desafio para a área de ES.

O DDS tem levado os pesquisadores na área de ES a defrontar-se com a necessidade de novos conhecimentos e uma abordagem mais multidisciplinar. O modelo de referência proposto é uma tentativa de contribuir na busca de respostas a uma nova classe de problemas que o ambiente de trabalho distribuído apresenta. Os resultados encontrados apresentam indícios de que a área de ES necessita de mais pesquisas voltadas para esta nova de classe de problemas que o DDS está trazendo. Do ponto de vista científico, a legitimação do modelo de referência proposto é decorrente do processo de pesquisa como um todo. Sendo assim, a principal contribuição é a proposta de um modelo de referência para projetos de desenvolvimento distribuído de software. O modelo soma-se aos existentes na literatura como mais uma contribuição para responder aos desafios da área.

Uma das principais limitações da pesquisa refere-se ao número de empresas estudadas na parte empírica do estudo, restringindo a generalização dos resultados obtidos. Deve-se, entretanto, destacar que os resultados do estudo de caso, principalmente os da identificação dos fatores, foram sustentados na base teórica estudada, o que permite um bom grau de segurança nas conclusões obtidas. Isto também é típico do tipo de pesquisa desenvolvida, exploratória e de base qualitativa, permitindo o uso de inferências nas conclusões obtidas. Identifica-se grande potencial de crescimento nesta linha de pesquisa, onde se sugere o refinamento do modelo de referência, e o aprofundamento do estudo dos níveis de capacidade do modelo proposto, detalhando a descrição de cada estágio.

Por fim, o DDS, ao acrescentar fatores como dispersão geográfica, dispersão temporal e diferenças culturais, acentuou alguns dos desafios existentes e acrescentou novos desafios ao processo de desenvolvimento. Entre estes desafios pode-se citar questões estratégicas, questões culturais, gestão do conhecimento e gerência de riscos como de grande importância. Por isso, o trabalho em ambientes de DDS é 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 [4, 6, 11]. Além disso, os riscos técnicos e tecnológicos estarão sempre presentes, e estudos relacionados com os fatores técnicos têm sido amplamente divulgados [2, 3, 8]. Por isso, o trabalho na prevenção de dificuldades e problemas decorrentes tanto dos fatores técnicos como dos não-técnicos deve ser sempre valorizado.

Referências Bibliográficas

1. Herbsleb, J. D., Moitra, D. "Global Software Development", IEEE Software, March/April, EUA, 2001, p. 16-20.
2. Karolak, D. W. "Global Software Development – Managing Virtual Teams and Environments". Los Alamitos, IEEE Computer Society, EUA, 1998, 159p.
3. Herbsleb, J.D., Grinter, R. "Splitting the organization and integrating the code: Conway's Law revisited", In: ICSE 1999, Carolina do Norte, EUA, 1999.
4. Kiel, L. "Experiences in Distributed Development: A Case Study", In: Workshop on Global Software Development at ICSE, Oregon, EUA, 2003, 4p.
5. Herbsleb, J.D., Mockus, A., Finholt, T.A. e Grinter, R. E. "An empirical study of global software development: distance and speed", In: ICSE 2001, Toronto, Canada.
6. Carmel, E. "Global Software Teams – Collaborating Across Borders and Time-Zones". Prentice Hall, EUA, 1999, 269p.
7. Vogel, D., Davison, R., Shroff, R., Qureshi, S. "Sociocultural Learning in Globally Distributed Teams", In: Informing Science Conference, Polônia, 2001.
8. Altmann, J., Weinreich R. "An Environment for Cooperative Software Development Realization and Implications", In: HICSS 1998, Havaí, EUA, 1998.
9. Biuk-aghai R. P. "Customizable Software Engineering Environments for Flexible Distributed Software Teams", In: APSEC 1998, Taipei, Taiwan, 1998.
10. Prikladnicki, R., Audy, J. L. N., Evaristo, R. "Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context", In: Workshop on Global Software Development at ICSE, Oregon, EUA, 2003.
11. Marquardt, M. J., Horvath, L. "Global Teams: how top multinationals span boundaries and cultures with high-speed teamwork". Davies-Black. Palo Alto, EUA, 2001.
12. Evaristo, J. R., Scudder, R., Desouza, K., Sato, O. "A Dimensional Analysis of Geographically Distributed Project Teams: A Case Study," Journal of Engineering Technology and Management, 2003.
13. Morstead, S., Blount, G. "Offshore Ready: strategies to plan & profit from offshore IT enabled services", ISANI Press, US, 2003.
14. Yin, Robert. "Estudo de Caso: planejamento e métodos". SP: Bookman, 2001, 205 p.
15. Krippendorff, K. "Content analysis: an introduction to its methodology", Sage, 1980.
16. Prikladnicki, R., Audy, J. L. N., Evaristo, R. "Global Software Development in Practice: Lessons Learned", Journal of Software Process: Practice and Improvement – Special Issue on Global Software Development, 2004.
17. Paulk, M. C., Curtis, B., Mary, B. C., and Weber, C. V. (1993) "Capability Maturity Model, Version 1.1," IEEE Software, Vol. 10, No. 4, July, USA.
18. Prikladnicki, R. "MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software". Dissertação de Mestrado, PPGCC – PUCRS, Brasil, 2003.
19. Lopes, L., Prikladnicki, R., Audy, J. L. N., "Distributed Requirements Specification: Minimizing the Effect of Geographic Dispersion", In: ICEIS 2004, Cidade do Porto, Portugal, 2004.
20. Prikladnicki, R., Yamaguti, M. H., "Risk Management in Global Software Development: A Position Paper", In: Workshop on Global Software Development at ICSE, Edimburgo, Escócia, 2004.
21. Prikladnicki, R., Yamaguti, M. H., Antunes, D. C. "Risk Management in Distributed Software Development: A Process Integration Proposal", In: PRO-VE 2004, 5th IFIP Working Conference on Virtual Enterprises at 18th IFIP World Computer Congress.

ⁱ Pesquisa realizada no CDPe, financiada com recursos da Lei de Informática (Lei Federal Brasileira nº 8.248/91), vinculada ao PPGCC da Faculdade de Informática da PUCRS.