

# Fatores que Afetam o Desenvolvimento Distribuído de Software

Rodrigo G. C. Rocha, Alan Kelon de O. Moraes, Silvio Lemos Meira

Centro de Informática – Universidade Federal de Pernambuco (UFPE)  
Caixa Postal 7851, Cidade Universitária – 50.732-970 – Recife – PE – Brasil  
{rgcr, akom, srlm}@cin.ufpe.br

**Abstract.** *For many years the software development has become essential to the global market. In the past decade, as a reflection of globalization, software companies began to distribute their development processes in different places, creating the Distributed Software Development. This approach inherited problems from the traditional way and for several reasons added other difficulties. This work aims at analyzing data of distributed projects in order to identify how and which factors - technical, human or organizational - influence the performance of projects with distributed teams, with special focus on their influence in the cancellation, delay of schedule and increased costs of projects.*

**Resumo.** *Há muitos anos o desenvolvimento de software se tornou fundamental para o mercado global. Na década passada, como reflexo da globalização, empresas de software começaram a distribuir seus processos de desenvolvimento em lugares diferentes, criando o desenvolvimento distribuído de software. Esta abordagem herdou os problemas existentes na forma tradicional e por diversas razões acrescentou outras dificuldades. O objetivo deste trabalho será levantar e analisar dados referentes a projetos distribuídos a fim de identificar que fatores, em qual eixo – técnico, humano ou organizacional – e como eles influenciam no desempenho dos projetos com equipes distribuídas, dando enfoque especial nas suas influências no cancelamento, atraso de cronogramas e aumento de custos dos projetos.*

## 1. Caracterização do problema

De acordo com pesquisa realizada, 44% das empresas já tiveram projetos cancelados devido a problemas na execução (Bennatan, 2006). Existem vários fatores técnicos que podem prejudicar o progresso dos projetos. Um deles é o desenvolvimento de software com equipes geograficamente distribuídas. Tal fator demanda boa comunicação entre os responsáveis pelo design das partes do software que serão desenvolvidos em diferentes localidades (Cataldo et al., 2006).

Projetos de software são formados por atividades temporárias, com objetivos, custos e responsabilidades definidas. Portanto fica visível que é necessário desenvolver métodos que auxiliem na identificação prematura dos fatores que levam ao fracasso dos projetos distribuídos.

## 2. Fundamentação teórica

Desenvolver software no mesmo espaço físico, na mesma organização ou até no mesmo país, tem se tornado cada vez mais custoso e menos competitivo (Audy e Prikladnicki, 2007). 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).

Brooks (1978) argumenta o software possui muitas características que lhe fazem unicamente apropriado para esta abordagem e o diferencia de outros setores, pois pode ser replicado, transmitido, corrigido e usado sobre distâncias ilimitadas com custo virtual zero. Contudo, também é sujeito a mudanças, é intangível e pode tornar-se potencialmente complexo. Desta forma, o Desenvolvimento Distribuído tem se apresentado nos últimos anos como uma alternativa para o desenvolvimento de software. É um fenômeno que vem crescendo desde a última década e que 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 (Meyer,

2006).

Freitas (2005) afirma que existem diversas razões que motivam a adoção do DDS, sendo as principais:

- Necessidade de profissionais qualificados em áreas especializadas;
- Incentivos fiscais para o investimento em pesquisas em informática;
- Disponibilidade de mão-de-obra especializada e de custos reduzidos em países em desenvolvimento;
- Realização de etapas do desenvolvimento de software perto dos clientes;
- Redução dos prazos de entrega proporcionada pelo desenvolvimento round-the-clock;
- Formação de organizações e de equipes virtuais para aproveitar oportunidades de mercado;
- Necessidade de integrar recursos resultantes de aquisições e fusões organizacionais

Karolak (1998) define uma instância do DDS chamado-a de desenvolvimento global de software (Global Software Development – GSD) quando a distância física entre os desenvolvedores em um ambiente de DDS envolve mais de um país. O GSD é instanciado através de equipes globais de desenvolvimento de software (Global Software Teams).

Audy e Prikladnicki (2007) explicam que o desenvolvimento tradicional de software sempre se apresentou de forma complexa e que o Desenvolvimento Distribuído de Software acrescentou outros desafios ao processo ao adicionar fatores como dispersão física, distância temporal e diferenças culturais.

Como citado anteriormente, com o surgimento do DDS, novos desafios foram acrescentados ao desenvolvimento de software. Carmel (1999) afirma a existência de cinco fatores que podem levar ao fracasso uma equipe distribuída: (1) comunicação ineficiente; (2) falta de coordenação; (3) dispersão geográfica; (4) perda do espírito de equipe; e, (5) diferenças culturais, chamadas de forças centrífugas. Além disso, cita também a existência de seis fatores que podem levar a equipe ao sucesso: (1) infraestrutura de comunicação; (2) arquitetura do produto; (3) construção de uma equipe, (4) metodologia de desenvolvimento; (5) tecnologia de colaboração; e, (6) técnicas de gerência, chamados de forças centrípetas.

Komi-Sirvo e Tihinen (2002), baseados em suas pesquisas também apresentam alguns áreas problemáticas e fatores associados quando o desenvolvimento é distribuído: (1) ambientes e ferramentas de desenvolvimento; (2) comunicação; (3) engenharia de requisitos; (4) gerenciamento de projetos; (5) diferenças culturais e, (6) tempo e orçamento acima do estimado, entre outros. Os fatores citados por Carmel, Komi-Sirvo e Tihinen, entre outros fatores que podem influenciar diretamente e indiretamente o desenvolvimento de um projeto distribuído são descritos abaixo.

## **2.1 Gerência de Projeto**

Cleland e Ireland (2002) citam que as principais funções da gerência de projetos são: planejamento, organização, motivação, direção e controle. Algumas pesquisas indicam que os problemas de gerência de projetos em ambientes distribuídos não diferem significativamente dos problemas no desenvolvimento tradicional (co-localizado). Entretanto, no contexto distribuído é reforçada a importância das técnicas formais de gerenciamento de projetos, como, por exemplo, as apresentadas no PMBOK (Audy e Prikladnicki, 2007).

A gerência de projeto, no que diz respeito à coordenação e ao controle, se torna extremamente difícil no DDS. A integração entre as diversas partes e módulos do projeto deve acontecer de modo eficiente para que o fato de estarem longe fisicamente não interfira (Souza, 2007).

Prikladnicki (2003) afirma que o gerenciamento de projetos de DDS exige uma adaptação de algumas técnicas utilizadas em projetos co-localizados, de forma a suportar e reduzir as dificuldades impostas pela dispersão da equipe.

## **2.2 Processos de Software**

Evaristo (et al., 2005) afirmam que um processo de desenvolvimento é fundamental, já que a sincronização de atividades é um dos objetivos principais e que quando as equipes distribuem o processo de desenvolvimento em diversas localidades, a falta de sincronização pode se tornar crítica.

Angionini, Sanna e Soro (2005) relatam a proximidade entre Metodologias Ágeis e DDS citando algumas características comuns para ambos, como *feedback* contínuo, *releases* frequentes, valorização da comunicação e padrões de codificação. Além disso, Angionini, Sanna e Soro apresentam e descreve o MAAD (Methodology for Agile Distributed Development), que é um processo baseado em metodologias ágeis, criado inicialmente com foco na comunidade open source. Essa metodologia foi idealizada a partir da avaliação de outros processos de desenvolvimento de software voltado para o ambiente distribuído, como é o caso do DXP (Distributed eXtreme Programming) (Kircher, 2001).

### **2.3 Distância socio-cultural**

Souza (2007) relaciona o fator socio-cultural principalmente no que se refere à confiança entre as equipes envolvidas. Para que um projeto tenha um desenvolvimento eficiente, é necessário que os integrantes localizados em ambientes diferentes tenham confiança entre os participantes e também no projeto.

Quando o contexto se trata de desenvolvimento global de software existem diferenças culturais, por se tratar de pessoas espalhadas ao redor do mundo que possuem diferentes hábitos, crenças, atitudes com relação à hierarquia, senso de tempo e estilos de comunicação. Souza (2007) fala que as diferenças culturais podem influenciar em diversas decisões dentro do projeto devido a determinadas tradições e costumes de cada país e que há necessidade de que as diferenças culturais sejam minimizadas ao máximo para que não interfiram no resultado final do produto.

Audy e Prikladnicki (2007) asseguram que muitas empresas trabalham com o conceito de *liaison*, ou seja, uma pessoa que desempenha o papel de ponte entre duas ou mais culturas, justamente por entender ou já as ter vivido.

### **2.4 Motivação**

De acordo com Olson e Olson (2004), a motivação dos indivíduos também difere de país para país dependendo das suas culturas. Em países onde existe a valorização do individualismo, as pessoas buscam ganho material e reconhecimento pessoal. Já, em outros, onde a ênfase está voltada ao coletivo, busca-se relacionamentos pessoais e família. Os sistemas de recompensa devem levar em conta esses valores, recompensando cada grupo com valores monetários ou dias de folga de acordo com seus valores.

Equipes são formadas por pessoas, logo, são unidades sociais frágeis que por diversas razões (fraca comunicação, distância temporal e distância geográfica, infra-estrutura utilizada, falta de comunicação informal, diferença cultural e o tamanho do grupo, entre outros) se tornam vulneráveis. Evaristo e Scudder (2000) asseguram que a confiança é fundamental para o time manter o espírito de equipe, que é difícil, porém essencial manter a confiança no desenvolvimento de um projeto distribuído.

### **2.5 Distância Geográfica e Temporal**

Quando a distância entre envolvidos em projetos distribuídos ultrapassa 30 metros, a frequência de comunicação diminui para o nível idêntico ao de envolvidos que estão distribuídos a centenas/milhares de metros (Herblesb et al. 2001 apud Allen 1977).

Evaristo e Scudder (2000) afirmam que o alto nível de dispersão proporciona maior dificuldade para monitorar o comportamento entre diferentes grupos com relação a outros grupos. Audy e Prikladnicki (2007) em seu trabalho, citam que é necessário entender o nível de distância existente, pois, com isso é possível ajudar na identificação de possíveis fontes de dificuldades ou apenas caracterizar melhor a distribuição física das equipes envolvidas em projetos distribuídos. E desta forma

apresentam quatro níveis de dispersão física:

- Mesma localização física: situação em que a empresa possui todos os atores em um mesmo local e as reuniões face a face podem acontecer sem dificuldade;
- Distância nacional: as pessoas envolvidas nessa situação estão no mesmo país, porém estados e/ou cidades diferentes. Já surgem dificuldades para realizar reuniões face a face;
- Distância continental: atores participando em países do mesmo continente;
- Distância global: atores localizados em países e continentes diferentes. Nessa situação assim como na distância continental reuniões face a face se tornam bem mais difíceis de acontecer, geralmente quando acontecem são no início do projeto;

Além da dispersão física, a dispersão temporal exerce efeitos sobre o DDS. Participantes de uma equipe estão dispersos no tempo quando há diferença nos horários de trabalho, fusos horários e/ou ritmos de trabalho que diminuam o tempo disponível para interação síncrona (Audy e Prikladnicki, 2007).

## **2.6 Complexidade e Tamanho do Projeto**

Herbsleb (2000) afirma que o nível de complexidade do projeto também pode afetar a performance de projetos distribuídos e que a dimensão do projeto também pode ser fator de complexidade. Souza (2007) cita que normalmente projetos com desenvolvimento distribuído de software são complexos e de grande porte, devido ao fato do investimento inicial neste tipo de projeto ser elevado, e podendo envolver diversos países. A utilização de metodologias tradicionais e mais completas é necessária nesse aspecto.

## **2.7 Comunicação**

Diversos pesquisadores Carmel (1999), Audy e Prikladnicki (2007), Trindade (2008), Souza (2007), entre outros, garantem que a comunicação é fator fundamental para o desenvolvimento de projetos distribuídos.

Souza (2007) assegura que a comunicação, em todos os seus sentidos, é um dos grandes problemas do desenvolvimento distribuído. É preciso que haja meios eficientes que ultrapassem as barreiras impostas pelo desenvolvimento em locais diferentes. A definição de interfaces de comunicação formal pode ser obtida por meio de modelos de processo bem definidos, com marcos (milestones) e métricas bem estabelecidas.

Rocha et al (2008) citam em sua experiência que algumas ferramentas foram fundamentais para a comunicação entre os membros, tais como comunicadores instantâneos ou messengers e um website, mas, que ferramentas comuns de comunicação (e-mail, messengers, etc) precisariam evoluir para poder suportar melhor o trabalho distribuído.

## **2.8 Ambientes e Ferramentas de Desenvolvimento**

Komi-Sirvo e Tihinen (2002), afirmam que embora a infra-estrutura técnica disponível atualmente pareça proporcionar o apoio adequado para o desenvolvimento distribuído, pesquisas realizadas com empresas que possuem projetos distribuídos, indicaram que ferramentas e ambientes de desenvolvimento ainda não apóiam de maneira eficiente o desenvolvimento distribuído.

Komi-Sirvo e Tihinen (2002) citam que a incompatibilidade de ferramentas e versões usadas por sites de desenvolvimento diferentes é muito comum, e o estabelecimento de um ambiente uniforme é uma tarefa desafiadora. Além disso, as ferramentas de desenvolvimento são baseadas no pressuposto que as redes de comunicações são extremamente rápidas, o que nem sempre ocorre.

## **3. Metodologia e estado atual do trabalho**

Será realizada uma Revisão Sistemática, que tem como objetivo principal identificar os diversos fatores inseridos que influenciam os projetos de software no contexto distribuído, a partir disto, identificar um

dos fatores que possua uma importante lacuna ou maior carência, para assim, projetar uma outra pesquisa mais aprofundada visando suprir tal carência/necessidade. Após a revisão sistemática será realizado o delineamento do escopo da pesquisa de campo e do estudo de caso, logo, aplicar a pesquisa, o estudo de caso, em seguida coletar e analisar os dados. O estado atual do trabalho se resume no estado da arte, onde a revisão bibliográfica foi feita em revistas científicas, artigos científicos e livros, abrangendo importantes periódicos e conferências da IEEE, ACM, Springer e Elsevier.

#### 4. Trabalhos relacionados

Komi-Sirviö e Tihinem (2005) descreve em sua pesquisa quais os principais fatores que afetam o DDS, alguns deles: Ambiente e Ferramentas de Desenvolvimento, Comunicação, Conhecimento do Projeto, Gerência do Projeto, Diferenças Culturais, entre outros, além de apresentar um gráfico onde exibe a porcentagem dos fatores que mais influenciam. Smite (2007) apresenta em seu trabalho o que ele chama de as mais perigosas ameaças globais, onde apresenta gráficos e descreve a gravidade de cada ameaça caracterizando pela sua frequência de ocorrência e efeito sobre o projeto através de diferentes critérios, como: orçamento, cronograma, a satisfação dos clientes e outros.

#### 5. Resultados esperados

O objetivo deste trabalho será levantar e analisar dados referentes a projetos distribuídos a fim de identificar que fatores, em qual eixo – técnico, humano ou organizacional – e como eles influenciam no desempenho dos projetos com equipes distribuídas, dando enfoque especial nas suas influências no cancelamento, atraso nos cronogramas e aumento de custos dos projetos. Isso proporcionará a descoberta das causas do fracasso desses trabalhos e quais são os principais requisitos observados para o sucesso dos projetos de software neste modelo.

Os possíveis resultados esperados:

- **Identificar os principais fatores que influenciam o desempenho de projetos distribuídos :** Observar os principais fatores relatados por empresas que utilizam o DDS.
- **Classificação dos mesmos nos eixos técnico, humano e organizacional:** Classificar os fatores em técnico, humano e organizacional, possibilitando uma maior facilidade para o estudo aprofundado de algum fator específico ou de todos.
- **Identificação das relações entre fatores:** identificar as relações entre tais fatores, como eles contribuem ou atrapalham os projetos, criando-se assim um catálogo;
- **Estudo de Caso:** através de um estudo de caso realizar a avaliação do catálogo em projetos de empresas da região que exportem software, por exemplo, empresas do Porto Digital;
- **Análise do Estudo de Caso:** validando as hipóteses definidas e mapeando os resultados esperados.

Espera-se, com a realização da pesquisa, incentivar outros estudos que favoreçam a diminuição de falhas desses projetos. Os resultados dessa pesquisa podem ser de grande relevância para que as empresas locais utilizem processos que se adequem às limitações técnicas identificadas.

#### 6. Referências

- Allen, T. J. (1977) *Managing the Flow of Technology*. Cambridge, MA: MIT Press.
- Angioni, M., Sanna, R., Soro, A. (2005) *Defining a Distributed Agile Methodology for na Open Source Scenario*. Proceedings of the First International Conference on Open Source Systems, Genova. p. 209- 214.
- Audy, J.; Prikladnicki, R. (2007) *Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas*. Rio de Janeiro: Elsevier.
- Bennatan, E. M. (2006) *Catastrophedisentanglement: getting software projects back on track*. Cambridge, Pearson Education.
- Brooks, F. P. (1978). *The Mythical Man-Month: Essays on Softw.* 1st. Addison-Wesley Longman

Publishing Co., Inc.

- Carmel, E. (1999) *Global Software Teams: Collaboration Across Borders and Time Zones*. Prentice-Hall, EUA.
- Cataldo, Marcelo e Bass, Matthew e Herbsleb, James D & Bass, Len. (2006). *Managing Complexity in Collaborative Software Development: On the Limits of Modularity*. Workshop on Supporting the Social Side of Large-Scale Software Development, Conference on Computer Supported Cooperative Work (CSCW'06), Banff, Alberta, Canada.
- Cleland, D., Ireland, L. (2002). *Gerência de projetos*. 1º edição. Rio de Janeiro: Reichmann & Affonso Editores, 324 p.
- Evaristo, R., Audy, J. L. N., Prikladnicki, R., Pilatti, L., Lopes L. (2005) “Innovation in Information Systems Education - V: The Management of Outsourcing: Development of a Module with Implications for the IT Curriculum”. *Communications of the Association for Information Systems*, Volume 15, Article 21.
- Evaristo, Roberto; Scudder, Richard. (2000) Geographically distributed project teams: a dimensional analysis. In: *HICSS, 2000, Havaí. Proceedings...*, EUA, p. 1-15.
- Herbsleb, J. D.; Moitra, D. (2001). Guest editors' introduction: global software development. *IEEE Software*.
- Herbsleb, J. D.; Mockus, A.; Ingholt T. A.; Grinter, R. E. (2001). “An empirical study of speed and communication in globally-distributed software development”, *IEEE Transactions on Software Engineering*, 29:6, pp. 481-494.
- Herbsleb, J. D. (2007). *Global Software Engineering: The Future of Socio-technical Coordination*. *IEEE Computer Science*. p188-198.
- Karolak, D. W. (1998) *Global software development – managing virtual teams and environments*, *IEEE Computer Society*, EUA.
- Kircher, M., Jain, P., Corsaro, A., Levine D. (2001). *Distributed eXtreme Programming*, *Proceeding of XP 2001*, Villasimius, Italia. p. 66-71.
- Kircher, M. (2001). *eXtreme Programming in Open-Source and Distributed Enviroments*, *JAOO (Java And Object-Orientation) conference*, Aarhus, Dinamarca.
- Komi-Sirvo, S; Tihinen M. (2005). *Lessons Learned by Participants of Distributed Software Development*. *Journal Knowledge and Process Management*, vol. 12 n° 2 p. 108–122.
- Meyer, B. (2006). *The unspoken revolution in software engineering*. *IEEE Computer*.
- Olson, J. S., Olson, G. M. (2004). *Culture Surprises in Remote Software Development Teams*. *ACM Queue*, New York, v. 1, n. 9, pp 52-59.
- Prikladnicki, 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.
- Rocha, R. Arcoverde D. Brito, R. Aroxa, B. Costa, C. Silva, F. Q. B. Albuquerque, J. Meira, S. R. L. (2008). *Uma experiência na adaptação do RUP para pequenas equipes de desenvolvimento distribuído*. *II Workshop de Desenvolvimento Distribuído de Software (SBES-SBBD)*. p. 87-96.
- Smite, Darja. (2007). *Global Software Development Improvement*. *University of Latvia*. Tese.
- Souza, M. (2007) “*Análise do Processo de Desenvolvimento de Software para o Desenvolvimento Distribuído de Software*”. *Dissertação*. Pelotas.
- Trindade, Cleyton C. Moraes, Alan Kelon O. Meira, Silvio Lemos. (2008). *Comunicação em Equipes Distribuídas de Desenvolvimento de Software: Revisão Sistemática*. 5<sup>th</sup> *Experimental Software Engineering*. Salvador.