

O Gerenciamento de Projetos em ambientes de Desenvolvimento Distribuído de Software

Catarina S. Costa

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851, Cidade Universitária – 50.732-970 – Recife – PE – Brasil

csc@cin.ufpe.br

***Abstract.** The companies are redefining its processes to survive and aligning them with global policies. Considering this business context, a new environment has been encouraged to create new possibilities of competition and cooperation which overcome geographical boundaries. This new tendency is called Distributed Software Development (DSD). Under the perspective of project management, the DSD has been studied with the goal of searching for models that enables the effective management of geographically dispersed teams. This research aims to conduct a study on theories, models and techniques in the context of project management in environments DSD*

***Resumo.** As empresas estão redefinindo os seus processos para sobreviverem e se alinharem às políticas globais. Neste contexto, um novo ambiente tem sido incentivado e tem criado novas possibilidades de competição e cooperação que supera os limites geográficos. Esta nova tendência é chamada de Desenvolvimento Distribuído de Software (DDS). No gerenciamento de projetos, o DDS tem sido estudado com o objetivo de pesquisar modelos que permite uma gestão eficaz das equipes geograficamente dispersas. Esta pesquisa visa realizar um estudo sobre teorias, modelos e técnicas no contexto da gestão de projetos em ambientes DDS.*

1. Introdução

A gerência de projetos de software é uma atividade de gerenciamento global de todo o desenvolvimento, estando presente em todas as fases, desde o início até a implantação do software. É fundamental que haja a adoção de boas práticas de gerenciamento em todas as atividades do projeto, desde o planejamento até a entrega final do produto.

De acordo com Torreão [2005], atualmente, o gerenciamento de projetos vem se fortalecendo cada vez mais. As organizações estão conferindo maior importância a esta disciplina para maximizar o sucesso em seus projetos.

Muitos dos problemas que afetam os projetos de desenvolvimento de software são de ordem gerencial e não técnicos [Branco e Belchior, 2002]. Embora o desenvolvimento de software tenha evoluído consideravelmente, ainda há diversas dificuldades tais quais: entregas fora do prazo estipulado ou com custos superiores ao orçado [Lopes, Majdenbaum e Audy, 2003].

Em paralelo, o desenvolvimento de software passa nos últimos anos por transformações, que geralmente são motivadas pela globalização dos negócios, as

empresas estão distribuindo seus processos de desenvolvimento dentro e fora de seus países.

Neste cenário, o Desenvolvimento Distribuído de Software (DDS) ganha cada vez mais força, motivado por três fatores ligados ao ambiente de negócios: a globalização, o crescimento da importância dos sistemas de informação nas empresas e os processos de terceirização que geram o ambiente propício a esse cenário de desenvolvimento [Audy e Prikladnicki, 2008].

Segundo Kiel [2003], há toda uma razão para acreditar que haverá uma contínua pressão para que as empresas adotem alguma abordagem global de desenvolvimento de software. Com a distribuição geográfica de recursos e investimentos, surge uma nova tendência, em que usuários e equipes de desenvolvimento estão em locais físicos diferentes, às vezes com culturas diferentes.

Neste contexto, novos problemas surgem, principalmente relacionados a distância física entre os participantes do processo e os já tradicionais problemas inerentes ao desenvolvimento ganham contornos mais críticos.

Segundo Prikladnicki [2004], alguns estudos observam algumas dificuldades em se implantar, executar e monitorar projetos em ambientes de DDS devido a fatores não-técnicos tais como fatores sociais, culturais, comportamentais, psicológicos, lingüísticos e políticos; por exemplo: confiança, diferenças culturais, idioma, entre outros.

O cenário de desenvolvimento distribuído de software provoca mudanças no gerenciamento de projetos, pois agora novas variáveis são adicionadas, como: distância física e temporal. Portanto, uma atenção especial deve ser dedicada já que o gerenciamento de projetos de DDS exige a adaptação de algumas das técnicas utilizadas em projetos co-localizados [Audy e Prikladnicki, 2008].

Tem-se firmado também o entendimento de que a utilização de metodologias científicas em projetos de software vem de sobremaneira auxiliar no processo de gerenciamento [Branco e Belchior, 2002].

Nesse sentido, este estudo busca através de uma pesquisa em diversos trabalhos que tratam do gerenciamento e um cenário de DDS, determinar quais aspectos do gerenciamento de projetos tradicional são alterados ou afetados em um contexto de desenvolvimento distribuído de software, além de discutir técnicas e modelos criados nos últimos anos para atender a esse novo cenário de desenvolvimento.

Além da seção introdutória, este artigo está organizado da seguinte forma: na Seção 2 são abordados os conceitos de Gerenciamento de Projetos e Desenvolvimento Distribuído de Software, que são referências básicas para a realização e compreensão deste estudo; na Seção 3 são apresentados trabalhos que abordam de alguma forma o gerenciamento de projetos em ambientes de desenvolvimento distribuído de software; na Seção 4 é realizada análise dos trabalhos apresentados; na Seção 5 são apresentadas as conclusões e recomendações para trabalhos futuros.

2. Fundamentação Teórica

Os conceitos de gerenciamento de projetos e desenvolvimento distribuído de software formam a base teórica para o entendimento desse estudo. Essa base conceitual é descrita nas próximas seções

2.1. Gerenciamento de Projetos

A gerência de projetos visa à coordenação de todo o trabalho de desenvolvimento de um produto, do seu início até sua implantação, visto que, gerenciar projetos implica em planejar, organizar e controlar os recursos e tarefas orientadas a um objetivo único, em um ambiente de restrições de tempo e custo.

Heldman [2006] afirma que o gerenciamento abrange uma série de ferramentas e técnicas utilizadas por pessoas para descrever, organizar e monitorar o andamento das atividades do projeto, e que os gerentes de projetos são os responsáveis pela administração dos processos envolvidos e pela aplicação das ferramentas e técnicas necessárias ao cumprimento das atividades do projeto.

De acordo com o PMBOK® (*Project Management Body of Knowledge*), guia de responsabilidade do PMI (*Project Management Institute*), o gerenciamento de projetos consiste na aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos requisitos [PMI, 2004].

O gerenciamento de projetos também envolve negociação, solução de problemas, política, comunicação, liderança e estudo da estrutura organizacional [Frame, 1995]. Os gerentes de projetos devem ter algumas habilidades para que os projetos sejam eficientes, como: (1) uma boa comunicação, tanto escrita como oral; (2) aptidões organizacionais e de planejamento; (3) habilidades para a elaboração de orçamentos; (4) habilidades para resolução de conflitos; (5) habilidades de negociação e influência; (6) habilidades de liderança; (7) habilidades para formação e motivação de equipes [Heldman, 2006].

Uma grande atenção deve ser dedicada a essa atividade, pois falhas no gerenciamento do projeto são causas freqüentes do insucesso de projetos. A importância de um bom gerenciamento de projetos é evidente, e na literatura existem algumas abordagens para sua realização, sendo que uma das referências mais conhecidas na área é o PMBOK®.

2.2. Desenvolvimento Distribuído de Software

Na última década, observou-se um grande investimento na conversão de mercados nacionais em mercados globais na busca por novos clientes e oportunidades. Conseqüentemente, as concorrências locais também se transformaram em globais, tornando o mercado de software cada vez mais competitivo. Nesse cenário, o DDS torna-se uma alternativa interessante para integrar e coordenar equipes geograficamente distribuídas.

Isso levou a um grande crescimento nesta área, com diversas empresas distribuindo seus processos de desenvolvimento de software, dentro de seu país ou em outros países, na busca por menores custos fiscais e recursos mais qualificados que nem sempre estão localmente disponíveis [Herbsleb e Moitra, 2001].

Além destes, outros fatores contribuíram para acelerar a busca pelo DDS, entre eles [Prikladnicki, 2003]: a) necessidade de recursos globais para serem utilizados a qualquer hora; b) vantagem de estar perto do mercado local, incluindo o conhecimento dos clientes e as condições locais; c) rápida formação de organizações e equipes virtuais para explorar as oportunidades locais; d) grande pressão para o desenvolvimento *time-to-market* (velocidade no trabalho, tempo entre a concepção e a comercialização do produto) utilizando as vantagens do fuso horário diferente, no desenvolvimento conhecido como *follow-the-sun* (24 horas contínuas).

Segundo Audy e Prikladnicki [2008], as principais características que diferenciam o desenvolvimento co-localizado do desenvolvimento distribuído são: dispersão geográfica (distância física), dispersão temporal (diferença de fuso horário) e diferenças socioculturais (idioma, tradições, costumes, normas e comportamento).

Mesmo com diversos fatores contribuindo para o crescimento do DDS; assim como no desenvolvimento co-localizado, construir sistemas de software não é uma tarefa simples, e no cenário de desenvolvimento distribuído a complexidade tende a aumentar.

Prikladnicki [2003] afirma que o desenvolvimento distribuído criou uma nova classe de problemas a serem resolvidos pelos pesquisadores na área de desenvolvimento de software. Estas mudanças estão causando um grande impacto não apenas no mercado propriamente dito, mas na maneira como os produtos estão sendo criados, modelados, construídos, testados e entregues para os clientes.

3. O Gerenciamento de Projetos em ambientes de DDS

Motivados pela globalização, entre outros fatores citados anteriormente, muitas equipes de projeto passaram a trabalhar de forma distribuída, e o gerenciamento, considerado um fator importante para o sucesso de projetos centralizados, agora deve atender também aos desafios impostos por ambientes de desenvolvimento distribuído.

De acordo com Espindola [2005], o DDS adiciona novas dificuldades ao desenvolvimento de software ao acrescentar fatores como dispersão geográfica, temporal e possíveis diferenças culturais, o que pode dificultar as comunicações no projeto.

Segundo Prikladnicki [2003], a estrutura utilizada em equipes co-localizadas pode perder a efetividade em ambientes distribuídos e necessita de adaptações. Ele acrescenta ainda que uma equipe de DDS deve possuir uma estrutura flexível para permitir a distribuição de tarefas e a tomada de decisões de forma efetiva.

Neste contexto, a Gerência de Projetos, passa a exigir adaptações de técnicas utilizadas em projetos co-localizados, de forma a reduzir dificuldades impostas pela dispersão das equipes, principalmente as de comunicação [Audy e Prikladnicki, 2008].

Diante disto, atualmente é possível encontrar na literatura alguns trabalhos voltados para o gerenciamento de projetos em ambientes de desenvolvimento distribuído de software, estudos, modelos, metodologias, entres outros, criados no intuito de atender a esse novo cenário de desenvolvimento e a todos os desafios acentuados por ele. Nas próximas seções são apresentadas algumas dessas abordagens.

3.1. Abordagem de Zaroni [2002]: Modelo de Gerência de Projeto Baseado no PMI para ambientes de Desenvolvimento de Software Fisicamente Distribuído

Zaroni [2002], afirma que há uma tendência crescente por utilizar equipes geograficamente dispersas para se desenvolver software e cita alguns dos desafios encontrados nesse ambiente, como: (1) a atribuição das tarefas do desenvolvimento aos membros da equipe; (2) a comunicação e colaboração entre desenvolvedores; (3) o compartilhamento dos documentos e artefatos do projeto e a consistência destes; (4) a necessidade de suportar plataformas heterogêneas; (5) o acompanhamento do progresso dos grupos distintos; e, (6) a integração entre os módulos desenvolvidos por estes grupos.

Tendo em vista isso, o trabalho afirma que a gerência de projetos de software em ambiente distribuído, deve atender a algumas características consideradas fatores de sucesso nesse tipo de ambiente:

Comunicação: na gerência de projetos em ambiente distribuídos, a comunicação é fundamental para o bom andamento e execução do projeto. A falta do gerenciamento de comunicação pode levar as equipes distantes a desconhecerem informações relevantes sobre o projeto. Devido a isso, é necessário existir um fluxo de informações contínuo e eficaz entre os integrantes das equipes.

Gerenciamento da Equipe: equipes são geralmente compostas por pessoas que têm diferentes idéias de como solucionar um determinado problema, e isso pode prejudicar o andamento do projeto, causando problemas que poderiam ser melhor gerenciados em um ambiente de trabalho em grupo. Em um ambiente distribuído o gerenciamento requer um processo de comunicação padrão.

Credibilidade: a gerência de projetos de software precisa lidar com a credibilidade existente entre os participantes do projeto. Ter credibilidade é ter confiança, segurança e firmeza no trabalho da equipe com um todo. Em um ambiente distribuído essa característica torna-se ainda mais relevante, já que essa credibilidade entre aos participantes fisicamente distantes é importante para o bom fluxo de informação entre as equipes distribuídas.

Conflitos: qualquer ambiente de gerência ou desenvolvimento de software é propício a ocorrer conflitos. No ambiente distribuído, os conflitos apresentam mais complexidade na sua resolução, e acontecem por, além dos problemas comuns do ambiente centralizado, como, não entendimento por parte da equipe de suas funções, falta de comunicação e uma estimativa irreal, por diferenças culturais e a distribuição física das equipes que compõem o projeto.

Cooperação: em um ambiente distribuído a gerência de projetos de software, exige um grau maior de cooperação entre os participantes do projeto. É necessário gerenciar o trabalho em equipe para se chegar a um objetivo comum.

Processos Definidos: a gerência de projeto de software, em um ambiente distribuído, requer regras e processos bem definidos, o que auxilia na coordenação e entendimento por parte das equipes. Deve existir um suporte para a definição de processos das atividades gerenciadas.

Diferenças Culturais: geralmente não estão presentes em ambientes centralizados, mas são comuns em ambientes distribuídos, já que as equipes podem ser compostas por pessoas de locais (cidade, estado, país, continente) diferentes. Deve-se compreender bem o nível de diferenças culturais existentes entre as equipes, tendo em vista que determinadas ações podem ser mal interpretadas pelo simples fato de fazer parte da cultura de uma equipe, e não da outra.

O trabalho expõe que um processo formal para a gerência de projeto de software deve ser adotado e seguido pelo gerente de projeto, principalmente em um ambiente de DDS. É importante que todas as equipes do projeto estejam cientes do funcionamento deste processo, realizando as atividades necessárias para que não ocorra a perda de informações ou conflitos durante as fases do projeto.

Após identificar todos esses fatores relacionados a ambientes de desenvolvimento distribuído, a pesquisa analisa as disciplinas do guia PMBOK, buscando responder quais disciplinas do modelo do PMI exigem mais preocupação por parte do gerente de projeto, além de propor mais rigor nos processos tendo em vista o ambiente distribuído:

Gerência de Comunicação: Em um ambiente de DDS deve-se manter um fluxo de comunicação muito intenso entre os participantes do projeto, reuniões com clientes e usuários são necessárias para o entendimento do projeto.

Gerência do Escopo: É relevante controlar o que está ou não incluso no projeto, é importante definir o escopo no início do projeto, proporcionando uma gerência mais estável e realista.

Gerência de Recursos Humanos: É importante ter conhecimento de todas as pessoas envolvidas com o projeto, independentemente da localização, levando-se em conta fatores como: relacionamentos interpessoais e as diferenças culturais. As funções e responsabilidades devem estar bem definidas.

Gerência de Integração: No momento da formalização do projeto por todos os envolvidos, deve-se assegurar que todos os elementos do projeto estão adequadamente coordenados. A integração é necessária nessa fase, pois participantes, independentemente da localização, devem estar cientes do entendimento do projeto.

Gerência de Tempo: Em um ambiente distribuído é importante a definição e acompanhamento das atividades que irão compor o projeto. O gerente de projeto deve estar focado no controle do cronograma, determinando sua alteração, gerenciando as mudanças quando elas ocorrerem.

Gerência dos Riscos: Nesse cenário vários riscos podem surgir, mas os principais riscos são: descuido na comunicação e a possibilidade de algum risco ser ignorado. É necessário minimizar riscos através de contatos frequentes com todos os participantes do projeto.

Gerência da Qualidade: A elaboração de um plano de qualidade, através da identificação de quais padrões de qualidade são relevantes para o projeto e determinar como satisfazê-los é importante.

Gerência de Custos: É necessário um consenso entre patrocinadores e responsáveis pelo projeto, deve-se realizar uma estimativa de custo dos recursos

necessários (pessoas, equipamentos e materiais) para a realização das atividades do projeto.

Além disso, a pesquisa propõe uma extensão ao modelo proposto pelo PMI, adicionando 4 (quatro) novas disciplinas: Gerência do Planejamento, Gerência da Propriedade Intelectual, Gerência de Aprendizagem e Gerência de Conflitos.

Por fim, Zanoni [2002], propõe um modelo de gerenciamento de projetos para ambientes de desenvolvimento distribuído de software e leva em consideração questões que envolvem a distribuição geográfica das pessoas.

As características principais do modelo, que possui seis fases, como mostra a Figura 1, são: (1) Possui ciclo de vida do tipo espiral; (2) Utiliza o processo de desenvolvimento de sistemas OO, com linguagem de especificação UML (*Unified Modeling Language*) e processo UP (*Unified Process*); e, (3) Incorpora a abordagem processual do PMI expandindo as áreas de gestão indicadas.

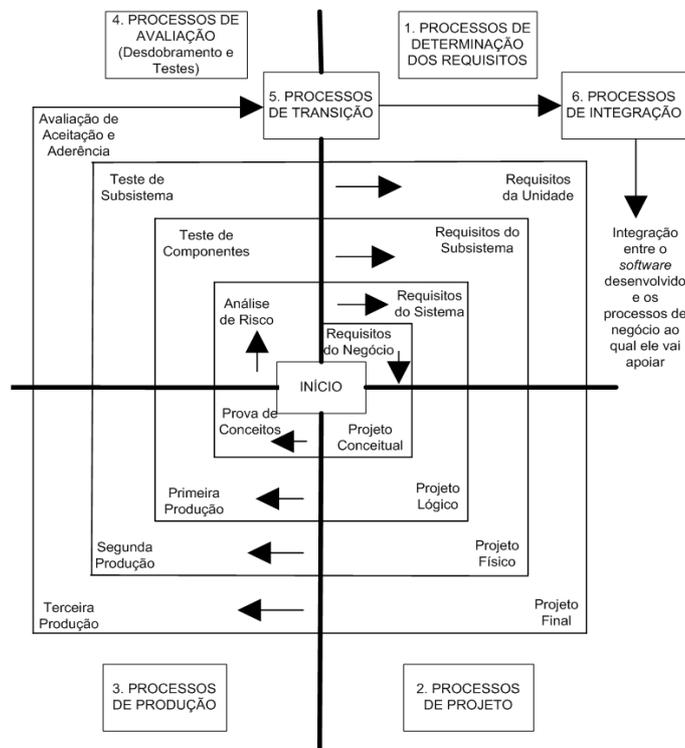


Figure 1. Ciclo de Vida do modelo proposto por Zanoni [2002].

As fases que compõem o modelo são: Análise de Requisitos, Projeto (Exploração e Definição), Processos de Produção, Avaliação (Desdobramento e Testes), Processos de Transição e Processos de Integração.

3.2. Abordagem de Prikladnicki [2003]: MuNDDoS: Um modelo de Referência para Desenvolvimento Distribuído de Software

Em seu trabalho, Prikladnicki [2003], propõe um modelo de referência para a área de desenvolvimento distribuído, contemplando as dimensões técnicas e não-técnicas e os fatores envolvidos em cada uma delas, embora não tenha como foco o gerenciamento de

projetos, algumas lições indicaram o gerenciamento como uma área que deve ter uma atenção especial quando o cenário é distribuído.

O modelo MuNDDoS (Maturidade No Desenvolvimento Distribuído de Software) visa para facilitar o desenvolvimento distribuído, permitindo também a identificação de fraquezas e oportunidades de melhorias nos projetos.

O modelo é composto por três partes, como mostra a Figura 2: (1) planejamento estratégico; (2) planejamento tático e operacional; e, (3) aprendizado. No planejamento estratégico busca-se uma visão estratégica realizada pela matriz que identifica e prioriza os projetos a serem desenvolvidos. O planejamento tático-operacional é realizado para definir quais as unidades distribuídas que irão desenvolver cada projeto. No aprendizado são coletados dados para uma avaliação das atividades realizadas e estratégias adotadas.

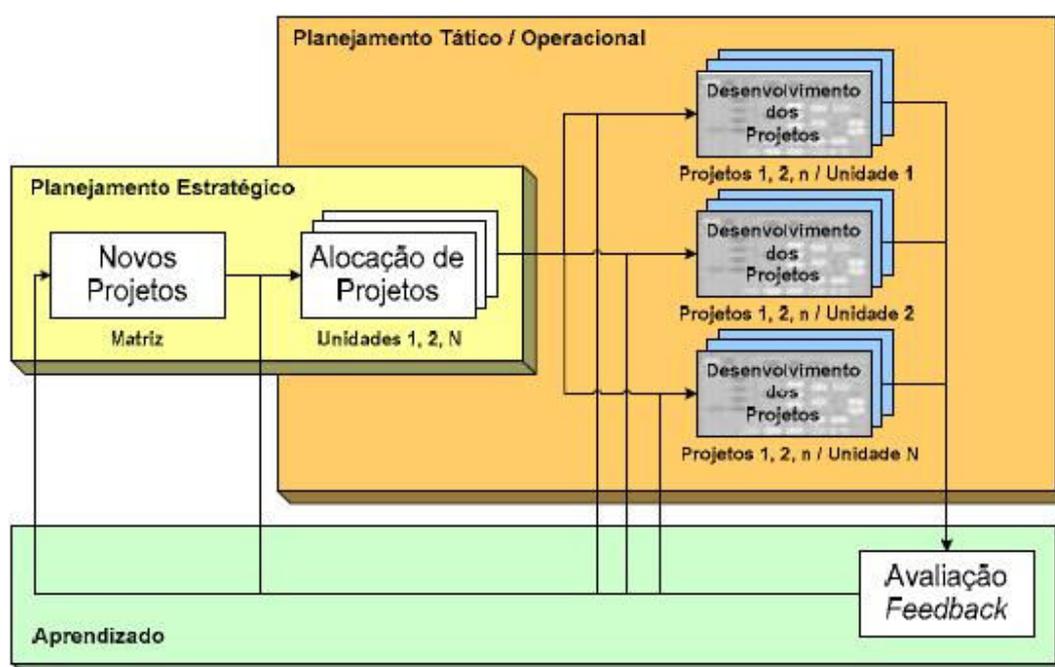


Figure 2. Modelo proposto por Prikladnicki [2003].

De acordo com a pesquisa, a gerência de projetos requer um esforço adicional em um ambiente de DDS, pois, segundo entrevistas realizadas com Gerentes de Projetos, estes citaram que algumas práticas tradicionais de gerenciamento de projetos precisam ser adaptadas para se adequarem às características do desenvolvimento distribuído. A pesquisa lista algumas dificuldades de dimensões técnicas e não-técnicas encontradas no desenvolvimento distribuído.

A engenharia de requisitos (os requisitos devem ser passados com o maior nível de detalhes possíveis, sem margens para falsas interpretações), os processos de desenvolvimento de software (nem sempre as equipes estão ligadas por um mesmo processo, muitos problemas podem surgir principalmente na gerência do projeto) a gerência de configuração (muitas vezes os artefatos não possuem a mesma versão, nem o mesmo conteúdo nos diferentes locais onde o projeto está sendo desenvolvido).

Além destas, a gestão do conhecimento (dificuldades no sentido de compartilhar informações em ambientes distribuídos devido ao pouco investimento nesta área), as

barreiras de comunicação e idioma (dificuldades de comunicação ocorrem com frequência, incluindo dificuldades devido a diferença de idiomas), as diferenças culturais e contexto (principalmente em projetos com equipes em países diferentes, as diferenças podem dá origem a diversas dificuldades) e a confiança (dificuldades relacionadas à criação de uma atmosfera de confiança e respeito entre equipes distribuídas).

Além de apresentar algumas dificuldades, o trabalho, baseado na literatura e em estudos de caso, propõe algumas soluções e fatores críticos de sucesso em projetos distribuídos:

O gerenciamento de expectativas (definir claramente papéis e responsabilidades dos integrantes das equipes), a integração das equipes (integração face-a-face entre as equipes, na medida do possível), uma comunicação aberta e *feedback* (boa infraestrutura de comunicação), um processo de desenvolvimento de software (definição dos processos e padrões de trabalho) o gerenciamento dos riscos (identificação e ações de mitigação podem minimizar diversas dificuldades).

O trabalho também cita como fatores de sucesso, a Engenharia de requisitos (documentação e aprovações formais dos artefatos de projeto), a aquisição da confiança (treinamentos, planejamentos, entre outras atividades de integração das equipes remotamente ou face-a-face, quando possível, visam principalmente à aquisição de confiança e o conhecimento dos participantes), treinamentos (nivelamento do conhecimento, aprendizagem contínua), o planejamento e *engagement* (uma clara definição do planejamento inicial e do *engagement* - engajamento das equipes no projeto) e a infra-estrutura (boa infra-estrutura de comunicação e ferramentas de suporte ao desenvolvimento).

Segundo a pesquisa, o desenvolvimento distribuído é considerado mais complexo que o desenvolvimento centralizado devido principalmente a falta de contato pessoal e a obrigatoriedade de haver um relacionamento distribuído, aumentando a necessidade de comunicação. Projetos distribuídos necessitam de um maior controle, e conseqüentemente de um maior investimento na gerência do projeto e gerenciar riscos nesse ambiente é uma tarefa essencial e constante.

Prikladnicki [2003], afirma que com os resultados de sua pesquisa pode concluir que a gerência de projetos e conseqüente gerência de riscos requerem um esforço e passos adicionais. Além disso, um maior esforço na gerência de projetos repercute em uma maior necessidade de comunicação.

3.3. Abordagem de Enami [2006]: Um Modelo de Gerenciamento de Projetos para um Ambiente de DDS

Enami [2006], apresenta um modelo de gerenciamento de projetos (MGP) para o desenvolvimento distribuído de software (DDS). O objetivo do modelo é contribuir para a melhoria do controle gerencial em busca de soluções para os problemas que este tipo de desenvolvimento pode trazer. Para isso, foram usados como referência os seguintes MGPs: *Project Management Body of Knowledge* (PMBOK), *Capability Maturity Model Integration* (CMMI), Modelo de Gerência de Projeto Baseado no *Project Management Institute* (PMI) para Ambiente de Desenvolvimento de Software Fisicamente

Distribuído, o Modelo de Maturidade no DDS (MuNDDoS) e também outros trabalhos relacionados.

O MGP proposto trata os três níveis organizacionais: estratégico, tático e operacional. O MGP está mais focado no nível tático para dar apoio ao gerente de projeto na execução de suas funções. O modelo apresenta: (1) orientação para minimizar os problemas advindos de diferenças culturais e dispersão geográfica; (2) comunicação entre os *stakeholders*; (3) obtenção do comprometimento dos *stakeholders* primários; (4) necessidade de transporte de artefatos entre os participantes do projeto; e, (5) padronização de documentos e procedimentos. Veja na Figura 3 um esquema com os componentes do MGP.

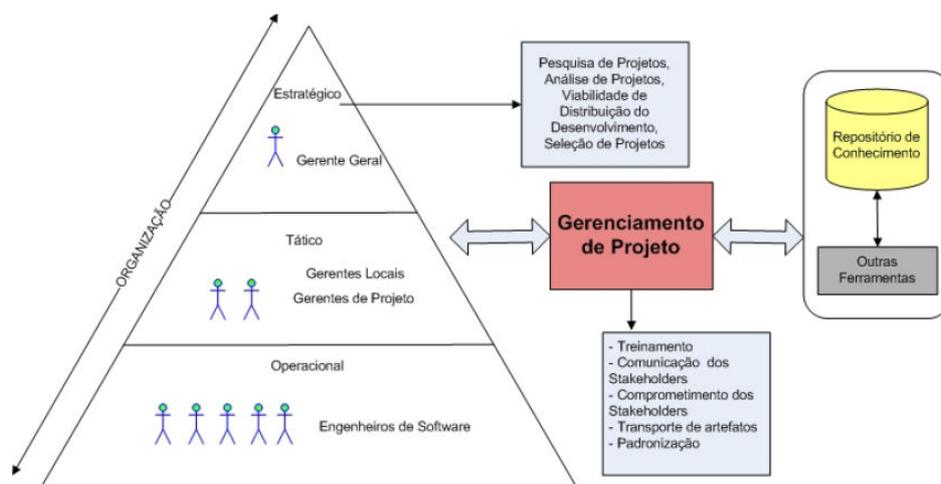


Figura 3. Componentes do MGP proposto.

O MGP faz parte do Projeto *Distributed Software Engineering Environment (DiSEN)* e apresenta os seguintes elementos: gerência de *stakeholders*, gerência do conhecimento, gerência de riscos, gerência de requisitos. Além de orientação para minimizar ou eliminar os problemas advindos de diferenças culturais e dispersão geográfica em um ambiente de DDS, a propriedade intelectual, e por fim, uma ferramenta de apoio ao gerenciamento de projeto. Para cada elemento, a dissertação detalha como deve ser feita a gerência de projetos em um ambiente de desenvolvimento distribuído de software.

Além destes elementos, o modelo também define métricas para medição e acompanhamento da aplicação do modelo; assim como, modelos de documentos que podem ser utilizados.

3.4 Abordagem de Komi-Sirvi e Tihinen [2002]: Lessons Learned by Participants of Distributed Software Development

O trabalho de Komi-Sirvi e Tihinen [2002], apresenta um estudo realizado com o objetivo de levantar e compartilhar lições aprendidas para obter um melhor entendimento da natureza do processo de desenvolvimento de software quando realizado em um ambiente distribuído e dos problemas que podem estar associados com tais processos distribuídos. Algumas recomendações práticas desenvolvidas pela indústria para superar estes problemas são também superficialmente comentados.

Segundo o estudo, o DDS permite que a produção de software ocorra independentemente da localização geográfica de indivíduos e organizações envolvidos. Infelizmente, projetos de DDS costumam herdar os mesmos problemas que projetos co-localizados enfrentam. Logo, possuem problemas com qualidade, tempo e custo que muitas vezes tornam-se mais difíceis de lidar em função da distribuição. Estudos recentes mostram que a distância física existente entre sites de desenvolvimento cria atrasos no trabalho; como também causam problemas na coordenação de tarefas, no gerenciamento de projetos e na comunicação.

A pesquisa realizada por Komi-Sirvi e Tihinen, contou principalmente com a participação de empresas da Finlândia, mas também de outras da Holanda e dos Estados Unidos. No total, foram 27 organizações consideradas. 53% destas são empresas com mais de 500 funcionários, 35% entre 50 e 500 funcionários e 13% com menos de 50 funcionários.

Os participantes tinham a opção de escolher até oito áreas de problemas diferentes no DDS. A lista de problemas foi construída a partir de uma revisão da literatura, ou seja, os participantes marcavam os problemas que eles haviam enfrentado em seus projetos e também descreviam com mais detalhes estes. Também havia a possibilidade de identificar problemas não presentes na lista. Contudo, só dois participantes utilizaram esta opção. Além da descrição dos detalhes dos problemas enfrentados, os participantes também descreveram as soluções adotadas para superar estes, assim como fizeram uma auto-avaliação do nível de sucesso destas soluções. Veja na Figura 4 a distribuição das respostas ao longo das oito áreas de problemas.

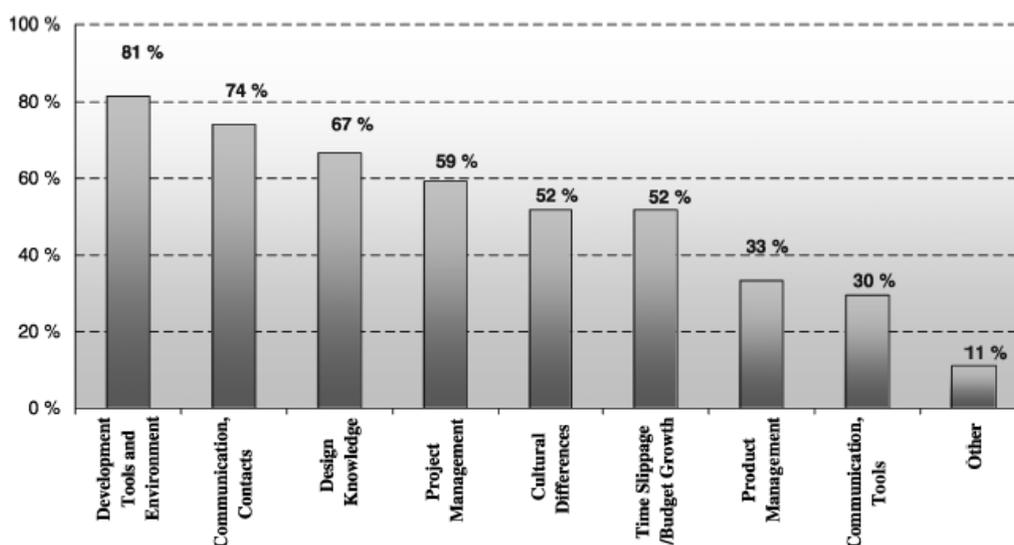


Figura 4. Áreas de problemas em DDS.

Os resultados do trabalho mostram que a área mais problemática está relacionada com ferramentas e ambientes de desenvolvimento de software, mais especificamente com a incompatibilidade de ferramentas e versões usadas por sites de desenvolvimento diferentes. Este problema é mais acentuado em grandes organizações que empregam mais de 500 pessoas.

Problemas de comunicação aparentam ser extremamente comuns em todas as organizações, de forma que esta área de problema foi classificada como a segunda. No

entanto, uma análise mais detalhada das respostas mostra que o papel da comunicação é ainda maior do que aparenta ser inicialmente. Ao estudar as razões por trás de outros problemas, a falta ou a baixa qualidade das comunicações é frequentemente mencionada como causa principal. Quando um mecanismo eficiente de captura e compartilhamento de conhecimento é dificultado pela distância, torna-se vital o apoio de soluções substitutas que apoiem estes mecanismos.

O gerenciamento e a engenharia de requisitos apareceram também como áreas bastante problemáticas em projetos de desenvolvimento distribuído de software, causando vários erros.

Baseado na análise descrita no artigo, não há dúvida sobre a mensagem deste estudo: desenvolvimento distribuído de software com sucesso requer tanto engenharia de software disciplinada e estruturada, como também soluções de gerenciamento do conhecimento considerando, particularmente, o gerenciamento das comunicações e um substituto efetivo para a comunicação face-a-face.

Uma execução cuidadosa das atividades iniciais do projeto – incluindo o planejamento (divisão de tarefas, cronograma e entregáveis), a determinação exata de regras comuns, responsabilidades, ferramentas utilizadas e sessões de *kick-off* – pode contribuir significativamente para uma implementação de sucesso de um projeto de DDS.

3.5. Abordagem de Gilberg e Grabow [2005]: Quality Management in Distributed Software Development

O trabalho de Gilberg e Grabow [2005], descreve um sistema de gestão da qualidade que tem sido feito especialmente e aplicado para avaliar e melhorar a qualidade da segurança de produto de software (software para a realização das eleições através da Internet). Neste desenvolvimento, existem organizações e parceiros envolvidos que estão geograficamente separados.

O objetivo do projeto descrito, é projetar, implementar e validar um sistema eletrônico de votação baseado na Internet que forneça todos os serviços necessários para a organização e a realização de um procedimento de votação. Este processo pode variar de referendos e sondagens até eleições gerais.

Os autores afirmam que muitas vezes as falhas nos software são comuns, porém a situação deve ser completamente diferente quando se trata de softwares críticos, como sistema de gerenciamento de voo e centrais nucleares, onde pequenas falhas podem ter conseqüências gravíssimas.

A qualidade é interpretada e entendida de maneira diferente por pessoas diferentes, o que gera grande problema quando estas pessoas estão em culturas diferentes. O primeiro obstáculo quando se trata de qualidade em ambiente de desenvolvimento distribuído é que as equipes distribuídas não possuem processos de desenvolvimento iguais, onde, cada parceiro, cada equipe faz suas tarefas de acordo com seus processos e procedimentos internos.

Assim é aconselhado pelos autores que o produto deve ser o centro de referência entre as equipes e as mesmas devem se concentrar numa abordagem de gestão de qualidade igual para todos.

No caso desse projeto, o modelo deveria ser capaz de lidar com duas peculiaridades: (1) o e-VOTE, o projeto em questão, é desenvolvido por equipes geograficamente distantes; e, (2) é um sistema inovador e um produto de segurança crítica.

Desta forma quando se tratando de ambientes de DDS é aconselhável criar um modelo próprio de qualidade para o projeto, um modelo genérico para que todas as equipes envolvidas possam seguir sem muitos problemas de adaptações.

O modelo proposto pelos autores é um modelo hierarquicamente estruturado, onde existe um nó raiz representando a qualidade geral do produto a ser desenvolvido. Este nó tem vários filhos cujo número pode variar em função do respectivo produto. Estes filhos constituem características qualitativas, que correspondem aos fatores de qualidade. Cada característica de qualidade representa um aspecto intrínseco ao software para contribuir na sua qualidade global (por exemplo, usabilidade, segurança). Veja a estrutura do modelo na Figura 5.

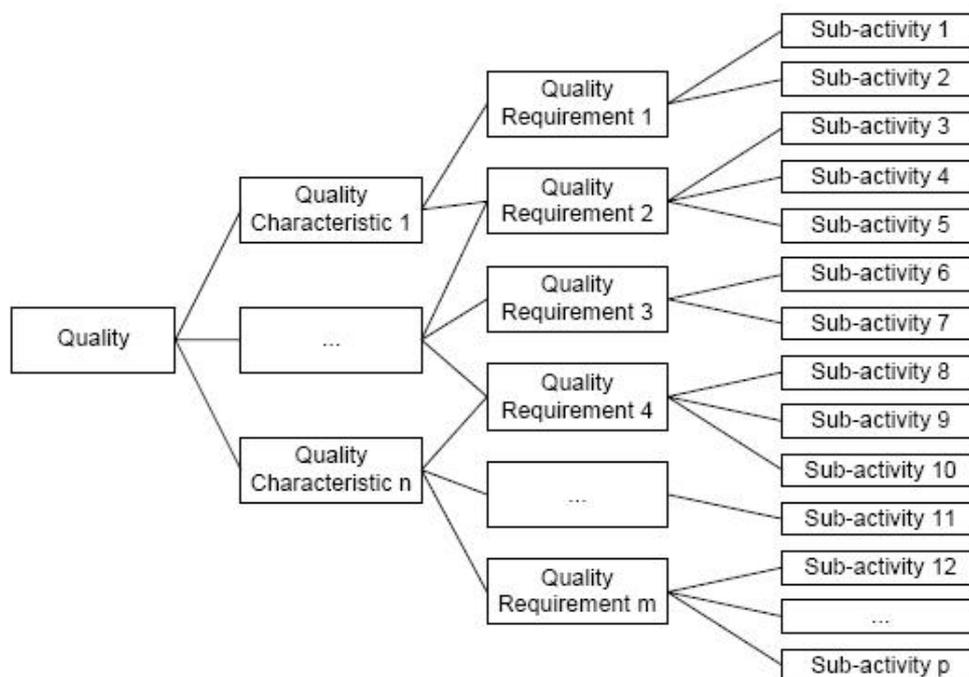


Figura 5. Estrutura do modelo de qualidade.

Além disso, é associado a cada característica de qualidade um número variável de requisitos de qualidade, que representam exigências concretas, por exemplo: o sistema deve apoiar diferentes idiomas e o usuário deve ser capaz de mudá-lo. Por fim, o mecanismo usado para avaliar o cumprimento dos requisitos de qualidade é uma abordagem mais geral que dá uma descrição detalhada das ações a serem realizadas. Essas ações são chamadas de sub-atividades, onde cada uma define uma ação concreta que deve ser exercida pelo respectivo avaliador.

É importante para que o modelo funcione, haver a criação de uma estrutura organizacional adequada para a gestão da qualidade, juntamente com as atribuições e responsabilidades para cada membro da equipe.

Os autores aconselham destacar um responsável, que terá o papel central para a gestão da qualidade, ou seja, um coordenador da garantia da qualidade (CGQ). Este será responsável por definir as estratégias globais de qualidade, gerar o modelo de qualidade a ser seguido por todos e responsável ainda pela condução das avaliações de qualidade no projeto.

Além disso, em cada equipe, é identificado um membro, ao qual é atribuída a responsabilidade de avaliação da qualidade, ficando responsável pela execução e conclusão do processo de avaliação, são conhecidos como gerentes locais da qualidade (GLQ). Eles têm o discernimento necessário nas suas organizações, e tem o papel de localizar os avaliadores adequados, além de serem o elo entre o coordenador da qualidade (CGQ) e avaliadores.

O trabalho também recomenda o agrupamento de todas as questões relacionadas com qualidade dentro de um manual de qualidade a ser seguido por todos os demais participantes do projeto. O manual de qualidade é o principal documento que descreve a estratégia de qualidade, bem como todas as responsabilidades e os procedimentos necessários para facilmente avaliar a qualidade do sistema.

Na primeira fase do projeto o coordenador da garantia da qualidade (CGQ) envia as formas de avaliação dos requisitos de qualidade para os gerentes de qualidade local de cada organização participante do projeto. Eles enviarão esses formulários para os avaliadores finais. O respectivo avaliador local realiza as ações descritas nas sub-atividades, fiscalizando o cumprimento de cada sub-atividade, e finalmente, realiza uma avaliação dos resultados (requisito é cumprido ou não cumprido), em seguida os formulários preenchidos retornam para o coordenador da gestão de qualidade (CGQ).

O coordenador examina todas as análises, e a partir das avaliações, prepara um documento que consolida o conjunto de informações sobre as avaliações. Para cada característica de qualidade, apresenta o número de exigências que não foram realizadas. Posteriormente, o documento é enviado para todos os locais e os requisitos não completos são discutidos com o objetivo de melhoria.

A metodologia de gestão da qualidade garante que os objetivos definidos no início do projeto não se percam durante o seu desenvolvimento. Especialmente em um projeto com ambiente de desenvolvimento distribuído em que existe a possibilidade de dependências de tarefas e que ninguém se sinta responsável por uma determinada demanda.

4. Comparação entre as abordagens

Todas as abordagens apresentadas nesse trabalho, falam de alguma maneira do gerenciamento de projetos em ambientes de desenvolvimento distribuído. A tabela 1 apresenta uma comparação entre os trabalhos descritos anteriormente.

Table 1. Comparação entre as abordagens apresentadas

	Abordagem de Zanoni [2002]	Abordagem de Prikladnicki [2003]	Abordagem de Enami [2006]	Abordagem de Komi-Sirvi e Tihinen [2002]	Abordagem de Gilberg e Grabow [2005]
Trabalho	Dissertação apresentada a PUC do Rio Grande do Sul	Dissertação apresentada a PUC do Rio Grande do Sul	Dissertação apresentada a Universidade Estadual de Maringá	Survey apresentado no Journal Knowledge and Process Management	Artigo escrito para o GIWIS - German-Italian Workshop on Information Systems
Objetivo	Propor um modelo de gerenciamento de projetos para o desenvolvimento distribuído de software	Propor um modelo de referência para a área de desenvolvimento distribuído de software	Apresentar um modelo de gerenciamento de projetos para o desenvolvimento distribuído de software.	Levantar e compartilhar lições aprendidas do processo de desenvolvimento em um ambiente distribuído	Descrever um modelo de qualidade para desenvolvimento distribuído de produtos de software de alta segurança
Alcance	Projetos de software que utilizem desenvolvimento espiral, UML, UP, e exista distribuição física dos participantes	Projetos com DDS e empresas que buscam melhorar o nível de maturidade e capacidade	Criado para um Ambiente de DDS integrado ao ambiente <i>DiSEN</i>	Empresas de Software em um contexto de DDS	Empresas de Software em um contexto de DDS

5. Conclusões e Trabalho Futuros

O estudo buscou apresentar, baseado em vários trabalhos que abordam o desenvolvimento de software com equipes geograficamente dispersas, através de desafios e lições aprendidas, o que muda no gerenciamento de projetos em um contexto de DDS.

Nesse sentido, realizou-se um levantamento teórico em alguns trabalhos que abordam o assunto em questão, para se identificar os principais conceitos, desafios e boas práticas encontrados em um ambiente de DDS relacionados ao gerenciamento de projetos de software.

Com base em tais trabalhos foi possível constatar que a gerência de projetos requer um esforço adicional em um ambiente de DDS, pois, com a equipe geograficamente distribuída, as estratégias de gerenciamento devem ser adaptadas, de forma a dar suporte aos novos desafios que este cenário apresenta.

Por fim, para trabalhos futuros recomenda-se que a partir da identificação das dificuldades e boas práticas de gerenciamento de projetos em ambiente de DDS, e da análise de como alguns modelos atendem a esse cenário de desenvolvimento, propor uma abordagem para o gerenciamento distribuído, fornecendo um maior suporte às atividades do ambiente de DDS.

References

- Audy, J.; Prikładnicki, R. (2008) *Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas*. Rio de Janeiro: Elsevier.
- Branco, E. C.; Belchior A. D. (2002) Um modelo para avaliação da qualidade da gerência de projetos de software. I Simpósio Brasileiro de Qualidade de Software, Gramado, RS, Brasil.
- Enami, L. N. M. (2005) Um modelo de gerenciamento de projetos para um ambiente de desenvolvimento distribuído de software. Dissertação de Mestrado, Universidade Estadual de Maringá, PR, Brasil.
- Espindola, R.; Lopes L.; Prikładnicki, R.; Audy, J. (2005) Uma abordagem baseada em gestão do conhecimento para gerência de requisitos em desenvolvimento distribuído de software. VIII Workshop em Engenharia de Requisitos, Porto, Portugal.
- Frame, J. Davidson. (1995) *Managing Projects In Organizations*, São Francisco: Jossey-Bass inc.
- Gilberg, J. e Grabow, H. (2005) *Quality Management in Distributed Software Development*. GIWIS - German-Italian Workshop on Information, Regensburg.
- Heldman, K. (2006) *Gerência de Projetos: guia para o exame oficial do PMI*. Rio de Janeiro: Elsevier.
- Herbsleb, J. D.; Moitra, D. (2001) *Global Software Development*. IEEE Software.
- Kiel, L. (2003) *Experiences in Distributed Development: A Case Study*. Workshop on Global Software Development at ICSE, Oregon, EUA.
- Komi-Sirvi, S; Tihinen M. (2005) *Lessons Learned by Participants of Distributed Software Development*. *Journal Knowledge and Process Management*, vol. 12 nº 2 p. 108–122.
- Kotlarsky, J. (2005) *Management of Globally Distributed Component-Based Software Development Projects*. Tese de Doutorado. Erasmus University Rotterdam.
- Lopes, L.; Majdenbaum A.; Audy, J. (2003) Uma proposta para processo de requisitos em ambientes de desenvolvimento de software. VI Workshop em Engenharia de Requisitos do SBC, Anais do WER'03, v. 1. p. 78-92. Piracicaba, SP, Brasil.
- PMI (2004) *Conjunto de Conhecimentos em Gerenciamento de Projetos*, 3.ed., Pennsylvania: Project Management Institute Publications.
- Prikładnicki, R. (2003) *Munddos: Um Modelo de Referência para Desenvolvimento Distribuído de Software*. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, Brasil.
- Prikładnicki, R. (2004) *Munddos: Um modelo de referência para desenvolvimento distribuído de software*. XVIII Simpósio Brasileiro de Engenharia de Software, Brasília, DF, Brasil.
- Torreão, P. G. B. C. (2005) *Project Management Knowledge Learning Environment: Ambiente de Aprendizado para Educação em Gerenciamento de Projetos*. Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife, PE, Brasil..
- Zanoni, R. (2002) *Modelo de Gerência de Projeto Baseado no PMI para ambientes de Desenvolvimento de Software Fisicamente Distribuído*. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS, Brasil.