



UM MAPEAMENTO SISTEMÁTICO DA ATIVIDADE DE TESTES NO AMBIENTE DE DESENVOLVIMENTO DISTRIBUÍDO

Trabalho de Conclusão de Curso

Engenharia da Computação

Augusta Raiana Marques de Figueredo

Orientador: Prof. Gustavo Henrique Porto de Carvalho



**UNIVERSIDADE
DE PERNAMBUCO**

**Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação**

Augusta Raiana Marques de Figueredo

**UM MAPEAMENTO SISTEMÁTICO DA
ATIVIDADE DE TESTES NO AMBIENTE
DE DESENVOLVIMENTO DISTRIBUÍDO**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Junho de 2011.

De acordo

Recife

____/____/____

Gustavo Henrique Porto de Carvalho

Dedico este trabalho a meus filhos: Linus e Nara, as minhas amadas, mãe e irmã, e a meu querido e saudoso pai.

Agradecimentos

Agradeço a minha mãe por todos os puxões de orelha que recebi e que me tornaram mais paciente em relação aos problemas que a vida nos impõe. E ao meu saudoso pai, por ter me ensinado tudo que sei sobre o mundo.

Agradeço a minha irmã por sempre estar presente em todas as minhas angústias e alegrias, tornando não somente a escrita desta monografia como todas as atividades de minha vida, tarefas prazerosas.

Agradeço aos meus filhos, Linus e Nara, que gentilmente (forçadamente) cederam seus horários de brincar para que eu pudesse ler os artigos deste mapeamento.

Agradeço a Andersson, meu fiel amigo, que sempre leu todas as minhas produções, de textos de msn a esta monografia, e que fortaleceu meu espírito para que eu pudesse ter preparo para lidar com os desafios destes 5 anos de faculdade.

Agradeço a Jamerson, por ter em meros 3 meses reconstruído meu coração e ter me dado paz de espírito. O que me foi essencial para eu me dedicar por completo a este trabalho.

Agradeço ao meu orientador, Prof. Gustavo Carvalho, por me aceitar como sua orientanda, por aceitar o desafio de me orientar na execução desta monografia e por me motivar com mãos de ferro e paciência de Jó, alguém tão indisciplinada como eu. Somente com um grande mestre eu teria conseguido.

Tá terminando!!!

Resumo

Entre as atividades de verificação e validação, a atividade de testes consolida-se como uma das evidências da confiabilidade do software e tentativa de manter um nível de qualidade. Casey (2009) afirma que em um cenário de desenvolvimento exposto a diversos tipos de interferências e desafios, os testes poderão garantir a única medida real de confiança de que o software tenha atingido um nível desejado de qualidade. A necessidade de investigar a atividade de teste no contexto dos desafios do Desenvolvimento Distribuído de Software já foi abordada por diversos pesquisadores. Sengupta et al. (2004) propõe a utilização dos artefatos de teste como meio de comunicação. Zage et al. (2005) apresenta uma ferramenta para apoio da atividade de testes num ambiente global. Bondi e Ros (2009) relatam sua experiência na formação de uma equipe de teste distribuída e descreve os problemas e soluções a respeito do teste em DDS. O tema deste trabalho está centrado no Desenvolvimento Distribuído de Software, tendo como problema a ser abordado: Como adaptar o processo de testes em um ambiente de desenvolvimento constituído de equipes distribuídas?

Palavras-chaves: Desenvolvimento Distribuído de Software; Testes

Abstract

Among the activities of verification and validation, the testing activity assesses the reliability of the software in an effort to maintain a standard level of quality. Casey (2009) states that a development scenario exposed to many types of interference and challenges is the only real guarantee, to ensure with measure of confidence, that the software tested has reached the desired level of quality. Several researchers have approached the need to investigate the testing activity in the context of Distributed Software Development. Sengupta et al. (2004) proposes the use of test artifacts as a means of communication. Zage et al. (2005) presents a tool to support the testing activity in a global environment. Bondi and Ros (2009) report their experience in forming a distributed test team and describe the problems and solutions regarding the DDS test. This work focuses on Distributed Software Development, specifically addressing the problem: How to adapt the testing process in a development environment composed of distributed teams.

Sumário

Capítulo 1 Introdução	1
1.1 Motivação.....	1
1.2 Objetivos.....	2
1.3 Estrutura do trabalho	3
Capítulo 2 Fundamentação Teórica	4
2.1 Desenvolvimento Distribuído de Software.....	4
2.1.1 Classificação do DDS.....	5
2.1.2 Motivações e Desafios	9
2.1.3 Karolak (1998).....	10
2.1.4 Carmel (1999).....	10
2.1.5 Evaristo e Scudder (2000).....	13
2.2 Teste de Software	15
2.2.1 Taxonomia de Termos relacionados a Teste de Software.....	16
2.2.2 IEEE Standard 829: Padrão para Documentação dos Testes	17
2.2.3 Processo de Testes.....	18
Capítulo 3 Metodologia	25
3.1 Mapeamento Sistemático da Literatura	25
3.2 Ciclo da Pesquisa.....	27
3.2.1 Processo do Mapeamento Sistemático	27
3.2.1.1 Questões da Pesquisa.....	28
3.2.1.2 Estratégia de Busca.....	28
3.2.1.3 String de Busca.....	28
3.2.2 Seleção dos Estudos.....	29
3.2.2.1 Critérios de Inclusão e Exclusão	29
3.2.3 Processo de Seleção dos Estudos Primários.....	30
3.2.4 Estratégia para Extração dos Dados.....	32
Capítulo 4 Resultados	34
4.1 Análise Quantitativa do Mapeamento Sistemático	34
4.2 Análise Qualitativa do Mapeamento Sistemático	40
Capítulo 5 Conclusão	48

5.1	Limitações e Ameaças de Validade.....	48
5.2	Conclusão e Trabalhos futuros.....	48
	Bibliografia.....	52
	Apêndice A.....	56
	Apêndice B.....	64
	Apêndice C.....	69
	Apêndice D.....	81
	Apêndice E.....	82

Índice de Figuras

Figura 1. Classificação quanto ao nível de dispersão dos autores	6
Figura 2. Relacionamento entre as partes envolvidas	8
Figura 3. Classificação das empresas quanto ao nível de relacionamento	8
Forças centrípetas	10
Forças centrífugas	10
Figura 4. Forças que influem no DDS	10
Figura 5. Dimensões	13
Figura 6. Relacionamento entre o processo de desenvolvimento e o processo de testes	18
Figura 7. Fase de Planejamento da atividade de Testes	19
Figura 8. Fase de Preparação da atividade de Testes	20
Figura 9. Fase de Projeto da atividade de Testes	21
Figura 10. Fase de Execução da atividade de Testes	22
Figura 11. Modelo-V de Craig e Jaskie (2002)	24
Figura 12. Etapas da pesquisa	27
Figura 13. Visualização entre os artigos candidatos e relevantes retornados por ano	35
Figura 14. Relevância dos estudos primários	36
Figura 15. Classificação quanto ao local de produção	36
Figura 16. Relação de publicações no IGSE	36
Figura 17. Produções relevantes para esta pesquisa do IGSE vs Ano	36
Figura 18. Classificação dos artigos primários quanto a seu tipo	37
Figura 19. Classificação dos artigos primários quanto a seu tipo	39
	XI

Índice de Tabelas

Tabela de Símbolos e Siglas	XIV
Tabela 1. String utilizada no Mapeamento	29
Tabela 2. Tabela de qualidade X quantidade de artigos	37
Tabela 3. Tabela de Qualidade X local de publicação.....	38
Tabela 4. Áreas chaves X artigos X quantidade de trabalhos- qualidade	47

Tabela de Símbolos e Siglas

DDS	Desenvolvimento Distribuído de Software
EP	Estudo Primário
ICGSE	Internacional Conference on Global Software Engineering
TI	Tecnologia da Informação

Capítulo 1

Introdução

Neste Capítulo são descritas as questões que levaram a realização deste trabalho, os objetivos a serem atingidos com ele, a metodologia seguida no seu desenvolvimento e a sua organização. Esses componentes são detalhados da seguinte maneira:

1.1 Motivação – são apresentadas informações relativas ao cenário de Desenvolvimento Distribuído, entrando na particularidade da realidade da atividade de Testes quando realizadas neste cenário;

1.2 Objetivos – são apresentadas informações sobre a importância deste trabalho para a área de pesquisa;

1.3 Estrutura do trabalho – são apresentadas informações sobre como está organizado a estrutura de tópicos deste trabalho.

1.1 Motivação

Na busca de vantagens competitivas, empresas de software têm lançado mão de práticas de desenvolvimento que tragam benefícios ao seu produto final. Uma dessas práticas é o Desenvolvimento Distribuído de Software (DDS) que segundo Carmel (1999) “é um modelo de desenvolvimento de software onde os envolvidos em um determinado projeto estão dispersos fisicamente”. Para Prikladnick (2003), fatores como acesso a recursos globais mais qualificados, menores custos e incentivos fiscais têm impulsionado organizações a investirem em DDS e a optarem por distribuírem seus processos de desenvolvimento. Foi a partir da década de 90 que esta abordagem se difundiu entre as empresas de desenvolvimento de software (Krishna et al, 2004), e desde então vem criando novas formas de competição e cooperação que vão além das fronteiras territoriais.

Segundo Mak (2007), os benefícios potenciais do desenvolvimento distribuído são muito atraentes para serem ignorados, por tal razão muitas empresas adotam o DDS como uma tentativa de diminuir seus custos. Porém, a falta de adaptação no

seu processo de gerenciamento associado à inexperiência com equipes distribuídas aumenta exponencialmente a taxa de insucesso do projeto (Pichler, 2007). Diversos estudos relatam as mudanças que o cenário de DDS adiciona a já complexa tarefa de gerenciar projetos co-localizados, variáveis como: distância física, temporal e sociocultural devem ser consideradas (Audy; Prikladnicki, 2007; Binder 2007; Carmel, 1999; Enami, 2006; Liang, 2008; Prikladnick, 2003; Vavassori, 2002; Zaroni, 2002). A possibilidade de ineficiência das técnicas de gerenciamento de projetos tradicionais no contexto de DDS impulsiona a necessidade de adaptação dos processos, métodos e ferramentas específicos para tal contexto (Nidiffer E Dolan, 2005).

Entre as atividades de verificação e validação, a atividade de testes consolida-se como uma das evidências da confiabilidade do software e tentativa de manter um nível de qualidade. Casey (2009) afirma que um cenário de desenvolvimento exposto a diversos tipos de interferências e desafios, os testes poderão garantir a única medida real de confiança de que o software tenha atingido um nível desejado de qualidade. A necessidade de investigar a atividade de teste no contexto dos desafios do DDS já foi abordada por diversos pesquisadores. Sengupta et al. (2004) propõe a utilização dos artefatos de teste como meio de comunicação. Zage et al. (2005) apresenta uma ferramenta para apoio da atividade de testes num ambiente global. Bondi e Ros (2009) relatam sua experiência na formação de uma equipe de teste distribuída e descreve os problemas e soluções a respeito do teste em DDS.

O tema deste trabalho está centrado no Desenvolvimento Distribuído de Software, tendo como problema a ser abordado: **O que muda nas atividades de testes de software quando o cenário de desenvolvimento é distribuído?**

1.2 Objetivos

Este trabalho tem como objetivo geral investigar e reunir de forma sistemática o conhecimento sobre como adaptar a etapa de testes em um ambiente de desenvolvimento distribuído.

O objetivo geral pode ser decomposto nos seguintes objetivos específicos:

- Realizar um mapeamento sistemático da literatura sobre a adaptação da etapa de testes quando executada em projetos no desenvolvimento distribuído de software.
- Identificar desafios e boas práticas de apoio a etapa de testes com o objetivo de elaborar um guia que auxilie na execução do processo de testes no contexto de projetos de desenvolvimento distribuído de software.

1.3 Estrutura do trabalho

O trabalho está organizado da seguinte maneira:

No **Capítulo 2** é apresentado o referencial teórico, que contém a revisão das teorias bases da dissertação. Inicialmente é apresentado o Desenvolvimento Distribuído de Software, e os principais conceitos que envolvem o tema. Em seguida, são apresentados os conceitos relacionados aos Testes de Software.

No **Capítulo 3** é descrita a metodologia empregada para realizar o estudo, a classificação da pesquisa junto ao quadro metodológico, as principais etapas da pesquisa, o processo para a realização do Mapeamento Sistemático da Literatura, com o protocolo definido. Por fim, é descrita a forma de análise dos dados extraídos e sintetizados pelo mapeamento sistemático.

No **Capítulo 4** são apresentados os resultados. Inicialmente o capítulo apresenta uma análise dos dados gerais do mapeamento sistemático como as principais fontes, o número de estudos retornados, a distribuição temporal dos estudos, a avaliação da qualidade dos mesmos, entre outras informações.

Por fim, no **Capítulo 5** são apresentadas as conclusões do trabalho, onde se realiza uma análise das limitações e ameaças à validade da pesquisa, propostas para trabalhos futuros a partir dos resultados, além de uma análise final do trabalho e lições aprendidas sobre o processo do mapeamento sistemática.

Capítulo 2

Fundamentação Teórica

A base teórica necessária para a realização da pesquisa e o entendimento do estudo é apresentada nesse capítulo. O capítulo está organizado da seguinte forma:

2.1 Desenvolvimento Distribuído de Software – nesta seção são apresentados alguns conceitos sobre o Desenvolvimento Distribuído de Software (DDS), como: as principais características, cenários, razões para o crescimento e pesquisas importantes para a área.

2.2 Testes de Software – nesta seção são apresentados os conceitos sobre Testes de Software, como: motivação para sua realização, ciclo de vida da atividade de testes durante o desenvolvimento de um software, suas características e tipos de testes.

2.1 Desenvolvimento Distribuído de Software

A globalização do mercado tem exibido uma mudança cada vez mais aparente na forma como os softwares estão sendo desenvolvidos. Na busca por um diferencial competitivo, empresas de TI têm aderido a novas formas de desenvolvimento (Liang, 2008). O avanço da economia, a modernidade dos meios de comunicação, a disponibilidade de mão de obra especializada e de custos reduzidos em países em desenvolvimento, associados a incentivos fiscais e a possibilidade do desenvolvimento *round-the-clock*, tornaram o desenvolvimento distribuído de software (DDS) uma abordagem de desenvolvimento cada vez mais constante nas empresas que buscam seu diferencial no mercado (Carmel,1999).

Esta abordagem de mercado tem repercutido não somente no âmbito dos negócios como também na maneira como os produtos estão sendo modelados, construídos, testados e entregues aos clientes. A abordagem clássica de desenvolvimento de software, com equipes co-localizadas, possui diversos desafios associados à natureza do software, e soluções são elaboradas para diminuir os impactos desses desafios: técnicas de gerenciamento de projetos incentivo a

comunicação entre a equipe, estímulo para o senso de equipe. De tal forma que problemas de projetos podem ser resolvidos durante as conversas informais no trabalho, porém mesmo assim essa abordagem de desenvolvimento possui grandes dificuldades e baixas taxas de sucesso (MacGregor et al, 2005).

Segundo Audy e Prikladnicki (2007), entender as diferenças e os desafios da abordagem distribuída para a de projetos com equipes co-localizadas é entender que o desenvolvimento distribuído irá possuir todos os desafios de um projeto desenvolvido tradicional, além das mudanças impostas pelo cenário distribuído. Para Komi-Sirvio e Tihinen (2005) problemas associados à qualidade, tempo e custo são intensificados quando tratados num ambiente de desenvolvimento distribuído. Diversos estudos relatam as mudanças que o cenário de DDS adiciona a já complexa tarefa de gerenciar projetos co-localizados, variáveis como: distância física, temporal e sócio-cultural devem ser consideradas (Audy; Prikladnicki, 2007; Binder 2007; Carmel, 1999; Enami, 2006; Liang, 2008; Prikladnick, 2003; Vavassori, 2002; Zanoni, 2002).

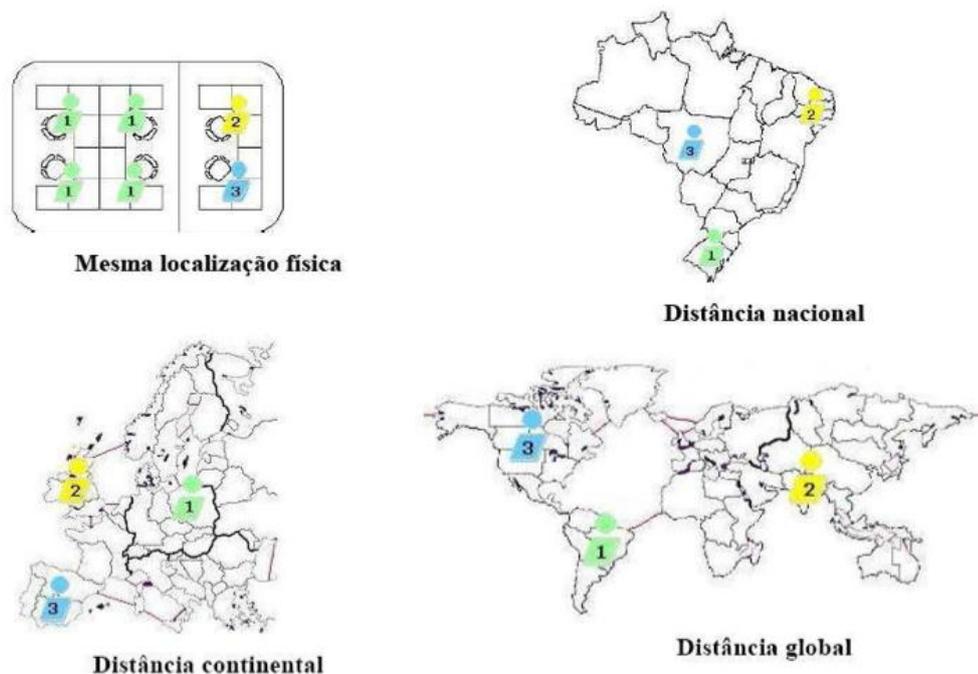
Pichler (2007) afirma que muitas empresas aderem ao DDS e constituem suas equipes como se todos os participantes estivessem na mesma sala ou mesmo prédio, quando na verdade essas equipes podem estar até mesmo em continentes diferentes e sendo impactados por desafios associados a diferentes culturas, fusos e idiomas. Um projeto de DDS requer diferentes mecanismos para controlar, monitorar e coordenar suas atividades. A possibilidade de ineficiência das técnicas de gerenciamento de projetos tradicionais no contexto de DDS impulsiona a necessidade de adaptação dos processos, métodos e ferramentas específicos para tal contexto (Nidiffer; Dolan, 2005).

2.1.1 Classificação do DDS

Diversos autores já caracterizaram em diferentes formas os tipos de DDS que podem existir, levando em conta diferentes tipos de variáveis que influenciam o projeto. Carmel (1999) propôs que a diferença de uma equipe tradicional para uma global estava centrada em três variáveis: fuso horário, distância física e cultural. Evaristo e Scudder (2000) propõem uma classificação mais aprofundada na qual leva em conta variáveis que influenciam no desempenho do projeto: características do projeto, complexidade, diferenças culturais, a existência de diferentes

metodologias dentro do projeto e o nível de interesses das partes envolvidas. Cockburn (2002) propõe uma classificação que leva em conta: o tipo de projeto (comercial ou de código aberto), o tamanho da equipe e os papéis existentes em cada local.

Audy e Prikladnicki (2007) caracterizam o nível de dispersão dos diversos atores envolvidos no processo ao longo do projeto em quatro níveis: mesma localização, distância nacional, distância continental e distância global e expõem que as dificuldades de um cenário com equipes co-localizadas podem ser bem diferentes de um cenário com equipes em distância global. Na Figura 1 é ilustrada essa caracterização.

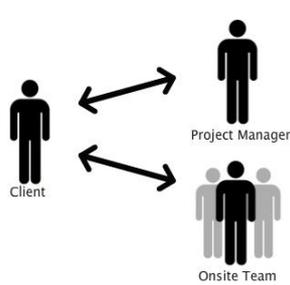


Fonte: Audy e Priklanicki (2007)

Figura 1. Classificação quanto ao nível de dispersão dos autores

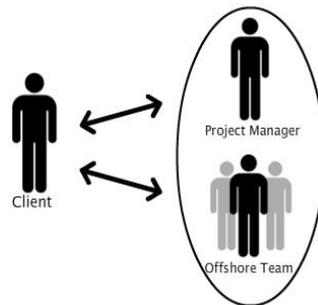
Um projeto de software a ser desenvolvido possui a participação obrigatória de diferentes atores: cliente, gerente, equipe de desenvolvimento, equipe de testes e outros. Produzir um software no cenário distribuído permite a virtualização de qualquer uma das partes envolvidas. Do caso 1 até o caso 6 é apresentado seis dessas possibilidades. No Caso1 temos a interação de um projeto co-localizado tradicional, com o cliente podendo ter contato direto com o gerente e com o time. No caso 2 temos a interação de um projeto quando o cliente não possui interação física direta nem com time nem com o gerente. No caso 3 temos uma equipe com

interação direta com o cliente e compartilhando o desenvolvimento com uma equipe virtual. No caso 4 temos uma equipe com interação virtual com o cliente e compartilhando o desenvolvimento com outra equipe virtual. No caso 5 temos a interação de um projeto co-localizado (gerente, equipe de testes e o cliente) compartilhando atividades de desenvolvimento e de testes com um grupo formado por equipe virtual de testes e outra equipe virtual de desenvolvimento. No caso 6 temos uma equipe em contato direto com o cliente, compartilhando atividades com uma equipe virtual e esta compartilhando suas atividades com outra equipe dispersa globalmente.



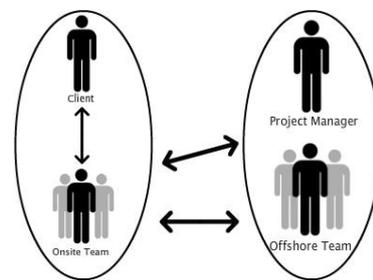
Fonte: Elaboração Própria

Caso 1



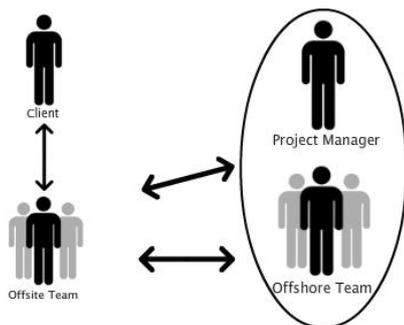
Fonte: Elaboração Própria

Caso 2



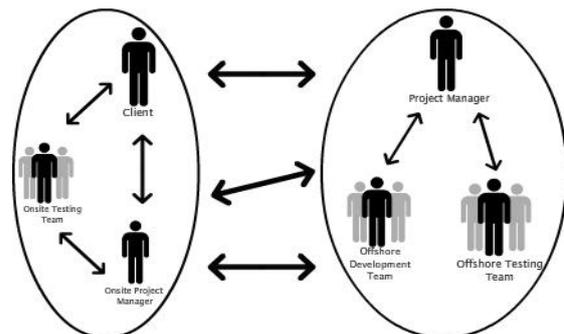
Fonte: Elaboração Própria

Caso 3



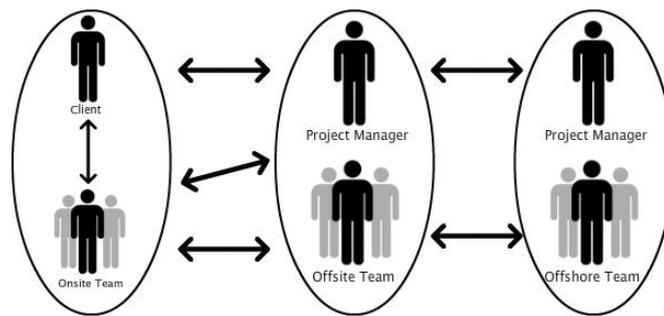
Fonte: Elaboração Própria

Caso 4



Fonte: Elaboração Própria

Caso 5



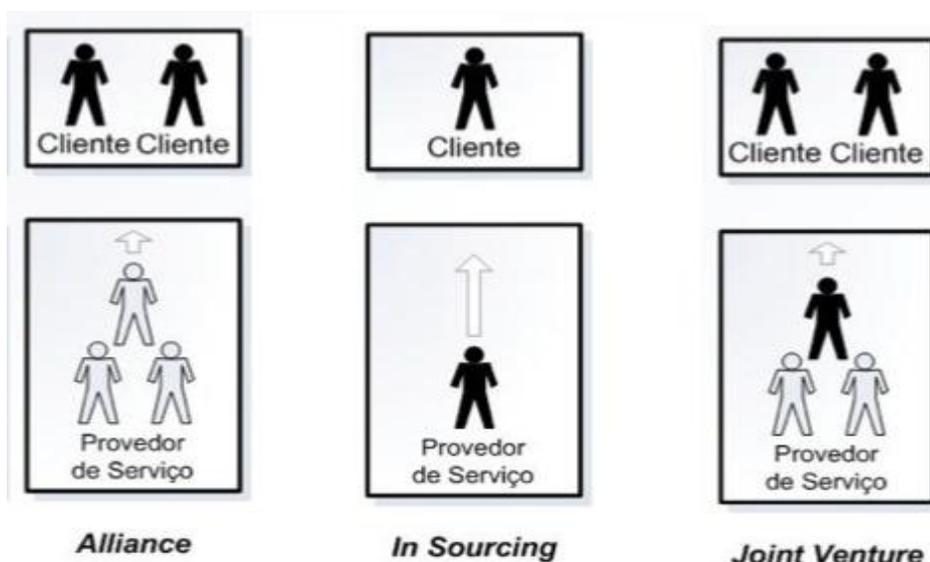
Fonte: Elaboração Própria

Caso 6

Figura 2. Relacionamento entre as partes envolvidas

Quanto ao nível de relacionamento entre as empresas envolvidas são consideradas três formas principais (Figura 3):

- 1) Terceirização (*outsourcing*): a empresa delega uma ou mais atividades para uma empresa externa.
- 2) Filial (*insourcing*): a empresa cria um departamento e delega para ele atividades do desenvolvimento de software
- 3) Colaboração (*joint-venture*): acordo temporário entre empresas para execução de um determinado projeto.



Fonte: Prikladnicki e Audy (2007)

Figura 3. Classificação das empresas quanto ao nível de relacionamento

2.1.2 Motivações e Desafios

Segundo Cockburn (2002), o processo de desenvolvimento de software é tradicionalmente caracterizado por pessoas localizadas lado a lado, permitindo um fluxo constante de informações e idéias. Herbsleb et al (2001) afirma porém que quando a distância entre os desenvolvedores, ou equipe de desenvolvedores, ultrapassam 30 metros a comunicação dentro da equipe se torna equivalente a comunicação entre os desenvolvedores situados a milhares de distância. A abordagem tradicional de desenvolvimento com equipes co-localizadas tem se tornado cada vez mais custosa e menos competitiva. Segundo Carmel et al (2001), diversas organizações começaram a investir no desenvolvimento distribuído, sobretudo, por conter custos e pela rápida disponibilidade de recursos técnicos qualificados. Freitas (2005) expõe a existência de diversos fatores que motivam a adoção de DDS no segmento de TI, sendo as principais:

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

Mesmo com diversos fatores contribuindo para o crescimento do DDS, desenvolver software não é uma tarefa trivial e o cenário DDS adiciona novas classes de problemas. A literatura apresenta artigos associados aos desafios que o desenvolvimento de software adiciona. Alguns desses trabalhos são discutidos a seguir.

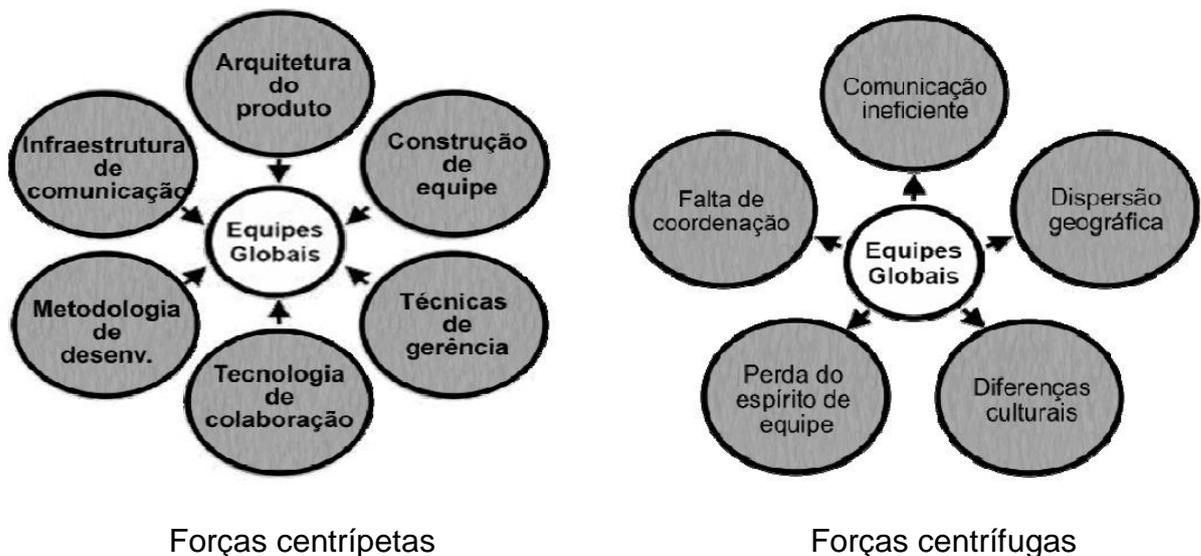
2.1.3 Karolak (1998)

Um modelo específico para projetos de DDS é proposto seguindo uma abordagem adequada ao ciclo de vida do desenvolvimento tradicional (Visão e Escopo, Requisitos, Modelagem, Implementação, Teste, Entrega e Manutenção).

Nesse trabalho ele levanta atividades a serem realizadas durante todo o ciclo de vida do projeto, desde o planejamento até a etapa de manutenção do software a ser desenvolvido, porém não especifica como implantar cada atividade.

2.1.4 Carmel (1999)

Esse trabalho leva em conta os principais desafios e características inerentes a formação de equipes globais. O autor propõe a existência de seis fatores que podem levar a equipe distribuída ao sucesso (Figura 4), chamados de forças centrípetas, e cinco fatores que podem levar ao fracasso, chamadas de forças centrífugas.



Fonte: Carmel (1999)

Figura 4. Forças que influem no DDS

Como fatores de sucesso, o autor sugere:

- 1) **Infra-estrutura de comunicação** - o canal de comunicação entre equipes distribuídas deve ser confiável e com conexões de alta velocidade. Essa

infra-estrutura deve possibilitar a comunicação formal e estimular a comunicação informal. O ambiente de desenvolvimento distribuído deve amparar a equipe no ciclo de produção do software, possibilitando a confiança de que os artefatos e informações recebidas estejam corretos.

- 2) **Arquitetura do produto** - o princípio da modularidade deve fundamentar a arquitetura de um produto que seja desenvolvido no cenário do DDS, tornando possível um menor grau de interdependência dos módulos a serem desenvolvidos.
- 3) **Construção de uma equipe** - conhecer os integrantes do projeto, e seus respectivos papéis e responsabilidade, provê ao grupo mecanismos de controle e coordenação facilitando atividades de comunicação e relacionamento.
- 4) **Metodologia de desenvolvimento** - a metodologia a ser utilizada num projeto de DDS deve ser única entre todos os locais envolvidos, isso possibilita consistência nos artefatos gerados e nos procedimentos. A falta de sincronização das atividades durante o ciclo de desenvolvimento amplia a ineficiência de uma comunicação já complexa.
- 5) **Tecnologia de colaboração** – são imprescindíveis para projetos distribuídos, pois são utilizadas para suporte das atividades do ciclo de desenvolvimento e possibilitam ampliar a comunicação entre os membros dispersos.
- 6) **Técnicas de gerência** - aspectos de técnicas de gerência como: gerência de conflitos, formas de reconhecimento e bonificação, coleta de métricas, entre outros, devem ser levados em conta e adaptados para a realidade do cenário de DDS.

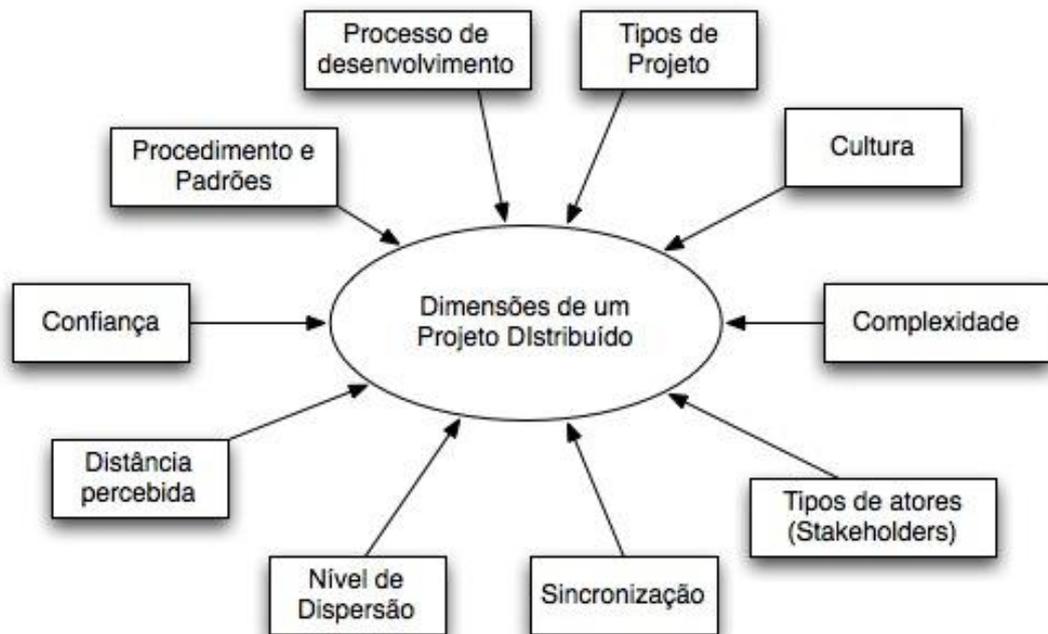
Como fatores de fracasso, o autor sugere:

- 1) **Comunicação ineficiente** - as pessoas utilizam de várias formas de troca de informações realizadas dentro do projeto para obter um entendimento comum. A dispersão das equipes expõe um problema chave no desenvolvimento distribuído, ela impacta diretamente na comunicação (formal, informal, síncrona, assíncrona) de um time.

- 2) **Falta de coordenação** - coordenar o desenvolvimento das atividades de cada local integrante do grupo envolvido se torna uma tarefa não-trivial quando coexistem diferentes culturas, de idiomas e tecnologias.
- 3) **Dispersão geográfica** - O desenvolvimento distribuído lida com dois tipos de dispersão: dispersão geográfica, relativo à distância física entre os envolvidos; e a dispersão temporal, relativo aos diferentes fusos horário nos quais cada equipe estará inserida. A dispersão geográfica impacta sobre a comunicação dentro do time, tornando mais complexa a formação de um sentimento de grupo e diminuindo a empatia necessária para atividades de colaboração. Essa distância atrasa a resolução de problemas, pois torna difícil tomar ciência do projeto como um todo ou saber quem pode resolver determinado problema. A dispersão temporal influi na comunicação síncrona/assíncrona, quando integrantes remotos encontram dificuldades na resolução de suas atividades e ficam na espera para que a área gerencial volte ao horário comercial para ter suas dúvidas respondidas. Isto impõe um gargalo na produção. Fatores como: diferentes fusos, idiomas, culturas entre outros, fazem com que a coordenação de atividades se torne mais complexa.
- 4) **Perda do espírito de equipe** - Manter o senso de equipe em um grupo co-localizado é um tarefa que requer atividades que incentive o sentimento de pertencer a uma comunidade/grupo. O senso de equipe estimula a colaboração nas atividades e com isto reduz o tempo de produção, porém para conseguir manter este relacionamento entre os integrantes é necessário o fortalecimento da empatia entre o grupo e uma ampla comunicação informal. Ambos são fatores críticos dentro do cenário DDS. A distância entre as equipes, a comunicação escassa, diferentes idiomas e variedade cultural intensificam a dificuldade de manter o espírito de equipe.
- 5) **Diferenças culturais** - Problemas associados às diferenças culturais existentes entre os integrantes de um projeto podem ser encontrados até mesmo em equipes co-localizadas. Quando inserido no contexto global, a diversidade de cultura, motivações e comportamento perante determinada situação se torna mais variável, aumentando a possibilidade de conflitos.

2.1.5 Evaristo e Scudder (2000)

Nesta pesquisa é proposto um modelo que detalha as diferentes dimensões que implicam no contexto do DDS. Os autores detalham as vantagens e desvantagens desse tipo de cenário auxiliando no entendimento dos problemas que afetam o sucesso do projeto (Figura 5).



Fonte: Elaboração Própria

Figura 5. Dimensões

Dimensões que afetam o DDS:

- 1) **Tipo de Projeto** – cada projeto é único e possui diferentes necessidades e níveis de criticidade. O tipo de projeto a ser desenvolvido influi diretamente na forma como ele será concebido, projetado, construído e testado. Diferentes técnicas de gerenciamento podem ser adotadas dependendo da característica de cada projeto.
- 2) **Tipos de Atores (stakeholders)** – um projeto sofre influência de vários grupos de *stakeholders*, que podem ser desenvolvedores, pessoas ligadas à gerência, clientes, ou qualquer outra pessoa que tenha ligação com o projeto. Num cenário de DDS esta influência se intensifica em decorrência dos stakeholders possuírem possivelmente diferentes culturas, aumentando a dinâmica de visões do produto.

- 3) **Processo de Desenvolvimento** – um processo envolve uma série de etapas compostas por um conjunto de atividades, procedimentos, artefatos e tecnologia. Ele permite a noção do andamento do projeto e a facilidade de saber o que deve ser produzido em cada etapa, portanto um processo único facilita a troca de informações e comunicação dentro da equipe, pois o desenvolvedor sabe a qualquer momento em que ponto o projeto está. A sincronização das atividades facilita a coordenação e controle do projeto, facilitando questões relacionadas a gerência do plano e controle de qualidade.
- 4) **Procedimentos e Padrões** – diferentes padrões e procedimentos podem ser adotados pela empresa e deverão compor a cultura da organização. A padronização desses aspectos facilitam a troca de informações e artefatos dentro da equipe.
- 5) **Sincronização** – a sincronização permite a paralelização de atividades dentro do projeto, esta capacidade é potencializada quando tratada no ambiente de desenvolvimento distribuído de software, pois os locais de desenvolvimento possuem suas atividades modularizadas no objetivo de que o projeto seja desenvolvido com o menor grau de interdependências.
- 6) **Complexidade** – está intrinsecamente relacionada às dificuldades encontradas em conduzir o desenvolvimento do produto. Diversos fatores podem potencializar ou amenizar a complexidade: unicidade do processo de desenvolvimento, comunicação da equipe, técnicas de gerência.
- 7) **Nível de Dispersão** – está relacionada com distância física entre os envolvidos. Pode estar relacionada com a equipe, ou com os *stakeholders*. O grau de dispersão irá influenciar na complexidade do projeto, pois quanto maior a dispersão maiores problemas relacionados à empatia entre o time, comunicação informal e fatores associados ao idioma e cultura.
- 8) **Distância Percebida** – é a capacidade de visualização dos integrantes como participantes da equipe. Um projeto co-localizado pode permitir a interação semelhante à de um projeto global, isto irá depender das técnicas gerenciais aplicadas no projeto.
- 9) **Cultura** – este aspecto é o mais variável dentro de projetos de DDS. As diferenças culturais propiciam diferentes visões de um determinado

assunto, porém deve ser estritamente tratada no objetivo de evitar ambigüidades ou conflitos.

- 10) **Confiança** – é o fator subjetivo, porém conseqüente do grau de coesão, colaboração e senso de equipe.

2.2 Teste de Software

Teste de software é uma das técnicas de Garantia de Qualidade de Software e que consiste de várias atividades com o objetivo de avaliar se um determinado software se comporta conforme foi especificado. Dijkstra (1979) afirmou que os testes podem assegurar a presença de defeitos porém não sua ausência. Apesar do uso das melhores práticas e técnicas de desenvolvimento, todos os defeitos são de produção humana, tornando os testes fundamentais durante todo o ciclo de desenvolvimento. O teste de software possibilita: encontrar, o mais rápido possível, defeitos no produto, permitindo que esses possam ser corrigidos durante o ciclo de desenvolvimento do software; aumentar a confiança da corretude e qualidade do produto final (Myers, 1979).

Conforme Black (2000), os custos de correção de defeitos aumentam exponencialmente conforme a fase de desenvolvimento na qual o defeito foi encontrado. Para assegurar que os defeitos sejam encontrados o quanto antes é necessário ter um processo de testes bem definido. Mats (2001) afirma que a atividade de testes sofre com os seguintes principais problemas:

- a) atrasos no cronograma do projeto, deixando a equipe de teste impossibilitada de completar os testes planejados devido à redução de recursos e tempo; carência na rastreabilidade de casos de teste entre diferentes versões do sistema, dificultando o reuso e repetição dos testes após modificações dos requisitos;
- b) teste manual ou não-padronizado, resultando em um grande esforço a cada início de uma nova atividade de teste;
- c) incerteza sobre o que está sendo testado, devido à falta de definições dos objetivos e escopo para as atividades de teste;

- d) ausência de critérios para seleção dos casos de teste, definição de sua completude e estabelecimento de um ponto de parada, dentre outros, dificultando a revelação das falhas no produto.

O teste de software é considerado uma das práticas mais custosas do processo de desenvolvimento de um software sendo, por isso, muitas vezes uma das atividades dispensadas na tentativa de economizar, o que adiciona ao projeto do software uma alta probabilidade de insucesso. Para contornar os altos custos é necessário um gerenciamento associado a um bom planejamento e controle dessa atividade (McGregor e Sykes, 2001). Dias e Travassos (2006) expõe que os cinco problemas anteriormente citados demonstram a carência das empresas nas atividades de planejamento e controle dos testes.

2.2.1 Taxonomia de Termos relacionados a Teste de Software

Nessa seção são definidos os principais termos relacionados a Teste de Software no objetivo de auxiliar o entendimento deste trabalho. Os termos são:

- **Caso de Teste** – é um cenário particular com sua entrada, saídas, restrições e comportamentos esperados, refere-se a uma condição particular a ser testada (Craig e Jaskiel, 2002).
- **Procedimento de Teste** – é a descrição do conjunto de passos a serem executados quando realizando um caso de teste (Craig e Jaskiel, 2002).
- **Critério de Teste** – é utilização na seleção e avaliação dos casos de teste e podem ser utilizados como: **Critério de Cobertura dos Testes** (segundo Rocha et al (2001), permite identificar quais partes do programa devem ser executadas, determinando um percentual de elementos necessários a serem executados do conjunto de casos de teste; **Critério de Adequação de Casos de Teste** (segundo Rocha et al (2001), permite verificar se um determinado conjunto de casos de teste satisfaz os requisitos estabelecidos); **Critério de Geração de Casos de Teste** (segundo Rocha et al (2001), utilizado para gerar um conjunto de casos de teste a partir do critério definido.

2.2.2 IEEE Standard 829: Padrão para Documentação dos Testes

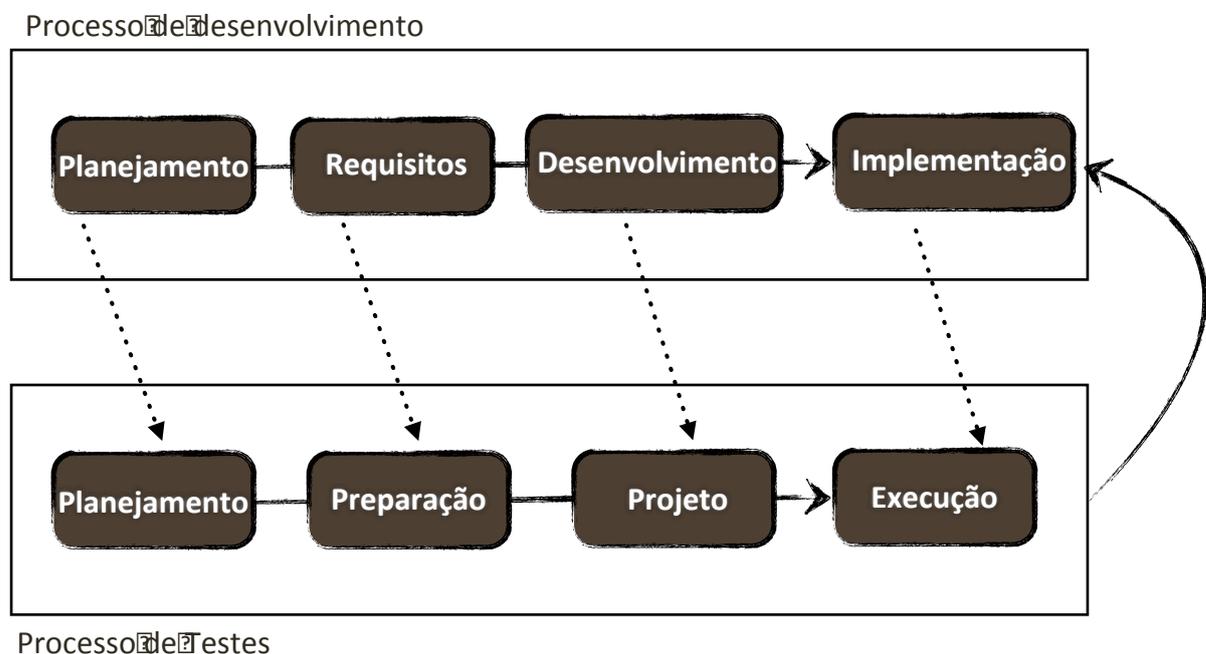
O IEEE Standard 829-1998 (1998) descreve um conjunto de documentos para as atividades de preparação, execução e registro dos testes em um software. Os documentos são descritos, a seguir, de acordo com (Crespo et al., 2004).

- 1) **Plano de Teste** – contém o planejamento para execução dos testes, incluindo o objetivo, escopo, abordagem de teste a ser seguida, recursos físicos e humanos e cronograma das atividades de teste. Nele estão identificados os itens, características e funcionalidades a serem testadas, além das tarefas a serem realizadas e os riscos associados às atividades de teste.
- 2) **Especificação de Projeto de Teste** – é o detalhamento do Plano de Teste para avaliação de um (ou vários) item de teste em relação a uma característica (ou uma combinação de características) de teste que tenha sido especificada no Plano de Teste. Este documento também identifica os casos e os procedimentos de teste, se existirem, e apresenta os critérios para aprovação do(s) item (ou itens) de teste avaliado(s) neste projeto de teste específico. Cada característica (ou combinação de características) a ser avaliada durante os testes deve possuir um documento de Especificação de Projeto de Teste específico.
- 3) **Especificação de Caso de Teste** - define um caso de teste, incluindo dados de entrada, resultados esperados, ações e condições gerais para a sua execução. Cada caso de teste identificado nos documentos de Especificação de Projeto de Teste deve possuir um documento de Especificação de Caso de Teste.
- 4) **Especificação de Procedimento de Teste** - especifica os passos para executar um ou um conjunto de casos de teste. Cada procedimento de teste identificado nos documentos de Especificação de Projeto de Teste deve possuir um documento de Especificação de Procedimento de Teste.
- 5) **Histórico dos Testes** - Apresenta registros cronológicos dos detalhes relevantes relacionados à execução dos testes.
- 6) **Relatório de Incidente de Teste** - Documenta qualquer evento anormal que ocorra durante a execução dos casos e procedimentos de teste e que requeira análise posterior.

- 7) **Relatório de Resumo de Teste** - apresenta de forma resumida os resultados obtidos durante os testes e provê avaliações baseadas nesses resultados.
- 8) **Relatório de Encaminhamento de Item de Teste** - Identifica os itens encaminhados para teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste.

2.2.3 Processo de Testes

Para Pfleeger (2004), um processo pode ser definido como uma série de passos sistematizados, envolvendo atividades, prazos, e recursos, produzindo uma saída esperada. O processo de testes possui características próprias em relação ao processo de desenvolvimento em um projeto de software, e esse deve ser realizado em paralelo ao processo de desenvolvimento para que todas as atividades de desenvolvimento estejam asseguradas quanto ao nível de qualidade (McGregor e Sykes, 2001). Os processos de teste e desenvolvimento se interligam em um ciclo de realimentação constante (Figura 6) que se repete até finalização do produto.

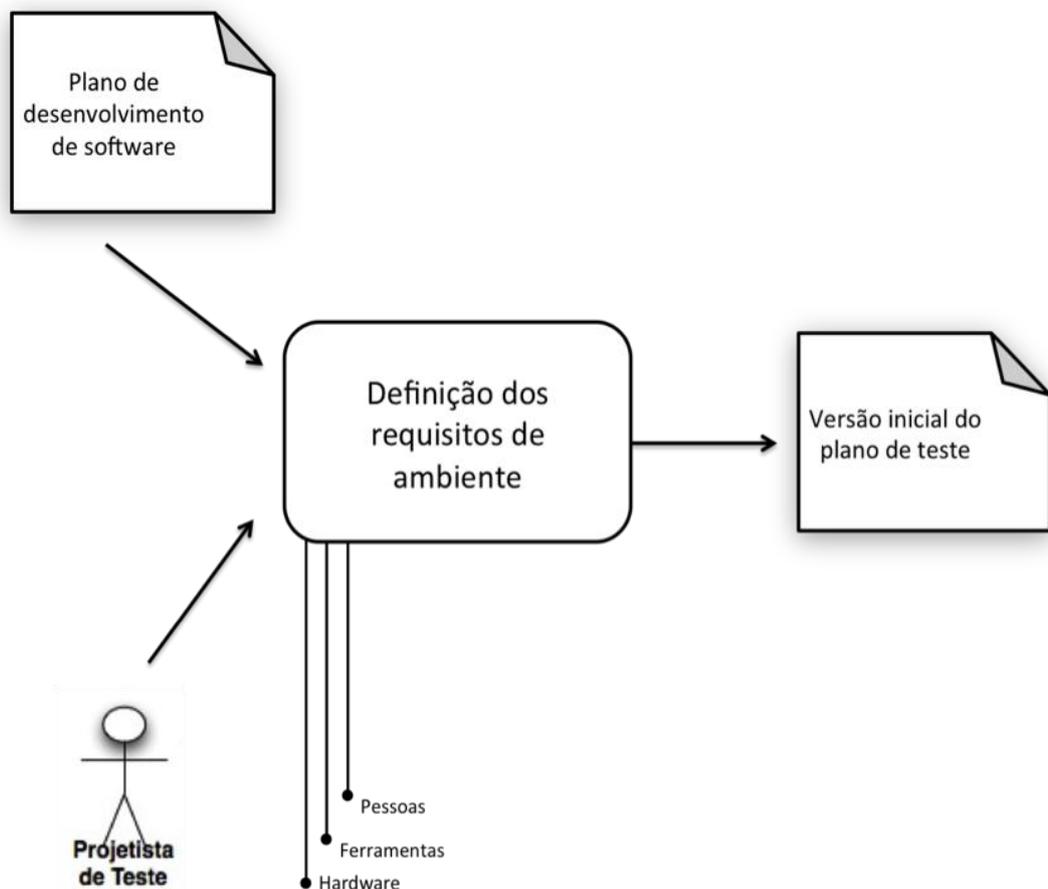


Fonte: Elaboração Própria

Figura 6. Relacionamento entre o processo de desenvolvimento e o processo de testes

O processo de Testes é realizado em fases. São elas: Planejamento, Preparação, Projeto e Execução. Abaixo serão detalhas cada uma dessas fases.

1. **Planejamento** – é a fase onde acontece o levantamento de requisitos e riscos, seleção dos módulos a serem testados e identificação dos indicadores de desempenho assim como a definição da equipe e elaboração de um cronograma. Como artefatos desta fase são produzidos o plano de teste e o cronograma preliminar. Na Figura 7 é esboçada a interação das entradas desta fase (atores e artefatos), a tarefa principal e o artefato produzido ao seu término.

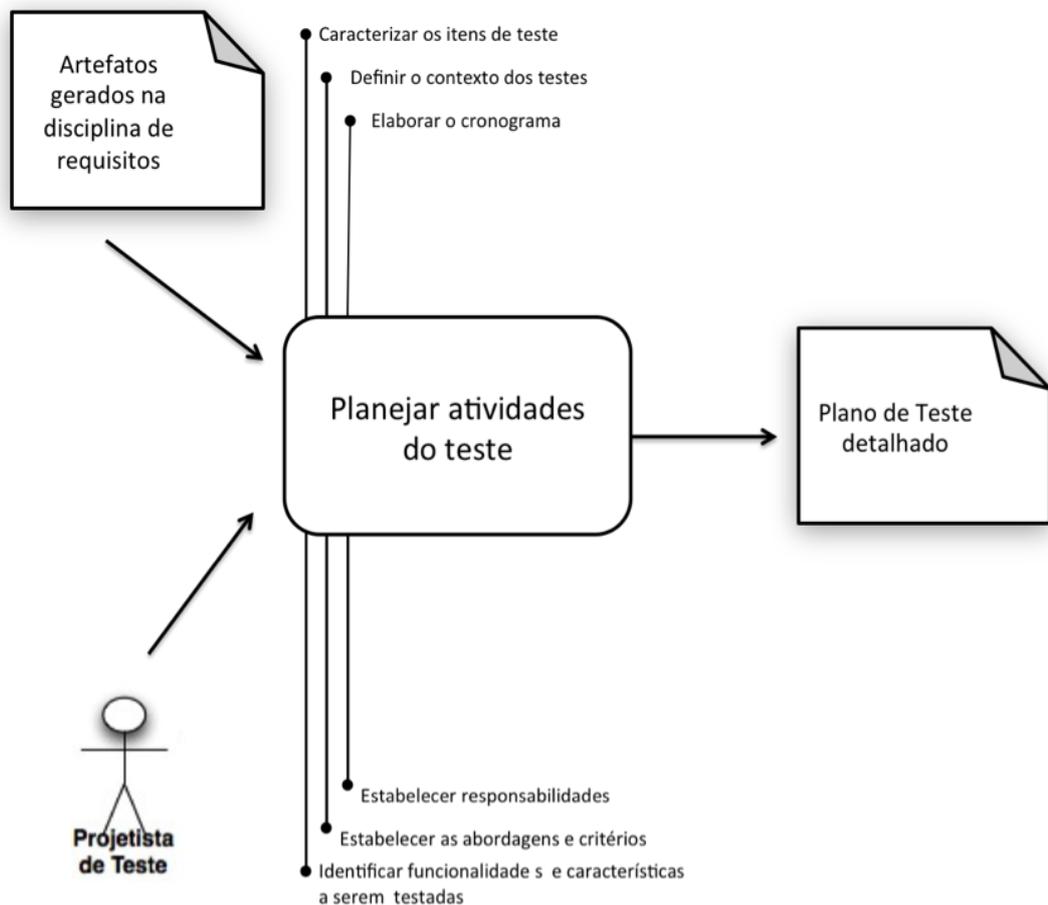


Fonte: Elaboração Própria

Figura 7. Fase de Planejamento da atividade de Testes

2. **Preparação** – é a fase onde são executadas as atividades: determinação da seqüência dos componentes para integração; levantamento da configuração do ambiente para os testes e controle das mudanças;

estabelecimento do ambiente de testes de integração, verificação e validação; instalação e configuração das ferramentas de teste; e, elaboração dos projetos de testes. Nesta fase são gerados três artefatos: o projeto de testes; relatório de equipe e suas respectivas atividades; e, ferramentas de testes disponíveis. Na Figura 8 é esboçada a interação das entradas desta fase (atores e artefatos), a tarefa principal e o artefato produzido ao seu término.

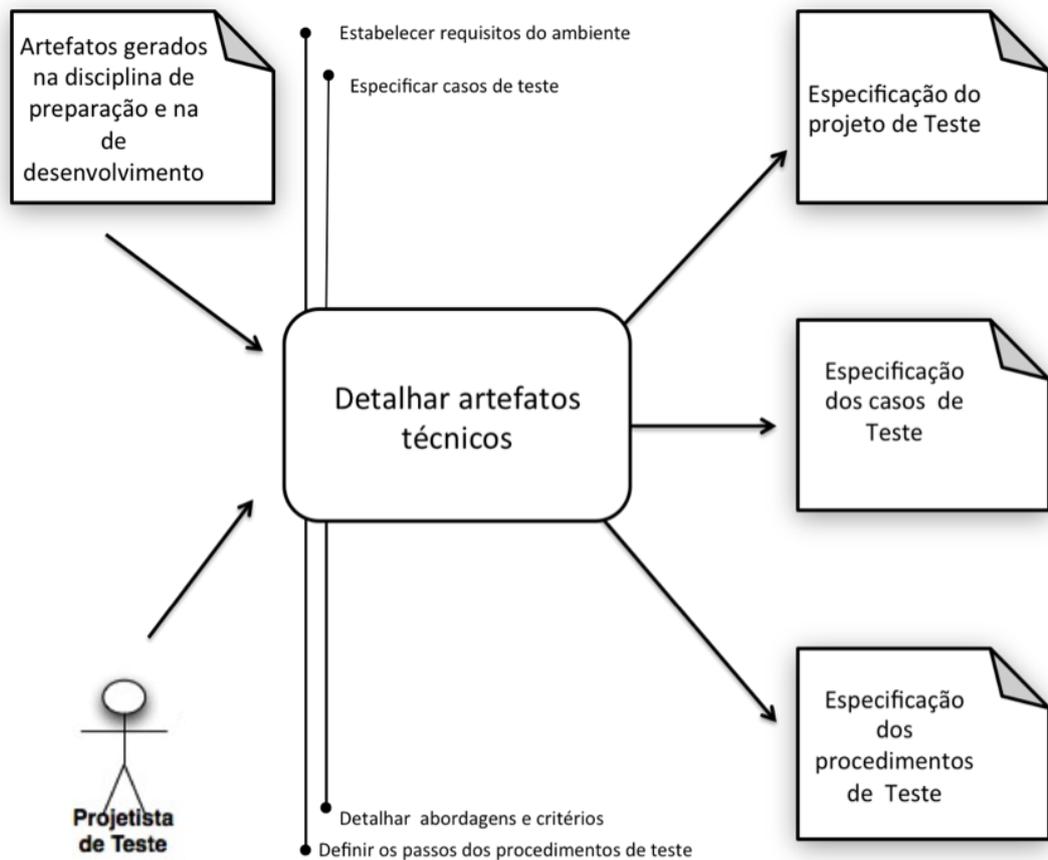


Fonte: Elaboração Própria

Figura 8. Fase de Preparação da atividade de Testes

3. **Projeto** – é a fase onde são definidas as atividades da equipe de testes e estas atividades irão depender de qual técnica foi adotada. As principais técnicas são: Teste Funcional, Teste Estrutural e o Teste Baseado em Erros. O resultado dessa fase tem como artefato a Especificação do Projeto de Teste, com o refinamento do Plano de Teste; e os casos e roteiros de testes

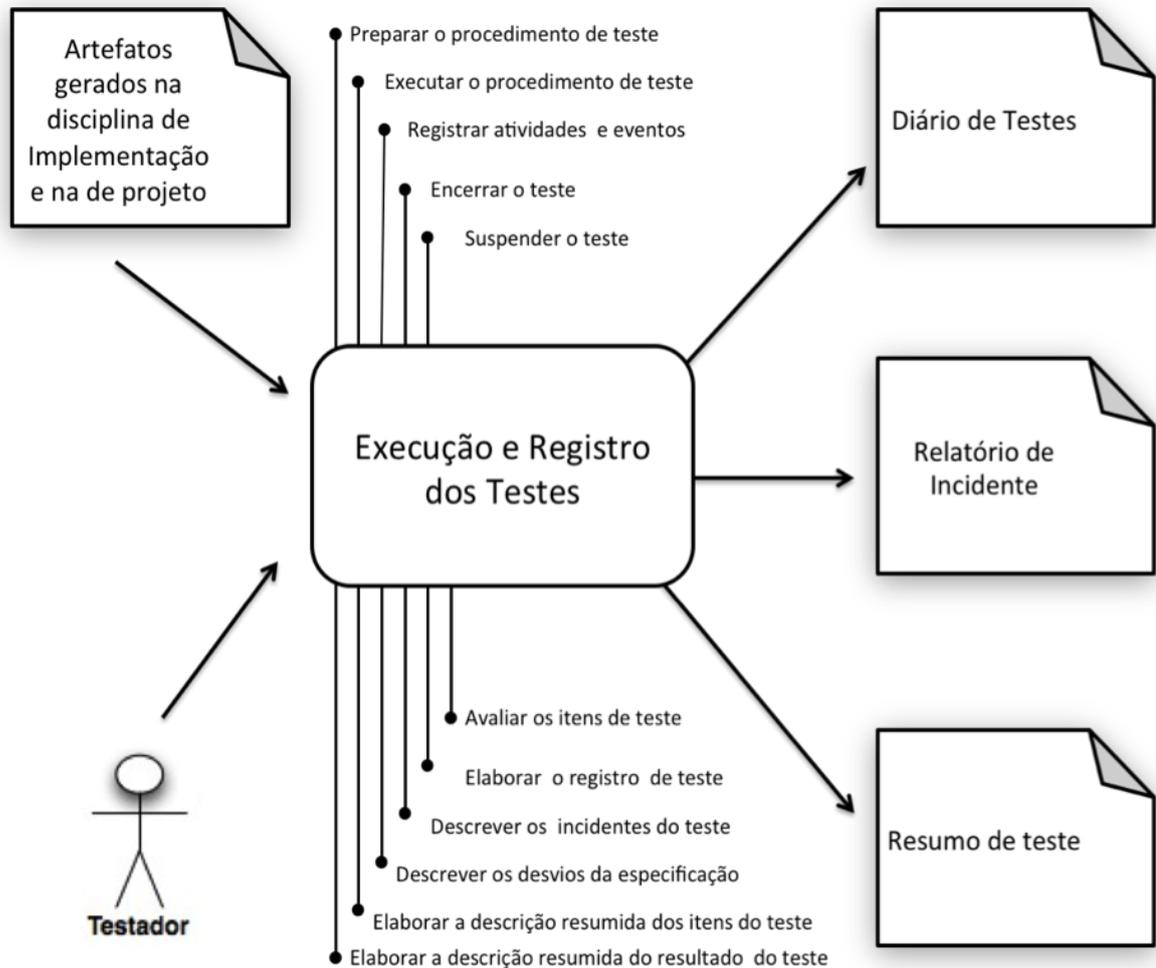
são elaborados com seus critérios de aprovação. Na Figura 9 é esboçada a interação das entradas desta fase (atores e artefatos), a tarefa principal e o artefato produzido ao seu término.



Fonte: Elaboração Própria

Figura 9. Fase de Projeto da atividade de Testes

4. **Execução** – é a fase composta pelas seguintes atividades: execução dos testes de unidade, integração, sistemas e aceitação; elaboração da documentação formal de erros; análise dos resultados; e, solicitação das correções e alterações. Durante esta fase são produzidos: o registro de testes, registro de consolidação de testes, registro de defeitos conhecidos, diários de testes, relatório de conformidades e não-conformidades e o relatório de incidente de testes. Na Figura 10 é esboçada a interação das entradas desta fase (atores e artefatos), a tarefa principal e o artefato produzido ao seu término.



Fonte: Elaboração Própria

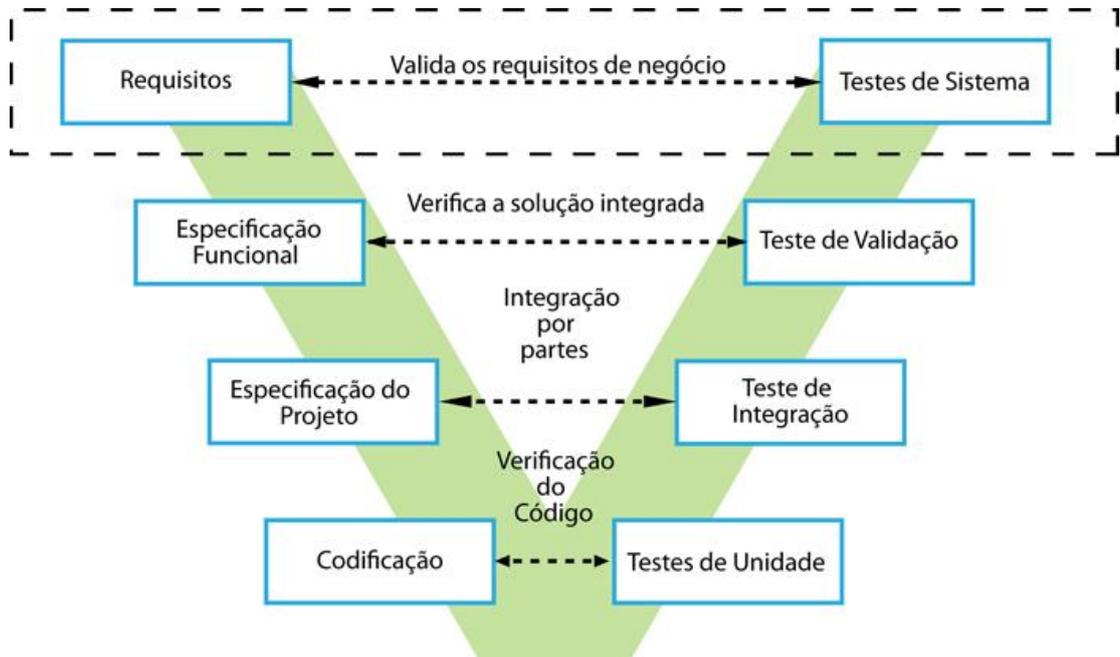
Figura 10. Fase de Execução da atividade de Testes

Durante todo o processo de desenvolvimento, o processo de testes pode ter diferentes atividades associadas à fase do processo de desenvolvimento. A literatura possui diferentes nomenclaturas para as atividades de testes, neste trabalho será utilizada a definida por Pfleeger (2004). As atividades são definidas em cinco estágios:

1. **Teste de Unidade** – tem como cerne as menores unidades de um programa. Seu objetivo é identificar erros de programação, algoritmos mal implementados ou estruturas de dados incorretas.

2. **Teste de Integração** – tem como objetivo verificar se os componentes do sistema ao serem integrados trabalham conforme está descrito nas especificações do sistema e do projeto do programa. Pfleeger (2004) expõe diversas estratégias para fazer a integração dos componentes, podendo citar a integração bottom-up², top-down³, big-bang⁴, entre outras e ela será escolhida de acordo com as características e tamanho do sistema.
3. **Teste de Sistema** – tem como objetivo assegurar que o sistema se comporte como foi especificado. Tem como etapas o teste de desempenho, funcional, de aceitação e de instalação. O teste funcional assegura que o funcionamento do sistema esteja de acordo com o que foi especificado nos requisitos. O teste de desempenho, também chamado teste de performance, assegura que o sistema atende aos tempos de respostas e níveis de desempenho acordados. O teste de aceitação assegura os objetivos de negócio e de requisitos relacionados à funcionalidade e usabilidade, são realizados pelo usuário. O teste de instalação ocorrerá após o teste de sistema ser validado e é realizado o teste de instalação para verificar que o sistema após instalado funciona como devido (Pfleeger, 2004).
4. **Teste de Regressão** – tem como objetivo assegurar a continuidade do funcionamento do sistema em caso de alterações ou manutenções.

O modelo-V de Craig e Jaskie (2002) (Figura 11) relaciona a estrutura de um processo de desenvolvimento de software com as atividades de teste. Eles expõem no seu modelo o paralelismo das atividades do processo de testes em relação ao processo de desenvolvimento. Enquanto o processo de desenvolvimento está na fase de Especificação de Requisitos, pode ser elaborado o planejamento dos Testes de aceitação. Na fase de Projeto de Alto nível, pode ser elaborado os Testes de Sistema. Na fase de Projeto elaborado, podem ser elaborados os Testes de Integração e na fase de Codificação, podem ser elaborados os Testes de Unidade. Todas essas etapas envolvem as fases de Planejamento, Preparação e Especificação, somente após a etapa de codificação é que pode ser iniciada a execução dos testes elaborados.



Fonte: Elaboração Própria baseada em Craig e Jaskie (2002)

Figura 11. Modelo-V de Craig e Jaskie (2002)

Capítulo 3

Metodologia

3.1 Mapeamento Sistemático da Literatura – nesta seção são apresentadas as características gerais das atividades sistemáticas de um Mapeamento;

3.2 Ciclo de vida da Pesquisa – nesta seção é apresentado o detalhamento das etapas desta pesquisa, o processo de mapeamento sistemático e todos os passos que se seguiram para a realização da mesma.

3.1 Mapeamento Sistemático da Literatura

O Mapeamento sistemático é uma revisão abrangente de estudos primários, numa determinada área, buscando identificar quais evidências estão disponíveis nesta área (Kitchenham, 2007). Para Petersen (2008), a realização de um mapeamento além de prover uma visão geral de uma área de pesquisa, possibilita conhecer também as frequências de publicações ao longo do tempo, quantidade e os tipos de pesquisa dentro dela, possibilitando identificar tendências.

Travassos et al (2007) destaca que a revisão sistemática envolve três fases principais: planejamento, execução e análise dos resultados. Sendo que este processo não é puramente seqüencial, podendo ocorrer bastante iteração entre as fases.

Um mapeamento sistemático começa com a definição do protocolo, no qual se especifica os métodos que serão utilizados. Assim, este documento apresenta o protocolo de um mapeamento sistemático da literatura, parte de um trabalho de conclusão de curso, cujo objetivo principal é adaptar o processo de testes quando o desenvolvimento do software é distribuído. Este estudo busca identificar desafios e práticas de apoio a execução da etapa de testes tendo como foco produzir um guia que auxilie no planejamento das atividades de testes no cenário de desenvolvimento distribuído, no qual fatores como, distância física e temporal são presentes, tornando

a atribuição, coordenação, sincronização e acompanhamento de atividades uma tarefa ainda mais complexa

Planejando um Mapeamento Sistemático

A primeira fase do Mapeamento sistemático foca na necessidade de um protocolo detalhado para descrever o processo e os métodos que serão aplicados. Identificar as questões da pesquisa se traduz como o ponto mais importante da etapa de planejamento do Mapeamento, pois toda pesquisa será limitada pelo escopo da questão ser respondida (Dybå et al., 2007). Kitchenham (2007) afirma que a realização de uma revisão deve estar assegurada de sua necessidade e o protocolo formulado deve abordar alguns pontos: objetivo da revisão, as fontes pesquisadas na busca dos estudos primários, as restrições da pesquisa, os critérios definidos de inclusão/exclusão e como serão aplicados na validação de qualidade dos estudos, o processo de extração de dados e como eles foram sintetizados.

Além disso, o protocolo de revisão deve ser avaliado para garantia de que o planejamento é viável. Para isso, muitos pesquisadores sugerem a consulta de especialistas para revisão do protocolo e/ou testar a execução do protocolo.

Conduzindo um Mapeamento Sistemático

As fases de seleção dos estudos primários, da extração e da avaliação dos dados constituem a etapa de execução do mapeamento sistemático. Na seleção são utilizados os critérios de inclusão e exclusão definidos na etapa de planejamento e na extração e avaliação são preenchidos os formulários criados também nessa etapa.

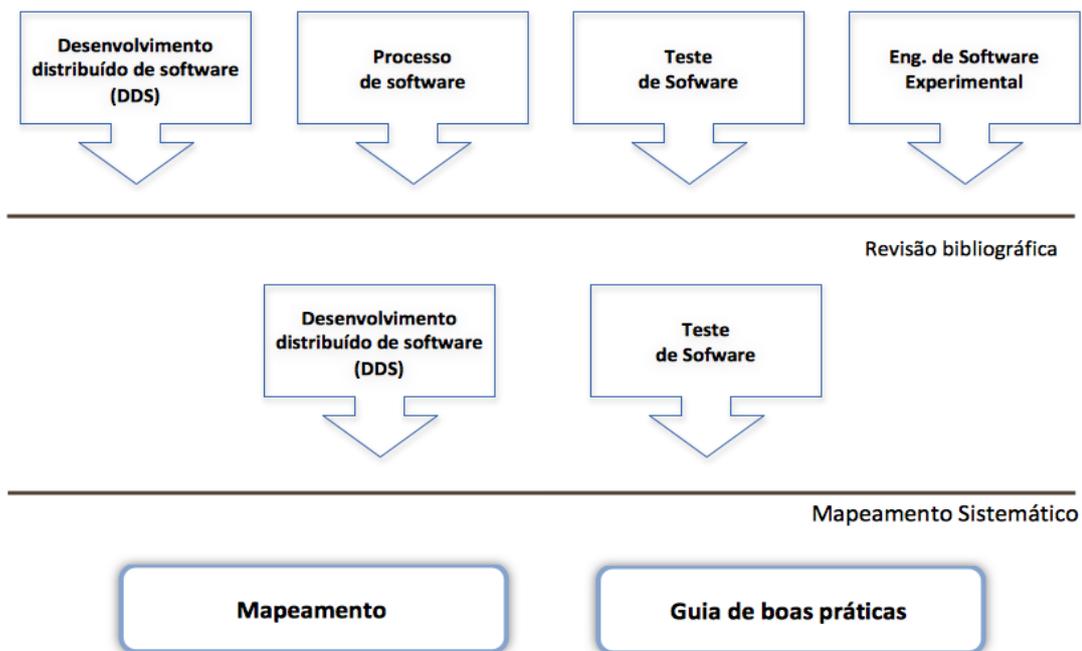
Apresentando os Resultados

A última etapa do mapeamento sistemático consiste, de acordo com a análise e síntese dos dados, na escrita do relatório do mapeamento. Posteriormente, os resultados são apresentados, com informações tabuladas de forma consistente com as questões de pesquisa, utilizando recursos como tabelas, destacando similaridades e diferenças entre os resultados, isto é, ressaltando as possíveis análises e combinação de dados.

3.2 Ciclo da Pesquisa

Na realização deste trabalho foi utilizada uma metodologia composta por duas etapas, conforme ilustrado na Figura 12 primeira fase corresponde à etapa na qual foi realizado uma revisão bibliográfica nos assuntos de importância para esta pesquisa. Com esta revisão foi possível obter maior aprofundamento sobre os quatro grandes campos, verificar a relevância do tema e formular o problema do estudo especificado no Capítulo 1. Na segunda fase da pesquisa foi realizado um mapeamento sistemático da literatura, inicialmente foi elaborado o planejamento, com a definição de um protocolo que foi seguido durante toda a pesquisa.

Ao final do mapeamento, é proposto um guia de boas práticas que auxilia na realização do processo de teste no cenário de desenvolvimento distribuído.



Fonte: Elaboração Própria

Figura 12. Etapas da pesquisa

3.2.1 Processo do Mapeamento Sistemático

Um mapeamento sistemático da literatura foi direcionado no objetivo de explorar e encontrar o maior número de trabalhos primários relevantes e que pudessem responder as questões de pesquisa. Ao analisar esses trabalhos foi possível identificar categorizações a respeito de como está sendo realizada a

atividade de testes no cenário de desenvolvimento distribuído. Essa seção discute alguns tópicos do protocolo de pesquisa definido que guiou este estudo.

3.2.1.1 Questões da Pesquisa

A questão cerne dessa pesquisa é investigar: **o que muda nas atividades de testes de software quando o cenário de desenvolvimento é distribuído?** A pesquisa preza pela abordagem que apóie com boas práticas e ferramentas eficazes a realização das atividades de testes no contexto do DDS.

3.2.1.2 Estratégia de Busca

Travassos et al (2007) afirma que a geração de uma estratégia de busca ocorre de forma interativa e se inicia durante o desenvolvimento do protocolo, com o levantamento das palavras-chave, definição da *string* e escolha das máquinas de busca.

Termos chaves da pesquisa

Os termos e sinônimos identificados são apresentados abaixo:

- **Desenvolvimento Distribuído de Software:** Distributed software development, Global software development, Collaborative software development, Globally distributed work, Distributed development, Distributed teams, Global software teams, Globally distributed development, Geographically distributed software development, Offshore software development, Offshore outsourcing, Dispersed teams;
- **Software testing:** Test team, Testing team, Test management, Test process.

3.2.1.3 String de Busca

Segundo Kitchenham (2007), as strings de busca são construídas a partir das estruturas das questões e adaptações. Geralmente, são necessárias de acordo com as necessidades específicas de cada base de dados. As strings de busca foram geradas a partir da combinação dos termos chave e sinônimos usando OR (ou) e

AND (e), de acordo com as peculiaridades desta biblioteca digital (IEEE). A string utilizada para a questão é listada a seguir (Tabela 1):

Tabela 1. String utilizada no Mapeamento

String da Pesquisa
("Distributed software development" OR "Global software development" OR "Collaborative software development" OR "Globally distributed work" OR "Distributed development" OR "Distributed teams" OR "Global software teams" OR "Globally distributed development" OR "Geographically distributed software development" OR "Offshore software development" OR "Offshore outsourcing" OR "Dispersed teams") AND ("Test team" OR "Testing team" OR "Test management" OR "Test process")

3.2.2 Seleção dos Estudos

Os estudos que podem fazer parte dessa pesquisa são:

- Artigos em Jornais;
- Revistas;
- Conferências;
- Congressos

Uma vez que estudos potencialmente candidatos a se tornarem estudos primários tenham sido obtidos, eles precisam ser analisados para que a sua relevância seja confirmada e trabalhos com pouca relevância sejam descartados. Segundo Travassos et al (2007) critérios de inclusão e exclusão devem ser baseados nas questões de pesquisa. Logo, alguns critérios de inclusão e exclusão são definidos nas próximas subseções, baseados nos trabalhos de Kitchenham (2007) e Travassos et al (2007).

3.2.2.1 Critérios de Inclusão e Exclusão

Durante a etapa de elaboração da estratégia de busca são estabelecidos quais critérios serão utilizados como limitantes na seleção de estudos. Fundamentado nesses critérios o pesquisador poderá restringir quais artigos estão

inseridos no contexto da pesquisa. Nas próximas subseções são definidos estes critérios.

A inclusão de um trabalho é determinada pela relevância em relação às questões de investigação, determinada pela análise do título, palavra-chave e *abstract*. O seguinte critério de inclusão foi definido:

a) Apresenta uma adaptação da atividade de testes direcionada para o cenário de desenvolvimento distribuído de software.

A partir também da análise do título, palavra-chave, e abstract, serão excluídos os estudos que se enquadrem em alguns dos casos abaixo:

a) Estudos que não estejam disponíveis livremente para consulta na web ou Portal da Capes;

b) O estudo tem como tema "testes remotos." Teste remoto refere-se a testes de sistemas fisicamente distantes, testes feitos para embasamentos de sistemas distribuídos, o que foge ao escopo deste trabalho.

c) Estudos claramente irrelevantes para a pesquisa, de acordo com a questão de investigação levantada

d) O recurso não está em Inglês.

3.2.3 Processo de Seleção dos Estudos Primários

O processo de seleção dos Estudos transcorre a respeito da dinâmica de atividades relacionadas à realização do mapeamento, o qual é descrito abaixo.

Dois pesquisadores iniciam as buscas de acordo com a estratégia de busca descrita nas seções anteriores no objetivo de identificar os potenciais estudos primários e a partir da leitura dos títulos dos trabalhos e palavra-chave que a pesquisa retorna, excluem trabalhos que claramente são irrelevantes para as questões investigadas. Cada pesquisador elabora uma lista com os estudos candidatos e comparam entre si formando uma lista unificada. Cada pesquisador chega então a uma lista de potenciais estudos primários. As duas listas são então comparadas e os pesquisadores chegam a uma única lista de potenciais candidatos. Se houver qualquer discordância na inclusão ou exclusão de um estudo, o mesmo deve ser incluído. Após a lista estar unificada, é realizada a leitura do resumo e

conclusão de seus artigos levando em conta os critérios de inclusão e exclusão para então formar a lista final de estudos primários.

Os estudos incluídos serão documentados através do Formulário A e posteriormente no Formulário C. E todos os trabalhos excluídos e o critério que definiu sua exclusão serão documentados no Formulário B. Posteriormente, cada estudo primário será lido e através do Formulário C, a extração dos dados e avaliação da qualidade dos trabalhos será realizada.

Formulário A

O formulário A deverá ser utilizado para armazenar dados relativos aos trabalhos incluídos no estudo.

Formulário B

O formulário B deverá ser utilizado para armazenar dados relativos aos trabalhos não incluídos no estudo.

Formulário C

O formulário C deverá ser utilizado para extração dos dados relativos aos trabalhos incluídos no estudo. Ele é composto por três partes que auxiliaram no processo de extração: Formulário de Coleta de Dados, Questões da Pesquisa e Avaliação da qualidade.

Critérios de Avaliação

Para a realização da avaliação dos estudos primários, algumas questões são definidas, essas questões estão disponíveis nessa seção e fazem parte do Apêndice E. No formulário existem perguntas relacionadas ao processo de condução dos estudos primários e se seus objetivos foram alcançados.

Para a avaliação da qualidade dos estudos é utilizada a adaptação da escala Likert (1932) que permite respostas gradativas através da opinião dos pesquisadores. O pesquisador pode usar os seguintes níveis de concordância ou discordância (concordo, neutro e discordo) para responder as questões dos critérios de qualidade. Para a avaliação, devem ser consideradas as seguintes observações:

- Concordo (2): deve ser concedido no caso em que o trabalho atenda totalmente aos critérios da questão;
- Neutro (1): deve ser concedido no caso em que o trabalho não deixe claro se atende ou não a questão;
- Discordo (0): deve ser concedido no caso em que o trabalho não atenda de forma alguma aos critérios de avaliação, isto é, não existe nada no trabalho que atenda aos critérios da questão.

3.2.4 Estratégia para Extração dos Dados

Para Kitchenham (2007), o objetivo desta etapa é criar formas de extração dos dados para registrar com precisão as informações obtidas a partir dos estudos primários. Deve, portanto, ser projetado como esta estratégia ocorrerá para posterior coleta das informações necessárias à pesquisa. Esta deve ser projetada para coletar as informações necessárias às questões. Um formulário eletrônico é sugerido por vários trabalhos, pois segundo especialistas, o uso pode facilitar a análise posterior. Logo, para apoiar a extração e registro dos dados e posterior análise, será utilizada a ferramenta gratuita Mendeley como gerenciador de referências.

No formulário A, são listados os trabalhos incluídos, com apenas as informações que identificam o trabalho e dados que serão apresentados em forma de gráficos nos resultados da revisão. No formulário B, são listados os trabalhos excluídos e o motivo que levou a exclusão. Já o Formulário C foi usado para extrair as informações gerais e realização da avaliação da qualidade, algumas das informações necessárias são listadas abaixo:

- ID (identificador), Título, Autores, Data de Avaliação, Fonte de pesquisa, Tipo de publicação (Jornal ou Conferência), Data da Publicação;
- Modelo de Negócio - Offshore Insourcing (Internal Offshoring), Offshore Outsourcing (Offshoring), Onshore Insourcing (Demanda doméstica interna), Onshore Outsourcing (Outsourcing), conforme classificação de Audy e Prikladnicki, (2007). Para os trabalhos que o modelo de negócio não esteja claramente definido, será utilizado o termo Distributed (distribuído);

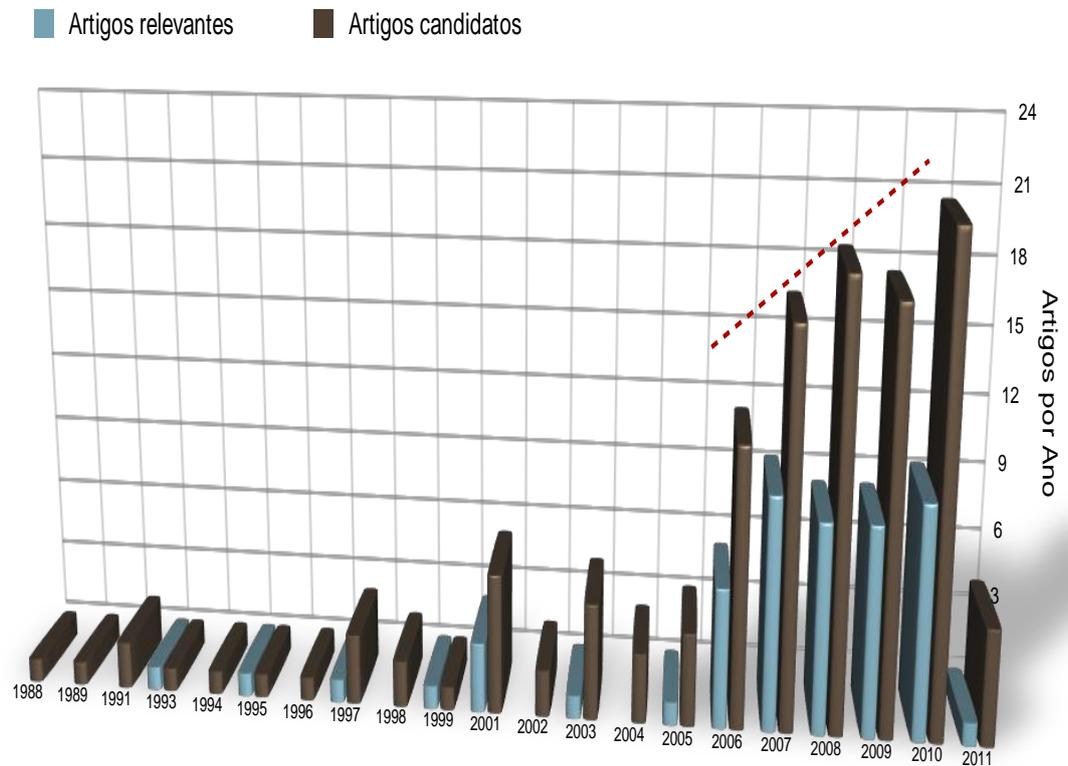
- Foco - Atividades genéricas de testes em DDS, Organização, Planejamento, Controle, Colaboração e Comunicação;

Capítulo 4

Resultados

4.1 Análise Quantitativa do Mapeamento Sistemático

O Mapeamento sistemático foi realizado seguindo a metodologia citada no capítulo anterior. No processo de busca por artigos candidatos, através do uso da *string* na base de dados do IEEE, foram retornados 132 artigos sendo 126 os com conteúdo disponível. Durante o processo de condução do mapeamento, após esses artigos serem analisados, seguindo os critérios de inclusão e exclusão, restaram 55 artigos primários a serem incluídos. A Figura 13 mostra a disposição dos artigos relevantes e dos 55 artigos primários ao longo dos anos e é notório que a partir do ano de 2006 houve um crescimento de publicações a respeito de testes no contexto de DDS.

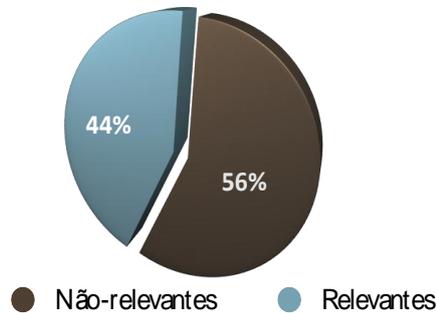


Fonte: Elaboração Própria

Figura 13. Visualização entre os artigos candidatos e relevantes retornados por ano

Os 55 artigos primários representam 44% de todos os artigos retornados (Figura 14), sendo destes 85% publicados em Anais e 13% em periódicos (Figura 15). A produção na área de testes quando inseridos no contexto de DDS vem sendo impulsionada por uma necessidade de mercado e isto repercute nas crescentes produções e congressos associados ao tema. Desses 55 artigos 27% foram publicados no *Internacional Global Software Engineering* (IGSE), o qual demonstra uma boa frequência de publicações relevantes para o tema desta pesquisa (Figura 16 e 17).

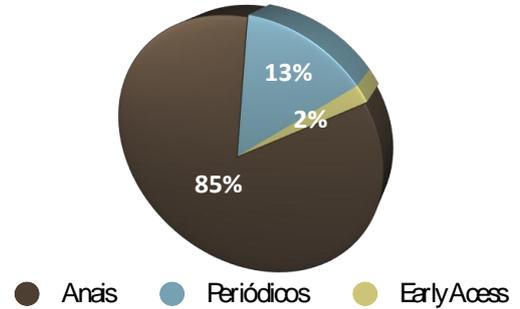
% dos artigos quanto a relevância para esta pesquisa



Fonte: Elaboração Própria

Figura 14. Relevância dos estudos primários

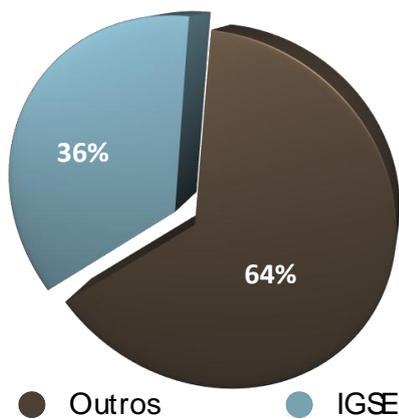
% dos artigos quanto ao local de publicação



Fonte: Elaboração Própria

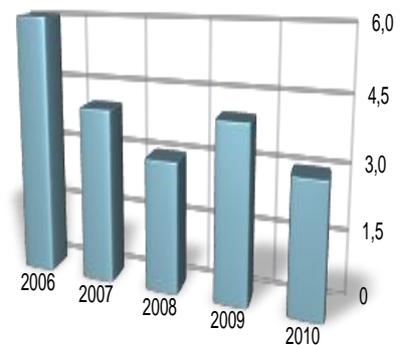
Figura 15. Classificação quanto ao local de produção

% dos artigos relevantes publicados no IGSE



Fonte: Elaboração Própria

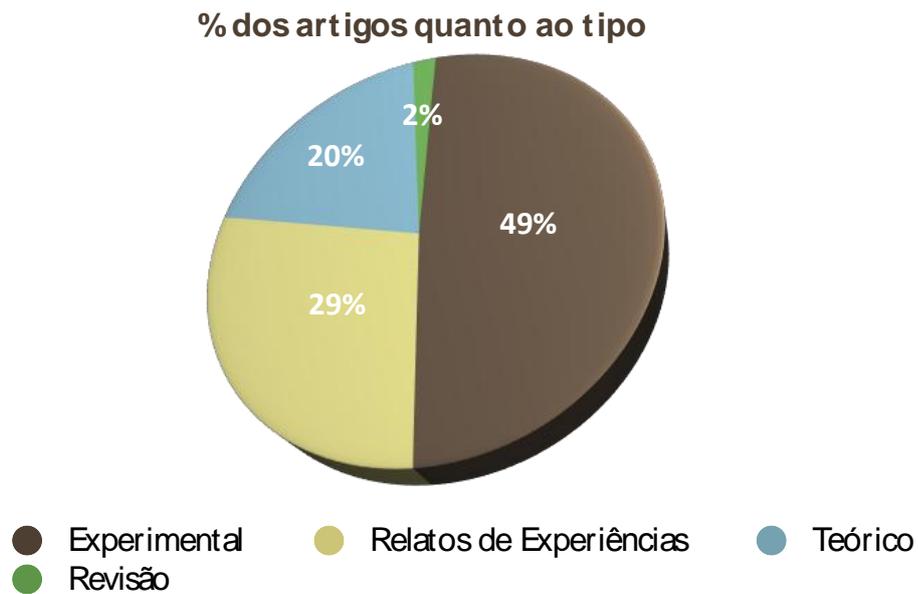
Figura 16. Relação de publicações no IGSE



Fonte: Elaboração Própria

Figura 17. Produções relevantes para esta pesquisa do IGSE vs Ano

Outras análises puderam ser feitas (Figura 18) : dos 55 artigos, foi retornado apenas uma revisão da literatura, 49% possuíam caráter de estudos experimentais e 29% associados a relatos de experiência. Mostrando que a área está em expansão sobre o conhecimento a respeito de testes quando realizado no cenário de DDS.



Fonte: Elaboração Própria

Figura 18. Classificação dos artigos primários quanto a seu tipo

Os resultados da avaliação da qualidade são apresentados pela Tabela 2. A pontuação máxima que um estudo poderia alcançar de acordo com os critérios de qualidade definidos no Capítulo 2, são 10 pontos. Baseado em Beecham et al. (2007) os valores foram divididos em 5 faixas de nota, e a esse valor é associada uma classificação: Excelente, Muito Boa, Boa, Média e Baixa.

Tabela 2. Tabela de qualidade X quantidade de artigos

	Baixa [0,26%)	Média [26%,46%)	Boa [46%,65%)	Muito Boa [65%,86%)	Excelente [86%,100%]	TOTAL
Nº de estudos primários	1	0	10	12	32	55
%	2%	0	18%	22%	58%	100

Fonte: Elaboração Própria, com dados da revisão sistemática, baseado em Beecham et al. (2007)

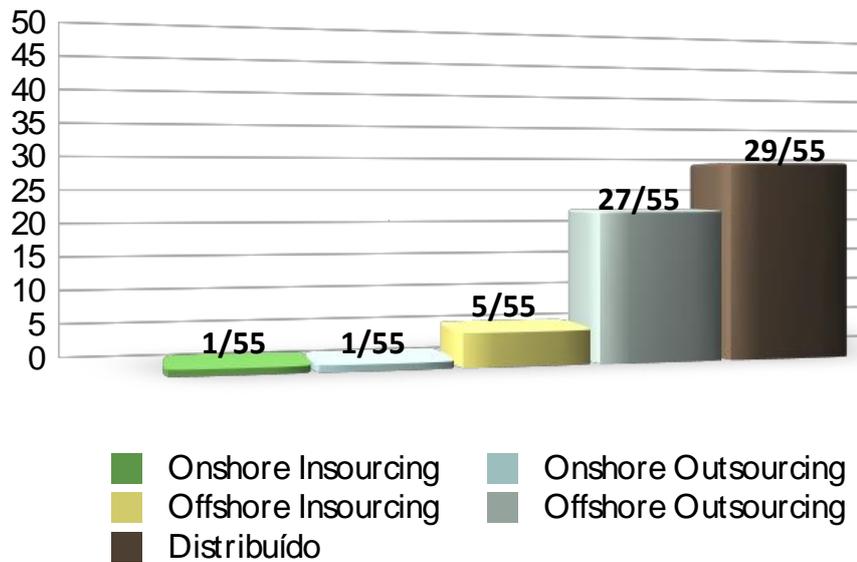
Como pode ser observado, um trabalho está na faixa baixa, enquanto 10 estudos (18%) estão na faixa Boa, 12 estudos (22%) estão na muito boa, e 32 estudos (58%) na faixa Excelente. Portanto, os trabalhos analisados apresentam

qualidade acima da média de acordo com os critérios utilizados. Dentre os estudos classificados com qualidade Excelente, 1 está classificado como *Early Access* (sem local de publicação definido ainda), 1 é proveniente de periódicos e 29 de anais de eventos. Quanto aos estudos classificados com qualidade Muito Boa, 4 foram publicados periódicos, enquanto 8 foram publicados em anais de eventos. Dos estudos classificados com qualidade Boa, 2 foram publicados em periódicos e 9 em anais de eventos. Por fim, dos classificados como baixa qualidade, 1 foi publicado em anais de eventos. Essas informações podem ser visualizadas na 3. Todos os valores da avaliação da qualidade se encontram no Apêndice A, junto a lista de estudos primários.

Tabela 3. Tabela de Qualidade X local de publicação

Qualidade	Periódicos	Anais de Evento	Early Access	TOTAL
Excelente	1	29	1	31
Muito Boa	4	8	0	12
Boa	2	9	0	11
Média	0	0	0	0
Baixa	0	1	0	1
				55

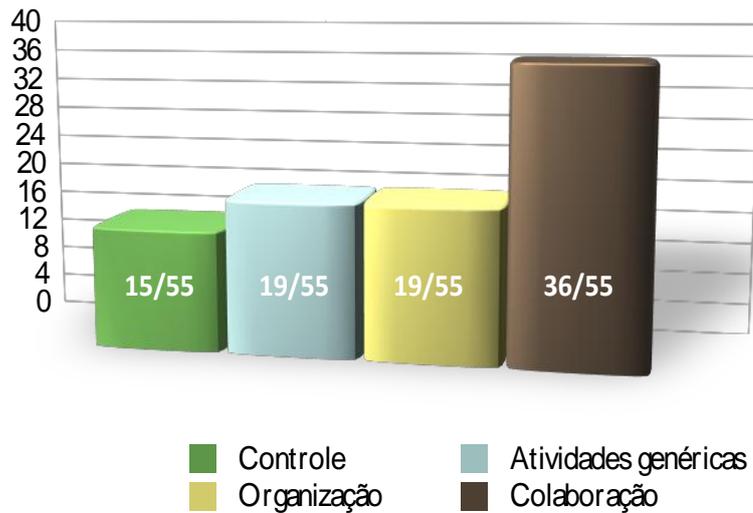
Outra análise que pode ser derivada desse mapeamento foi quanto ao tipo de negócio (Figura 19): *offshore insourcing*, *offshore outsourcing*, *onshore insourcing*, *onshore outsourcing* e distribuído. Dos 55 artigos relevantes 9% possuíam como um dos modelos de negócio o *offshore insourcing*, 40% o *offshore outsourcing*, 2% o *onshore insourcing*, 2% o *onshore outsourcing* e 57% distribuído ou seja a maioria não descreveu o modelo de negócio envolvido, considerando que as soluções propostas para um tipo de negócio irá se aplicar a todos os outros tipos.



Fonte: Elaboração Própria

Figura 19. Classificação dos artigos primários quanto a seu tipo

Uma análise que pode ser derivada desse mapeamento foi quanto ao foco abordado, foram levantados dados sobre 4 áreas criadas baseado no survey de Dias Neto et al (2006), no qual são avaliadas quais práticas de testes são utilizadas num cenário de desenvolvimento real, são elas: controle, organização, colaboração e atividades genéricas. Dos 55 artigos relevantes (Figura 20), 25% entraram em detalhes das atividades genéricas de testes, 35% centraram nas atividades voltadas a organização, 27% em etapas de controle (medição e análise), 58% abordaram a colaboração (utilização de ferramentas e compartilhamento de informações) sendo 49% focados em atividades voltadas à comunicação. Com isso podemos observar que atividades voltadas a área de colaboração possuem um grande destaque dentre os artigos primários, isso mostra que apesar das soluções descritas serem perfeitamente aplicáveis em projetos tradicionais co-localizados, esta área se mostra crítica quando realizada no contexto do cenário distribuído.



Fonte: Elaboração Própria

Figura 20. Classificação dos artigos primários quanto ao foco

4.2 Análise Qualitativa do Mapeamento Sistemático

Alguns trabalhos despontaram como fontes para classificação das categorias deste trabalho: Casey (2009) investigou em sua pesquisa a questão "Como é possível a atividade de testes em um ambiente de equipe distribuída global ser efetivamente realizada?" em dois projetos com equipes de teste distribuídas para um nó irlandês e um nó indiano. No entanto, além do resultado interessante de interação de equipes distribuídas que consistem, neste caso de testadores, consequências e soluções concretas para a organização do teste, o processo de teste, e artefatos não são apresentados.

Sengupta et al. (2004) propõem Test-Driven Global de Desenvolvimento de Software como uma solução para manter o conhecimento sobre os requisitos e suas mudanças consistentes em todos os locais envolvidos. Este utiliza os artefatos de teste como meio de comunicação. Ele aborda o tópico de engenharia de requisitos e não especificamente resolve o problema de teste.

Com sua ferramenta Ambiente de Teste de Acesso Global (GATE) Zage et al. (2005) apresentam uma solução para gerenciar as informações relacionadas com o teste e, portanto, o apoio eficaz em testes em DDS. A ferramenta oferece suporte, mas

não apontam soluções sobre como estabelecer uma organização de teste em DDS e o processo não é descrito em detalhes. Copstein e de Oliveira (2001) investiga uma direção similar. Eles propõem uma ferramenta baseado no workflow para gestão do processo de teste, com o objetivo de apoiar a consistência entre os documentos para permitir a atribuição distribuídas, controle das atividades de teste e automatizar algumas destas. Este achado é relevante para a gestão de testes em projetos de DDS. No entanto, o trabalho leva em conta apenas as atividades de design do plano de teste e execução de plano de teste, perdendo outras atividades como especificação de teste. Além disso, o cenário do projeto descrito no documento considera apenas uma opção para a distribuição de tarefas: A equipe de desenvolvimento está localizado em um site e a equipe de teste em outro site. A opção dos membros da equipe de teste distribuída não é abordada.

Outra trabalho de extrema importância foi o de Teste baseado em modelos como uma abordagem para enfrentar os desafios da transferência do domínio de conhecimentos e problemas de comunicação em um ambiente global que foi investigado por Bruegge et al (2006). O projeto comparou quatro diferentes realizações de um processo de teste. Os procedimentos variaram na atribuição de tarefas para a equipe co-localizada e para uma outro local e em quais atividades de teste são realizados de forma tradicionais ou baseado em modelos. Eles avaliaram os resultados usando critérios diferentes. Conclusão do projeto foi que a usando o modelo de offshore e a abordagem baseada em modelos, o projeto de teste e execução de atividades são altamente offshorable. Os resultados descritos são muito interessantes e fornece uma visão prática sobre a eficácia dos testes baseados em modelos no contexto offshore. Focalizando na automação de teste offshoring, Tervonen e Mustonen (2009) investigou três casos, de outsourcing, de atividades de automação de teste de colaboradores offshore. Eles identificaram classes principais e riscos classificados: qualidade insuficiente é influenciada pelo colaborador offshore não está familiarizado com ferramentas de teste e software, barreira da língua e as questões de rede. Na classe de riscos temos a falta de apoio para a equipe offshore e problemas de prazo. Eles também apresentaram ações corretivas para enfrentar esses riscos. A investigação focalizou offshoring no contexto do outsourcing.

Bondi e Ros (2009) relatam sua experiência na formação de uma equipe de teste de desempenho estabelecidos na Índia por um engenheiro de performance, em

Nova Jersey (EUA). Os principais desafios foram as diferenças culturais, a falta de conhecimento e experiência em testes de desempenho dos membros da equipe e questões organizacionais. Eles superaram esses usando frequentes scrum meeting, bem como treinamento, a seleção cuidadosa de líderes de equipe no local da Índia e o uso de ferramentas automatizadas para a execução e acompanhamento de resultado de testes de desempenho. Sua conclusão foi que, com estas medidas, a equipe de teste de desempenho remoto aumentou a sua produtividade. A investigação descreve os problemas e soluções a respeito do teste em DDS. Porém, ela foca no treinamento e não diretamente descrever o processo de testes.

Uma análise qualitativa dos 55 artigos relevantes expõem 5 áreas chaves mais abordadas como soluções para um projeto que realize testes no contexto de DDS: comunicação, sentimento de equipe, processo e metodologia, infraestrutura e gestão de conhecimentos. Foi realizada a análise qualitativa e abaixo será detalhado as soluções propostas nos artigos primários para cada área

Área 1 - Comunicação:

Black(2000), Craig e Jaskiel(2002) abordam modos de organizar as equipes de testes, tais como: equipe independente de testes, equipe integrada de testes, desenvolvedores que desempenham atividades dos testadores, terceirização, etc. Muitas empresas adotam uma destas sem se preocupar com os por menores do tipo escolhido, um dos fatores que deve ser planejado e bem estruturado é a comunicação entre os envolvidos para que atividade de testes não seja realizada isolada do desenvolvimento. Da análise qualitativa foi formulada a seguinte lista com as principais necessidades associadas a comunicação:

- Necessidade de implantar uma cultura de colaboração e comunicação fluente entre os envolvidos, sendo de grande auxílio a realização de treinamentos na utilização de sistemas de comunicação. A forma como os times irão se comunicar não deve ser restrita à quantidade de mecanismos de comunicação ofertados;
- Uma boa comunicação com o cliente irá facilitar a realização dos testes de aceitação, uma comunicação fluente entre a equipe pode reduzir o tempo de realização de determinados testes. A utilização de sessões de *Brainstorm* para criar listas de idéia a serem testadas. Num time

offsite a reunião física é impossibilitada, porém através de reuniões de grupo num determinado objetivo se torna possível integrar diferentes partes envolvidas no projeto. As sessões de *Brainstorm* além de permitir uma melhor visualização das funcionalidades a serem testadas pelo grupo permite o compartilhamento de conhecimento entre os envolvidos e fortalecimento do senso de equipe. Essas reuniões podem ocorrer através de múltiplos canais de comunicação: chat, teleconferência, ferramentas integradas ao ambiente para este objetivo, entre outras. Um fator que pesa sobre esse tipo de comunicação é a sua natureza síncrona, em equipes com participantes localizados em diferentes fusos irá existir a necessidade de agendamento para que possa ter interações mais informais entre os envolvidos;

- A comunicação informal é fator de carência associado ao cenário DDS. Promover reuniões de grupo além de gerar motivação e senso de equipe, possibilita entender o que está sendo desenvolvido como um todo e entender o que os outros membros envolvidos estão produzidos. A partir dessa comunicação, a construção na confiança entre os envolvidos é fortalecida e o trabalho é realizado de forma mais rápida e com melhor qualidade.
- Comunicação através de troca de mensagens com auxílio de wikis e portais web;
- Definição de interfaces de comunicação formal obtida por meio de modelos de processos bem definidos, com marcos e métricas bem estabelecidas;
- Inicializar projeto com reunião presencial para que todos os membros das equipes se conheçam e para que o gerente de projeto deixe bem definido, de forma objetiva o papel e responsabilidade de cada um dentro do projeto;
- Reuniões face a face sempre que possível. Podendo ser utilizado ferramentas de apoio quando a não possibilidade de reunião física;

- Manter um idioma padrão comum a todos.

Área 2 – Sentimento de equipe:

A elaboração de um projeto com times distribuído convive com a realidade de isolamento físico entre as partes, a empatia necessária para colaboração dentro do grupo só irá existir a partir do momento que os envolvidos trocam experiências e se relacionam de modo a gerar confiança e respeito. Uma forma de iniciar esse processo de interação é:

- Realizar reuniões presenciais como ponto de partida para os projetos.
- Co-alocação temporária da equipe de testes com o cliente ou de algum componente da equipe de testes com a equipe de desenvolvimento. Este envolvimento reduz diferenças culturais, alinha expectativas e possibilita que a equipe de testes tenha um maior conhecimento do produto como um todo. Esta prática pode ser usada como uma estratégia de sincronização das atividades, treinamento e fortalecimento da comunicação.
- Sensibilização do senso de equipe através de notificações automáticas onde membros da equipe podem ver o trabalho de outros membros;
- Co-alocação temporária;
- Reuniões síncronas frequentes através de ferramentas como teleconferência, videoconferência, dentre outras. Desta forma diminui-se a distância entre as equipes do projeto, reduzindo as chances de causar desmotivação.

Área 3 – Processo e Metodologia:

A adoção de um processo único é tida como uma solução para a sincronização das atividades, uniformidade de padrões para artefatos gerados e percepção do ciclo de produção. Os artigos deste grupo confirmam a necessidade do processo de teste ocorrer de forma paralela ao desenvolvimento, consolidando a interação entre a equipe de testes e a equipe de desenvolvimento offsite, as seguintes práticas são propostas nesse objetivo:

- Utilização de metodologias ágeis na formação de equipes multitarefas que apesar de distantes fisicamente possuem reuniões de acompanhamento diariamente, integração contínua e incremental, desenvolvimento iterativo, com entregas e metas frequentes do grupo, política de colaboração;
- Adoção de metodologias de especificação de fácil compreensão e que possuam semântica para transmissão de informações entre os envolvidos;
- Definição de um idioma único para melhor integração dos envolvidos e formalização do processo;
- Utilização de uma especificação formal para minimizar os problemas de compreensão e dependência da comunicação;
- Distribuição de tarefas para o grupo e a possibilidade de que os participantes consigam visualizar em que cada membro está trabalhando, isto é uma prática que facilita o planejamento de alterações e permite em caso de dúvidas ou problema identificar os possíveis membros que podem dar auxílio em determinada tarefa;
- Utilização dos princípios do *Model Driven Development* (MDD) permitindo que os modelos de processo façam parte da documentação. O MDD facilita a portabilidade e quando associado ao Teste Dirigido por Modelos (MDT) possibilita uma maior confiabilidade e produtividade no processo.
- Utilização de modelos arquiteturais independentes dos detalhes de implementação (como por exemplo o *Computational Independent Model* – CIM e *Platform Independent Model*- PIM)
- Utilização de uma arquitetura baseada em componente para melhor distribuição das atividades.
- Definição de uma infra-estrutura para compartilhar informações do projeto e incentivo da colaboração por meio de group awareness.

Área 4 - Infraestrutura:

A necessidade de uma infraestrutura confiável que apoie o processo de construção do software é abordado como um fator relevante para o sucesso de um projeto distribuído. Uma infraestrutura segura inclui as ferramentas de software necessárias, largura de banda necessária e os serviços: locais, de centros de dados, e de redes locais. As seguintes práticas são propostas:

- Utilização de ferramentas extensíveis por meio de plug-ins para auxílio da integração de ferramentas utilizadas no ciclo de vida;
- Utilização de um framework para unir as ferramentas utilizadas no ciclo de vida e auxiliar o processo de gestão de configuração;
- Compatibilidade de ferramentas e versões usadas por sites de desenvolvimento diferentes
- Estabelecimento de um ambiente uniforme e com características bem documentadas no intuito de ser replicado com facilidade pelos diferentes sites;
- Utilização de ferramentas que possibilitem o acompanhamento do processo de testes de software e a documentação realizada ao longo dos testes usando o padrão IEEE-829;

Área 5 – Gestão de Conhecimento:

Um grande problema no processo de desenvolvimento distribuído é o controle e a consistência das versões dos artefatos produzidos. Manter os artefatos atualizados e realistas para para todos os envolvidos é um desafio. Sendo, portanto, uma boa prática a criação de uma base de conhecimento para reunir as informações em um único lugar permitindo às equipes terem acesso ao acompanhamento do projeto de maneira centralizada, além de ter nessa base as informações dos membros das diversas equipes do projeto distribuído. É de grande importância a institucionalização dos resultados obtidos para todos do projeto. E também, abrir um canal para comentários informais entre os integrantes da equipe;

Na tabela abaixo são categorizados os 55 artigos relevantes dentre essas áreas e em detalhes são listadas a quantidade e qualidade do grupo.

Tabela 4. Áreas chaves X artigos X quantidade de trabalhos- qualidade

Contexto	Artigos	Quantidade de Trabalhos - Qualidade
Comunicação	EP_4, EP_6, EP_8, EP_9, EP_12, EP_17, EP_18, EP_19, EP_21, EP_22, EP_26, EP_30, EP_31, EP_32, EP_33, EP_34, EP_35, EP_36, EP_37, EP_38, EP_42, EP_43, EP_44, EP_49, EP_53, EP_55	26 (17 Excelente, 4 Muito Boa, 4 Boa e 1 baixa)
Sentimento de equipe	EP_8, EP_10, EP_12, EP_17, EP_18, EP_21, EP_29, EP_33, EP_37, EP_43	10 (7 Excelente, 2 Muito Boa, e 2 Boa)
Processo e Metodologia	EP_1, EP_2, EP_3, EP_4, EP_5, EP_6, EP_7, EP_8, EP_9, EP_10, EP_11, EP_14, EP_15, EP_18, EP_19, EP_20, EP_21, EP_23, EP_24, EP_25, EP_26, EP_27, EP_30, EP_33, EP_35, EP_38, EP_40, EP_41, EP_46, EP_50, EP_54	31 (16 Excelente, 9 Muito Boa, 5 Boa e 1 baixa)
Infra-estrutura	EP_4, EP_10, EP_11, EP_13, EP_15, EP_17, EP_21, EP_34, EP_39, EP_48, EP_51, EP_52	12 (7 Excelente, 4 Muito Boa, e 1 Boa)
Gestão de conhecimentos	EP_15, EP_22, EP_29, EP_36, EP_45, EP_47	6 (5 Excelente e 1 Boa)

Capítulo 5

Conclusão

5.1 Limitações e Ameaças de Validade

Uma das limitações deste trabalho está associada ao fato deste mapeamento ter utilizado apenas uma base de dados, a IEEE. O que pode acarretar um mapeamento limitado pelas características da base. Durante todo período de Planejamento do Mapeamento desta pesquisa, o portal do IEEE se mostrou um fiel obstáculo. Por ser um processo sistemático, todos os passos precisam ser bem fundamentados e documentados, porém apesar de utilizar uma string para procura o portal oferecia diferentes resultados dependendo do local onde a pesquisa fosse feita. Um dos principais problemas foi o fato da primeira *string* gerada ser mais ampla no intuito de obter mais artigos candidatos e produzir um número ínfimo se comparado quando utilizada uma string mais restrita. A *string* retornou 135 artigos, sendo apenas 16 considerados relevantes, e a após uma nova pesquisa com uma string mais restrita, foram retornados 126 artigos, dos quais 55 foram considerados relevantes.

Outra limitação deste trabalho foi o pouco tempo disponível para sua realização. Realizar um mapeamento sistemático requer um bom conhecimento nas áreas pesquisadas, o que implicou em um alto esforço inicial para entender a área inicialmente, conhecer seus problemas individualmente para depois poder verificar como novos desafios impactam na área. O período de quatro meses teve que servir para entendimento da área, e toda produção do mapeamento.

5.2 Conclusão e Trabalhos futuros

O desenvolvimento de software sempre se apresentou de forma complexa, com diversos desafios e problemas inerentes ao processo. Ao buscar ampliar sua competitividade, empresas de software adotaram o DDS e distribuíram seus processos de desenvolvimento de software. Esta mudança causou impacto não

apenas no mercado, mas na maneira como os produtos são criados, modelados, construídos, testados e mantidos.

A análise dos artigos primários propiciou entender os desafios encontrados quando as atividades de testes são realizadas no cenário de DDS e conseguir responder a questão cerne desta pesquisa: **o que muda nas atividades de testes de software quando o cenário de desenvolvimento é distribuído?**

Esta questão buscou encontrar práticas adotadas como soluções para os desafios adicionados pelo contexto DDS. Abaixo são exposta a lista de práticas propostas ao se realizar testes no cenário de desenvolvimento distribuído.

Lista de Práticas

- Utilização de sessões de *Brainstorm* na elaboração conjunta de cenários de teste que guiarão a especificação dos respectivos casos de teste;
- Utilização de uma arquitetura baseada em componentes, possibilitando testar o componente independente do seu contexto. Esta prática auxiliada por um processo de modelagem possibilita a geração de um representação gráfica do projeto, o que permite à equipe a visualização do projeto como um todo e utilização de técnicas de teste estrutural para identificar casos de teste e realizar testes em relação a especificação, ou seja, sem o acesso ao seu código fonte. Essa prática permite que a equipe inicie o processo de testes o quanto antes, sem o gargalo da espera pela integração de código;
- Modelagem do projeto baseado em cenários, esta abordagem reflete o ponto de vista do usuário final possibilitando a criação de testes mais próxima da necessidade real do usuário;
- Atrelar o processo de teste ao processo de desenvolvimento, com a finalidade de aproveitar artefatos já produzidos e de evitar que problemas existentes na especificação do software sejam propagados para etapas posteriores do desenvolvimento;
- Descrição formal dos requisitos e do sistema como um todo através da linguagem formal de especificação OCL, facilitando o entendimento e uma possível automação dos testes;

- Utilizar um processo de testes bem definido que aborde os métodos de teste a serem utilizados, isto possibilita a uniformidade dentro da equipe, automação dos testes, possuir uma forma de representação única e padronizada e definir a ordem de integração e realização de testes

Dentre os estudos primários dessa pesquisa, apenas 49% são experimentais, o que demonstra que o número de estudos está aumentando podendo possibilitar um mapeamento mais detalhado das necessidades desta atividade neste cenário.

Esta pesquisa buscou além de contribuir com a apresentação de alguns conceitos sobre desenvolvimento distribuído de software, contribuir com desmitificação de que apenas boas ferramentas e infra-estrutura são necessárias para realizar testes. A atividade de teste foi considerada inicialmente como a atividade do ciclo de produção mais propícia para o cenário de DDS por ser a mais sistematizada, porém os fatores humanos relacionados à boa comunicação, motivação pessoal, sentimento de equipe foram considerados de alta criticidade para uma boa interação e produção dentro das equipes de testes globais. A grande diferença observada dos resultados encontrados para os aplicativos num cenário co-localizado é a criticidade de se aplicar a solução. Projetos distribuídos possuem o mesmo caráter dos projetos tradicionais porém com a ausência do convívio físico e as limitações culturais e de distâncias impostas, adotar boas práticas se torna medida de mitigação e não aperfeiçoamento. As práticas descritas neste capítulo podem ser adotadas no cenário tradicional porém devem ser aplicadas num cenário distribuído para que a atividade de testes obtenha sucesso.

Algumas lições aprendidas neste trabalho são listadas abaixo, como forma de ajudar outros pesquisadores em trabalhos futuros:

- Um mapeamento sistemático exige tempo, dedicação e paciência. A etapa de planejamento do mapeamento é de extrema importância para traçar os objetivos a serem alcançados, para que durante a etapa de condução da pesquisa nenhuma informação deixe de ser coletada.
- Muito cuidado ao fazer as *strings* de busca e fazer verificações se os artigos que elas retornam estão realmente de acordo com os utilizados de referência.

Como proposta para trabalhos futuros a serem realizados temos:

- A criação de um processo que aborde as práticas propostas por esta pesquisa;
- Realização de um estudo de caso que consiga comprovar melhorias trazidas ao projeto ao se adotar estas práticas;
- Criação de uma ferramenta que auxilie a execução das práticas propostas.

Finalmente, os resultados desta pesquisa contribuem para uma discussão sobre testes em desenvolvimento distribuído de software e para área de testes, pois através desse mapeamento pode-se observar que é uma área em crescimento e que seus desafios individuais se somam. Automatizar tarefas, gerenciar documentos e oferecer boa infra-estrutura para as atividades são ações consideradas de fundamental importância para qualquer atividade realizada num cenário de desenvolvimento tradicional, considerada de alta importância para as atividades de testes e considerada crítica quando a atividade de testes é realizada no cenário de desenvolvimento distribuído. Os resultados deste mapeamento apresentam à comunidade acadêmica uma necessidade de abordar etapas mais detalhadas das atividades de testes, porém sem deixar de lado os fatores humanos.

Bibliografia

- Audy, J.; Prikladnicki, R. *Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas*. Rio de Janeiro: Elsevier, 2007.
- Binder, J. C. *Global Project Management: Communication, Collaboration and Management Across Borders*. Gower Publishing, 2007.
- Bondi, A.B.; Ros, J.P. Experience with training a remotely located performance test team in a quasi-agile global environment. In *ICGSE 2009: Fourth IEEE International Conference on Global Software Engineering*, pages 254–261. IEEE Computer Society, 2009.
- Black, P.E.; Okun, V.; Yesha, Y. Mutation of Model Checker Specifications for Test Generation and Evaluation, in *Proceedings of the 1st Workshop on Mutation Analysis. Mutation Testing for the New Century*. San Jose, California, pp. 14–20, 2000.
- Bruegge, B.; Dutoit, A. H.; Wolf, T. Sysiphus: Enabling informal collaboration in global software development. In *IEEE International Conference on Global Software Engineering (ICGSE 2006)*, 2006.
- Casey, V. *Software Testing and Global Industry: Future Paradigms*. Cambridge Scholars Publishing, United Kingdom, 2009.
- Carmel, E. *Global Software Teams – Collaborating Across Borders and Time-Zones*. Prentice Hall, EUA, 1999.
- Carmel, E.; Agarwal, R. Tactical approaches for alleviating distance in global software development. *IEEE Software*, 18(2):22–29, 2001.
- Cockburn, A. *Agile Software Development*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2002.
- Craig, R.D.; Jaskiel, S. P. *Systematic Software Testing*. Artech House Publishers, Boston, 2002.

- Copstein, B.; Oliveira, F.M. Management of a Distributed Testing Process using Workflow technologies: A Case Study. In *Seventh Workshop on Empirical Studies of Software Maintenance*, pages 62–64, 2001.
- Crespo, A. N.; Silva, O. J.; Borges, C. A.; Salviano, C. F.; Argollo, M.; Jino, M. Uma metodologia para teste de Software no Contexto da Melhoria de Processo. In: *III Simpósio Brasileiro de Qualidade de Software (SBQS 2004)*, Brasília, 2004.
- Dijkstra E. *Programming COnsidered as a Human Activity*. Classics in Software Engineering. New York: Yourdon Press, 1979.
- Dias-Neto, A. C.; Travassos, G. H. Maraká: Uma Infra-estrutura Computacional para Apoiar o Planejamento e Controle de Testes de Software”. In: *Simpósio Brasileiro de Qualidade de Software*, Vila Velha-ES, 2006.
- Dias-Neto, A.C.; Subramanyan, R.; Vieira, M.; Travassos, G.H. Characterization of Model-based Software Testing Approaches. *Rel. Tec. ES-713/07*, PESC-COPPE/UFR, 2007.
- Dybå T. et al. *Applying Systematic Reviews to Diverse Study Types: An Experience Report*. First International Symposium on Empirical Software Engineering and Measurement, ESEM , 2007.
- Enami, L. N. M. Um modelo de gerenciamento de projetos para um ambiente de desenvolvimento distribuído de software. *Dissertação de Mestrado*, Universidade Estadual de Maringá, PR, Brasil, 2006.
- Evaristo, R.; Scudder, R. Geographically distributed project teams: a dimensional analysis. In: *HICSS, Havaí. EUA*, p. 1-1, 2000.
- Freitas, A. V. *APSEE-Global: a Model of Processes Management of Distributed Software Processes*. Faculdade de Informática – UFRS – RS- Brazil, 2005.
- Herbsleb, J. D.; Moitra, D. *Global Software Development*. IEEE Software Magazine, IEEE Computer Society, EUA, 2001.
- Liang, H. *Distributed Software Development*. Leading Edge Forum. CSC Experience e Results, 2008.
- Likert, R. A Technique for the Measurement of Attitudes. *Archives of Psychology* 140: pp. 1-55, 1932.

- Macgregor, E. et al. The impact of intercultural factors on global software development. Proc. Canadian Conference on Electrical and Computer Engineering, p. 920- 926, 2005.
- Mak, D. K. M.; Kruchten, P. B. NextMove: A Framework for Distributed Task Coordination. Proc. 18th Australian Software Engineering Conference ASWEC 2007, p. 399-408, 2007.
- Mats, L. The top five software-testing problems and how to avoid them. EDN Europe, Vol. 46 Issue 2, p37, 3p, 2001.
- Mcgregor, J. D.; Sykes, D. A. A practical guide to testing object-oriented software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2001.
- Myers, G. The Art of Software Testing. New York: John Wiley, 1979.
- Nidiffer, K. E.; Dolan, D. Evolving distributed project management. IEEE Software., vol. 22, p. 63-72, 2005.
- Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic Mapping Studies in Software Engineering. 12th International Conference on Evaluation and Assessment in Software Engineering, 2008.
- Pfleeger, S. L. Engenharia de Software – Teoria e Prática. Prentice Hall, 2a Edição, São Paulo, 2004.
- Pichler, H. Be successful, take a hostage or "outsourcing the outsourcing Manager". Proc. Second IEEE International Conference on Global Software Engineering, ICGSE, p. 156-161, 2007.
- Prikladnicki, R. MuNDDoS - Um modelo de referência para desenvolvimento distribuído de software. Dissertação de Mestrado, PUC/RS, Porto Alegre, RS, Brasil, 2003.
- Rocha, A. R. C.; Maldonado, J. C.; Weber, K. C. Qualidade de software – Teoria e prática. Prentice Hall, São Paulo, 2001.
- Karolak, D. W. "Global Software Development – Managing Virtual Teams and Environments". Los Alamitos, IEEE Computer Society, EUA, 1998, 159p.

Kitchenham, B.; Charters, S. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, Keele University and University of Durham, July 2007.

Komi-sirviö, S.; Tihinen M. Lessons Learned by Participants of Distributed Software Development. *Journal Knowledge and Process Management*, vol. 12 no 2 p. 108–122, 2005.

Sengupta, B.; Chandra, S.; Sinha, V. Test-driven global software development. In *Proceedings of The 3rd International Workshop on Global Software Development, co-located with ICSE 2004*, 2004.

Tervonen, I.; Mustonen, T. Offshoring Test Automation: Observations and Lessons Learned. In *ICGSE 2009: Fourth IEEE International Conference on Global Software Engineering*, pages 226–235. IEEE Computer Society, 2009.

Vavassori, F. B. Metodologia para o Gerenciamento Distribuído de Projetos e Métricas de Software. Dissertação de Mestrado, UFSC, Florianópolis, SC, Brasil, 2002.

Krishna, S.; Sahay, S.; Walsham, G. Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4):62–66, 2004.

Zage, D.; Zage, W.; Wilburn, C. Test management and process support for virtual teams. Technical report, Ball State University, April 2005.

Zanoni, R. Modelo de Gerência de Projeto Baseado no PMI para ambientes de Desenvolvimento de Software Fisicamente Distribuído. Dissertação de Mestrado, PUC/RS, Porto Alegre, RS, Brasil, 2002.

Apêndice A

Estudos Primários Selecionados

ID	Ano	Fonte	Referência	Qualidade
EP_01	1993	IEEE	Concurrent-development process model	9
EP_02	1995	IEEE	Management of distributed concurrent development for large scale software systems	10
EP_03	1998	IEEE	Process definition for capturing legacy system requirements	10
EP_04	1998	IEEE	Web-based Agile software development	8
EP_05	1999	IEEE	A meta-model for distributed software development	7
EP_06	2001	IEEE	Using extreme programming in a maintenance environment	5
EP_07	2001	IEEE	CORBA-based e-commerce application testing architecture	6

ID	Ano	Fonte	Referência	Qualidade
EP_08	2001	IEEE	Improving validation activities in a global software development	10
EP_09	2005	IEEE	Project management model for a physically distributed software development environment	10
EP_10	2006	IEEE	Agile offshore techniques - a case study	8
EP_11	2006	IEEE	Managing Software Performance in the Globally Distributed Software Development Paradigm	7
EP_12	2006	IEEE	Experience Developing Software Using a Globally Distributed Workforce	8
EP_13	2006	IEEE	Tool Support for Distributed Software Engineering	8
EP_14	2006	IEEE	How BMC is scaling agile development	7
EP_15	2006	IEEE	A Test Specification Method for Software Interoperability Tests in Offshore Scenarios: A Case Study.	10

ID	Ano	Fonte	Referência	Qualidade
EP_16	2006	IEEE	IBM Software Development Leveraging Geographically Distributed Teams - The Interactive Solution Marketplace 2.0 (ISM) Case Study	5
EP_17	2007	IEEE	Globalizing Software Development in the Local Classroom	10
EP_18	2007	IEEE	Of Deadlocks and Peopeware - Collaborative Work Practices in Global Software Development	9
EP_19	2007	IEEE	Building Effective Global Software Test Teams through Training	1
EP_20	2007	IEEE	Offshore Software Development Models	9
EP_21	2007	IEEE	Process Investigations for the Global Studio Project Version 3.0	10
EP_22	2007	IEEE	The Social Context of Software Maintenance	10

ID	Ano	Fonte	Referência	Qualidade
EP_23	2007	IEEE	Distributed Scrum: Agile Project Management with Outsourced Development Teams	8
EP_24	2007	IEEE	Grid Middleware Development in Large International Projects - Experience and Recommendations	8
EP_25	2007	IEEE	Managing Offshore Outsourcing of Knowledge-intensive Projects - A People Centric Approach	7
EP_26	2007	IEEE	Skoll: A Process and Infrastructure for Distributed Continuous Quality Assurance	7
EP_27	2008	IEEE	Requirements Engineering in the Development of Large-Scale Systems	6
EP_28	2008	IEEE	The Impact of Fear on the Operation of Virtual Teams	9
EP_29	2008	IEEE	Challenges and Solutions in Test Staff Relocations within a Software Consultancy Company	9

ID	Ano	Fonte	Referência	Qualidade
EP_30	2008	IEEE	Identifying Pitfalls of System Integration -- An Exploratory Study	10
EP_31	2008	IEEE	Adopting Agile in Distributed Development	6
EP_32	2008	IEEE	Coordination Implications of Software Architecture in a Global Software Development Project	10
EP_33	2008	IEEE	Experience of Collaborative Requirement Management in Dual-Shore Software Maintenance Projects	10
EP_34	2008	IEEE	A Hundred Days of Continuous Integration	10
EP_35	2008	IEEE	Usage of SCRUM Practices within a Global Company	10
EP_36	2009	IEEE	Experience with Training a Remotely Located Performance Test Team in a Quasi-agile Global Environment	10

ID	Ano	Fonte	Referência	Qualidade
EP_37	2009	IEEE	Investigating Collaboration Driven by Requirements in Cross-Functional Software Teams	10
EP_38	2009	IEEE	The 24-Hour Knowledge Factory: Can It Replace the Graveyard Shift?	7
EP_39	2009	IEEE	TAMRI: A Tool for Supporting Task Distribution in Global Software Development Projects	10
EP_40	2009	IEEE	Applicability of Software Reliability Growth Modeling in the Quality Assurance Phase of a Large Business Software Vendor	6
EP_41	2009	IEEE	Skoll: A Process and Infrastructure for Distributed Continuous Quality Assurance	10
EP_42	2009	IEEE	Offshoring Test Automation: Observations and Lessons Learned	10
EP_43	2009	IEEE	The Failure of the Off-shore Experiment: A Case for Collocated Agile Teams	6

ID	Ano	Fonte	Referência	Qualidade
EP_44	2009	IEEE	An Empirical Approach for the Assessment of Scheduling Risk in a Large Globally Distributed Industrial Software Project	10
EP_45	2009	IEEE	Practice of project teaching method in computer course	10
EP_46	2010	IEEE	Practice of project teaching method in computer course	10
EP_47	2010	IEEE	Study on beta testing of web application	5
EP_48	2010	IEEE	OffshoreQA: A Framework for Helping Software Development Outsourcing Companies Comply with ISO 9001:2008	10
EP_49	2010	IEEE	Foundations for Event-Based Process Analysis in Heterogeneous Software Engineering Environments	6
EP_50	2010	IEEE	Status Communication in Agile Software Teams: A Case Study	10

ID	Ano	Fonte	Referência	Qualidade
EP_51	2010	IEEE	An Industrial Survey on Contemporary Aspects of Software Testing	10
EP_52	2010	IEEE	Empirical Study of Tool Support in Highly Distributed Research Projects	5
EP_53	2010	IEEE	Technologies and Tools for Distributed Teams	10
EP_54	2010	IEEE	A Viability Study of an Integrated Approach of Software Development and Test to Distributed Teams	10
EP_55	2010	IEEE	The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor?	10

Apêndice B

ID	Autor	Tipo
EP_01	Aoyama, M.;	Journals
EP_02	Aoyama, M.;	Conferences
EP_03	Markel, S.D.;	Conferences
EP_04	Aoyama, M.;	Journals
EP_05	Goldman, S.;	Conferences
	Munch, J.;	
	Holz, H.;	
EP_06	Poole, C.;	Journals
	Huisman, J.W.;	
EP_07	Probert, R.L.;	Conferences
	Li, W.;	
EP_08	Ebert, C.;	Conferences
	Parro, C.H.; Suttels,	
	R.; Kolarczyk, H.;	
EP_09	Zanoni, R.;	Conferences
	Audy, J.L.N.;	

ID	Autor	Tipo
EP_10	Danait, A.;	Conferences
EP_11	Gaurav Caprihan	Conferences
EP_12	Alberto Avritzer; Thomas Ostrand; Elaine J. Weyuker	Conferences
EP_13	Hans Spanjers; Maarten ter Huurne; Bas Graaf; Marco Lormans; Dan Bendas; Rini van Solingen;	Conferences
EP_14	Gat, I.;	Conferences
EP_15	Brahim Sakout Andaloussi; Andreas Braun;	Conferences
EP_16	Vito Vitale;	Conferences
EP_17	Richardson, I.; Moore, S.; Paulish, D.; Casey, V.; Zage, D.;	Conferences
EP_18	Avram, G.;	Conferences
EP_19	Hackett, M.;	Conferences
EP_20	Faiz, M.F.; Qadri, U.;	Conferences
	Ayyubi, S.R.;	
EP_21	Avritzer, A.; Hasling, W.; Paulish, D.;	Conferences

ID	Autor	Tipo
EP_22	Sillito, J.; Wynn, E.;	Conferences
EP_23	Jeff Sutherland; Anton Viktorov; Jack Blount; Nikolai Puntikov;	Conferences
EP_24	Kunszt, P.;	Conferences
EP_25	Jensen, M.; Menon, S.; Mangset, L.E.; Dalberg, V.;	Conferences
EP_26	Porter, A.; Yilmaz, C.; Memon, A.M.; Schmidt, D.C.; Natarajan, B.;	Journals
EP_27	Konrad, S.; Gall, M.;	Conferences
EP_28	Casey, V.; Richardson, I.;	Conferences
EP_29	Larsson, D.; Bertilsson, H.; Feldt, R.;	Conferences
EP_30	van Moll, J.H.; Ammerlaan, R.W.M.;	Conferences
EP_31	Sureshchandra, K.; Shrinivasavadhani, J.;	Conferences

ID	Autor	Tipo
EP_32	Avritzer, A.; Paulish, D.; Yuanfang Cai;	Conferences
EP_33	Bin Xu; Yujuan Dou; Xiaohu Yang; Xiejun Liang; Ma, A.;	Conferences
EP_34	Miller, A.;	Conferences
EP_35	Cristal, M.; Wildt, D.; Prikladnicki, R.;	Conferences
EP_36	Bondi, A.B.; Ros, J.P.;	Conferences
EP_37	Marczak, S.; Kwan, I.; Damian, D.;	Conferences
EP_38	Gupta, A.;	Journals
EP_39	Lamersdorf, A.; Munch, J.;	Conferences ICGSE 2009
EP_40	Beckhaus, A.; Karg, L.M.; Hanselmann, G.;	Conferences

ID	Autor	Tipo
EP_41	Porter Adam; Yilmaz Cemal; Memon M.; Schmidt C.; Natarajan Bala;	Early Access
EP_42	Tervonen, I.; Mustonen, T.;	Conferences
EP_43	Cohen, B.; Thias, M.;	Conferences
EP_44	Avritzer, A.; Lima, A.;	Conferences ICGSE 2009
EP_45	Zhang Yinnan; Wang Xiaochi;	Conferences
EP_46	Zhu Zemin;	Conferences
EP_48	Annous, H.; Livadas, L.; Miles, G.;	Conferences ICGSE
EP_48	Sunindyo, W.D.; Moser, T.; Winkler, D.; Biffel, S.;	Conferences
EP_49	Downs, J.; Hosking, J.; Plimmer, B.;	Conferences

ID	Autor	Tipo
EP_50	Causevic, A.; Sundmark, D.; Punnekkat, S.;	Conferences
EP_51	Prause, C.R.; Reiners, R.; Dencheva, S.;	Conferences ICGSE
EP_52	Rodríguez, J.P.; Ebert, C.; Vizcaino, A.;	Journals
EP_53	Eldh, S.; Brandt, J.; Street, M.; Hansson, H.; Punnekkat, S.;	Conferences
EP_54	Leal, G.C.L.; Huzita, E.H.M.; da Silva, C.A.; Delamaro, M.E.;	Conferences
EP_55	de Souza, Cleidson R. B.; Redmiles, David F.;	Journals

Apêndice C

Estudos Primários Excluídos

ID	Ano	Fonte	Referência
----	-----	-------	------------

ID	Ano	Fonte	Referência
01	1988	IEEE	An environment for the integration and test of the Space Station distributed avionics systems
02	1989	IEEE	A distributed M-ary hypothesis testing problem with correlated observations
03	1991	IEEE	An algorithm for determining the decision thresholds in a distributed detection problem
04	1991	IEEE	Multi-process constrained estimation
05	1994	IEEE	System reliability policy at Motorola Codex
06	1996	IEEE	Configuration management strategy for distributed and diverse software development environments
07	1998	IEEE	Distributed and concurrent development environment via sharing design information

ID	Ano	Fonte	Referência
08	1998	IEEE	An experience in setting-up a configuration management environment
09	1998	IEEE	Infusion of autonomy technology into space missions: DS1 lessons learned
10	2001	IEEE	Evaluating the software test strategy for the 2000 Sydney Olympics
11	2001	IEEE	Empirical studies to identify defects prevention opp
12	2001	IEEE	Better validation in a world-wide development environment
13	2002	IEEE	Test manager: the test automation component for the maintenance of large-scale systems
14	2002	IEEE	The importance of life cycle modeling to defect detection and prevention

ID	Ano	Fonte	Referência
15	2003	IEEE	About the development of a point of sale system: an experience report
16	2003	IEEE	Increasing understanding of the modern testing perspective in software development projects
17	2003	IEEE	Reengineering legacy application to e-business with modified Rational Unified Process
18	2003	IEEE	Author Index
19	2004	IEEE	Preserving distributed systems critical properties: a model-driven approach
20	2004	IEEE	Lesson learned from attempts to implement daily build
21	2004	IEEE	Guide to the Software Engineering Body of Knowledge

ID	Ano	Fonte	Referência
22	2005	IEEE	Observations and lessons learned from automated testing
23	2005	IEEE	Distributed collaborative environments for systems engineering
24	2005	IEEE	Main effects screening: a distributed continuous quality assurance process for monitoring performance degradation in evolving software systems
25	2006	IEEE	Software Engineering Expert System for Global Development
26	2006	IEEE	Generating a Test Strategy with Bayesian Networks and Common Sense
27	2006	IEEE	Products

ID	Ano	Fonte	Referência
28	2006	IEEE	Experiences of Requirements Engineering for Two Consecutive Versions of a Product at VLSC
29	2006	IEEE	The Test Community of Practice Experience in Brazil
30	2006	IEEE	Adaptation in Distributed Projects: Collaborative Processes in Digital Natives and Digital Immigrants
31	2007	IEEE	Sprinting toward Open Source Development
32	2007	IEEE	Risk Mitigation Tactics for Planning and Monitoring Global Software Development Projects
33	2007	IEEE	Experiences in Large-Scale, Component Based, Model-Driven Software Development

ID	Ano	Fonte	Referência
34	2007	IEEE	Incorporating industrial experience to IEC 61499 based development methodologies and toolsets
35	2007	IEEE	A Framework to Enable Offshore Outsourcing
36	2007	IEEE	Test Automation in Practice
37	2007	IEEE	Exploring Trust among Globally Distributed Work Teams
38	2008	IEEE	An empirical study of software developers' management of dependencies and changes
39	2008	IEEE	Requirements Engineering in the Development of Large-Scale Systems
40	2008	IEEE	Best Practices for International eSourcing of Software Products and Services

ID	Ano	Fonte	Referência
41	2008	IEEE	Interorganizational Workflow Collaboration Based on Local Process Views
42	2008	IEEE	ESB Enablement of an International Corporate Acquisition, an Experience Report
43	2008	IEEE	Linking Requirements and Testing in Practice
44	2008	IEEE	IEEE Standard for Software and System Test Documentation
45	2008	IEEE	A Script-Based Testbed for Mobile Software Frameworks
46	2008	IEEE	A Production Support Infrastructure for Live Student Projects
48	2008	IEEE	Towards a Selection Model for Software Engineering Tools in Small and Medium Enterprises (SMEs)

ID	Ano	Fonte	Referência
48	2009	IEEE	Agile Tool Market Growing with the Philosophy
49	--	IEEE	---
50	2009	IEEE	SOArMetrics: A Toolkit for Testing and Evaluating SOA Middleware
51	2009	IEEE	On educating globally distributed software development — A case study
52	2009	IEEE	Understanding Coordination in IT Project-Based Environments: An Examination of Team Cognition and Virtual Team Efficacy
53	2009	IEEE	Semi-automated Test Planning for e-ID Systems by Using Requirements Clustering
54	2009	IEEE	Skoll: A Process and Infrastructure for Distributed Continuous Quality Assurance

ID	Ano	Fonte	Referência
55	2009	IEEE	Key barriers for global software product development organizations
56	2010	IEEE	Work Item Tagging: Communicating Concerns in Collaborative Software Development
57	2010	IEEE	A training tool for Global Software Development
58	2010	IEEE	Preparing Students and Engineers for Global Software Development: A Systematic Review
59	2010	IEEE	Exploitation of social semantic technology for software development team configuration
60	2010	IEEE	Supporting Hardware-Related Software Development with Integration of Development Tools

ID	Ano	Fonte	Referência
61	2010	IEEE	Tools to Support Global Software Development Processes: A Survey
62	2010	IEEE	An Evaluation Framework For Distributed Collaboration Tools
63	2010	IEEE	Establishing a Service-Oriented Tool Chain for the Development of Domain-Independent MBT Scenarios
64	2010	IEEE	Off-Site Commitment and Voluntary Turnover in GSD Projects
65	2010	IEEE	Effort Estimation in Global Software Development Projects: Preliminary Results from a Survey
66	2010	IEEE	Supporting Hardware-Related Software Development with Integration of Development Tools

ID	Ano	Fonte	Referência
68	2011	IEEE	An Empirical Investigation of Client Managers' Responsibilities in Managing Offshore Outsourcing of Software-Testing Projects
68	2011	IEEE	Process Mining Software Repositories
69	2011	IEEE	Evaluation and Measurement of Software Process Improvement - A Systematic Literature Review

Apêndice D

Detalhe da Busca na Fonte do IEEE

String de Busca		
Biblioteca digital: IEEEExplore	Data da busca: 29/04/2011	Período: não limitado
Pesquisador:	Augusta Marques e Andersson Melo	
String de Busca:	("Distributed software development" OR "Global software development" OR "Collaborative software development" OR "Globally distributed work" OR "Distributed development" OR "Distributed teams" OR "Global software teams" OR "Globally distributed development" OR "Geographically distributed software development" OR "Offshore software development" OR "Offshore outsourcing" OR "Dispersed teams") AND ("Test team" OR "Testing team" OR "Test management" OR "Test process")	
Comentários:	Retornou 126 resultados	

Apêndice E

Formulários utilizados nesse trabalho:

Formulário A

Trabalhos incluídos					
ID	Fonte	Título	Autor	Tipo	Ano

Formulário B

Trabalhos excluídos						
ID	Fonte	Título	Autor	Tipo	Ano	Critério de Exclusão

Formulário C

Formulário de Coleta		
ID:	Pesquisador:	Data da avaliação:
Título do Trabalho:		
Autores:	Tipo:	Ano:
Modelo de Negócio: <ul style="list-style-type: none"> ● Offshore Insourcing ● Offshore Outsourcing ● Onshore Insourcing ● Onshore Outsourcing ● Distribuído 	Foco: <ul style="list-style-type: none"> ● Atividades genéricas de testes em DDS ● Organização ● Controle ● Colaboração ● Comunicação 	

Questões da Pesquisa
Quais os principais desafios encontrados na realização das Atividades de Teste quando no cenário do desenvolvimento distribuído de software?
Quais as melhores práticas a serem adotadas na realização das Atividades de Teste quando no cenário do desenvolvimento distribuído de software?

Avaliação da Qualidade		
Item	Critérios de Qualidade	Valores
Introdução/ Planejamento		
1	Os objetivos ou questões do estudo são claramente definidos (incluindo justificativas para a realização do estudo)?	
2	O tipo de estudo está definido claramente?	
Desenvolvimento		
3	Existe uma clara descrição do contexto no qual a pesquisa foi realizada?	
Conclusão		
4	O estudo relata de forma clara e não ambígua os resultados?	
5	Os objetivos ou questões do estudo são alcançados?	