

# Uma Análise Sociotécnica de Processos de Software sob a Perspectiva do Desenvolvimento com Equipes Distribuídas

Gislaine Camila Lapasini Leal<sup>1</sup>, César Alberto da Silva<sup>2</sup>,  
Elisa Hatsue Moriya Huzita<sup>2</sup>, Tania Fátima Calvi Tait<sup>2</sup>

<sup>1</sup>Departamento de Engenharia de Produção

<sup>2</sup>Departamento de Informática – Universidade Estadual de Maringá (UEM)  
Maringá, PR – Brazil

gclleal2@uem.br, cesaralberto91@hotmail.com, {ehmhuzita, tait@din.uem.br}

**Abstract.** *A growing number of companies that are distributing their software development processes around the world in search of increased productivity, cost reduction and improvements in quality. This strategy added new factors to the development in relation to processes of communication, coordination and control of projects. This paper aims to contribute to the Software Engineering, with some sociotechnical analysis of development processes from the perspective of development with distributed teams.*

**Resumo.** *O número de empresas que estão distribuindo seus processos de desenvolvimento de software ao redor do mundo, em busca de aumento de produtividade, redução de custos e melhoria na qualidade, é cada vez maior. Essa nova estratégia acrescentou novos fatores ao desenvolvimento no que se refere aos processos de comunicação, coordenação e controle dos projetos. Este trabalho tem por objetivo contribuir com a Engenharia de Software, através de uma análise sociotécnica de alguns processos de desenvolvimento sob a perspectiva do desenvolvimento com equipes distribuídas.*

## 1. Introdução

O processo de software é definido como sendo um conjunto ordenado de atividades para o gerenciamento, desenvolvimento e manutenção de software, e deve estar alinhado com as condições organizacionais [Fuggetta 2000]. Os processos de software são, conforme literatura corrente [Sengupta et al. 2006], [Rocha et al. 2008], [Paasivaara and Lassenius 2004], voltados para o desenvolvimento de projetos coloados.

O Desenvolvimento Distribuído de Software (DDS) é uma abordagem recente para o desenvolvimento dos projetos de software que vem ao encontro das necessidades da globalização. Essa nova estratégia adicionou ao desenvolvimento de software alguns desafios relacionados às diferenças culturais, dispersões geográficas, coordenação e controle, comunicação e espírito de equipe.

As questões sociotécnicas relacionadas a processos de software tem adquirido, cada vez mais, relevância, como pode ser visto em [Santos 2008] e [Junior and Santos 2009], uma vez que o software não é apenas um conjunto de objetos técnicos. Eles são projetados, construídos e usados por pessoas [Damasevièius 2007]. No

contexto de DDS os fatores culturais, conflitos comportamentais e a comunicação entre os *stakeholders* tornam mais explícitas as questões sócio-culturais [Cibotto et al. 2009].

Neste contexto, é relevante analisar sob a perspectiva sociotécnica alguns processos de software utilizados no desenvolvimento com equipes distribuídas. O texto encontra-se estruturado em cinco seções, além da introdução. A Seção 2 descreve a metodologia adotada. O Desenvolvimento Distribuído de Software é apresentado na Seção 3. Na Seção 4 são retratados alguns conceitos sobre processos de desenvolvimento. Na Seção 5 são apresentados os aspectos sociotécnicos sob os quais os processos selecionados são analisados. Por fim, na Seção 6 são realizadas as considerações finais.

## **2. Metodologia de Pesquisa**

A pesquisa conduzida neste trabalho se caracteriza como sendo de base qualitativa e do tipo exploratória. Não há a preocupação de quantificar os dados e o estudo objetiva proporcionar uma maior familiaridade com o problema com vistas a torná-lo explícito [Silva and Menezes 2000].

A metodologia adotada é composta pelas seguintes etapas:

1. revisão inicial da literatura, que teve como objetivo formar um referencial teórico consistente para a continuidade do estudo;
2. identificação dos processos de desenvolvimento utilizados com equipes distribuídas;
3. definição dos critérios para análise dos processos considerando os aspectos sociotécnicos envolvidos;
4. investigação sociotécnica dos processos selecionados.

## **3. Desenvolvimento Distribuído de Software**

No cenário atual, as mudanças ocorrem rapidamente, exigindo das organizações um esforço muito maior para se manterem competitivas. Para tanto, seus processos e produtos precisam estar em constante e acelerada evolução para acompanhar as demandas do mercado. A globalização retira as barreiras que definem um mercado, trazendo para o mesmo patamar os competidores locais e globais [Audy and Prikladnicki 2008]. Com isso, tem-se observado um grande investimento na conversão de mercados nacionais em mercados globais, criando novas formas de competição e colaboração [Herbsleb et al. 2000].

Em busca de vantagem competitiva, diversas organizações optaram por distribuir o processo de desenvolvimento de software configurando o Desenvolvimento Distribuído de Software (DDS) [Huzita et al. 2008]. Nessa estratégia o processo é acelerado devido a: redução de custos; acesso à mão de obra e recursos; avanços na infraestrutura (Internet e desenvolvimento e integração de ferramentas); vantagens de novos mercados; rápida formação de equipes virtuais; e, melhoria do *time-to-market*, com o uso do desenvolvimento "em torno do sol" [Begel and Nagapan 2008].

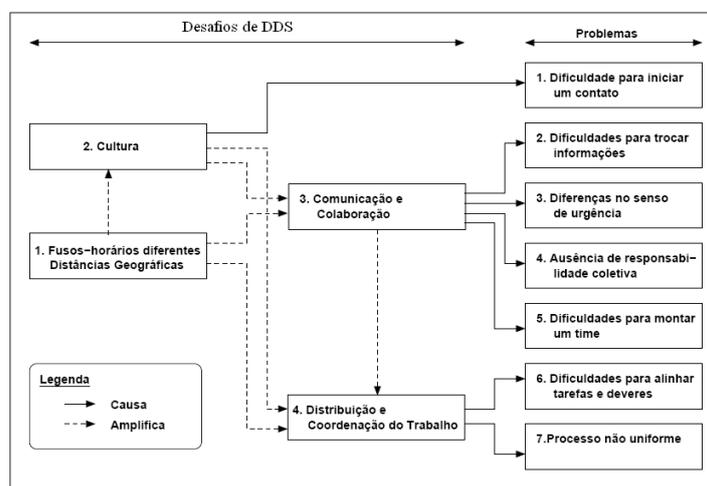
O DDS tem sido caracterizado, principalmente, pela colaboração e cooperação entre departamentos de organizações e pela criação de grupos de desenvolvedores que trabalham em conjunto, mas estão localizados em cidades ou países diferentes, distantes temporal e fisicamente [Audy and Prikladnicki 2008]. Segundo Carmel [Carmel 1999], os projetos de DDS consistem de times trabalhando juntos para atingir objetivos comuns em um mesmo projeto, porém dispersos geograficamente.

Os times distribuídos enfrentam desafios relacionados a fatores técnicos e não técnicos. Os fatores técnicos referem-se a problemas de conectividade em rede e às diferenças entre os ambientes de desenvolvimento e teste. Quanto aos fatores não técnicos, têm-se as questões de confiança, comunicação, conflitos e diferenças culturais [Damian 2002].

A divisão dos fatores entre técnicos e não-técnicos tem sido discutida na área de Engenharia de Software, conforme colocado por [Cukierman et al. 2007]. Segundo estes autores, não se pode menosprezar os aspectos não técnicos devendo, também, os mesmos serem tratados sob a abordagem sócio-técnica.

Os principais desafios encontrados em DDS relacionam-se às diferenças culturais, dispersões geográficas, coordenação e controle, comunicação e espírito de equipe. As diferenças culturais, geralmente, acentuam os problemas de comunicação, e podem levar a estágios de frustração, descontentamento e até mesmo desentendimento entre as equipes. Holmstrom et al. [Holmstrom et al. 2006], ressaltam que a cultura pode ter um efeito determinante na forma como as pessoas interpretam determinadas situações, e como elas reagem a estas situações. As diferenças culturais envolvem a cultura organizacional, cultura nacional, idioma, política, motivação individual e a ética no trabalho das equipes distribuídas [Carmel 1999].

A Figura 1 apresenta os desafios do DDS e suas correlações. As linhas tracejadas apontam os desafios amplificados por outros desafios e as linhas contínuas apontam para os problemas decorrentes desta amplificação. As grandes distâncias geográficas afetam negativamente a comunicação, e, conseqüentemente, da mesma forma, na coordenação e no controle [Clerc et al. 2007].



**Figura 1. Desafios em DDS e suas correlações [Clerc et al. 2007].**

A coordenação consiste no ato de integrar cada atividade com cada unidade organizacional. A integração geralmente requer comunicação intensa e constante. O controle é o processo de adesão aos objetivos, políticas, padrões e níveis de qualidade. Para Mockus e Herbsleb [Mockus and Herbsleb 2001], a coordenação se torna um problema porque os processos de cada time distribuído são diferentes, isto é, não existe um processo uniforme.

A comunicação é o fator de mediação que afeta diretamente a coordenação e o

controle, sendo considerada o componente essencial de todas as práticas de colaboração no processo de desenvolvimento de software. Ela representa a troca de informações não ambíguas e completas, isto é, o emissor e o receptor conseguem chegar a um entendimento comum. Segundo Herbsleb et al. [Herbsleb et al. 2000], a comunicação informal exerce um papel chave no sucesso das equipes distribuídas. O impacto que a falta de comunicação eficiente pode causar em ambientes DDS é o resultado de um baixo nível de confiança entre as equipes e a perda da visibilidade sobre o andamento dos trabalhos [Damian 2002].

#### 4. Processo de Desenvolvimento de Software

A estrutura de um processo de software é, geralmente, composta pelos seguintes elementos: Ator, Papel, Artefato e Atividade [OMG 2005]. O Ator representa as entidades que executam um processo ou assumem um papel durante a execução de uma tarefa. Um papel significa um grupo de responsabilidades que executam uma atividade específica do processo de software. Um artefato é uma porção representativa de informação do produto que resulta de uma atividade e pode ser utilizado posteriormente como matéria-prima para gerar novos artefatos. Dessa forma, uma atividade pode consumir artefatos (de entrada) e gerar novos artefatos (de saída).

Segundo [Berger 2003], o uso de um processo padrão como base para o planejamento permite aos gerentes de projeto definir planos em conformidade com os padrões de qualidade e procedimentos da organização. Uma equipe de desenvolvimento que trabalha sem um processo bem definido acaba funcionando de maneira *ad hoc*, o que compromete o seu sucesso. Por outro lado, organizações maduras, que empregam um processo bem definido, podem desenvolver sistemas complexos de maneira consistente e previsível, independente de quem o produziu [Kruchten 2003].

[Teixeira and Cukierman 2007] ressaltam que um processo de desenvolvimento de software deve contemplar os fatores não-técnicos envolvidos pois, o sucesso de um projeto não se relaciona apenas com o cumprimento de atividades e boas práticas e que as falhas não decorrem apenas desses fatores. Um processo não é uma atividade puramente técnica, mas também social à medida que contempla a estrutura organizacional e cultural. As estruturas sociais influenciam e determinam o comportamento no trabalho e geração dos artefatos [Damasevièius 2007]. Desse modo, um processo de desenvolvimento deve contemplar os fatores não-técnicos.

Os processos que foram experimentados no desenvolvimento com equipes distribuídas, foram:

- *Rational Unified Process (RUP)*: é um processo iterativo, incremental e adaptável de desenvolvimento de software que pode ser customizado para diversos tipos e tamanhos de produtos e projetos de software. Papéis, atividades, artefatos, disciplinas e fluxo de trabalho constituem-se em elementos do RUP. Com isto, RUP permite estruturar os projetos dividindo e definindo claramente as atividades (tarefas e papéis) e, também, definindo uma coleção de artefatos sobre os quais se baseia a comunicação entre os desenvolvedores [Kruchten 2003].
- *Extreme Programming (XP)*: é uma metodologia de desenvolvimento ágil, com iterações curtas, e se baseia em quatro valores: comunicação, simplicidade, *feedback* e coragem. Tem por objetivo tornar o projeto flexível, diminuir o custo a

possíveis mudanças e utilizar o código gerado como um indicador de progresso do projeto. [Beck 2004].

- *Agile Unified Process (AUP)*: é uma versão simplificada do RUP, que utiliza técnicas e conceitos de métodos ágeis para descrever de uma maneira simples e fácil de compreender o desenvolvimento de software [Ambler 2008].
- *Extended Workbench Model*: tem sido amplamente utilizado na Siemens. Nessa abordagem há uma pequena equipe central que realiza o planejamento do projeto, definindo os papéis e responsabilidades dos times remotos, e as atividades no início de cada fase, por exemplo, análise de requisitos, projeto de arquitetura e plano de projeto. As equipes distribuídas são responsáveis pelas atividades de desenvolvimento e teste [Paulish 2007].
- *Local Agile Game-based Process (LAGPRO)*: é um processo para ser utilizado por equipes locais para reforçar a coordenação e motivação em um contexto de desenvolvimento distribuído de software. É um processo iterativo, em que cada iteração é dividida em fases, as quais têm duração de duas semanas. A estrutura e os princípios gerais do LAGPRO são baseados no AUP, simplificando suas disciplinas e fases. Além disso, o LAGPRO busca introduzir motivação em equipes utilizando elementos de jogos e competição. Segundo Ribeiro et al. [Ribeiro et al. 2006], as tentativas de execução do processo falharam devido à rejeição por parte dos membros do time com relação à característica competitiva.

## 5. Investigação Sociotécnica dos Processos

Os processos RUP e XP foram escolhidos por serem amplamente utilizados, sendo uma metodologia tradicional e uma ágil, respectivamente. Em [Rocha et al. 2008] é apresentado um relato de experiência sobre a adaptação do RUP para pequenas equipes de desenvolvimento distribuído.

O AUP por ser uma simplificação do RUP que utiliza conceitos de métodos ágeis. E os demais, o *Extended Workbench Model* e o LAGPRO por serem utilizados no contexto de DDS, conforme pode ser visto em [Paulish 2007], [Ribeiro et al. 2006], e [Urdangarim 2008].

A seguir será apresentada a análise dos processos de desenvolvimento de software mencionados acima. Para tanto, foi identificado um conjunto de características sociotécnicas, tomando por base as características e necessidades do desenvolvimento distribuído de software e os elementos básicos de um processo de software.

Os elementos utilizados para analisar os processos sob a perspectiva sociotécnica visando identificar qual dos processos melhor atende as necessidades dos projetos DDS, são:

- (1) **Diversidade**: Identifica se o processo analisado oferece mecanismos para controle, coordenação e comunicação das diferenças (cultural e idioma) entre as equipes envolvidas no projeto.
- (2) **Colaboração**: Caracteriza se o processo permite o uso de ferramentas de colaboração entre os times como meio de compartilhamento de informações.
- (3) **Confiança**: Identifica se o processo é capaz de promover confiança entre os membros dos diferentes times.

- (4) Papéis: Identifica se o processo possui uma organização rígida na definição dos papéis a serem desempenhados.
- (5) Atividades: Identifica se o processo define as atividades que devem ser realizadas em cada fase.
- (6) Acompanhamento: Identifica se o processo caracteriza-se por acompanhar periodicamente a evolução dos trabalhos dos times remotos.
- (7) Requisitos: Caracteriza se o processo analisado adota algum método formal para especificação dos requisitos, que tem por objetivo amenizar os problemas de ambiguidades nos artefatos e com isso, minimizar a comunicação.
- (8) Centralizado: Identifica se o processo analisado apresenta uma centralização do controle das atividades do projeto.
- (9) Descentralizado: Identifica se o processo analisado é caracterizado pela descentralização do controle das atividades do projeto.
- (10) Artefatos: Identifica se o processo apresenta a definição dos artefatos requeridos e gerados em cada fase.

Na Tabela 1 os processos identificados são analisados em relação aos critérios mencionados acima. Os fatores sociais são representados pelos critérios de 1 à 6, e os demais relacionam-se aos fatores técnicos.

**Tabela 1. Investigação Sociotécnica dos Processos**

Processo/Critérios	1	2	3	4	5	6	7	8	9	10
RUP				x	x				x	x
XP		x	x	x	x	x			x	x
AUP				x	x				x	x
<i>Extended Workbench Model</i>				x		x	x	x	x	x
LAGPRO		x	x	x		x	x	x	x	x

Na Tabela acima, é possível observar que nenhum dos processos atende todas as necessidades dos projetos desenvolvidos com equipes geograficamente distribuídas. Além disso, no DDS a necessidade de contemplar pessoas é de extrema relevância, como pode ser evidenciado pelos critérios de 1 à 6.

O RUP, o XP e o AUP definem o conjunto mínimo de elementos que compõem um processo. Entretanto, não apresentam procedimentos para tratar as questões relacionadas aos principais problemas de DDS, que são representados pelos critérios 1, 2, 3, 6, 7, 8 e 9.

O *Extended Workbench* e o LAGPRO, embora tenham sido definidos para serem utilizados por projetos que são desenvolvidos de modo distribuído, não possuem mecanismos para explicitar a diversidade, tratam apenas a especificação formal dos requisitos. No *Extended Workbench*, também, não são contemplados os fatores relacionados à colaboração e confiança. Além disso, não foram encontrados relatos sobre os artefatos gerados. E, no caso do LAGPRO, também, não foram identificadas as atividades.

Diante do exposto, é possível evidenciar a necessidade de um processo que atenda essas novas demandas sociotécnicas geradas pela abordagem distribuída.

## 5.1. Características Sociotécnicas para um Processo de Desenvolvimento Distribuído de Software

A partir dos estudos realizados foram identificadas algumas características relacionadas a fatores técnicos e não-técnicos que devem ser contempladas por um processo de desenvolvimento quando da adoção de equipes distribuídas.

As características relacionadas aos fatores técnicos são:

- especificação formal de testes para amenizar os problemas de ambiguidade e reduzir a necessidade de comunicação [[Mullick et al. 2006], [Avritzer et al. 2007] e [Avritzer et al. 2008]];
- uso de arquitetura baseada em componente para distribuição do desenvolvimento ou distribuição por fases, por exemplo desenvolvimento e teste, para reduzir a comunicação entre as equipes [Lings et al. 2007];
- adoção de metodologias de especificação que não apresentem uma curva de aprendizagem alta e que possuam semântica para transmitir informações aos desenvolvedores, no caso a UML por ser conhecida tanto no meio acadêmico como industrial [Sengupta et al. 2006];
- integração contínua para amenizar os eventuais problemas de integração [[Vanzin et al. 2005] e [Paasivaara and Lassenius 2004]]; e,
- definir infraestrutura que possibilite a colaboração, controle da documentação e controle de versão dos artefatos; desenvolver a aplicar suítes de teste; e, estabelecer método para controle da documentação [[Paasivaara and Lassenius 2004] e [Taylor et al. 2006]].

Sob a perspectiva social foram identificadas as seguintes questões:

- gerenciamento híbrido (centralizado-descentralizado) em relação ao controle das atividades [[Mullick et al. 2006], [Avritzer et al. 2007] e [Avritzer et al. 2008]];
- encorajamento da comunicação entre desenvolvimento e equipe de teste de integração oferecendo artefatos com informações relevantes para as duas equipes [Vanzin et al. 2005];
- para auxiliar na percepção, destaca-se a necessidade de identificar as equipes envolvidas, o que será realizado em cada local e as responsabilidades de cada membro [Lings et al. 2007];
- desenvolvimento iterativo e com entregas frequentes para fornecer uma maior visibilidade do projeto aos gerentes [[Paasivaara and Lassenius 2004] e [Taylor et al. 2006]];
- definir uma linguagem para formalização do processo e interação entre as equipes [Vanzin et al. 2005]; e,
- definir um líder para estimular a confiança e compromisso entre os membros.

## 6. Considerações Finais

A crescente busca por maior competitividade tem levado as empresas a adotarem o DDS. Tentando realizar desenvolvimento a baixo custo, empresas têm atravessado fronteiras formando um mercado global. Essa mudança de paradigma tem causado impacto no *marketing*, na distribuição e na forma de concepção, de produção, de projeto, de teste e de entrega do software aos clientes [Herbsleb et al. 2000].

A distribuição física das equipes agrava problemas existentes e inerentes a gerência do desenvolvimento. Diferenças culturais, de linguagem, de fuso horário entre outros aspectos aumentam a complexidade na comunicação, na coordenação e no controle durante o desenvolvimento.

Os processos de software e seus artefatos apresentam diversas perspectivas: a tecnológica, a social, a psicológica e outras. Segundo [Damasevièius 2007], a perspectiva sociotécnica possibilita uma análise mais profunda sobre a relação entre os métodos, técnicas, ferramentas, ambiente de desenvolvimento e estrutura organizacional. [Damasevièius 2007] destaca, também, que é difícil dissociar os aspectos sociais dos tecnológicos, pois eles são mutuamente interdependentes.

Em um processo de desenvolvimento, principalmente num cenário de DDS, compreender a perspectiva sociotécnica possibilita extrair informações relacionadas a: estrutura da equipe de desenvolvimento, relação com o ambiente, compreensão da comunicação e compartilhamento de conhecimentos. Os resultados obtidos deste tipo de análise podem ser utilizados para educar os membros das equipes, difundir as melhores práticas, melhorar o desempenho do processo e a qualidade dos artefatos gerados [Damasevièius 2007].

Este artigo investigou alguns processos que apresentam relatos de uso no desenvolvimento com equipes distribuídas e os analisou, tanto em termos de fatores técnicos quanto não-técnicos. Para tanto foram analisadas técnicas e métodos que compõem os principais processos de software existentes na literatura, que contribuíram para o entendimento das principais atividades, atores e artefatos que os utilizam.

## Referências

- Ambler, S. (2008). *The agile unified process (aup)*.
- Audy, J. and Prikładnicki, R. (2008). *Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas*. Elsevier, Rio de Janeiro, RJ.
- Avritzer, A., Hasling, W., and Paulish, D. (2007). Process investigations for the global studio project version 3.0. In *ICGSE '07: Proceedings of the International Conference on Global Software Engineering*, pages 247–251, Washington, DC, USA. IEEE Computer Society.
- Avritzer, A., Paulish, D., and Cai, Y. (2008). Coordination implications of software architecture in a global software development project. In *WICSA '08: Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, pages 107–116, Washington, DC, USA. IEEE Computer Society.
- Beck, K.; Andres, C. (2004). *Extreme Programming Explained: Embrace Change*. Addison Wesley Professional.
- Begel, A. and Nagapan, N. (2008). Global software development: Who does it? In *IEEE Internacional Conference on Global Software Engineering*, pages 195–199, Los Alamitos, CA, USA.
- Berger, P. M. (2003). Instanciação de processos de software em ambientes configurados na estação taba. Master's thesis, COOPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro - Rio de Janeiro.

- Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- Cibotto, R. A. G., Pagno, R. T., Tait, T. F. C., and Huzita, E. H. M. (2009). Uma análise da dimensão sócio-cultural no desenvolvimento distribuído de software. In *V Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, Ouro Preto, MG.
- Clerc, V., Lago, P., and van Vliet, H. (2007). Global software development: Are architectural rules the answer? In *ICGSE '07: Proceedings of the International Conference on Global Software Engineering*, pages 225–234, Washington, DC, USA. IEEE Computer Society.
- Cukierman, H. L., Teixeira, C., and Prickladnicki, R. (2007). Um olhar sociotécnico sobre a engenharia de software. *Revista de Informática Teórica e Aplicada*, XIV(2).
- Damasevičius, R. (2007). Analysis of software design artifacts for socio-technical aspects. *INFOCOMP Journal of Computer Science*, 6(4):07–16.
- Damian, D. (2002). Workshop on global software development. In *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, pages 667–668, New York, NY, USA. ACM.
- Fuggetta, A. (2000). Software process: a roadmap. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, pages 25–34, New York, NY, USA. ACM.
- Herbsleb, J. D., Mockus, A., Finholt, T. A., and Grinter, R. E. (2000). Distance, dependencies, and delay in a global collaboration. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 319–328, New York, NY, USA. ACM.
- Holmstrom, H., Conchuir, E. O., Agerfalk, P. J., and Fitzgerald, B. (2006). Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *Global Software Engineering, 2006. ICGSE '06. International Conference on*, pages 3–11.
- Huzita, E. H. M., Silva, C. A., Wiese, I. S., Tait, T. F. C., Quinaia, M., and Schiavone, F. L. (2008). Um conjunto de soluções para apoiar o desenvolvimento distribuído de software. In *II Workshop de Desenvolvimento Distribuído de Software*, pages 101–110, Campinas, SP.
- Junior, A. F. S. and Santos, R. P. (2009). Aspectos sociotécnicos do desenvolvimento de software utilizando scrum em um caso prático. In *V Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, Ouro Preto, MG.
- Kruchten, P. (2003). *The Rational Unified Process: An Introduction*. Addison-Wesley.
- Lings, B., Lundell, B., Agerfalk, P. J., and Fitzgerald, B. (2007). A reference model for successful distributed development of software systems. In *ICGSE '07: Proceedings of the International Conference on Global Software Engineering*, pages 130–139, Washington, DC, USA. IEEE Computer Society.
- Mockus, A. and Herbsleb, J. (2001). Challenges of global software development. In *METRICS '01: Proceedings of the 7th International Symposium on Software Metrics*, page 182, Washington, DC, USA. IEEE Computer Society.

- Mullick, N., Bass, M., Houda, Z., Paulish, P., and Cataldo, M. (2006). Siemens global studio project: Experiences adopting an integrated gsd infrastructure. In *Global Software Engineering, 2006. ICGSE '06. International Conference on*, pages 203–212.
- OMG (2005). Software process engineering metamodel specification. An Adopted Specification of the Object Management Group, Inc.T.
- Paasivaara, M. and Lassenius, C. (2004). Using iterative and incremental processes in global software development. *IEEE Seminar Digests*, 2004(912):42–47.
- Paulish, D. J. (2007). Scalable distributed organizations for ultra-large-scale software. In *ULS '07: Proceedings of the International Workshop on Software Technologies for Ultra-Large-Scale Systems*, page 4, Washington, DC, USA. IEEE Computer Society.
- Ribeiro, M. B., Czekster, R. M., and Webber, T. (2006). Improving productivity of local software development teams in a global software development environment. In *Global Software Engineering, 2006. ICGSE '06. International Conference on*, pages 253–254.
- Rocha, R., Arcoverde, D., Brito, D., Arôxa, B., Costa, C., Silva, F. Q. B., J., A., and Meira, S. R. L. (2008). Uma experiência na adaptação do rup em pequenas equipes de desenvolvimento distribuído. In *II Workshop de Desenvolvimento Distribuído de Software*, pages 81–90, Campinas, SP.
- Santos, R. P. (2008). Uma visão sociotécnica sobre o desenvolvimento de software com extreme programming. In *IV Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, Florianópolis, SC.
- Sengupta, B., Chandra, S., and Sinha, V. (2006). A research agenda for distributed software development. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 731–740, New York, NY, USA. ACM.
- Silva, E. L. and Menezes, E. M. (2000). *Metodologia da pesquisa e elaboração de dissertação*. Laboratório de Ensino a Distância da UFSC.
- Taylor, P. S., Greer, D., Sage, P., Coleman, G., McDaid, K., and Keenan, F. (2006). Do agile gsd experience reports help the practitioner? In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 87–93, New York, NY, USA. ACM.
- Teixeira, C. A. N. and Cukierman, H. L. (2007). Por que falham os projetos de implantação de processos de software? In *III Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, pages 1–12, Porto de Galinhas, PE.
- Urdangarim, R. (2008). Uma investigação sobre o uso de práticas extreme programming no desenvolvimento global de software. Master's thesis, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre - Rio Grande do Sul.
- Vanzin, M., Ribeiro, M., Prikładnicki, R., Ceccato, I., and Antunes, D. (2005). Global software processes definition in a distributed environment. In *Software Engineering Workshop, 2005. 29th Annual IEEE/NASA*, pages 57–65. IEEE Computer Society.