

# Desafios nas Fases do ciclo de vida de Projetos Distribuídos

Rodrigo Rocha<sup>1</sup>, Catarina Costa<sup>1</sup>, Rafael Prikladnicki<sup>2</sup>, Ryan Ribeiro de Azevedo<sup>1,3</sup>, Ivaldir H. F. Junior, Silvio Meira<sup>1</sup>

<sup>1</sup>Centro de Informática – Universidade Federal de Pernambuco (UFPE)  
50.732-970 – Recife – PE – Brasil

<sup>2</sup>Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul  
90.619-900 – Porto Alegre – RS – Brasil

<sup>3</sup>Universidade Federal do Piauí (UFPI)  
64049-550 – Picos – PI – Brasil

{rgcr, csc, rra2, ihfj, srlm}@cin.ufpe.br, rafaelp@puccrs.br

**Abstract.** *The Distributed Software Development inherited the problems existent on the traditional development and for many reasons, added other difficulties. This paper presents the challenges of each stage of the software development life cycle on companies. To identify the challenges of the development stages, a field research was performed on the domestic software market, through a questionnaire the companies have cited the problems they face. This way, it's possible to view some of the challenges the domestic industry faces on each development stage.*

**Resumo.** *O Desenvolvimento Distribuído de Software herdou os problemas existentes no desenvolvimento tradicional e por diversas razões acrescentou outras dificuldades. Este artigo apresenta quais são os desafios de cada fase do ciclo de vida do desenvolvimento de software das empresas. Para identificar os desafios das fases do desenvolvimento foi realizada uma pesquisa de campo no mercado nacional de software, que através de um questionário as empresas citaram os problemas que enfrentam. Dessa forma, é possível visualizar alguns desafios que a indústria nacional enfrenta para cada fase do desenvolvimento.*

## 1. Introdução

Nos últimos anos, o software se tornou um componente vital nos negócios, as empresas necessitam de software para gerenciar e controlar suas ações, lucros e ganho de competitividade entre seus concorrentes. Diante da demanda por software e pelo crescimento do número de empresas que desenvolvem essas soluções, 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 (DDS) (Herbsleb, 2007), também conhecido como desenvolvimento global de software (para projetos que envolvem outros países).

Diversos autores afirmam que o desenvolvimento de sistemas é uma atividade complexa (The Standish Group, 2001), dessa maneira, a abordagem distribuída herdou os problemas existentes na forma tradicional e por diversas razões acrescentou outras dificuldades (Prikladnicki, 2004). Existem vários fatores que podem prejudicar o

progresso dos projetos, e um deles é o desenvolvimento de software com equipes geograficamente distribuídas. Diante deste cenário, essa nova forma de fazer negócio ou software, demanda uma alta dependência de comunicação entre os *stakeholders*, ou mais especificamente os responsáveis pelo design das partes do software que serão desenvolvidas em diferentes localidades (Cataldo et al., 2006).

Inúmeras empresas tentam adotar técnicas, metodologias e ferramentas para que possam lidar da melhor forma com as variáveis do contexto distribuído, variáveis trazidas pela distância geográfica, temporal e cultural entre os membros da equipe global de software. Contudo, é necessário que existam mais estudos para que sejam concebidas novas técnicas, metodologias e ferramentas para ambientes distribuídos (Lopes et al, 2003) (Rocha et al, 2008). Em diversos momentos, as empresas que utilizam esse conceito de desenvolvimento necessitam definir quais metodologias utilizarão em um determinado projeto e suas respectivas práticas, pois estas não provêm de informações que sejam capazes de determinar qual metodologia é mais adequada para determinado tipo de projeto ou quais as práticas mais indicadas para o mesmo.

Além disso, as soluções que são utilizadas para o desenvolvimento tradicional visando diminuir os problemas não têm o resultado esperado quando os participantes do projeto estão separados geograficamente. Dessa maneira, a realização de um projeto distribuído se torna mais difícil pelos fatores que afetam o desenvolvimento tradicional e outros acrescentados pela distribuição geográfica e temporal. Tendo em vista isso, o objetivo principal deste trabalho é identificar quais são os desafios enfrentados pelas equipes distribuídas durante as fases do ciclo de vida do desenvolvimento distribuído, para isto, este trabalho realizou uma pesquisa de campo com algumas empresas do cenário nacional. A pesquisa foi realizada em 8 empresas localizadas em 5 diferentes estados.

Este trabalho está dividido da seguinte forma: a Seção 2 descreve a metodologia utilizada; na Seção 3 são apresentados os resultados, e com isso os desafios das fases do ciclo de vida são expostos; e por fim, as considerações finais.

## **2. Metodologia Utilizada**

Esta Seção descreve a metodologia utilizada para a realização da pesquisa de campo Pfleeger e Kitchenham (2001), a qual pode ser classificada como quanti-qualitativa (Flick, Kardorff e Steinke, 2000). Foi elaborada uma pesquisa de campo (exploratória) realizada nos meses de janeiro e fevereiro de 2010, através de um questionário com questões abertas, ou seja, do tipo não estruturado, no qual os participantes sabiam que o objetivo da pesquisa era identificar os modelos de colaboração que a indústria nacional utiliza e quais são os principais desafios enfrentados pelos desenvolvedores de software que trabalham no cenário distribuído. A amostra foi do tipo não probabilística (não aleatória) intencional, onde a mesma foi escolhida intencionalmente (Marconi e Lakatos, 1996).

Dez projetos foram analisados, no qual nove pessoas de oito empresas distintas responderam ao questionário. Um funcionário de uma determinada empresa respondeu a dois questionários, referente à sua participação nos dois últimos projetos distribuídos que participou naquela organização. Em uma outra empresa, dois funcionários que

participaram de projetos diferentes responderam às questões. Os seis outros projetos foram analisados com base nas respostas de seis funcionários de empresas diferentes.

Os participantes eram de matrizes ou filiais de cinco estados do país, tendo a predominância em Pernambuco (4) e Paraíba (3). Para obter os resultados, após os questionários serem aplicados, houve a tabulação e organização das respostas por afinidade de idéias, resultando como conclusão uma descrição sumária dos relatos de tais profissionais em relação aos problemas que enfrentaram em projetos distribuídos.

### **3. Resultados Obtidos**

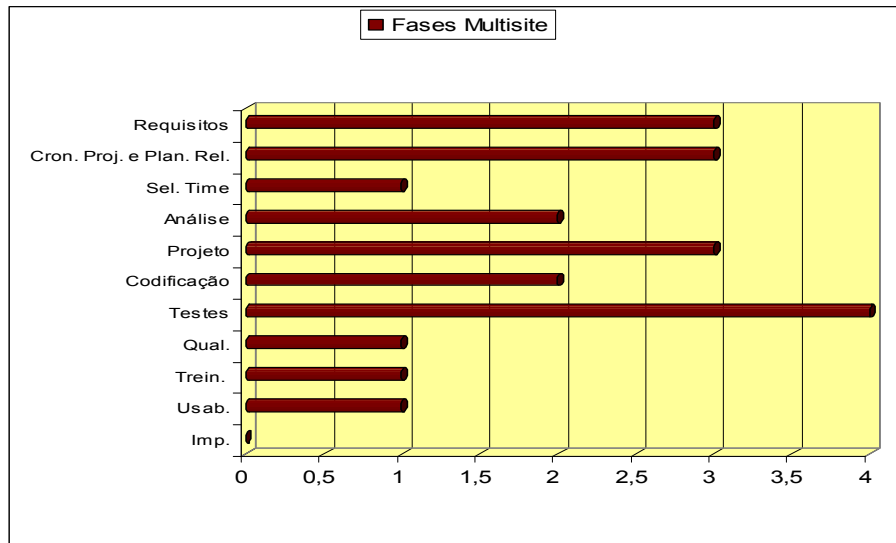
Nesta seção são apresentados alguns dos resultados obtidos (dados quantitativos e qualitativos) com a realização da pesquisa. A respeito da participação dos entrevistados em projetos distribuídos, dos nove questionados, 7 participaram de um a três projetos e 2 fizeram parte de quatro a seis projetos. Nenhum dos questionados participou de mais do que seis projetos.

Quando questionados à respeito de quanto tempo durou o projeto que o funcionário participou, é possível visualizar que um projeto durou de 1 a 3 meses, três projetos foram entre 4 a 6 meses, e dois duraram de 6 a 12 meses, outros dois projetos tiveram duração de 13 a 24 meses e, os dois projetos restantes acima de 24 meses. Algumas pessoas relataram que seu projeto envolvia países diferentes, sendo 60% (6) dos projetos não envolveram outros países, enquanto 40% (4) foram realizados com a participação de equipes de outros países.

No que se refere às atividades básicas realizadas no desenvolvimento distribuído de software da empresa para o projeto, foi questionado quais atividades foram realizada(s) e se cada atividade foi feita *onsite* (no cliente), *distribuído/offshore* (no *site* distante) ou *multisite* (em ambos). A fase de Requisitos foi realizada, igualmente, 6 *onsite* e 6 distribuído; na fase seguinte, Cronograma do Projeto e Planejamento de Releases, 5 foram *onsite* e 6 distribuídos; em Seleção do Time e Contrato prevaleceu distribuído, com 5, sobre *onsite* com 4; a fase de Análise conta com 6 *onsite* e 5 distribuídos; a de Projeto contém 6 em ambos; a fase de Codificação apresentou 4 *onsite* e 8 distribuídos; a fase de Testes foi a mais referida, sendo 8 *onsite* e 6 distribuídos; a Garantia de Qualidade foi em sua maioria *onsite*, com 7, sendo 3 distribuídos; já no Treinamento do Time Distribuído/*Offshore*, foram 2 *onsite* e 5 distribuídos; a Avaliação de Usabilidade apresentou 4 *onsite* e 3 distribuídos e, finalmente, a Implantação, com 4 *onsite* e 2 distribuídos.

No Gráfico 1 estão descritas todas as fases citadas anteriormente, porém retratando apenas as atividades que foram realizadas *multisite*, ou seja, *onsite* e distribuído ao mesmo tempo. A fase de Testes foi a que apresentou mais projetos *multisite*, com 4; já as fases de Requisitos, Cronograma do Projeto e Planejamento de Releases e a de Projeto tiveram 3 cada uma; Análise e Codificação com 2 e Seleção do Time e Contrato, Garantia de Qualidade, Treinamento do Time Distribuído/*Offshore* e a Avaliação de Usabilidade apresentaram apenas 1, enquanto que a fase de Implantação não apresentou projetos *multisite*.

**Gráfico 1 – Atividades realizadas Multisite**



### 3.1 Desafios das Fases do DDS

Numa pesquisa baseada em questionários abertos, é costume coordenar as respostas através de uma organização de idéias por temas afins, visto que o volume de informações retornado é grande. É uma forma de tipificar as respostas dos entrevistados. Portanto, uma vez definidos os temas, as idéias de cada entrevistado foram reorganizadas de acordo com o tema a que pertencem. Dessa maneira, as Tabelas 1, 2, 3, 4 e 5 apresentam os desafios encontrados (já classificados) em cada fase do desenvolvimento distribuído extraídos da pesquisa de campo. Tais desafios dizem respeito às atividades sendo executadas de forma *onsite*, distribuída e *multisite*.

A Tabela 1 apresenta os desafios encontrados durante a fase de Requisitos que é um momento importante do ciclo de vida, já que é a fase em que todos os objetivos e necessidades do cliente devem ficar bem claros. A maioria das respostas dos participantes citava a dificuldade na comunicação de uma forma geral, inclusive dificuldades entre a própria equipe. Um desafio interessante foi o fato de existirem clientes que não se comprometem ou que não dão prioridade ao projeto.

Tabela 1 – Fase Requisitos

Requisitos	
Temas Afins	Desafios
Comunicação	Demora para resposta via e-mails
	Falhas na comunicação via áudio através de <i>chats</i>
	Custo de telefonemas
	Conferência via áudio não suportava muitos usuários
	Idioma utilizado
Encontros presenciais	Necessidade de mais encontros presenciais para esclarecimento de dúvidas cruciais
Equipe	Falta de entrosamento das equipes
	Concorrência entre os engenheiros do projeto
	Necessidade em alguns momentos que houvesse

	uma decisão tomada por alguém mais experiente na equipe, para que não restassem dúvidas ou especulações
Cliente	Falta de comprometimento e prioridade
	Aprovação dos requisitos por parte do cliente demandava tempo que, em alguns momentos não existia

Durante a atividade de Cronograma e Planejamento de *releases* (Tabela 2) os maiores desafios foram encontrados na própria equipe, principalmente para conseguir um consenso geral sobre as datas que são acordadas entre os times. Durante a fase de Seleção do Time e Contrato, existiram problemas de mão de obra, tanto por profissionais qualificados como especialistas em determinadas tecnologias. Outro desafio é definir o time e as funções de cada um.

**Tabela 2 – Fase Cronograma do Projeto e Planejamento de Releases**

<b>Cronograma do Projeto e Planejamento de Releases</b>	
Temas Afins	Desafios
Equipe	Alinhamento de todos os <i>stakeholders</i>
	Consenso geral sobre as datas acordadas
	Sincronizar <i>releases</i> entre sites que fossem integrar o software final
Dispersão	Sequenciar atividades com equipes distribuídas

Durante a fase de análise (Tabela 3), os principais problemas eram de comunicação, principalmente na troca de informação das equipes e na forma como isso ocorria, como por exemplo a definição dos meios de comunicação e os procedimentos necessário para tal. Na fase do Projeto, como em outras fases a comunicação novamente foi o principal desafio. Outro desafio citado pelos participantes foi a interação do cliente com os desenvolvedores durante essa fase.

**Tabela 3 – Fase Análise**

<b>Análise</b>	
Temas Afins	Desafios
Comunicação	Demora para resposta via e-mails
	Falhas na comunicação via áudio através de <i>chats</i>
	Conferência via áudio não suportava muitos usuários
	Dificuldade para organizar uma conferência via áudio
	Idioma utilizado
Processo de Software	Seguir adequadamente o processo
	O processo de software não abrangia momentos específicos
Cliente	Demora para validar alguns diagramas e decisões

A fase de codificação (Tabela 4) apresentou diversos problemas, e muitos deles foram citados por quase todos os participantes, como o problema de código fonte, por exemplo, no sentido de padronização de código e a questão do tamanho e complexidade do projeto que também foi bastante citado.

Tabela 4 – Fase Codificação

Codificação	
Temas Afins	Desafios
Comunicação	Demora para respostas em geral
	Idioma utilizado
Equipe	Falta de conhecimento por parte do time sobre quem é o maior especialista da equipe naquela parte do código fonte
	Falta de conhecimento, sobre quem está fazendo o que
Código fonte	Dificuldade em padronização de código
	Problemas com o entendimento da arquitetura do software por partes dos membros da equipe
	Tamanho e complexidade do projeto
Motivação	Falta de motivação dos membros das equipes
Delegação de tarefas	Dificuldade em delegar atividades para perfis desconhecidos

A Tabela 5 apresenta os desafios encontrados durante a fase de testes. É possível afirmar que deve haver um maior planejamento dos testes e também da forma que eles serão executados. Já que essa não é uma tarefa trivial quando o projeto é distribuído.

Tabela 5 – Fase Testes

Testes	
Temas Afins	Desafios
Comunicação	Demora para resposta via e-mails
	Idioma utilizado
<i>Stakeholders</i>	Definição de responsabilidades, difícil determinar a fronteira das atividades dos membros
Plano de Testes	Definir planos de testes e executá-los, de forma que todos os envolvidos participassem efetivamente dos testes
Ambiente	Problema de ambiente
Complexidade	Complexidade dos testes
Equipes	Dependência de equipes externas

Com relação à atividade de treinamento do time distribuído, os maiores problemas encontrados foram no idioma utilizado. Em alguns momentos, a documentação era incoerente com o treinamento e em certos pontos o treinamento era bastante superficial. Na fase de garantia de qualidade, os desafios encontrados foram a confiança, que vai diminuindo entre os *stakeholders* durante a execução do projeto, bem como problema para definir quais seriam os requisitos de qualidade e os parâmetros para a sua medição, já que muitas pessoas envolvidas geram muitos interesses diferentes. Durante a fase de implantação, um problema pode ser destacado, que é a dificuldade que existe para saber o contexto do cliente de forma remota, ou seja, quais são os recursos e tecnologias que eles possuem.

A partir dos resultados da pesquisa de campo, é possível afirmar que:

- Os fatores que afetam as fases do desenvolvimento distribuído de software são conhecidos pelas organizações e pelos pesquisadores do meio;
- A comunicação é um fator que gera problema em quase todas as fases do

desenvolvimento, o que pode tornar esse fator o mais importante;

- Algumas fases como a de codificação, por exemplo, apresentaram muitos problemas que na verdade não tem relação com o contexto distribuído, e sim com o próprio desenvolvimento de software. Tal como, o desafio: “Dificuldade em delegar atividades para perfis desconhecidos”;
- As fases que mais apresentaram desafios/problemas foram as de requisitos e codificação;
- Nas tabelas, algumas fases possuem menos problemas, isso porque as respostas foram repetidas/semelhantes e também pelo fato de que nem toda empresa utilizava aquela fase em seu(s) projeto(s);
- A fase de Testes é mais realizada de forma *multisite*, seguida das fases de Requisitos, Cronograma do Projeto e Planejamento de Releases e Projeto;
- Os fatores que afetam não são específicos de um projeto, eles são comuns entre projetos e empresas;
- Idioma se manteve como um problema constante nos projetos que envolviam outros países;
- Na fase de requisitos muitos participantes responderam que o time precisa de um melhor entrosamento;

Através da pesquisa algumas constatações foram feitas, como vistas acima. Essas constatações, assim como outras, servem como base para novas pesquisas e novos estudos, onde soluções podem ser propostas visando mitigar esses fatores. Como o caso do desafio “Falta de conhecimento por parte do time sobre quem é o maior especialista da equipe naquela parte do código fonte” que é um problema que pode ser auxiliado através da ferramenta Presley (Trindade et al, 2009), pois os *stakeholders* possuem conhecimento de quem fez tal rotina/funcionalidade, e assim podem trocar informações valiosas sobre o código fonte durante o projeto. Assim como esse, alguns desafios já possuem soluções, porém a maioria desses problemas ainda precisam ser solucionados.

#### **4. Considerações Finais**

No cenário distribuído, projetos de software assumem perspectivas diferentes e, conseqüentemente, novos riscos. Se não houver um bom conhecimento dos fatores que podem influenciar o projeto, o mesmo terá mais chances de não obter sucesso.

Seria ideal ter um número mais significativo de empresas participando, bem como de estados também. Dessa maneira, seria possível ter um panorama que representaria de fato a perspectiva nacional, no sentido dos problemas que projetos distribuídos de software enfrentam em suas fases de desenvolvimento. O questionário aberto se mostrou muito interessante, pois os participantes podem descrever suas respostas e isso torna de certa forma, as informações mais ricas. Por outro lado, quando os participantes visualizam um questionário aberto, eles percebem que terão que perder mais tempo, tentando lembrar e compilar todas as informações que vivenciaram, dessa forma, isso os inibe um pouco e pode também prejudicar a riqueza das respostas, já que os mesmos podem não se comprometer da melhor forma possível.

Os desafios encontrados nas atividades do desenvolvimento distribuído de software são diferentes do desenvolvimento tradicional de software, porém, conhecidos há algum tempo pela comunidade de DDS. Portanto, tais problemas, já conhecidos,

poderiam ter soluções já conhecidas, mas talvez por ser uma área ainda imatura, ainda está sendo explorada. Os resultados dessa pesquisa podem ser de grande relevância para que empresas nacionais e internacionais utilizem processos, técnicas, boas práticas e metodologias que se adequem às limitações identificadas. Espera-se também, com a realização da pesquisa, incentivar outros estudos que favoreçam a diminuição de falhas desses projetos.

Alguns trabalhos podem ser desenvolvidos futuramente a partir deste estudo, como: estudos e pesquisas mais aprofundados sobre os fatores que influenciam cada fase do desenvolvimento distribuído, a fim de identificar maneiras para lidar com tais problemas; classificação de todos os fatores existentes que influenciam o DDS em fatores humanos, técnicos e organizacionais, com objetivo de estabelecer uma maior consolidação desses fatores.

## Referências

- Cataldo, Marcelo., Bass, Matthew., Herbsleb, James., 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.
- Flick, U., von Kardorff, E. e Steinke, I. (2000). Was ist qualitative Forschung? Einleitung und Überblick. pp. 13-29. Reinbek: Rowolht.
- Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. IEEE Computer Science. p188-198.
- Lopes, Leandro; Prikladnicki, Rafael; Majdenbaum, Azriel; Audy, Jorge. (2003). Uma proposta para processo de requisitos em ambientes de desenvolvimento distribuído de software. In: 6th WER, Piracicaba. Brasil. p. 329-342.
- Marconi, Mariana de Andrade; Lakatos, Eva Maria. (1996). Técnicas de pesquisa. 3 ed. São Paulo : Atlas. 231 p.
- Pfleeger, S. and Kitchenham, B. (2001). Principles of survey research: part 1: turning lemons into lemonade. ACM SIGSOFT Software Engineering Notes, 26(6):44–45.
- Prikladnicki, R.; Audy, J. (2004). Munddos: Um Modelo de Referência para Desenvolvimento Distribuído de Software. 18º Simpósio Brasileiro de Engenharia de Software.
- 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.
- The Standish Group. (2001). CHAOS 2001: A Recipe for Success.
- Trindade, C., Moraes, A., Barbosa, Y., Albuquerque, J., Meira, S. (2009). Presley: uma Ferramenta de Recomendação de Especialistas para Apoio à Colaboração em Desenvolvimento Distribuído de Software. XXIII Simpósio Brasileiro de Engenharia Software, Fortaleza. XXIII Simpósio Brasileiro de Engenharia Software, Sessão de Ferramentas.