

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**AVALIAÇÃO DE ABORDAGENS DE GESTÃO
DO CONHECIMENTO APLICADAS A UM
PROCESSO DE TESTE DE SOFTWARE
ENXUTO**

ANA ELIZA PEDROSO DA SILVA

ORIENTADOR: PROF. DR. FABIANO CUTIGI FERRARI

São Carlos – SP

Maio/2015

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**AVALIAÇÃO DE ABORDAGENS DE GESTÃO
DO CONHECIMENTO APLICADAS A UM
PROCESSO DE TESTE DE SOFTWARE
ENXUTO**

ANA ELIZA PEDROSO DA SILVA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Engenharia de Software

Orientador: Prof. Dr. Fabiano Cutigi Ferrari

São Carlos – SP

Maio/2015

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

S586a Silva, Ana Eliza Pedroso da
Avaliação de abordagens de gestão do conhecimento
aplicadas a um processo de teste de software enxuto
/ Ana Eliza Pedroso da Silva. -- São Carlos :
UFSCar, 2016.
147 p.

Dissertação (Mestrado) -- Universidade Federal de
São Carlos, 2015.

1. Teste de software. 2. Processo de teste. 3.
TMMi. 4. Gestão do conhecimento. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado da candidata Ana Eliza Pedroso da Silva, realizada em 15/06/2015:



Prof. Dr. Fabiano Cutigi Ferrari
UFSCar



Profa. Dra. Sandra Camargo Pinto Ferraz Fabori
UFSCar



Prof. Dr. Ricardo de Almeida Falbo
UFES

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Fabiano Cutigi Ferrari, sou grata pela confiança e oportunidade em realizar este trabalho. Agradeço também pela disponibilidade e paciência em empreender as infindáveis revisões de meus textos. Por todo o conhecimento adquirido e compartilhado durante a elaboração desta dissertação.

Aos meus pais, Rosa e Joaquim, agradeço pelo carinho e por serem um exemplo na minha vida. Por me ensinarem a importância da educação, e por todos os esforços empreendidos para que eu pudesse chegar a este momento.

Ao Alexandre, meu amorê, agradeço por fazer parte da minha vida e por sempre acreditar em mim. Nada disso teria acontecido se não fosse por você! E aos seus pais, Olga e Clóvis, por me apoiarem nesta jornada e em todas as outras, e por me acolherem como uma filha em seus corações.

À minha irmã Alice, e aos meus bichinhos de estimação, Clarinha, Loro e Obi-Wan, agradeço pela companhia em tantos momentos prazerosos que vivencio ao voltar para casa.

Aos meus colegas do LaPES e amigos de São Carlos e de São Paulo, que de diversas e insólitas maneiras me ajudaram nesta fase da minha vida.

À CAPES, pelo apoio financeiro.

E por fim, agradeço à todos aqueles que inspiraram minha jornada profissional, que fizeram meus olhos brilhar e despertaram em mim a vontade de fazer mais e melhor.

Se dois homens vêm andando por uma estrada, cada um carregando um pão, e, ao se encontrarem, eles trocam os pães, cada homem vai embora com um. Porém, se dois homens vêm andando por uma estrada cada um carregando uma ideia, e, ao se encontrarem, eles trocam as ideias, cada homem vai embora com duas.

Autor desconhecido

RESUMO

Contexto: Para que a atividade de teste seja conduzida de maneira eficaz, deve-se possuir processos bem definidos e apoiados por modelos de maturidade como, por exemplo, o TMMi. Entretanto, para que os resultados obtidos tenham real impacto na qualidade e confiabilidade dos artefatos produzidos, é fundamental que exista a administração do conhecimento pertencente aos indivíduos envolvidos. Neste contexto, a Gestão do Conhecimento é essencial para ajudar os profissionais de teste de software a adquirirem e compartilharem o conhecimento individual e de grupo. Contudo, ressalta-se que as abordagens de Gestão do Conhecimento para teste de software ainda são incipientes e movidas por diferentes conceitos-chave, tais como *feedback*, lições aprendidas e documentos de conhecimento.

Objetivo: O objetivo deste trabalho é investigar o grau de aplicabilidade de abordagens de Gestão do Conhecimento aplicadas em um processo de teste de software enxuto, segundo o ponto de vista de profissionais com experiência em liderança e gestão de equipes de teste em empresas brasileiras.

Metodologia: Por meio de uma pesquisa da literatura, foram identificados três *frameworks* de Gestão do Conhecimento aplicados em teste de software, e a partir destes foram elaborados três processos de Gestão do Conhecimento. Estes processos de Gestão do Conhecimento foram avaliados por dois métodos, uma Análise Crítica aprofundada e um *Survey*, tendo como referencial um processo de teste enxuto. Uma avaliação adicional analisou a viabilidade da combinação dos conceitos-chave dos três processos elaborados em um único processo de Gestão do Conhecimento. Todas as avaliações empreendidas contaram com a participação de profissionais com experiência em liderança e gestão de equipes de teste.

Resultados: As avaliações realizadas indicam o processo de Gestão do Conhecimento mais aplicável, dentre os três investigados neste trabalho, para apoiar as atividades de uma equipe de teste. Tal processo é fundamentado na utilização de *feedback* ao longo de todo o ciclo de teste de software.

Conclusão: A junção de diferentes conceitos-chave aplicados em um processo de Gestão do Conhecimento leva a opiniões divergentes, particularmente em um cenário de teste de software. Combinar estes conceitos em um único processo pode não produzir uma solução melhorada, possivelmente devido à maior complexidade e esforço necessários para gerenciar o conhecimento dentro das equipes de teste.

Palavras-chave: Teste de Software, Processo de Teste, TMMi, Gestão do Conhecimento.

ABSTRACT

Context: A well-defined testing process is necessary to ensure that the required activities are conducted effectively, supported by maturity models such as TMMi. Even so, for this activities results have real impact on the quality and reliability of the produced artefacts, is fundamental manage the knowledge belonging to the involved individuals. In this context, Knowledge Management (KM) is essential to help software testing professionals acquire and disseminate individual and group expertise. However, existing KM approaches for software testing are still incipient and are driven by different key concepts, such as feedback, lessons learned and knowledge documents.

Objective: This work investigates the applicability degree of KM approaches in a streamlined software testing process, from the point of view of professionals with experience in leadership and testing team management in Brazilian companies.

Method: A literature search allowed us to identify three KM frameworks for software testing processes. Based on them, we devised their underlying KM processes, upon which we performed two evaluations by means of a critical, in-depth analysis and a survey. An additional investigation explored the combination of the three KM key concepts in a single KM approach for software testing. All evaluations relied on the opinion of experienced software testing professionals.

Results: Our results indicate the most applicable KM process, amongst the three addressed in our study, to support testing teams in their activities. Such process relies on the concept of feedback provided throughout the software testing lifecycle.

Conclusion: Different key concepts applied in KM process leads to diverging opinions regarding their applicability, in the particular scenario of software testing. Combining such concepts into a single KM process may not yield a better solution, possibly due to the augmented complexity and required effort to manage knowledge within testing teams.

Keywords: Software Testing, Testing Process, TMMi, Knowledge Management.

LISTA DE FIGURAS

2.1	Fases de Teste de Software	20
2.2	Processo de Teste de Software (adaptado do trabalho de Höhn (2011))	22
2.3	Níveis de Maturidade TMMi (adaptada do trabalho de Camargo (2012))	23
2.4	Estrutura dos Componentes do TMMi (adaptado de TMMi Foundation (2010))	24
2.5	Processo de Teste de Software Enxuto Baseado no TMMi Obtido por Camargo (2012)	26
2.6	Em destaque na última coluna, as duas fases do Processo Enxuto Baseado no TMMi (Adaptado do Trabalho de Camargo (2012))	27
2.7	Níveis de Maturidade MPT.Br (Adaptado de MPT.BR (2011))	28
2.8	Hierarquia do Conhecimento	31
2.9	Dinâmica da Criação do Conhecimento (Adaptado de Nonaka e Takeuchi (1995))	33
3.1	Representação Gráfica dos Elementos BPMN, de Acordo com o Documento Oficial BPMN (OMG, 2014)	43
3.2	Fluxograma <i>Processo A</i>	45
3.3	Fluxograma <i>Processo B</i>	47
3.4	Fluxograma <i>Processo C</i>	50
3.5	Notas <i>Processo A</i>	56
3.6	Notas <i>Processo B</i>	60
3.7	Notas <i>Processo C</i>	60
3.8	Análise Crítica Agrupada por Fase do Processo de Teste	61
3.9	Dependências Entre as Práticas	69

3.10	Análise Crítica Agrupada por <i>Processo</i>	70
4.1	Página de Caracterização dos Participantes	74
4.2	Página de Apresentação das Teorias	75
4.3	Página das Questões de Avaliação	76
4.4	Representação das Etapas do <i>Survey</i>	77
4.5	Caracterização dos Participantes	78
4.6	Participantes - GERAL	79
4.7	Participantes - <i>Processo A</i>	80
4.8	Participantes - <i>Processo B</i>	81
4.9	Participantes - <i>Processo C</i>	81
4.10	<i>Processo A</i>	82
4.11	<i>Processo B</i>	83
4.12	<i>Processo C</i>	84
4.13	Mediana <i>Processo A</i>	86
4.14	Mediana <i>Processo B</i>	86
4.15	Mediana <i>Processo C</i>	87
4.16	Nota <i>Survey</i> Agrupada por Fase do Processo de Teste	87
4.17	Dependências Entre as Práticas	93
4.18	Nota do <i>Survey</i> Agrupada por <i>Processo</i>	94
4.19	Notas Obtidas Pelos <i>processos</i> Sem a Seleção dos Perfis	95
4.20	Nota <i>Survey</i> Agrupada por Fase	95
4.21	Nota <i>Survey</i> Agrupada por <i>Processo</i>	96
5.1	Comparação Análise Crítica Versus <i>Survey</i> por Prática - <i>Processo A</i>	100
5.2	Comparação Análise Crítica Versus <i>Survey</i> por Prática - <i>Processo B</i>	102
5.3	Comparação Análise Crítica Versus <i>Survey</i> por Prática - <i>Processo C</i>	105
5.4	Dependências Entre as Práticas	107

5.5	Comparação Análise Crítica Versus <i>Survey</i> por Fase	108
5.6	Comparação Análise Crítica Versus <i>Survey</i> por <i>Processo</i>	109
5.7	<i>Processo Combinado</i>	111
5.8	Página de Avaliação do Novo <i>Survey</i>	115
5.9	Avaliação do <i>Processo Combinado</i> por Práticas	117
5.10	Avaliação do <i>Processo Combinado</i> por Fases do Processo de Teste	117
A.1	Questões Para Análise Crítica	128

LISTA DE TABELAS

2.1	Trabalhos Selecionados - Informações Principais	38
2.2	Trabalhos Selecionados - Informações Complementares	39
3.1	Questões para Análise Crítica	55
3.2	Análise Crítica da Fase Projeto de Casos de Teste do <i>Processo A</i>	57
3.3	Análise Crítica da Fase Projeto de Casos de Teste do <i>Processo B</i>	58
3.4	Análise Crítica da Fase Projeto de Casos de Teste do <i>Processo C</i>	59
A.1	Análise Crítica	129
A.2	Relação das Práticas	142
A.3	Estrutura TMMi Utilizada no Processo de Teste Enxuto Desenvolvido por Camargo, Ferrari e Fabbri (2013)	143
B.1	Tabela de Frequência - <i>Processo A</i>	145
B.2	Tabela de Frequência - <i>Processo B</i>	146
B.3	Tabela de Frequência - <i>Processo C</i>	147

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	14
1.1 Contexto e Motivação	14
1.2 Objetivo	15
1.3 Metodologia	16
1.4 Organização do Documento	17
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA	18
2.1 Considerações Iniciais	18
2.2 Conceitos Básicos de Teste de Software	18
2.2.1 Fases de Teste	19
2.2.2 Técnicas de Teste	19
2.3 Processo de Teste de Software	21
2.3.1 TMMi	23
2.3.2 MPT.Br	27
2.4 Conhecimento e sua Gestão	30
2.4.1 Gestão do Conhecimento	33
2.4.2 Gestão do Conhecimento em Engenharia de Software	34
2.4.3 Gestão do Conhecimento em Teste de Software	36
2.4.4 Abordagens de Gestão do Conhecimento aplicadas em Teste de Software Elegíveis	38
2.5 Considerações Finais	39

CAPÍTULO 3 – ANÁLISE CRÍTICA DOS PROCESSOS DE GESTÃO DO CONHECIMENTO	41
3.1 Considerações Iniciais	41
3.2 Padronização da Representação dos Processos de Gestão do Conhecimento . .	42
3.2.1 <i>Processo A (COLOMO-PALACIOS et al., 2014)</i>	43
3.2.2 <i>Processo B (ANDRADE et al., 2013)</i>	46
3.2.3 <i>Processo C (XUE-MEI et al., 2009)</i>	49
3.2.4 Comparação dos Processos em Termos das suas Principais Características	52
3.3 Análise Crítica	53
3.3.1 Metodologia	54
3.3.2 Resultados da Análise Crítica	56
3.3.3 Avaliação Geral	67
3.4 Considerações Finais	70
CAPÍTULO 4 – SURVEY SOBRE GESTÃO DO CONHECIMENTO EM PROCESSO DE TESTE	72
4.1 Considerações Iniciais	72
4.2 Planejamento e Projeto do Survey	73
4.3 Análise de Dados	77
4.3.1 Amostra e Caracterização dos Perfis	77
4.3.2 Análise dos Resultados	80
4.3.3 Avaliação Geral	92
4.4 Considerações Finais	96
CAPÍTULO 5 – COMPARAÇÃO DE RESULTADOS E INTEGRAÇÃO DE CONCEITOS-CHAVE	97
5.1 Considerações Iniciais	97
5.2 Comparação da Análise Crítica com os Resultados do Survey	98

5.2.1	Disparidade entre os Resultados	99
5.2.2	Avaliação Geral	106
5.3	Integração de Conceitos-Chave dos Processos de Gestão do Conhecimento Avaliados	109
5.3.1	<i>Processo Combinado</i>	110
5.3.2	Avaliação	114
5.4	Trabalhos Relacionados	118
5.5	Considerações Finais	119
CAPÍTULO 6 – CONCLUSÃO		120
6.1	Contribuições	121
6.2	Limitações e Ameaças à Validade	121
6.3	Trabalhos Futuros	122
REFERÊNCIAS		124
GLOSSÁRIO		127
ANEXO A – ANÁLISE CRÍTICA DAS ABORDAGENS DE GESTÃO DO CONHECIMENTO		128
A.1	Análise Crítica	128
A.2	Relação das Práticas do TMMi	142
A.3	Práticas do TMMi presentes no Processo de Teste Enxuto	142
ANEXO B – RESULTADOS DO SURVEY		144

Capítulo 1

INTRODUÇÃO

1.1 Contexto e Motivação

Há algumas décadas, estudos realizados por Beizer (1990) apontavam que 50% dos custos totais de um projeto de desenvolvimento de software eram gastos com atividades de teste. Investigações mais recentes mostram que este percentual gira em torno de 30% a 40% (TMMi Foundation, 2010) e, em alguns casos, pode ser ainda maior, devido à criticidade e complexidade do projeto (ANDRADE et al., 2013). Desta forma, é fundamental que a atividade de teste seja conduzida de maneira eficiente, sem que seu orçamento seja excedido. Contudo, é comum as organizações extrapolarem de 1,5 a 2 vezes seus orçamentos nas atividades de teste de software (KARHU; TAIPALE; SMOLANDER, 2009).

Para amenizar este problema, é importante que as organizações tenham processos definidos. É neste ponto que os modelos de maturidade, como o CMMI (*Capability Maturity Model Integration*) (Software Engineering Institute, 2010), as auxiliam, pois orientam a implementação e melhoria dos processos. Existem modelos de maturidade específicos para a área de teste de software, dentre os quais destaca-se o modelo internacional TMMi (*Test Maturity Model integration*) (TMMi Foundation, 2010). No contexto nacional, há o modelo de maturidade MPT.Br (*Melhoria do Processo de Teste Brasileiro*) (MPT.BR, 2011), que surge como alternativa aos principais modelos internacionais. Contudo, tais modelos são extensos, complexos, de difícil interpretação, e exigem das empresas grandes esforços em sua implementação, especialmente das empresas de pequeno porte. Nestes casos, processos de teste enxutos, que contemplam um subconjunto de atividades incluídas em modelos de maturidade, demandam menor esforço e tornam o processo de teste descomplicado (CAMARGO; FERRARI; FABBRI, 2013). Apesar disso, um processo bem estruturado não será efetivo se os envolvidos não tiverem o compromisso e conhecimento necessários para sua utilização.

Outro obstáculo rotineiro para as organizações é o baixo reuso dos artefatos de teste, e também do próprio conhecimento (ABDULLAH; ERI; TALIB, 2011). Diariamente, analistas de teste realizam atividades semelhantes e enfrentam problemas que se repetem, entretanto a experiência obtida acaba confinada no indivíduo, não sendo explicitada e compartilhada com o resto da equipe. Sendo assim, a repetição dos erros e o tempo perdido com esta situação tornam-se recorrentes ao longo dos projetos, mesmo que existam profissionais com a experiência necessária para solucionar problemas conhecidos. As práticas de sucesso podem ser repetidas se as organizações forem capazes de extrair o conhecimento de seus funcionários, e torná-lo acessível (ANDRADE et al., 2013). Conseqüentemente, processos bem definidos aliados às políticas de propagação do conhecimento podem gerar resultados eficazes na realização das atividades em uma organização de teste.

Na literatura, pode-se encontrar algumas iniciativas de aplicar técnicas de Gestão do Conhecimento em processos de teste de software como uma forma de aumentar a eficiência do processo (COLOMO-PALACIOS et al., 2014; ANDRADE et al., 2013; XU-XIANG; WEN-NING, 2010). Entretanto, a combinação de Gestão do Conhecimento e Engenharia de Software, em particular no tocante à atividade de teste, ainda requer amadurecimento para de fato trazer os benefícios esperados para as equipes envolvidas em processos de desenvolvimento de software.

Um ponto comum que merece atenção e é identificado na maioria dos trabalhos que propõem uma abordagem com apoio automatizado é a quantidade limitada de avaliações realizadas, que restringem a generalização dos resultados obtidos. Por exemplo, a abordagem automatizada proposta por Colomo-Palacios et al. (2014) foi avaliada em quatro ocasiões. No trabalho de Andrade et al. (2013), a abordagem proposta foi avaliada durante os testes de desempenho de um projeto de desenvolvimento *web* em apenas duas ocasiões. Há ainda casos como o relatado por Xue-Mei et al. (2009), no qual não houve avaliação da abordagem automatizada proposta.

Neste contexto, apresenta-se na próxima seção o objetivo do trabalho de mestrado aqui realizado.

1.2 Objetivo

O objetivo deste trabalho foi investigar o grau de aplicabilidade de abordagens de Gestão do Conhecimento em um processo de teste de software enxuto, segundo o ponto de vista de profissionais com experiência em liderança e gestão de equipes de teste. A intenção é identificar aquela abordagem que mais bem atende uma equipe de teste, possibilitando o compartilhamento

do conhecimento, a definição dos melhores profissionais e, principalmente, a melhoria contínua das atividades.

1.3 Metodologia

Para atingir o objetivo proposto, fez-se necessária a realização de algumas etapas, brevemente descritas a seguir:

1. Identificação de *frameworks* de Gestão do Conhecimento: Uma busca sistemática foi empreendida para viabilizar a identificação de três abordagens de Gestão do Conhecimento, cada uma implementando um *framework* aplicado em teste de software.
2. Elaboração de três *processos* de Gestão do Conhecimento: A partir de cada *framework* selecionado, um processo foi proposto utilizando uma notação padronizada na sua representação.
3. Realização de uma Análise Crítica: Uma análise aprofundada avaliou os *processos* elaborados, de acordo com a opinião de profissional da autora desta dissertação, que possui experiência em liderança e gestão de equipes de teste de software.
4. Aplicação de um *Survey*: Uma pesquisa foi realizada para avaliar os *processos* de Gestão do Conhecimento, com a participação de diversos profissionais experientes em liderança e gestão de equipes de teste de software.
5. Comparação dos resultados: Os resultados obtidos nos dois estudos realizados foram comparados visando à definição do grau de aplicabilidade de cada um dos *processos* de Gestão do Conhecimento em um processo de teste de software.
6. Análise da viabilidade da combinação dos conceitos-chave dos *processos* elaborados: Considerou-se a combinação de conceitos fundamentais de Gestão do Conhecimento em um único *processo*, o qual foi avaliado por meio de um segundo *Survey*, realizado com um subgrupo dos participantes do primeiro *Survey*.

Os resultados obtidos em cada uma das etapas são descritos ao longo deste documento, cuja organização é apresentada na próxima seção.

1.4 Organização do Documento

Este trabalho está organizado em 6 capítulos. O primeiro apresentou o contexto e a motivação para o trabalho proposto, bem como o objetivo pretendido e a metodologia utilizada para sua obtenção.

No Capítulo 2 apresenta-se uma visão geral sobre teste de software e processo de teste. Também abordam-se os conceitos básicos de Gestão do Conhecimento e suas aplicações em engenharia e teste de software.

No Capítulo 3 são introduzidos os *processos* de Gestão do Conhecimento elaborados. A primeira avaliação é apresentada, sendo ela uma Análise Crítica que tem como objetivo investigar a aplicabilidade destes *processos* em um processo de teste de software. Esta análise reflete a experiência da autora desta dissertação, a qual demonstra seu ponto de vista como uma profissional experiente em liderança e gestão de equipes de teste de software.

No Capítulo 4 aborda-se a segunda avaliação dos *processos* de Gestão do Conhecimento. O objetivo é o mesmo da Análise Crítica. No entanto, esta segunda avaliação conta com a opinião de diversos profissionais, também com experiência em liderança e gestão de equipes de teste de software.

A comparação dos resultados dos dois estudos é realizada no Capítulo 5. Neste capítulo, adicionalmente, também é investigada a combinação dos conceitos-chave dos *processos* de Gestão do Conhecimento explorados. Esta combinação é apresentada em um único *processo*. Uma terceira avaliação, desta vez com foco neste *Processo Combinado* é conduzida com a intenção de verificar a viabilidade da aplicação deste *Processo Combinado* em um processo de teste de software, dispondo novamente da participação de profissionais da área de teste de software. Esta investigação foi conduzida de maneira exploratória, e abre possibilidades para estudos futuros.

Por fim, no Capítulo 6 apresentam-se as contribuições, ameaças à validade deste trabalho, e também as intenções de trabalhos futuros.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

2.1 Considerações Iniciais

Neste capítulo apresenta-se uma visão geral sobre teste de software, abordando-se os conceitos básicos necessários para compreender as atividades de teste como, por exemplo, as principais fases e técnicas utilizadas. São abordadas as etapas pertencentes a um processo de teste, e dois dos modelos de referência existentes que apoiam a melhoria do processo de teste; um internacional, o TMMi, e um nacional, o MPT.Br.

Apresentam-se também o estudo do conhecimento e sua gestão. O intuito é informar sobre os aspectos gerais da Gestão do Conhecimento e fornecer apoio para a introdução das suas abordagens, com ênfase nas aplicações de Gestão do Conhecimento em Engenharia de Software e os benefícios de sua adoção. Por fim, relata-se uma pesquisa sobre abordagens de Gestão do Conhecimento aplicadas em teste de software.

2.2 Conceitos Básicos de Teste de Software

Teste de Software é o processo de executar um software ou trechos dele com a intenção de revelar os defeitos que foram introduzidos ao longo do desenvolvimento, e assegurar que o código escrito está de acordo com sua especificação (MYERS; SANDLER, 2004). As atividades de teste de software devem ser conduzidas durante todas as etapas do projeto, pois além de melhorar a qualidade e aumentar confiabilidade dos artefatos produzidos, quanto mais cedo um defeito for revelado, menor será o custo da sua correção.

Partindo do princípio de que todo software contém defeitos (MYERS; SANDLER, 2004), é um engano assumir que um software está correto se nenhum defeito foi revelado durante sua

execução. Para que a atividade de teste encontre todos os defeitos presentes, é desejável a realização do teste exaustivo, ou seja, todas as possíveis combinações de entradas devem ser avaliadas. Entretanto, sabendo que a aplicação do teste exaustivo é impraticável devido à infinidade de combinações de entradas, as atividades de teste são divididas em fases e algumas técnicas são aplicadas em cada uma delas. Desta forma, o teste se torna economicamente viável e é realizado em um tempo factível, assegurando a qualidade da atividade e a revelação do maior número de defeitos.

É importante que sejam selecionadas corretamente as técnicas para cada fase de teste a ser realizada. Estas fases e técnicas de teste serão discutidos nas próximas seções.

2.2.1 Fases de Teste

Por ser uma atividade extensa que acompanha todas as etapas de um projeto de desenvolvimento, o teste de software deve assumir uma característica diferente para cada etapa. Sendo assim, as atividades são divididas em fases com focos e objetivos distintos, sendo complementares umas às outras, revelando diferentes tipos de defeitos em cada abordagem. As fases de teste descritas nesta seção estão ilustradas na Figura 2.1.

- **Unidade:** O teste de unidade tem como foco avaliar a menor unidade de software individualmente, executando pequenos blocos de código ao invés de executar o software por completo. Esta atividade facilita a localização de um defeito encontrado, pois a porção de código a ser avaliada é pequena (MYERS; SANDLER, 2004).
- **Integração:** O teste de integração é realizado após a fase de teste de unidade, e tem como foco avaliar a interface entre as unidades. Por exemplo, a troca de parâmetros entre uma unidade e outra, dando sequência ao correto tratamento da informação. Esta atividade auxilia na validação da estrutura do software (MYERS; SANDLER, 2004).
- **Sistema:** O teste de sistema avalia as características funcionais e não funcionais do software e se estas foram implementadas de acordo com as especificações da documentação. É realizado após a fase de teste de integração (MYERS; SANDLER, 2004).

2.2.2 Técnicas de Teste

A impossibilidade da aplicação de testes exaustivos, aliada às necessidades econômicas da organização exige que variadas técnicas de teste sejam aplicadas ao longo das fases de teste

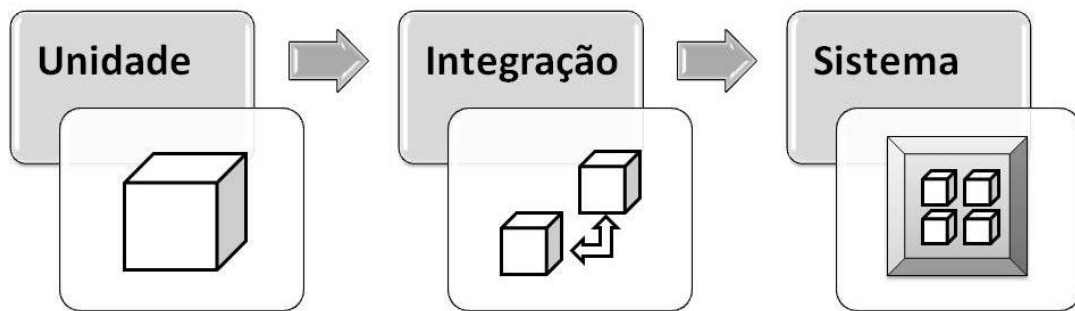


Figura 2.1: Fases de Teste de Software

existentes. As técnicas possuem diferentes estratégias para a seleção de dados de entrada e para a criação de casos de teste, e também distinguem-se na quantidade de esforço, nos custos, na eficácia e na maneira como o testador interage com o software. Ademais, as técnicas são divididas em critérios, e assim como as fases de teste, estes critérios também são complementares.

De acordo com Delamaro, Maldonado e Jino (2007), são três as principais técnicas de teste: Funcional, Estrutural e Baseado em Defeitos. Cada uma delas é brevemente descrita a seguir.

Teste Funcional

O teste funcional não considera detalhes de implementação, e, por isso, também é conhecido como teste caixa preta. Nesta técnica são fornecidas as entradas, e as saídas obtidas são avaliadas para verificar se estão de acordo com a especificação. Como a técnica funcional se baseia na especificação, é necessário que os requisitos estejam rigorosamente definidos e documentados com exatidão.

Por não considerar detalhes de implementação, esta técnica pode ser utilizada em todas as fases de teste, bem como em softwares desenvolvidos em qualquer paradigma de programação. É o teste mais comumente aplicado nas organizações por ter baixo custo em sua execução, entretanto é o de menor eficácia na revelação de defeitos se comparado às outras técnicas, como o teste estrutural e o teste baseado em defeitos (MATHUR, 2008).

Teste Estrutural

O teste estrutural, também conhecido como teste caixa branca, é dependente da implementação do software, ou seja, o conhecimento da estrutura do software fornece meios para que os caminhos lógicos sejam avaliados. Esta técnica utiliza um grafo para representar as instruções do software, no qual blocos de comandos indivisíveis são representados pelos

vértices e os desvios entre os blocos são representados pelas arestas. O teste estrutural complementa o teste funcional, por identificar diferentes tipos de defeitos (MYERS; SANDLER, 2004).

Teste Baseado em Defeitos

O teste baseado em defeitos considera os erros mais frequentes cometidos pelos programadores para gerar diversas versões do software original que contém propositalmente defeitos oriundos de tais erros de programação. Desta forma, ao comparar a diferença de comportamento entre a versão original e a versão com o defeito inserido, é possível mostrar que a versão original está correta (MORELL, 1990).

2.3 Processo de Teste de Software

Como comentado anteriormente, é necessário que as atividades de teste de software sejam conduzidas durante todo o desenvolvimento, assumindo diferentes perspectivas ao longo de seu ciclo. Para tanto é fundamental que as fases e técnicas de teste sejam aplicadas de maneira sistemática, sendo acompanhadas, medidas e melhoradas (METTE; HASS, 2008), ou seja, um processo de teste de software deve ser definido. Questões como escopo, esforço, prazo e alocação devem ser avaliadas no processo, os envolvidos devem ser orientados corretamente quanto aos seus papéis e devem possuir pleno conhecimento das etapas do processo.

As fases de um processo genérico de teste de software contemplam as seguintes atividades: planejamento, projeto, execução, finalização e acompanhamento (CAMARGO, 2012). As quatro primeiras fases são sequenciais e permeadas pela fase de acompanhamento. Todas estão ilustradas na Figura 2.2 e são sucintamente descritas a seguir:

- **Planejamento:** Nesta fase é definida a estratégia de teste, na qual devem ser especificados os recursos e ambientes necessários, os cronogramas a serem seguidos, os critérios de parada, os objetivos e resultados esperados da atividade de teste de software, bem como os produtos a serem entregues. O planejamento deve ser condizente com a realidade do projeto.
- **Projeto:** São criados os casos de teste, identificados os cenários, as condições de testes e revisado o ambiente. Durante esta fase também são especificados os riscos do projeto e definidas quais serão as técnicas, critérios e ferramentas a serem utilizadas.
- **Execução:** Fase em que os casos de teste são executados. Durante esta fase ocorrem diversas iterações, pois defeitos são detectados e corrigidos, então os casos de teste devem

ser reexecutados até que não surjam falhas, ou até que uma decisão estratégica relacionada a custo ou prazo seja tomada. Por isso é fundamental que seja realizado o gerenciamento dos defeitos.

- **Acompanhamento:** O acompanhamento do projeto de teste é realizado com o objetivo de fornecer visibilidade do mesmo através da organização e consolidação dos resultados utilizando métricas previamente estabelecidas. Ocorre a mitigação dos riscos e o replanejamento do projeto deve ser feito, caso necessário.
- **Finalização:** Fase onde informações e resultados são consolidados, analisados e compilados gerando o material de lições aprendidas que servirá como referência para futuros projetos. Ocorre a liberação de recursos e ambientes que não serão utilizados.

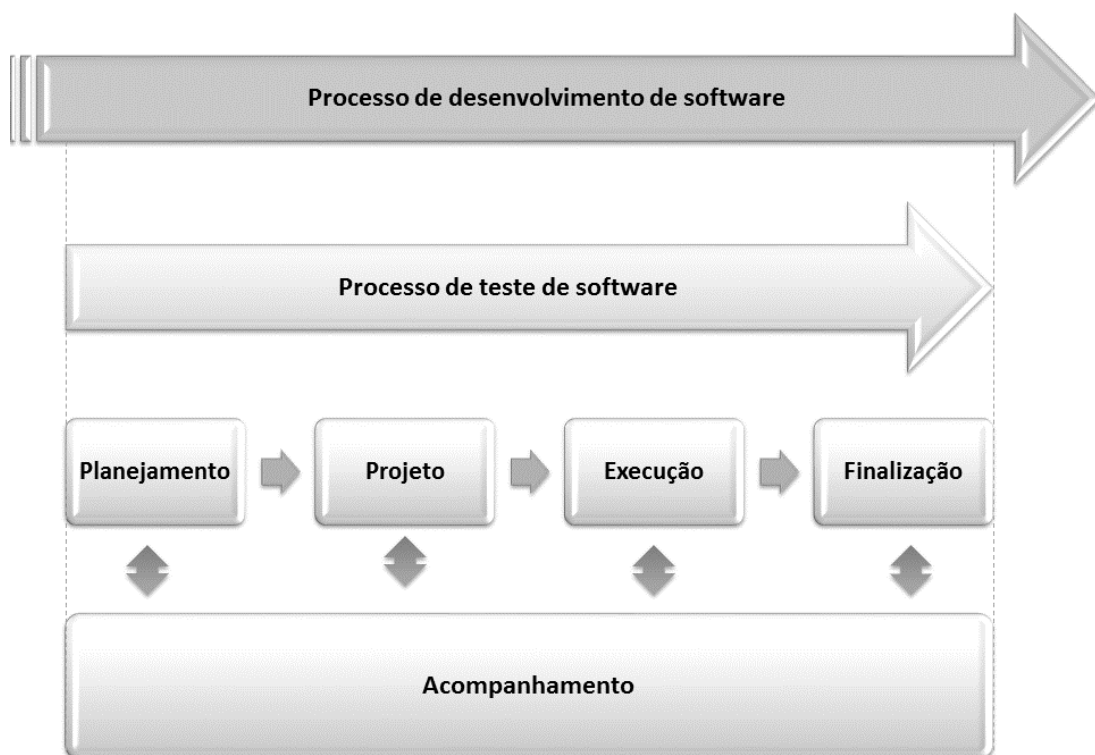


Figura 2.2: Processo de Teste de Software (adaptado do trabalho de Höhn (2011))

A evolução do processo de teste deve ser uma etapa constante a fim de alcançar melhores resultados em sua aplicação. Para isso existem modelos de maturidade que servem como um guia sobre o que um processo deve conter, auxiliam em sua avaliação, além de indicar quais são os passos necessários para implantação de melhorias, pois contam com as melhores práticas da área. Os modelos de maturidade de processo de teste que serão abordados nas próximas seções são o TMMi e o MPT.br.

2.3.1 TMMi

Para orientar a implementação e a melhoria de processos os modelos de maturidade podem ser utilizados, como o CMMI (*Capability Maturity Model Integration*), um modelo bastante difundido entre organizações que fornece diretrizes sobre as melhores práticas utilizadas. O CMMI tem como foco todo o processo de desenvolvimento de software e não se especializa em um domínio específico.

Assim como o CMMI, o TMMi (*Test Maturity Model integration*) é um modelo de maturidade especializado na melhoria do processo de teste de software. Um *framework* dividido em 5 níveis (Figura 2.3), que permitem a uma organização evoluir seu processo partindo de um ponto caótico, não gerenciado, para um ponto em que o processo é gerido, bem definido e otimizável. A divisão em níveis possibilita a introdução de mudanças de maneira incremental e facilita a avaliação do processo de uma organização.



Figura 2.3: Níveis de Maturidade TMMi (adaptada do trabalho de Camargo (2012))

Cada nível do TMMi (com exceção do primeiro) é composto por áreas de processo que agrupam diversas atividades relacionadas. Um determinado nível só é atingido quando são atendidas todas as suas áreas de processo e também as de seus níveis anteriores.

Uma área de processo possui objetivos específicos que indicam uma característica a ser implementada no processo. Estes objetivos específicos são divididos em práticas específicas que descrevem as atividades. Existem também objetivos genéricos, que podem estar relacionados a mais de uma área de processo, e são baseados em práticas genéricas, da mesma forma, relacionadas a mais de uma área de processo. A estrutura e organização dos componentes do TMMi

está ilustrada na Figura 2.4.

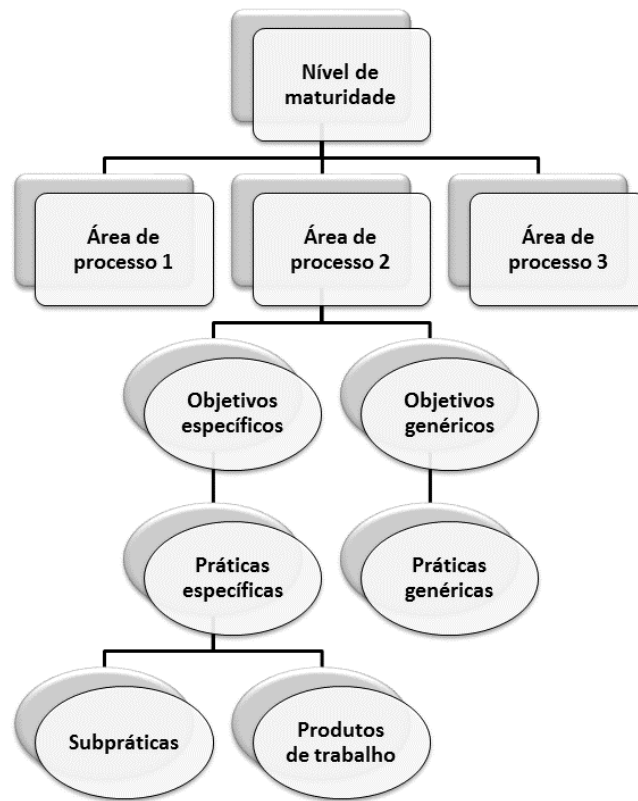


Figura 2.4: Estrutura dos Componentes do TMMi (adaptado de TMMi Foundation (2010))

Os níveis de maturidade do TMMi e suas respectivas áreas de processo são:

- **Nível 1 - Inicial:** Ambiente organizacional instável, sem definição de processos. Atividade de teste caótica que se resume a depuração de código. Não existe visibilidade sobre a qualidade e os riscos do software desenvolvido.
- **Nível 2 - Gerenciado:** Processo de teste torna-se independente e gerenciado, assegurando que as práticas existentes sejam mantidas. Atividades são monitoradas e controladas para evitar que desvios de planejamento ocorram. Áreas de processo deste nível:
 - Política e estratégia de teste
 - Planejamento de teste
 - Monitoramento e controle
 - Projeto e execução de teste
 - Ambiente de teste

- **Nível 3 - Definido:** O teste deixa de ser uma etapa que se segue ao código, revisões e treinamentos são realizados. Existe um processo padrão que é planejado desde o começo do projeto e integrado ao ciclo de vida de desenvolvimento. O escopo e rigor dos processos é maior que o aplicado no nível 2. Áreas de processo deste nível:
 - Organização do teste
 - Programa de treinamento
 - Ciclo de vida e integração
 - Teste não funcional
 - Revisão por pares

- **Nível 4 - Gerenciado Quantitativamente:** O teste é um processo inteiramente definido, fundamentado e mensurável. Todas as atividades do ciclo são submetidas a verificação. Métricas são armazenadas em repositórios auxiliando em tomadas de decisão e estimativas de custo e prazo. Áreas de processo deste nível:
 - Medição de teste
 - Avaliação de qualidade de software
 - Revisão por pares avançada

- **Nível 5 - Em Otimização:** Todo o processo é baseado na prevenção de defeitos, sendo controlado estatisticamente e previsível. Estabelecimento de uma equipe dedicada, responsável pelas constantes melhorias de processos. Áreas de processo deste nível:
 - Prevenção de defeitos
 - Otimização de processo de teste
 - Controle de qualidade

Apesar de ser largamente utilizado, o TMMi pode não refletir com realismo as práticas das organizações. Tendo em vista que o modelo retratado é muito extenso e fortemente dependente dos níveis existentes, existe dificuldade em sua adoção por empresas, especialmente as de pequeno porte, e são exigidos grandes esforços de adequação de processo (KASURINEN et al., 2011). Desta forma, existe a possibilidade de escolha de um processo de teste de software enxuto baseado no TMMi, como por exemplo o proposto por Camargo (2012). Mesmo sendo um processo enxuto (conta com 40% das práticas do modelo original), apresenta as 33 práticas mais relevantes para um processo de teste.

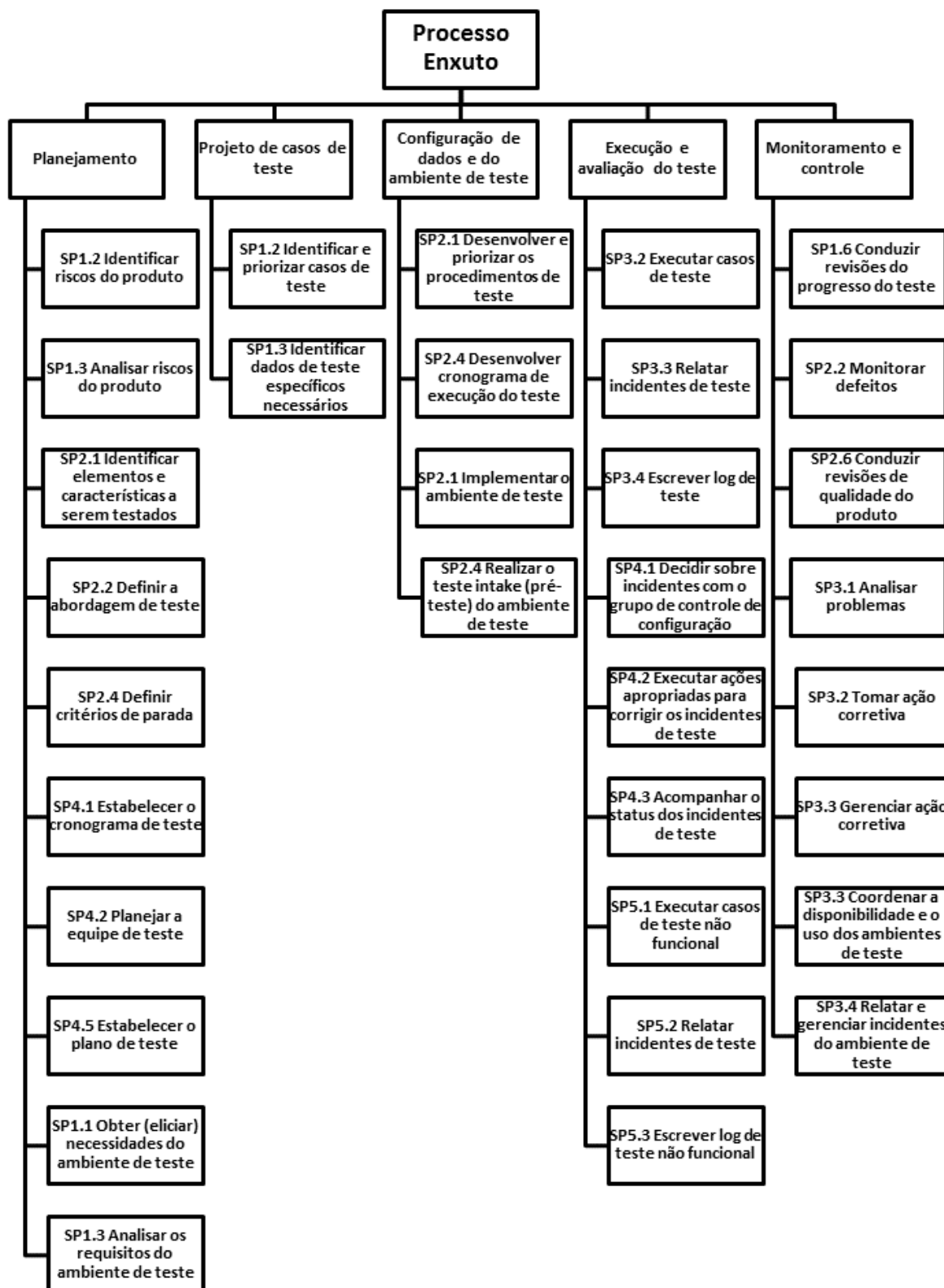


Figura 2.5: Processo de Teste de Software Enxuto Baseado no TMMi Obtido por Camargo (2012)

Na Figura 2.5 exibe-se o processo enxuto obtido por Camargo (2012) através de uma estrutura de árvore. Nesta árvore, os nós internos representam as fases do processo genérico de

teste, e as folhas representam as práticas selecionadas como relevantes. O modelo é de fácil compreensão e indica o que deve ser implementado prioritariamente.

É possível confrontar as práticas existentes no modelo do TMMi com as práticas existentes no processo enxuto de Camargo (2012) com base na Figura 2.6. Nesta figura, a fase do processo baseado no TMMi *Projeto de casos de teste* é representada pela árvore completa, na qual são detalhados as áreas de processo, objetivos específicos e práticas específicas. Em destaque estão as duas práticas específicas identificadas como relevantes contidas no processo enxuto.

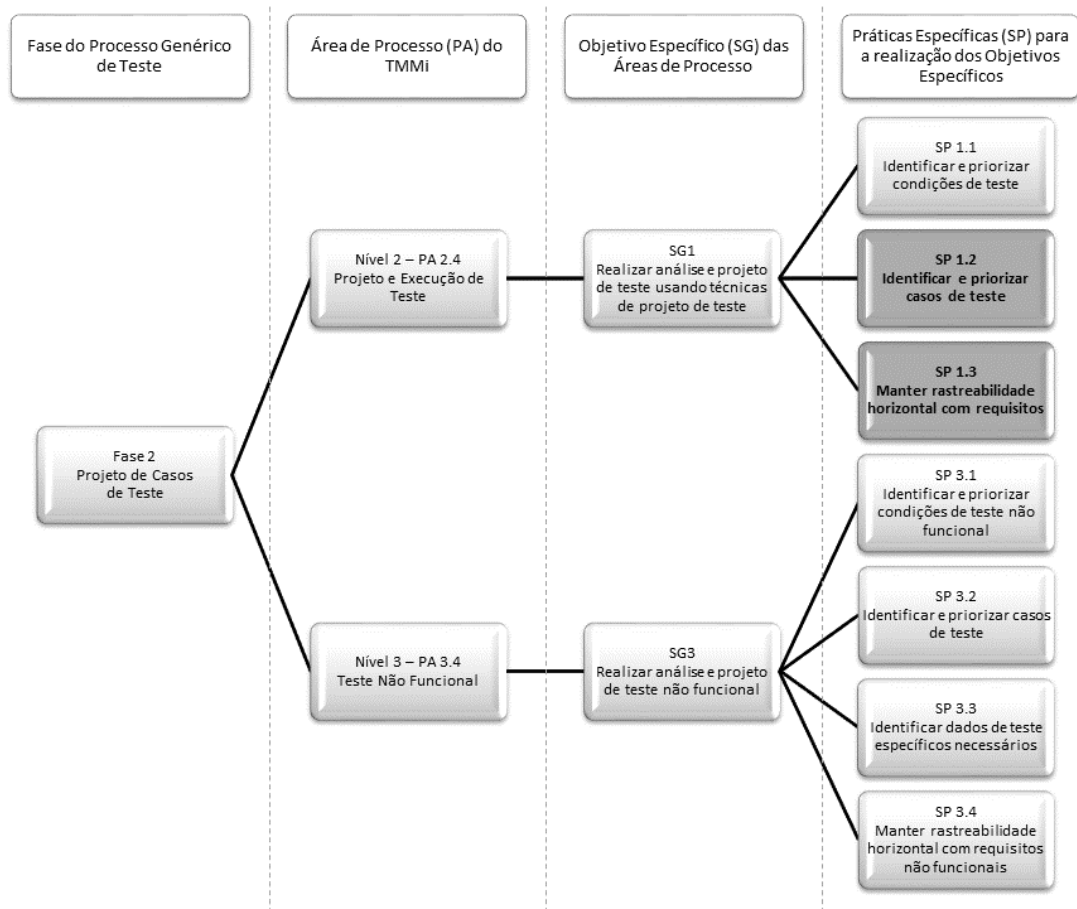


Figura 2.6: Em destaque na última coluna, as duas fases do Processo Enxuto Baseado no TMMi (Adaptado do Trabalho de Camargo (2012))

2.3.2 MPT.Br

Os modelos de maturidade auxiliam organizações a alcançarem melhores resultados através da evolução de seus processos. O MPT.Br (*Melhoria do Processo de Teste Brasileiro*) é um modelo nacional que foi concebido se tornar uma alternativa de baixo custo aos principais modelos internacionais com foco em pequenas e microempresas desenvolvedoras de software. É baseado em modelos de referência em teste de software como o TMMi, com diferencial por ser aderente

às metodologias ágeis.

O MPT.Br tem uma estrutura semelhante à do TMMi. O modelo é dividido em 5 níveis de maturidade (Figura 2.7) cada um composto por um conjunto de áreas de processo. As áreas de processo, por sua vez, são agrupamentos de práticas relacionadas que satisfazem um determinado objetivo. Práticas genéricas também fazem parte das áreas de processo. Assim como no TMMi, um nível é alcançado somente quando a organização atende à todas as suas áreas de processo e as de seus níveis anteriores.

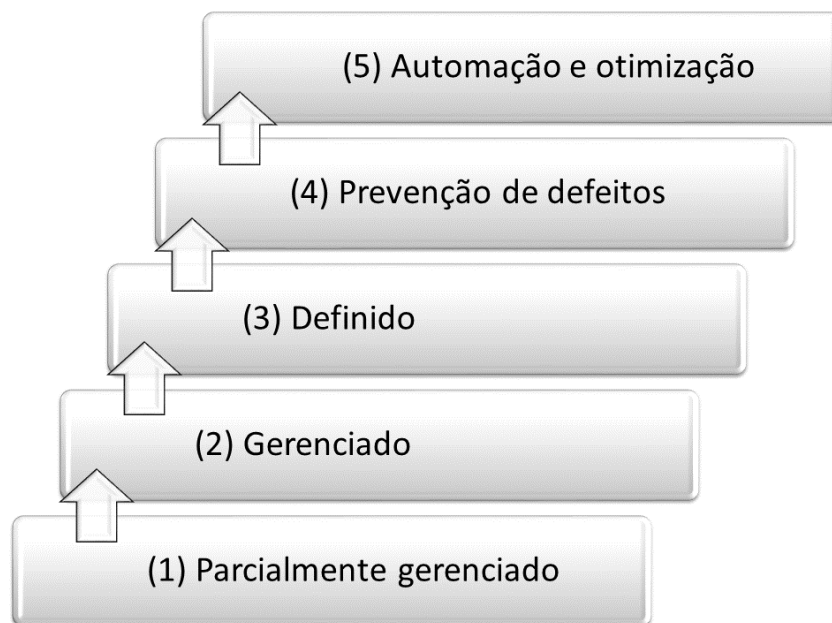


Figura 2.7: Níveis de Maturidade MPT.Br (Adaptado de MPT.BR (2011))

Os níveis de maturidade do MPT.Br e suas respectivas áreas de processo são:

- **Nível 1 - Parcialmente gerenciado:** Contempla o mínimo necessário para demonstrar que atividades de teste são aplicadas nos projetos de forma planejada e controlada. Áreas de processo deste nível:
 - Gestão do projeto de teste
 - Projeto e execução do teste
- **Nível 2 - Gerenciado:** Maior visibilidade do processo, que passa a ser monitorado e gerenciado. A gestão de mudanças passa a ser responsável pelo controle do escopo. Áreas de processo deste nível:
 - Gerência de requisitos de teste

- Evolução da gestão de projeto de teste
 - Evolução de projeto e execução do teste
- **Nível 3 - Definido:** É instituída a garantia de qualidade para auxiliar a definição dos processos, que passa a ser parte integrante do ciclo de vida do desenvolvimento. Fechamento do teste passa a ser aplicado de maneira sistemática, e novos tipos de teste são formalizados, o teste estático e o teste de aceitação. Papéis e responsabilidades são definidos e métricas são instituídas. Áreas de processo deste nível:
 - Fechamento do teste
 - Garantia da qualidade
 - Medição e análise de teste
 - Organização do teste
- **Nível 4 - Prevenção de defeitos:** Processo de gestão de defeitos e melhoria sistemática da qualidade do produto implantados, a fim de evitar que defeitos encontrados ocorram novamente pelas mesmas causas. Teste não-funcional passa a ser executado. A eficácia do teste e a qualidade do produto passam a ser analisados. Áreas de processo deste nível:
 - Avaliação da qualidade do produto
 - Gestão de defeitos
 - Teste não funcional
 - Evolução da organização do teste
- **Nível 5 - Automação e otimização:** Processo é controlado estatisticamente e está sob melhoria contínua. Automação da execução do teste e adoção de ferramentas *case* realizada de maneira sistemática. Áreas de processo deste nível:
 - Automação e execução do teste
 - Controle estatístico do processo
 - Gestão de ferramentas

Um dos principais diferenciais do MPT.Br é a aderência à metodologia ágil. Mesmo com a obrigatoriedade de processos, é possível para uma empresa que adota tal metodologia ser certificada, pois para a avaliação do MPT.Br as entrevistas e a observação da prática são mais

valiosas que a simples avaliação de evidências (FURTADO et al., 2012). Esta característica permite também a redução de custos e prazos com implementação, quando comparados a outros modelos de maturidade como, por exemplo, o TMMi.

A próxima seção introduz os conceitos de Gestão do Conhecimento, descreve os benefícios de sua utilização na área da Engenharia de Software e algumas de suas abordagens aplicadas em teste de software encontradas na literatura.

2.4 Conhecimento e sua Gestão

Devido à sua natureza, não existe um consenso para “o que é conhecimento”, sendo difícil determinar exatamente uma resposta para a questão, pois seu significado vai além da definição da palavra. Em uma definição filosófica, o conhecimento é tido como a “crença verdadeira justificada” (NONAKA; TAKEUCHI, 1995). Segundo o dicionário MICHAELIS (2014), o verbete “conhecimento” tem as seguintes definições: (1) Ato ou efeito de conhecer. (2) Faculdade de conhecer. (3) Ideia, noção; informação, notícia. (4) Consciência da própria existência. (5) Ligação entre pessoas que tem entre si algumas relações, menos estreitas que as de amizade. (6) Pessoa com quem se tem relações. (7) Saber, instrução, perícia; razoabilidade; circunspeção.

De acordo com estas definições, percebe-se uma íntima ligação entre conhecimento e informação. Tanto a informação quanto o conhecimento dependem do contexto, e são criados de maneira dinâmica no relacionamento interpessoal dos indivíduos.

Hierarquia do Conhecimento

O relacionamento entre informação e conhecimento foi extensamente estudado pelo americano R. L. Ackoff, que no ano de 1989 publicou em seu artigo intitulado *From data to wisdom* a proposta da hierarquia do conhecimento (Figura 2.8), mais conhecida como pirâmide *DIKW* (do inglês *Data, Information, Knowledge, Wisdom*). Ackoff (1989) propôs uma definição para cada nível da hierarquia, e a transformação que ocorre entre os níveis.

- **Dados** são símbolos que representam as propriedades dos objetos. São extraídos através da observação e não possuem utilidade até que sejam aplicados de maneira relevante.
- **Informação** contém descrições e significados relevantes, sendo inferida a partir dos dados.
- **Conhecimento** torna possível a transformação de informação em instruções. Pode ser



Figura 2.8: Hierarquia do Conhecimento

obtido através da transmissão por um detentor, seja por instrução ou por experiência.

- **Sabedoria** é a capacidade de aumentar a eficácia e agregar valor. Requer julgamento de quem a detém, e traz valores éticos e estéticos que são únicos e pessoais.

Na proposta clássica de Ackoff (1989) existe um nível intermediário entre conhecimento e sabedoria, a compreensão. Entretanto muitos autores consideram que a compreensão faz parte do processo da transformação que ocorre entre os níveis, ou seja, a compreensão dos dados gera informação, a compreensão da informação gera conhecimento, e a compreensão do conhecimento gera sabedoria. Sendo assim, a compreensão não é tratada separadamente como um nível na hierarquia (ROWLEY, 2007).

Dimensões do Conhecimento

O conhecimento pode somente ser criado por indivíduos, e existem maneiras diferentes de interação que ocorrem para promover a sua criação, a chamada dinâmica da criação do conhecimento. Tradicionalmente, no estudo do conhecimento, as duas dimensões existentes são conhecimento explícito e conhecimento tácito, que foram propostas inicialmente por Polanyi (1966). Nonaka e Takeuchi (1995) explicam as diferenças entre as dimensões do conhecimento:

- O conhecimento explícito pode ser transmitido facilmente através de linguagem formal entre indivíduos, ou seja, pode ser expresso em palavras e números de maneira sistemática.

- O conhecimento tácito é difícil de ser transmitido através de linguagem formal, pois é pessoal e está incorporado à experiência do indivíduo que o detém.

Em seu trabalho, Nonaka e Takeuchi (1995) defendem que o conhecimento humano é fruto da interação social entre as duas dimensões do conhecimento, tornando-as complementares. Essa interação é chamada de conversão do conhecimento, e explica a dinâmica da criação do conhecimento (Figura 2.9). São quatro os modos de conversão do conhecimento, os quais são discutidos a seguir, de acordo com a perspectiva apresentada em Nonaka e Takeuchi (1995).

- **De conhecimento tácito para conhecimento tácito - Socialização:** Baseado no compartilhamento de experiências. A partir do conhecimento tácito, de acordo com a experiência individual, a criação do conhecimento se dá através de modelos mentais ou habilidades técnicas compartilhadas. A troca de experiências é fundamental para que a aquisição do conhecimento ocorra.
- **De conhecimento tácito para conhecimento explícito - Externalização:** Tradução do conhecimento em conceitos explícitos, ou seja, através de metáforas, modelos, hipóteses e analogias o conhecimento tácito se torna articulável, provoca reflexões e se torna conhecimento explícito. A externalização é a principal forma de criação do conhecimento, pois a partir do conhecimento tácito é capaz de criar novos conceitos através do uso de metáforas e analogias. Desta forma, os conceitos explícitos criados podem ser modelados e expressos em linguagem sistemática de forma coerente.
- **De conhecimento explícito para conhecimento explícito - Combinação:** Em um sistema de conhecimento, indivíduos trocam conhecimento através de documentos, treinamentos, conversas ao telefone entre outros. Isso leva à combinação de diferentes conjuntos de conhecimento e à reconfiguração das informações existentes através da classificação e acréscimo do conhecimento explícito previamente existente, ou seja, levando a novos conhecimentos.
- **De conhecimento explícito para conhecimento tácito - Internalização:** Incorporação do conhecimento, ou "aprender fazendo". Trata-se de quando o conhecimento explícito é internalizado nas bases do conhecimento tácito dos indivíduos. Experiências obtidas através da socialização, externalização e combinação se tornam valiosas e são convertidas sob a forma de modelos mentais e *know-how* compartilhado. Neste modo de conversão, é importante que ocorra a verbalização e diagramação do conhecimento na forma de documentos, manuais ou histórias. Esta etapa auxilia na internalização das experiências

e também facilita a transferência do conhecimento explícito para outras pessoas, pois vivenciam indiretamente a experiência de outros indivíduos. É neste modo que se inicia uma nova espiral do conhecimento, ou seja, um novo ciclo de conversão do conhecimento.



Figura 2.9: Dinâmica da Criação do Conhecimento (Adaptado de Nonaka e Takeuchi (1995))

A criação do conhecimento ocorre pela interação contínua entre o conhecimento tácito e explícito, sendo um processo dinâmico moldado pelas diferenças entre os quatro modos de conversão do conhecimento.

2.4.1 Gestão do Conhecimento

Administrar o conhecimento e utilizá-lo como item estratégico para manter a competitividade tornou-se um dos principais objetivos das organizações, pois em tempos modernos, o maior valor é encontrado no intangível (TAKEUCHI; NONAKA, 2004). Para que esta seja uma estratégia de sucesso, as teorias de Gestão do Conhecimento são utilizadas.

A Gestão do Conhecimento é definida como um método que simplifica o processo de criação, compartilhamento, distribuição e entendimento do conhecimento de uma organização (DAVENPORT; PRUSAK; PRUSAK, 2000).

Técnicas de Gestão do Conhecimento

Em tempos remotos, o conhecimento de uma organização era armazenado no papel, ou, em não raros casos, nos próprios indivíduos. No entanto, a quantidade de informação e conhecimento que precisa ser armazenado cresceu drasticamente, além de haver alguns problemas fundamentais: (i) o acesso limitado que o papel oferece e seu difícil armazenamento; e (ii) o fato de que os indivíduos deixam a organização e carregam o conhecimento consigo (O'LEARY, 1998). Sendo assim, existe uma real importância na utilização de técnicas de Gestão do Conhecimento para aprimorar o acesso a informação e conhecimento de uma organização. Dentre essas técnicas, destacam-se:

- **Feedback:** Mecanismo que tem como objetivo criar conhecimento a partir da avaliação qualitativa ou quantitativa de um acontecimento/informação e, com base nesta avaliação, alterar e melhorar a cadeia de eventos na qual este acontecimento/informação foi originado (TIAN; NAKAMORI; WIERZBICKI, 2009; DINGSØYR; CONRADI, 2002).
- **Mapa de Conhecimento:** Permite a identificação dos funcionários de uma organização a partir de suas habilidades técnicas. É principalmente utilizado na alocação de recursos humanos e na busca de especialistas para a solução de problemas (HANSEN; NOHRIA; TIERNEY, 1999). Um exemplo de mapa de conhecimento são as *Páginas Amarelas*.
- **Repositório de Conhecimento:** Diferentemente de uma simples base de dados, os repositórios de conhecimento armazenam, além da informação em si, referências sobre o seu domínio (LIAO, 2003). Os repositórios de conhecimento facilitam o incremento de seu conteúdo e permitem a rastreabilidade de seu histórico (O'LEARY, 1998). Um exemplo de repositório de conhecimento são os *Documentos do Conhecimento*.
- **Repositório de Lições Aprendidas:** São bases utilizadas para armazenar informações de negócios e apoiam a operação de uma organização (O'LEARY, 1998). O conteúdo de uma lição aprendida pode ser definido como um tipo de conhecimento adquirido pela experiência (tanto positiva, quanto negativa) (SECCHI; CIASCHI; SPENCE, 1999).

2.4.2 Gestão do Conhecimento em Engenharia de Software

Nas organizações de software, os principais ativos não são máquinas ou prédios, mas sim o capital intelectual de seus funcionários. Como os recursos disponíveis não aumentam conforme crescem as necessidades de seus clientes, as organizações devem administrar o capital intelectual de maneira que a produtividade de seus funcionários aumente conforme a demanda.

O capital intelectual de uma organização consiste em ativos tangíveis e intangíveis. Os ativos *tangíveis* correspondem ao conhecimento explícito que normalmente inclui manuais, documentos, correspondências, patentes, licenças entre outros. Já os ativos *intangíveis* correspondem ao conhecimento tácito e não documentado, que inclui experiências, habilidades e conhecimento das pessoas da organização.

A atividade de desenvolvimento de software é dependente do conhecimento e da criatividade, e envolve diversas pessoas que trabalham em diferentes fases e tarefas, muitas vezes paralelamente. Este conhecimento é diversificado, de grandes proporções e em constante crescimento. Devido a estas características as organizações têm dificuldades em identificar o conteúdo e a localização do conhecimento. Melhorar a utilização do conhecimento e garantir que todos tenham acesso a informações corretas são as principais motivações da Gestão do Conhecimento na área da engenharia de software.

De acordo com Rus e Lindvall (2002), algumas das principais aplicações da Gestão do Conhecimento em engenharia de software são:

- **Diminuir o tempo e o custo do desenvolvimento melhorando a qualidade:** Evitar erros e reduzir retrabalho a fim de diminuir o tempo e o custo do desenvolvimento de projetos de software. Ao aplicar o conhecimento obtido em projetos de sucesso anteriores, as organizações garantem ganhos em produtividade, e maior probabilidade de sucesso nos projetos futuros.
- **Tomar decisões corretas:** Cada pessoa envolvida no processo de desenvolvimento de software precisa tomar decisões técnicas ou gerenciais, na maioria das vezes, baseado em sua própria experiência e conhecimento, ou através de contatos pessoais informais. Para que funcionários em toda a organização tomem decisões corretas, é necessário a definição de processos para o compartilhamento de conhecimento.
- **Compartilhar conhecimento sobre políticas e práticas locais:** Cada organização possui suas próprias políticas, práticas e cultura, que são relevantes em atividades técnicas, gerenciais e administrativas. Entretanto, este conhecimento normalmente existe como folclore organizacional, sendo transmitido de maneira informal e *ad-hoc*. Ou seja, nem todos têm acesso ao conhecimento necessário para a realização de suas atividades. Mesmo o compartilhamento informal sendo relevante para a organização, o compartilhamento formal do conhecimento é primordial, e garante que todos poderão acessá-lo.
- **Capturar o conhecimento sobre quem sabe o quê:** Os funcionários experientes são considerados a chave para o sucesso dos projetos, deixando as organizações dependentes

de seu conhecimento. No entanto, o acesso a estas pessoas pode ser difícil, e existe o risco de serem criadas graves lacunas de conhecimento ao deixarem a organização, e neste caso provavelmente ninguém estará ciente do conhecimento que foi perdido. Saber quais funcionários sabem é necessário para que uma estratégia de prevenção de perda do conhecimento valioso seja criada. Também é relevante para que equipes equilibradas sejam criadas para os projetos.

- **Colaboração e compartilhamento de conhecimento:** O desenvolvimento de software é uma atividade realizada em grupo, que muitas vezes se encontra geograficamente disperso, e trabalha em fusos-horários diferentes. Este grupo precisa se comunicar, colaborar e coordenar. A comunicação muitas vezes está relacionada com a transferência do conhecimento. A colaboração está relacionada com o compartilhamento mútuo do conhecimento.

2.4.3 Gestão do Conhecimento em Teste de Software

Os trabalhos de Rus e Lindvall (2002) e Bjørnson e Dingsøyr (2008) evidenciam uma diversidade de estudos e aplicações de Gestão do Conhecimento em Engenharia de Software, as quais no geral podem ser igualmente utilizadas na área de teste, pois tais atividades, assim como as de desenvolvimento, são dependentes do conhecimento daqueles que as executam, e apresentam problemas que podem ser solucionados de maneira semelhante (ANDRADE et al., 2013).

Entretanto, para entender as particularidades da área de teste de software com relação à Gestão do Conhecimento, no trabalho descrito nesta dissertação realizou-se uma pesquisa exploratória com a intenção de identificar quais técnicas de Gestão do Conhecimento são aplicadas nas atividades de teste.

O protocolo de busca utilizado para esta pesquisa considerou termos relacionados ao assunto, e restringiu aos trabalhos publicados na área da Computação. A *string* de busca utilizada é apresentada a seguir, aplicada na máquina de busca *Scopus*¹², limitada ao título, resumo e palavras-chaves dos estudos primários pesquisados.

(‘‘knowledge management’’) E (‘‘software testing’’ OU ‘‘testing process’’)

¹Base de dados de resumos e citações de literatura científica nas áreas de ciência, tecnologia, medicina, ciências sociais, artes e humanidades.

²<http://www.scopus.com> – Acessado em 10/04/2015.

A busca foi realizada no período de Fevereiro e Março de 2014 e retornou 65 trabalhos, os quais foram analisados através da leitura dos títulos e *abstracts*, sendo selecionados apenas 11 devido à sua relevância, ou seja, por abordarem o tema de Gestão do Conhecimento em teste de software. Trabalhos duplicados foram removidos, restando apenas as versões mais recentes. Outros 3 trabalhos foram incluídos por serem relacionados à Gestão do Conhecimento em Engenharia de Software e apresentarem conceitos importantes.

Os trabalhos selecionados foram analisados individualmente e as seguintes informações foram extraídas:

- Título do artigo
- Autores do artigo
- Periódico ou Congresso em que foi publicado
- Ano de publicação
- O trabalho descreve um *framework*?
- O trabalho é baseado em ferramenta automatizada? Caso positivo, qual?
- O trabalho descreve aplicação em estudo de caso? Quantas vezes foi aplicado?

A Tabela 2.1 contém as informações principais, como título e autores. As informações complementares são apresentadas na Tabela 2.2.

Os 11 primeiros trabalhos foram os selecionados através da busca na base *Scopus*. Os trabalhos #12, #13 e #14 apresentam conceitos relevantes sobre Gestão do Conhecimento em Engenharia de Software. Percebe-se que os meios de publicação são heterogêneos, não havendo concentração de publicações em um local específico. Dos 11 trabalhos sobre teste de software, 6 foram publicados entre 2010 e 2014. Não foram selecionadas publicações anteriores a 2007. Os trabalhos sobre Engenharia de Software foram publicados entre 2008 e 2012.

Dos trabalhos selecionados, 11 descrevem um *framework* para aplicação de técnicas de Gestão do Conhecimento em engenharia de software/teste de software. Destes, 4 implementam uma ferramenta automatizada para utilização do *framework* descrito. Nem todos os trabalhos que descrevem um *framework*, ou uma ferramenta, realizaram estudo de caso para sua avaliação.

Tabela 2.1: Trabalhos Selecionados - Informações Principais

#	Título do Artigo	Autores	Periódico ou Congresso	Ano
1	SABUMO-dTest: Design and evaluation of an intelligent collaborative distributed testing framework	Colomo-Palacios, R., López-Cuadrado, J.L., González-Carrasco, I., García-Peñalvo, F.J.	Computer Science and Information Systems	2014
2	Using ontology patterns for building a reference software testing ontology	Souza, E.F., Falbo, R.A., Vijaykumar, N.L.	Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC	2013
3	An architectural model for software testing lesson learned systems	Andrade, J., Ares, J., Martínez, M.-A., (...), Romera, J., Suárez, S.	Information and Software Technology	2013
4	A model of knowledge management system in managing knowledge of software testing environment	Abdullah, R., Eri, Z.D., Talib, A.M.	2011 5th Malaysian Conference in Software Engineering, MySEC 2011	2011
5	Knowledge management and software testing	Desai, A., Shah, S.	International Conference and Workshop on Emerging Trends in Technology 2011, ICWET 2011 - Conference Proceedings	2011
6	The PDCA-based software testing improvement framework	Li, X.-X., Zhang, W.-N.	2010 International Conference on Apperceiving Computing and Intelligence Analysis, ICACIA 2010 - Proceeding	2010
7	Research and implementation of knowledge management methods in software testing process	Liu, X.-M., Gu, G., Liu, Y.-P., Wu, J.	2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009	2009
8	Investigating the relationship between schedules and knowledge transfer in software testing	Karhu, K., Taipale, O., Smolander, K.	Information and Software Technology	2009
9	Documentation accountability: Approval to archiving	Gordon, D.L., Heaton, J.	AUTOTESTCON (Proceedings)	2008
10	A Standard Process for Data Mining based Software Debugging	Kou, G., Peng, Y.	Proceedings - 4th International Conference on Networked Computing and Advanced Information Management, NCM 2008	2008
11	The testing process - A decision based approach	Borner, L., Illes-Seifert, T., Paech, B.	2nd International Conference on Software Engineering Advances - ICSEA 2007	2007
12	Managing tacit knowledge for a software development process: A case study	Stevens, D.P., Hsu, S.H.Y., Zhu, Z.	Journal of Information and Knowledge Management	2012
13	Knowledge transfer in offshore outsourcing software development projects: An analysis of the challenges and solutions from German clients	Betz, S., Oberweis, A., Stephan, R.	Expert Systems	2012
14	Implementation specification for software process improvement supportive knowledge management tool	Alagarsamy, K., Justus, S., Iyakutti, K.	IET Software	2008

2.4.4 Abordagens de Gestão do Conhecimento aplicadas em Teste de Software Elegíveis

Para compor a próxima etapa deste trabalho, optou-se por escolher, dentre os trabalhos selecionados na pesquisa apresentada anteriormente, aqueles que descrevem um *framework* au-

Tabela 2.2: Trabalhos Selecionados - Informações Complementares

#	Descreve um framework?	Baseado em ferramenta? Qual?	Apresenta um estudo de caso? Quantas vezes foi aplicado?
1	Sim	SABUMO-dTest	Aplicado 4 vezes
2	Sim	-	Aplicado 1 vez
3	Sim	Protótipo	Aplicado 2 vezes
4	Sim	-	-
5	-	-	Aplicado 1 vez
6	Sim	-	Aplicado 3 vezes
7	Sim	Protótipo	-
8	-	-	Aplicado 5 vezes
9	Sim	-	-
10	Sim	-	-
11	Sim	-	Aplicado 4 vezes
12	-	-	Aplicado 1 vez
13	Sim	-	Aplicado 2 vezes
14	Sim	Protótipo	Aplicado 4 vezes

tomatizado, ou seja os quais um protótipo foi implementado. Além desta característica, devido ao contexto no qual esta dissertação está inserida, houve a necessidade de filtrar os trabalhos relacionados a teste de software. Os trabalhos elegíveis descrevem um *framework* de Gestão do Conhecimento que têm como objetivo principal atuar na disseminação do conhecimento dentro de uma equipe de testes. São eles:

1. *SABUMO-dTest: Design and Evaluation of an Intelligent Collaborative Distributed Testing Framework*, Colomo-Palacios et al. (2014)
3. *An Architectural Model for Software Testing Lesson Learned Systems*, Andrade et al. (2013)
7. *Research and Implementation of Knowledge Management Methods in Software Testing Process*, Xue-Mei et al. (2009)

2.5 Considerações Finais

Este capítulo inicialmente abordou o teste de software e seus principais conceitos. A partir de uma breve descrição sobre o objetivo fundamental da atividade de teste, que é revelar os

defeitos introduzidos durante o desenvolvimento, o texto abordou as técnicas que são utilizadas nas diversas fases de teste. A apresentação destes conceitos foi importante para o entendimento das seções posteriores, que trataram as características de um processo de teste de software. Para complementar o assunto foram discutidos dois modelos de maturidade (um internacional e um nacional) que podem ser adotados por uma organização com o intuito de melhorar o seu processo de teste.

Mais adiante foi exposta uma visão geral sobre Gestão do Conhecimento, abordando a hierarquia e dimensões do conhecimento existentes, além das teorias modernas sobre sua criação e compartilhamento. Esta introdução serviu como base para o entendimento dos conceitos básicos de Gestão do Conhecimento, que foram apresentados em seguida. Para finalizar, foi apresentada uma pesquisa sobre os trabalhos de Gestão do Conhecimento na área de teste de software, e a partir desta, foram selecionados três trabalhos para posterior análise.

No próximo capítulo estes três trabalhos serão descritos, e para cada um será elaborado um processo de Gestão do Conhecimento com aplicação em teste de software. Uma Análise Crítica destes três processos de Gestão do Conhecimento também irá compor o próximo capítulo.

Capítulo 3

ANÁLISE CRÍTICA DOS PROCESSOS DE GESTÃO DO CONHECIMENTO

3.1 Considerações Iniciais

Para suportar altas demandas de projetos e prazos por vezes agressivos, é necessária uma boa gestão da equipe de teste de software. Planejamento, estratégia, organização e controle são competências que um gestor deve adotar. Contudo, nem sempre tais competências serão a essência de um projeto de sucesso. Como visto no Capítulo 2, gerir conhecimento é um fator relevante que deve ser utilizado para aumentar a produtividade de equipes, através da geração e disseminação de conhecimento.

A fim de investigar quão relevante é a adoção de práticas de Gestão do Conhecimento em equipes de teste de software, uma Análise Crítica foi realizada, na qual três abordagens de Gestão do Conhecimento foram averiguadas. O objetivo desta análise foi identificar o grau de aplicabilidade de tais abordagens em um processo de teste, que contempla práticas essenciais dessa atividade.

As três abordagens selecionadas foram brevemente introduzidas na Seção 2.4.4, e o processo de teste utilizado foi o relatado na Seção 2.3.1. A análise teve por base a experiência na indústria da autora desta dissertação, que durante seis anos ocupou diversos cargos em uma equipe de teste de software de uma consultoria multinacional, iniciando como estagiária, chegando a função de coordenadora.

É importante ressaltar que o termo *processo* refere-se aos processos de Gestão do Conhecimento derivados dos *frameworks* selecionados neste trabalho.

3.2 Padronização da Representação dos Processos de Gestão do Conhecimento

A realização da Análise Crítica dos trabalhos selecionados exigiu a padronização da notação que os representa, uma vez que cada *framework* foi retratado de maneira diferente por seus autores. Para isto, as principais características apresentadas em cada trabalho foram identificadas e associadas aos fluxogramas correspondentes. A partir da coleta das informações necessárias, os *frameworks* foram interpretados e seus processos subjacentes propostos com a utilização da notação BPMN (OMG, 2014). O resultado desta padronização pode ser encontrado mais adiante, nas Figuras 3.2, 3.3 e 3.4.

Os autores dos referidos trabalhos foram contatados via e-mail, para que se verificasse se as representações dos *processos* elaborados a partir de seus *frameworks* estavam corretas. Esta etapa de verificação seria importante para confirmar se nenhum detalhe dos respectivos trabalhos teriam sido omitidos. Apenas o autor responsável pelo *framework* A (COLOMO-PALACIOS et al., 2014) retornou o contato. Em sua resposta, o autor indicou que o fluxograma BPMN estava de acordo com o *framework* proposto. Para a continuação da pesquisa, foi necessário assumir que os outros dois *processos* estavam representados corretamente.

A notação BPMN

A *Business Process Model and Notation* (BPMN) (OMG, 2014) do *Object Management Group* (OMG)¹ é uma notação que tem como objetivo principal representar um processo de maneira padronizada e facilmente compreensível por todos os envolvidos. A versão utilizada neste trabalho é a BPMN 2.0.

Para compor os fluxogramas foram selecionados os elementos mais básicos da notação BPMN: Armazenamento de Dados, Atividade, Eventos de Início e Fim, Fluxo de Sequência, Pontos de Decisão ou Junção, e Raia. A seguir, apresenta-se uma sucinta explicação sobre estes elementos, de acordo com o documento oficial BPMN (OMG, 2014). A representação gráfica dos elementos pode ser encontrada na Figura 3.1.

- **Armazenamento de Dados:** Mecanismo que representa o armazenamento de informação em uma atividade. Ele não representa uma ação de um ator específico, nem uma atividade em si, mas sim uma fluxo de informação que possui influência entre um grupo de atividades as quais é relacionado.

¹<http://www.omg.org/> – Acessado em 10/04/2015.

- **Atividade:** Termo genérico para qualquer ação realizada.
- **Evento:** Algo que acontece ao longo de um processo. Os eventos normalmente são um gatilho, ou possuem um resultado. Os Eventos utilizados neste trabalho são os de Início e Fim.
- **Fluxo de Sequência:** Indica a ordem em que as atividades serão executadas.
- **Ponto de Decisão ou Junção:** Utilizado para controlar a divergência ou convergência de um fluxo. Os Pontos de Decisão ou Junção utilizados neste trabalho são os Paralelos (decisão ou junção ocorrem paralelamente) e os Exclusivos (decisão ou junção são mutuamente exclusivos).
- **Raia:** Representa um ator no fluxograma. Todas as ações realizadas pelo ator devem ser dispostas dentro de sua Raia.

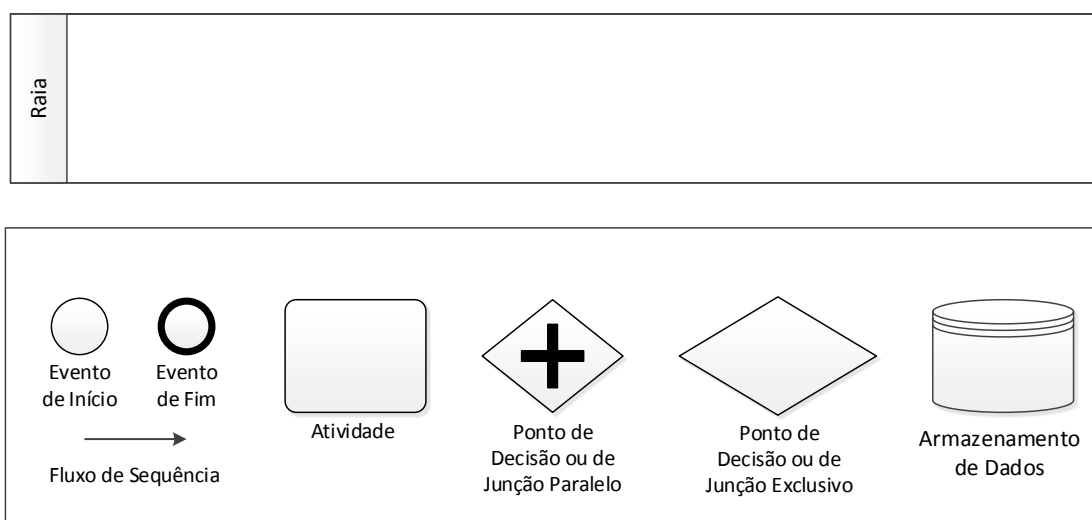


Figura 3.1: Representação Gráfica dos Elementos BPMN, de Acordo com o Documento Oficial BPMN (OMG, 2014)

3.2.1 *Processo A* (COLOMO-PALACIOS et al., 2014)

Desenvolvido por Colomo-Palacios et al. (2014), este *framework* é fundamentalmente baseado em retroalimentação (do inglês, *feedback*). O *processo* de Gestão do Conhecimento elaborado a partir deste *framework* será tratado como *Processo A*.

Observa-se que quase todas as ações realizadas possuem sua etapa de avaliação, o que possibilita a melhoria constante do conteúdo utilizado. Existem dois papéis bem definidos: *Contractor* (Gerente) e *Tester*, e cada um possui responsabilidades diferentes dentro do fluxo.

O *processo* se inicia com a definição das informações. O *Tester* preencherá o seu perfil (dados pessoais, contato, etc), e definirá suas competências (experiências, nível de conhecimento, entre outras informações relevantes para a realização de seu trabalho) - atividades 8 e 9, na Figura 3.2. O *Contractor* incluirá informações pertinentes aos planos de teste (chamados de “processos” neste *Processo A*), e aos itens pertencentes a este: ações (casos de teste) e competências (experiências requeridas para o teste) - atividades 2, 3 e 4.

Com o preenchimento das informações iniciais, o *Contractor* estará apto a buscar por um *Tester* que possua as competências necessárias para a execução de um determinado projeto - atividade 5. Uma vez que o *Tester* finalizou uma execução, ele irá avaliar o processo e seus itens (possibilitando que haja melhoria contínua do conteúdo) - atividades 7 e 11. O *Contractor* também avaliará a *performance* do *Tester* ao longo da execução (esta avaliação auxiliará uma decisão futura sobre a escolha do *Tester* em um novo processo) - atividade 10.

Com relação à teoria de Gestão do Conhecimento, esse *processo* implementa as seguintes técnicas: Páginas Amarelas (informações sobre o perfil do *Tester* e suas experiências), e *Feedback* (possibilidade de avaliar as ações executadas).

A representação BPMN do *Processo A* pode ser visualizada na Figura 3.2. Cada elemento apresentado no fluxograma foi enumerado, sendo descritos na lista a seguir:

1. **Escolher Atividade:** Escolha da atividade que será realizada (definir / alterar ações, competências e processo; ou selecionar perfil).
2. **Definir / Alterar Ações:** Especificação, realizada pelo *Contractor*, dos casos de teste que serão executados pelo *Tester* em um processo.
3. **Definir / Alterar Competências:** Especificação das competências (experiências) necessárias em um *Tester* para a realização de um processo.
4. **Definir / Alterar Processo:** Especificação, realizada pelo *Contractor*, das atividades de teste (plano de teste) que serão realizadas pelo *Tester*.
5. **Selecionar Perfil:** Escolha, realizada pelo *Contractor*, de qual *Tester* possui as competências necessárias para a execução do processo.

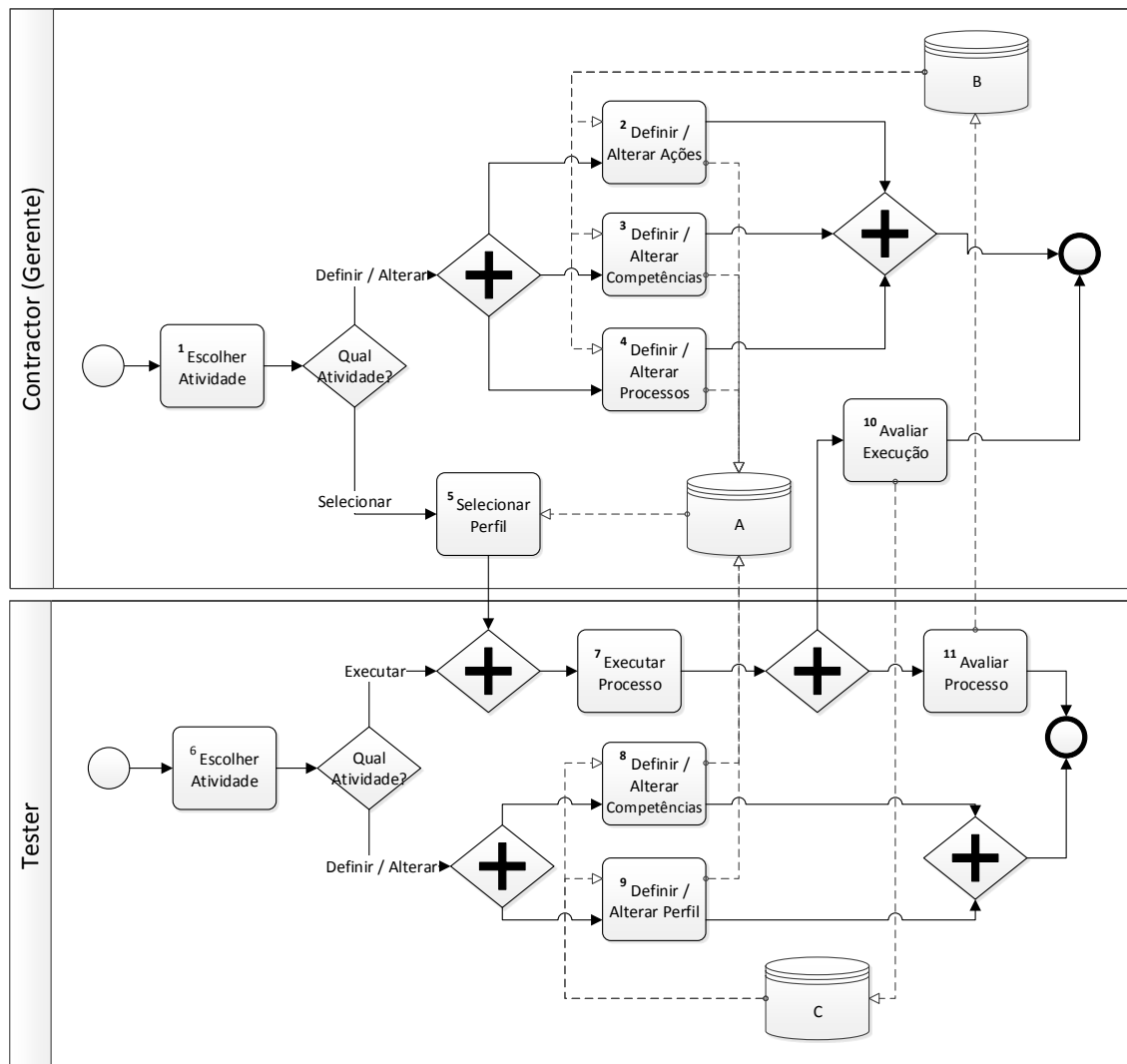


Figura 3.2: Fluxograma *Processo A*

6. **Escolher Atividade:** Escolha da atividade que será realizada (definir / alterar competências e perfil; ou executar processo).
7. **Executar Processo:** Após ser selecionado, o *Tester* seguirá instruções contidas no processo para a realização do teste.
8. **Definir / Alterar Competências:** Preenchimento das competências (experiências) que um *Tester* possui.
9. **Definir / Alterar Perfil:** Preenchimento dos dados pessoais do *Tester*.
10. **Avaliar Execução:** Avaliação, realizada pelo *Contractor*, da performance do *Tester* pela execução do processo.

11. **Avaliar Processo:** Avaliação, realizada pelo *Tester*, das instruções e ações contidas no processo.

Descrição dos elementos de Repositório de Dados:

A **Armazenamento das Definições:** Todas as informações definidas ao longo do processo são armazenadas em banco de dados para serem utilizadas na atividade 5. *Selecionar Perfil*.

B **Armazenamento da Avaliação do Processo:** As informações constatadas pelo *Tester* na atividade 11. *Avaliar Processo*, devem ser armazenadas em banco de dados para permitir a melhoria das descrições das ações, competências e processos.

C **Armazenamento da Avaliação do Processo:** O *Contractor* avaliará o desempenho do *Tester* após a execução do processo (atividade 10. *Avaliar Execução*). Essas informações serão utilizadas no aprimoramento das competências e perfil do *Tester*.

3.2.2 Processo B (ANDRADE et al., 2013)

Este *framework*, desenvolvido por Andrade et al. (2013), tem como base documentos chamados lições aprendidas (do inglês, *lessons learned*). O processo de Gestão do Conhecimento elaborado a partir deste *framework* será tratado como *Processo B*.

Uma *lesson learned* é um conjunto de informações sobre resultados de projetos. Estes resultados podem ser parciais ou finais, e o objetivo de uma *lesson learned* é servir de apoio para tomada de decisão em projetos futuros. As *lessons learned* podem servir como exemplos de o que se deve fazer (em casos de projetos de sucesso), ou de o que não se deve fazer (em casos de projetos fracassados).

No processo também são armazenadas informações sobre projetos, processos de teste, e sobre as equipes de teste - atividades 5, 6, 7 e 8 na Figura 3.3. Tais informações são centralizadas no chamado Mapa do Projeto - atividade 9. Este mapa deve ser consultado sempre que uma informação precisa ser recuperada - atividade 3. As *lessons learned* também são acessadas a partir do Mapa do Projeto - atividade 4.

Não existe distinção de papéis dentro do processo. Todos da equipe de teste podem, a qualquer momento, consultar e alterar as informações contidas no Mapa do Projeto.

A cada alteração ou inclusão no processo o sistema atualiza automaticamente o Mapa do Projeto, de maneira que as informações acessadas serão sempre as mais atuais - atividade 9. O sistema também exerce um papel dinâmico quanto à disseminação das *lessons learned* entre os

envolvidos em um projeto. Através dos cadastros dos projetos, as *lessons learned* relacionadas são identificadas e enviadas pelo sistema automaticamente aos Analistas de Teste envolvidos - atividades 10 e 11. Esta ação possibilita que o Analista de Teste possua a informação antes mesmo de ela ser necessária.

Com relação à teoria da Gestão do Conhecimento, esse *processo* implementa as seguintes técnicas: *Lessons Learned*, Páginas Amarelas (Mapa do Projeto) e *Feedback* (inclusão de novas informações nas *Lessons Learned*).

A representação BPMN do *Processo B* pode ser visualizada na Figura 3.3, sendo que cada elemento enumerado na figura é descrito a seguir:

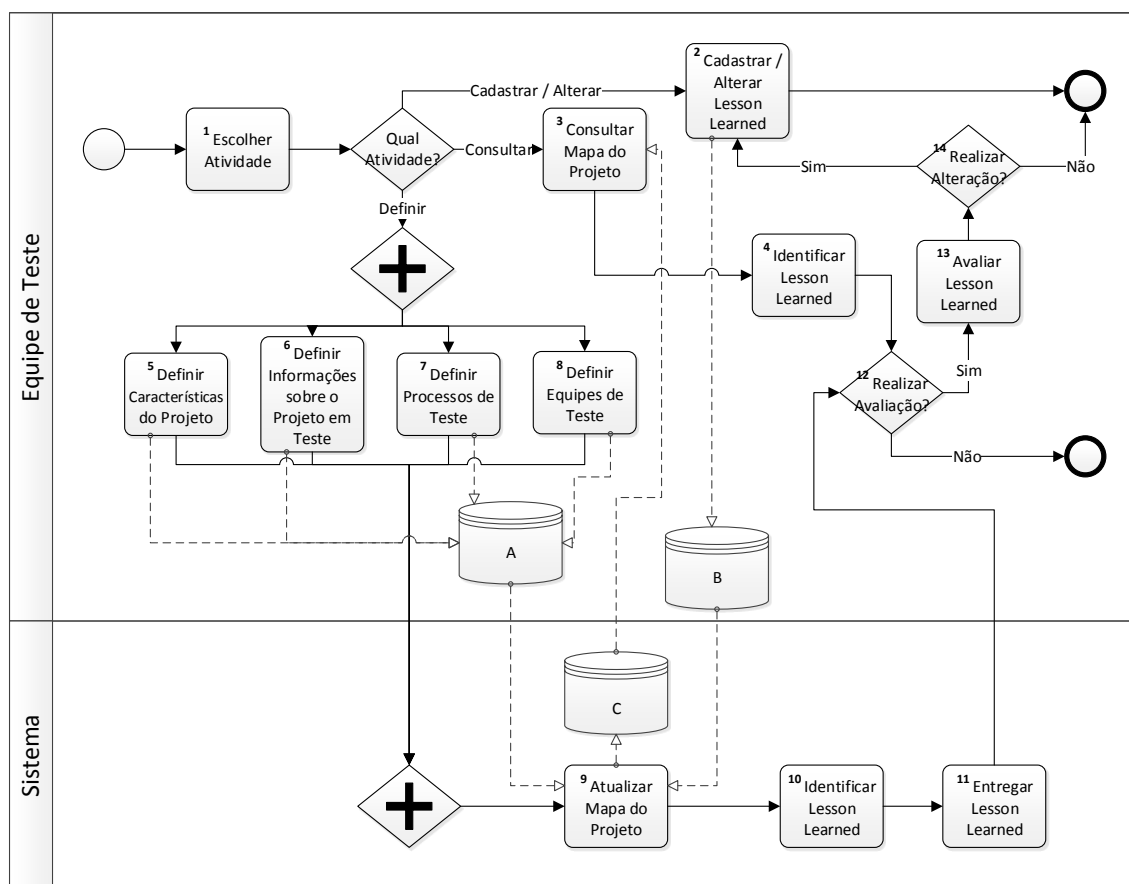


Figura 3.3: Fluxograma *Processo B*

1. **Escolher Atividade:** Escolha da atividade que será realizada (cadastrar / alterar *lesson learned*; ou consultar mapa do projeto; ou definir características, informações, processos e equipes).
2. **Cadastrar / Alterar *Lesson Learned*:** Preenchimento inicial (ou alteração) dos dados de

uma *lesson learned*.

3. **Consultar Mapa do Projeto:** Consulta às informações e detalhes sobre o projeto, para que seja possível definir quais são as *lessons learned* relacionadas.
4. **Identificar *Lesson Learned*:** Identificação e seleção de *lessons learned* que estão relacionadas ao projeto que será testado.
5. **Definir Características do Projeto:** Especificação das características e detalhes sobre o projeto.
6. **Definir Informações sobre o Projeto em Teste:** Especificação das informações do projeto que será testado.
7. **Definir Processos de Teste:** Especificação das atividades que devem ser realizadas durante o projeto que será testado.
8. **Definir Equipes de Teste:** Escolha dos profissionais que irão compor a equipe de teste.
9. **Atualizar Mapa do Projeto:** Atualização automática, realizada pelo sistema, de informações e detalhes do projeto.
10. **Identificar *Lesson Learned*:** Seleção automática, realizada pelo sistema, de *lessons learned* relacionadas ao projeto.
11. **Entregar *Lesson Learned*:** Entrega automática, realizada pelo sistema, de *lessons learned* selecionadas.
12. **Realizar Avaliação:** Verificação da necessidade de avaliar uma *lesson learned*.
13. **Avaliar *Lesson Learned*:** Avaliação do conteúdo (conteúdo relevante) de uma *lesson learned*.
14. **Realizar Alteração:** Verificação da necessidade de alterar uma *lesson learned*.

Descrição dos elementos de Repositório de Dados:

- A **Armazenamento das Definições:** Todas as informações definidas ao longo do processo são armazenadas em banco de dados para serem utilizadas na atividade 9. *Atualizar Mapa do Projeto*.

B Armazenamento de uma *Lesson Learned*: As informações contidas em uma *Lesson Learned* são armazenadas em base de dados e também são utilizadas na atividade 9. *Atualizar Mapa do Projeto*.

C Armazenamento da Avaliação do Processo: As informações recuperadas na atividade 3. *Consultar Mapa do Projeto* refletirão a atualização em base de dados do Mapa do Projeto.

3.2.3 Processo C (XUE-MEI et al., 2009)

Os elementos principais deste *framework*, desenvolvido por Xue-Mei et al. (2009), são os documentos de conhecimento. O *processo* de Gestão do Conhecimento elaborado a partir deste *framework* será tratado como *Processo C*.

Estes documentos armazenam toda informação que possa ser relevante para a atividade de teste de software. Neste *processo* existe distinção de papéis e responsabilidades: Gerente, *Tester* (Analista de Teste), e Analista de Conhecimento.

As primeiras informações populadas são as informações pessoais dos *Testers* (dados pessoais, contato, nível de experiência, etc) e as informações dos projetos (fornecidas pelos Gerentes) - atividades 2 e 10 da Figura 3.4. O Gerente é o responsável por montar as equipes de teste com base nas informações pessoais preenchidas pelos *Testers* - atividades 3 e 4. O papel do Gerente no *processo* se encerra com a definição das equipes. A partir deste ponto, toda interação será realizada entre o *Tester* e o Analista de Conhecimento.

A criação dos documentos de conhecimento é uma atividade vinculada aos *Testers*. Eles podem alterar ou avaliar informações contidas nos documentos - atividade 11. Caso necessário, os *Testers* também buscam informações pessoais de colegas, como por exemplo um contato de um especialista em determinado assunto - atividade 9.

Toda informação fornecida por um *Tester* (desde informações pessoais, a Documentos de Conhecimento) é avaliada pelo Analista de Conhecimento - atividades 12 e 15. É durante esta etapa de avaliação que será determinado se o conteúdo está correto, e se é relevante. Somente após esta verificação que tais informações são carregadas na base de dados. O Analista de Conhecimento funciona como um filtro, e seu papel é importante, devido a liberdade de temas que são aceitos nos documentos de conhecimento.

Os *Testers* também podem consultar documentos existentes em busca de informações que auxiliem a realização de suas atividades - atividade 6. E caso necessário, podem avaliá-los (ou incrementá-los), melhorando a qualidade da informação contida - atividades 8 e 11.

Com relação à teoria de Gestão do Conhecimento, esse *processo* implementa as seguintes técnicas: Documentos de Conhecimento, Páginas Amarelas (consulta de informações), e *Feedback* (Analista de Conhecimento, principalmente).

A representação BPMN do *processo* pode ser visualizada na Figura 3.4, e suas descrições são apresentadas como se segue.

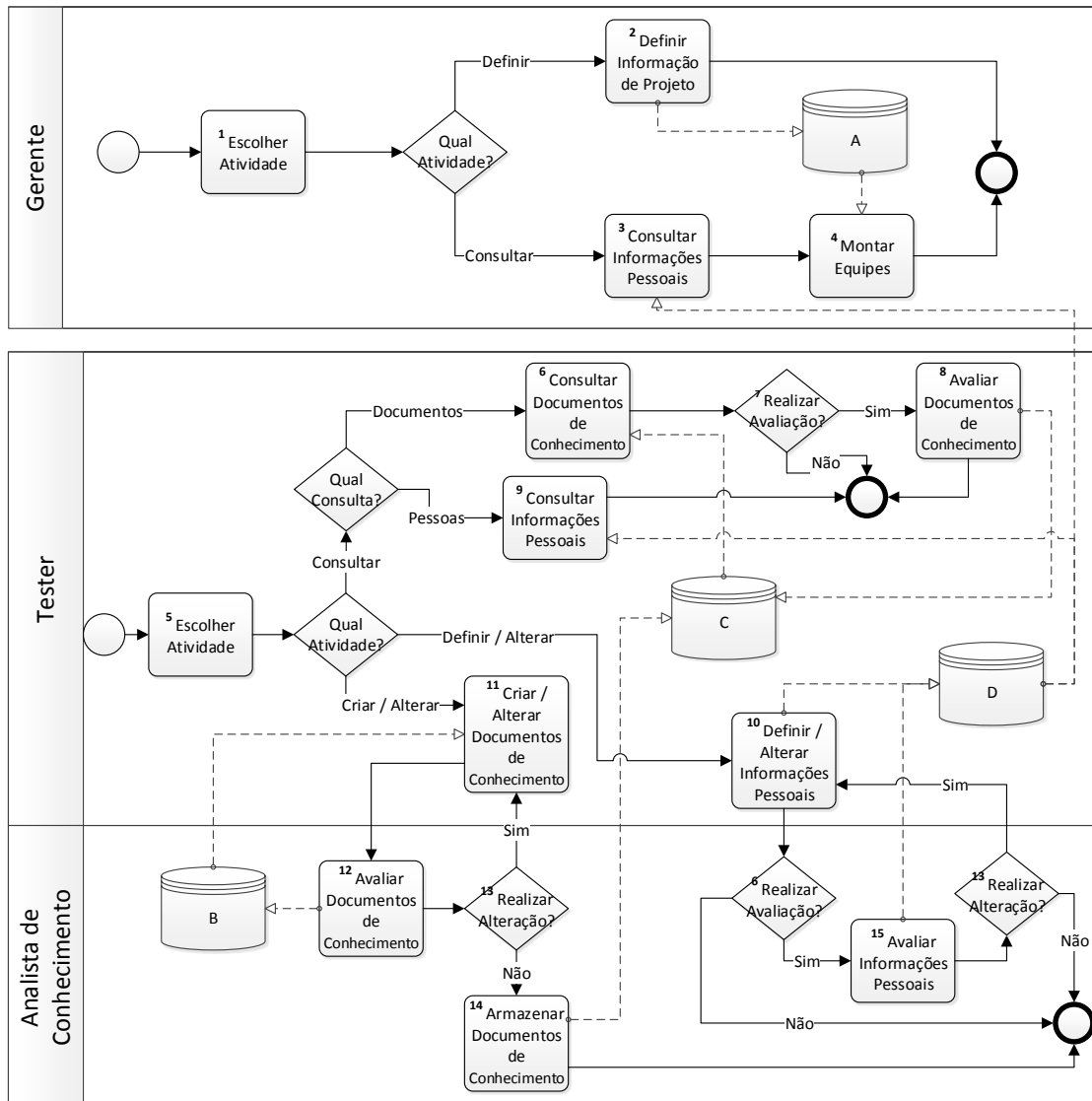


Figura 3.4: Fluxograma *Processo C*

1. **Escolher Atividade:** Escolha da atividade que será realizada (definir informação de projeto; ou consultar informações pessoais).
2. **Definir Informação de Projeto:** Especificação, realizada pelo Manager, das características e detalhes sobre o projeto.

3. **Consultar Informações Pessoais:** Consulta, realizada pelo Manager, às informações dos profissionais disponíveis para a realização do teste.
4. **Montar Equipes:** Escolha, realizada pelo Manager, dos profissionais que irão compor a equipe de teste.
5. **Escolher Atividade:** Escolha da atividade que será realizada (consultar documentos ou informações pessoais; ou criar / alterar documentos).
6. **Consultar Documentos de Conhecimento:** Consulta, realizada pelo *Tester*, aos documentos de conhecimento que contém informações relevantes para o teste.
7. **Realizar Avaliação:** Verificação, realizada pelo Analista de Conhecimento, da necessidade de avaliar uma informação.
8. **Avaliar Documentos de Conhecimento:** Avaliação, realizada pelo *Tester*, da completude das informações contidas nos documentos de conhecimento.
9. **Consultar Informações Pessoais:** Consulta, realizada pelo *Tester*, para localizar profissionais que possuem conhecimento para esclarecer dúvidas.
10. **Definir / Alterar Informações Pessoais:** Preenchimento dos dados pessoais e experiência do *Tester*.
11. **Criar / Alterar Documentos de Conhecimento:** Preenchimento inicial (ou alteração), realizada pelo *Tester*, de um documento que contém informações relevantes para a realização do teste, que podem ser reaproveitadas no futuro.
12. **Avaliar Documentos de Conhecimento:** Avaliação, realizada pelo Analista de Conhecimento, da relevância das informações contidas nos documentos de conhecimento.
13. **Realizar Alteração:** Verificação, realizada pelo Analista de Conhecimento, da necessidade de alterar um documento de conhecimento.
14. **Armazenar Documentos de Conhecimento:** Os documentos que realmente contém informações relevantes são armazenados na base de conhecimento para futuras consultas.
15. **Avaliar Informações Pessoais:** Avaliação, realizada pelo Analista de Conhecimento, da veracidade das informações relacionadas à experiência profissional, quando um *Tester* se declara especialista.

Descrição dos elementos de Repositório de Dados:

- A Armazenamento das Definições:** As informações definidas durante a atividade 2. *Definir Informação de Projeto* serão consideradas na escolha dos profissionais que irão compor a equipe.
- B Armazenamento provisório de um Documento de Conhecimento:** Quando um documento de conhecimento é criado durante a atividade 11. *Criar / Alterar Documentos de Conhecimento* o mesmo não é armazenado definitivamente em base de dados. Ele é submetido a uma primeira avaliação, e esta é armazenada para que as devidas correções sejam realizadas.
- C Armazenamento de um Documento de Conhecimento:** O documento de conhecimento, devidamente corrigido, é armazenado em base de dados definitiva.
- D Armazenamento das Informações Pessoais:** As informações fornecidas pelos *Testers* são armazenadas em base de dados e posteriormente avaliadas pelo Analista de Conhecimento. Elas são utilizadas na atividade 4. *Montar Equipes* e pelos *Testes* na identificação de especialistas.

3.2.4 Comparação dos Processos em Termos das suas Principais Características

Os *processos* apresentados nas seções anteriores possuem algumas características em comum, mas que são utilizadas de maneiras diferentes. São elas: *Páginas Amarelas* e *Feedback*.

Nos *Processos A* e *C*, a utilização das *Páginas Amarelas* ocorre de maneira semelhante. Há o preenchimento de informações pessoais pelos próprios *Testers*, que explicitam quais são suas experiências profissionais. Estas informações permitem que o profissional seja localizado por possuir uma experiência específica. Isto facilita a seleção de profissionais especializados para compor a equipe de teste. Diferentemente, o *Processo B* não contém informações específicas da experiência profissional dos *Testers*. As *Páginas Amarelas* são representadas por um Mapa do Projeto, que contém a referência dos *Testers* e dos projetos em que trabalharam. Existe a possibilidade de rastrear projetos semelhantes, e verificar o histórico de participação de cada profissional, entretanto, não há a possibilidade do próprio *Tester* incluir suas experiências.

O *feedback* é realizado em todos os *processos*. Conforme a necessidade, a avaliação da informação é realizada, e de acordo com esta avaliação ocorre ou não a alteração do conteúdo. Este ciclo de avaliação busca melhorar a qualidade da informação armazenada. No *Processo*

A existe maior influência do *feedback* nas decisões tomadas, pois seu resultado irá alterar tanto a maneira com que os processos serão definidos no futuro, quanto a seleção de perfis para execução dos testes. Ainda sobre o *feedback*, há uma etapa adicional apenas no *Processo C*, representada pelo Analista de Conhecimento. Ele é o responsável pela avaliação de toda informação incluída pelos *Testers*.

Quanto às características que se diferenciam em cada *processo*, estão o suporte à execução (*Processo A*), a utilização de *lessons learned* (*Processo B*), e de documentos de conhecimento e Analistas de Conhecimento (*Processo C*).

O suporte à execução é um facilitador presente somente no *Processo A*. As informações detalhadas sobre planos de teste, casos de teste e experiência requeridas são armazenadas e passíveis de avaliação. Este fator auxilia, principalmente, os *Testers* em suas atividades, além de possibilitar o conhecimento exato do *status* dos projetos.

Os outros *processos* se especializaram na utilização de documentos para auxiliar o trabalho executado pela equipe de teste. No caso do *Processo B*, os documentos chamados *lessons learned* armazenam informações sobre resultados dos projetos (tanto sucessos quanto fracassos). O escopo de uma *lesson learned* é bem definido, e considera informações mais gerenciais, que apoiam a tomada de decisão dos Gerentes à realização dos trabalhos dos *Testers*.

Os documentos utilizados pelo *Processo C* são chamados de documentos do conhecimento, que, diferentemente das *lessons learned*, possuem um escopo menos restrito. O intuito é que o *Tester* armazene nestes documentos de conhecimento toda informação que julgar relevante para seu trabalho. Entretanto, nem sempre tais documentos irão conter informações realmente relevantes (principalmente devido ao seu escopo irrestrito). Por isso, faz-se necessária a presença do Analista de Conhecimento para avaliar o conteúdo destes documentos. Este analista é um especialista no domínio, e irá decidir quais serão os documentos de conhecimento que serão armazenados na base de dados.

3.3 Análise Crítica

Uma vez que todos os *processos* foram estudados, entendidos e padronizados em uma mesma notação, o próximo passo foi organizar um método que pudesse guiar a Análise Crítica. O objetivo da análise foi avaliar o grau de aplicabilidade de cada um dos *processos* em um processo de teste composto por um conjunto de práticas TMMi, em particular o processo proposto por Camargo, Ferrari e Fabbri (2013), descrito no Capítulo 2 (Seção 2.3.1).

A metodologia, seguida para a Análise Crítica e os resultados obtidos são apresentados nas próximas seções.

3.3.1 Metodologia

A Escolha das Questões

Para auxiliar na análise de cada um dos *processos*, um grupo de questões foi criado. A definição deste grupo de questões exigiu ponderações acerca de Gestão do Conhecimento e Engenharia de Software. O intuito foi abranger, através das questões, temas como: tomada de decisão, conhecimento especializado, automatização, processos, e armazenamento de dados. Justifica-se a escolha desses temas pela sua relevância para o planejamento e realização de projetos de teste de software na indústria.

No cotidiano de uma empresa que realiza atividades de desenvolvimento de software, os fatores primordiais a serem atendidos são o custo e o prazo de um projeto (KARHU; TAIPALE; SMOLANDER, 2009). Portanto, as empresas buscam otimizar suas atividades a fim de respeitar tais fatores, e uma grande preocupação é como manter a qualidade de seus produtos dentro deste contexto.

Não há dúvida de que o alinhamento das atividades com *processos* e o investimento em *conhecimento especializado* implicam na otimização de custos e prazos de um projeto (assuntos largamente tratados no Capítulo 2). No entanto, alguns outros fatores podem trazer estes benefícios. É o caso da *automação* de atividades. Nem todas as tarefas são automatizadas em uma equipe de teste, principalmente devido ao custo envolvido em manter as automatizações atualizadas. Identificar quais atividades são passíveis de automatização pode contribuir com a tarefa de otimização (DUSTIN; GARRETT; GAUF, 2009).

A existência de um *repositório de dados* a ser consultado também é relevante, pois permite a reutilização de informação. Apesar de grande relevância, a taxa de reutilização de artefatos em uma equipe de teste ainda é baixa (ABDULLAH; ERI; TALIB, 2011). E a cultura da “não-reutilização” muitas vezes está relacionada à falta de organização e da inexistência de um repositório de dados centralizado.

Decisões erradas também podem impactar no orçamento de uma equipe de teste, mesmo quando todas as outras precauções foram cuidadosamente adotadas. É importante que se ofereçam ferramentas aos membros da equipe, para que eles possam balizar e identificar quais são os rumos corretos a serem traçados em suas atividades.

Em suma, ao relacionar estes temas no formato de questões, é possível identificar quais são os pontos de melhoria que podem ser implementados, e no caso desta Análise Crítica, como cada *processo* pode auxiliar na otimização das atividades de uma equipe de teste.

As questões formuladas são as apresentadas na Tabela 3.1.

Tabela 3.1: Questões para Análise Crítica

Questionamentos	
a	Atividade envolve tomada de decisão?
b	Atividade exige um ator com conhecimento especializado?
c	Atividade pode ser automatizada?
d	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?
e	Atividade requer o envolvimento de outras pessoas/equipes?
f	Atividade pode influenciar outras atividades (futuras)?

A Análise Crítica

Inicialmente, as práticas do TMMi pertencentes ao modelo de processo base foram analisadas sob a perspectiva de cada uma das questões propostas de acordo com a vivência e a experiência profissional da autora desta dissertação. O objetivo desta etapa foi identificar e entender quais são as características das atividades pertencentes a um processo de teste de software. É importante ressaltar que a Análise Crítica foi realizada de maneira independente à escrita da dissertação. A interpretação dos resultados dos estudos e suas conclusões não influenciaram no processo desta análise.

Com a análise das práticas, a etapa seguinte foi a avaliação da aplicabilidade dos *processos* com relação às práticas. A avaliação foi realizada com base nas respostas dadas às questões. Notas e justificativas foram atribuídas a cada uma das práticas.

Para a atribuição das notas, uma escala *Likert* de quatro graus foi criada, com os itens *Não Aplicável*, *Pouco Aplicável*, *Aplicável*, e *Extremamente Aplicável*. A escolha por quatro itens na escala evita um valor neutro, pois obriga o respondente a se posicionar quanto à intensidade de sua opinião (GARLAND, 1991).

3.3.2 Resultados da Análise Crítica

Como resultado desta avaliação, cada *processo* recebeu 33 notas (correspondendo a cada uma das práticas), variando conforme a escala definida. Com o intuito de facilitar a apresentação dos resultados, toda a análise foi tabulada. Exemplos podem ser obtidos nas Tabelas 3.2, 3.3 e 3.4, que correspondem aos *Processos A, B e C* respectivamente. Nas tabelas, as linhas representam as práticas, enquanto as colunas representam as análises das questões e dos *processos*. A análise completa de todas as práticas é encontrada no Anexo A.

Para que fosse possível a contabilização de notas dos *processos*, e até mesmo das fases do processo de teste (ou qualquer outro agrupamento), houve a necessidade de transformar a escala nominal em numeral. Cada grau da escala recebeu um valor de 1 (um) a 4 (quatro), no qual o valor 1 corresponde a *Não Aplicável*, 2 corresponde a *Pouco Aplicável*, 3 corresponde a *Aplicável* e 4 corresponde a *Extremamente Aplicável*.

O resultado obtido desta Análise Crítica é encontrado nos gráficos das Figuras 3.5, 3.6 e 3.7.

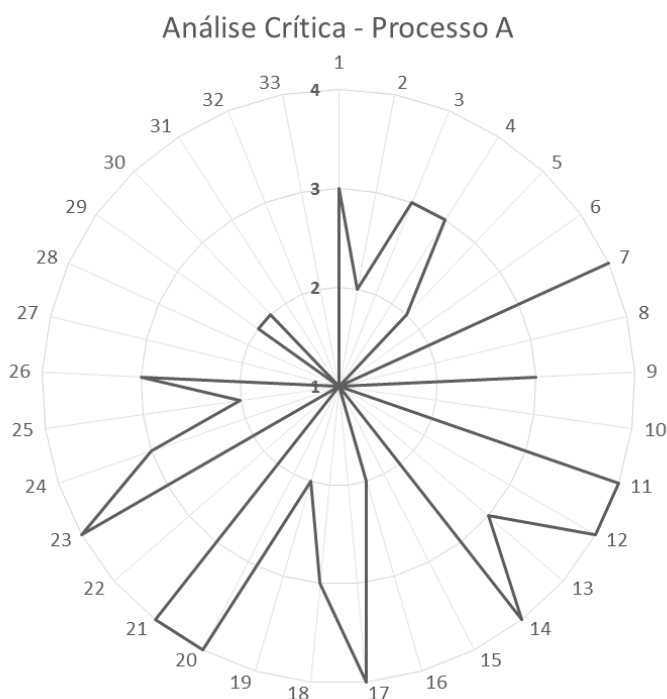


Figura 3.5: Notas Processo A

As raias são enumeradas de 1 a 33, correspondendo às práticas do processo de teste definido por (CAMARGO; FERRARI; FABBRI, 2013). Os círculos representam as notas obtidas por cada prática (de 1 a 4, conforme a escala apresentada). O centro corresponde à nota 1, e o círculo mais externo à nota 4.

Tabela 3.2: Análise Crítica da Fase Projeto de Casos de Teste do *Processo A*

Prática (TMMi)	Análise Crítica (Prática)		Análise Crítica (Processo A)
SP1.2 Identificar e priorizar casos de teste <i>Caso de teste são identificados e priorizados utilizando técnicas de design de teste</i>	Atividade envolve tomada de decisão?	Sim. Nesta atividade são definidas quais serão as entradas necessárias e as saídas esperadas. Além disso existe a tomada de decisão quanto à priorização dos casos de teste.	Este <i>processo</i> é preparado para receber os planos de teste. Oferece apoio para consulta dos planos de projetos anteriores, e também é possível encontrar pessoas com competências necessárias para a realização da atividade.
	Atividade exige um ator com conhecimento especializado?	Sim, conhecimento no domínio é fundamental.	
	Atividade pode ser automatizada?	Parcialmente. A criação dos casos de teste pode ser automatizada, mas a priorização necessita de intervenção humana.	
	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?	Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade.	
	Atividade requer o envolvimento de outras pessoas/equipes?	Não. A identificação dos casos de teste é uma atividade interna à equipe de testes.	
	Atividade pode influenciar outras atividades (futuras)?	Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Extremamente aplicável
SP1.3 Identificar dados de teste específicos necessários <i>São identificados os dados de testes específicos necessários para suportar as condições de teste e execução de casos de teste.</i>	Atividade envolve tomada de decisão?	Sim, decisão sobre a criação dos casos de teste são tomadas nesta atividade.	Este <i>processo</i> é preparado para receber os planos de teste. Oferece apoio para consulta dos planos de projetos anteriores, e também é possível encontrar pessoas com competências necessárias para a realização da atividade.
	Atividade exige um ator com conhecimento especializado?	Sim, conhecimento no domínio é fundamental para a correta identificação dos dados de teste.	
	Atividade pode ser automatizada?	Não. A seleção dos dados de teste específicos não pode ser automatizada, pois é uma atividade restrita ao escopo do projeto.	
	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?	Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade. Entretanto há a necessidade de adequar tais casos de teste para os requisitos individuais deste projeto.	
	Atividade requer o envolvimento de outras pessoas/equipes?	Não. A identificação dos dados de teste é uma atividade interna à equipe de testes.	
	Atividade pode influenciar outras atividades (futuras)?	Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Extremamente aplicável

Tabela 3.3: Análise Crítica da Fase Projeto de Casos de Teste do *Processo B*

Prática (TMMi)	Análise Crítica (Prática)		Análise Crítica (Processo B)
SP1.2 Identificar e priorizar casos de teste <i>Caso de teste são identificados e priorizados utilizando técnicas de design de teste</i>	Atividade envolve tomada de decisão?	Sim. Nesta atividade são definidas quais serão as entradas necessárias e as saídas esperadas. Além disso existe a tomada de decisão quanto à priorização dos casos de teste.	<i>O processo não apoia a criação de planos de teste diretamente, pois somente são armazenadas informações do projeto. Mas oferece meios de encontrar lessons learned relacionadas a projetos anteriores, que podem auxiliar em algumas questões. Também é possível encontrar especialistas através das páginas amarelas.</i>
	Atividade exige um ator com conhecimento especializado?	Sim, conhecimento no domínio é fundamental.	
	Atividade pode ser automatizada?	Parcialmente. A criação dos casos de teste pode ser automatizada, mas a priorização necessita de intervenção humana.	
	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?	Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade.	
	Atividade requer o envolvimento de outras pessoas/equipes?	Não. A identificação dos casos de teste é uma atividade interna à equipe de testes.	
	Atividade pode influenciar outras atividades (futuras)?	Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Aplicável
SP1.3 Identificar dados de teste específicos necessários <i>São identificados os dados de testes específicos necessários para suportar as condições de teste e execução de casos de teste.</i>	Atividade envolve tomada de decisão?	Sim, decisão sobre a criação dos casos de teste são tomadas nesta atividade.	<i>O processo não apoia a criação de planos de teste diretamente, pois somente são armazenadas informações do projeto. Mas oferece meios de encontrar lessons learned relacionadas a projetos anteriores, que podem auxiliar em algumas questões. Também é possível encontrar especialistas através das páginas amarelas.</i>
	Atividade exige um ator com conhecimento especializado?	Sim, conhecimento no domínio é fundamental para a correta identificação dos dados de teste.	
	Atividade pode ser automatizada?	Não. A seleção dos dados de teste específicos não pode ser automatizada, pois é uma atividade restrita ao escopo do projeto.	
	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?	Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade. Entretanto há a necessidade de adequar tais casos de teste para os requisitos individuais deste projeto.	
	Atividade requer o envolvimento de outras pessoas/equipes?	Não. A identificação dos dados de teste é uma atividade interna à equipe de testes.	
	Atividade pode influenciar outras atividades (futuras)?	Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Aplicável

Tabela 3.4: Análise Crítica da Fase Projeto de Casos de Teste do *Processo C*

Prática (TMMi)	Análise Crítica (Prática)		Análise Crítica (Processo C)
SP1.2 Identificar e priorizar casos de teste <i>Caso de teste são identificados e priorizados utilizando técnicas de design de teste</i>	Atividade envolve tomada de decisão?	Sim. Nesta atividade são definidas quais serão as entradas necessárias e as saídas esperadas. Além disso existe a tomada de decisão quanto à priorização dos casos de teste.	O <i>processo</i> não realiza a criação de planos de teste diretamente, entretanto, informações relevantes podem ser encontradas nos documentos de conhecimento. O <i>framework</i> também oferece meios de encontrar pessoas que possuem conhecimento necessário para realização da atividade.
	Atividade exige um ator com conhecimento especializado?	Sim, conhecimento no domínio é fundamental.	
	Atividade pode ser automatizada?	Parcialmente. A criação dos casos de teste pode ser automatizada, mas a priorização necessita de intervenção humana.	
	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?	Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade.	
	Atividade requer o envolvimento de outras pessoas/equipes?	Não. A identificação dos casos de teste é uma atividade interna à equipe de testes.	
	Atividade pode influenciar outras atividades (futuras)?	Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Extremamente aplicável
SP1.3 Identificar dados de teste específicos necessários <i>São identificados os dados de testes específicos necessários para suportar as condições de teste e execução de casos de teste.</i>	Atividade envolve tomada de decisão?	Sim, decisão sobre a criação dos casos de teste são tomadas nesta atividade.	O <i>processo</i> não realiza a criação de planos de teste diretamente, entretanto, informações relevantes podem ser encontradas nos documentos de conhecimento. O <i>framework</i> também oferece meios de encontrar pessoas que possuem conhecimento necessário para realização da atividade.
	Atividade exige um ator com conhecimento especializado?	Sim, conhecimento no domínio é fundamental para a correta identificação dos dados de teste.	
	Atividade pode ser automatizada?	Não. A seleção dos dados de teste específicos não pode ser automatizada, pois é uma atividade restrita ao escopo do projeto.	
	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?	Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade. Entretanto há a necessidade de adequar tais casos de teste para os requisitos individuais deste projeto.	
	Atividade requer o envolvimento de outras pessoas/equipes?	Não. A identificação dos dados de teste é uma atividade interna à equipe de testes.	
	Atividade pode influenciar outras atividades (futuras)?	Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Extremamente aplicável

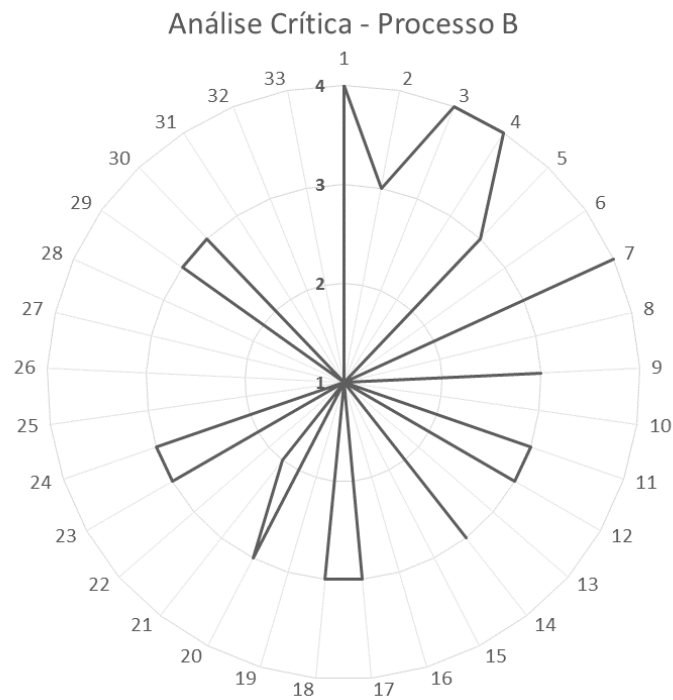


Figura 3.6: Notas Processo B

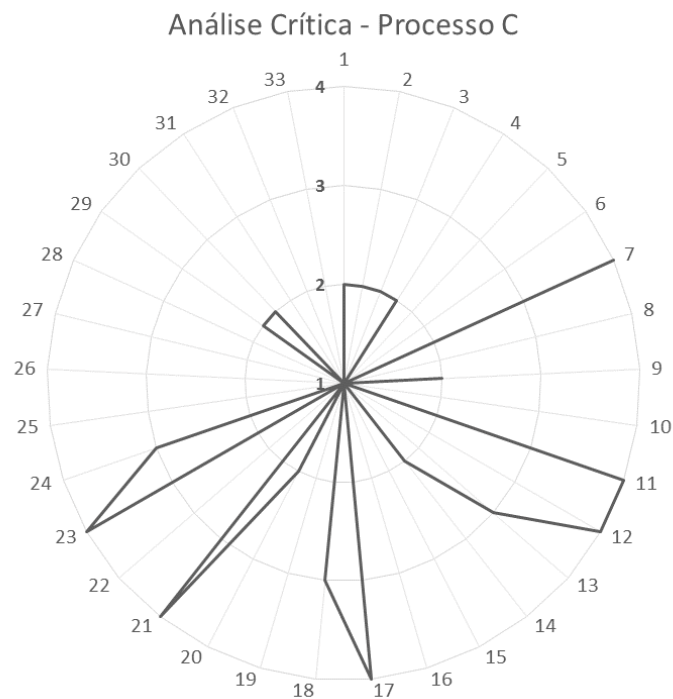


Figura 3.7: Notas Processo C

A relação das práticas é encontrada na Tabela A.2.

A Figura 3.8 ilustra as notas obtidas por fase do processo de teste. Foi utilizada a média aritmética para obtenção de tais notas, uma vez que todas as práticas possuem o mesmo peso.

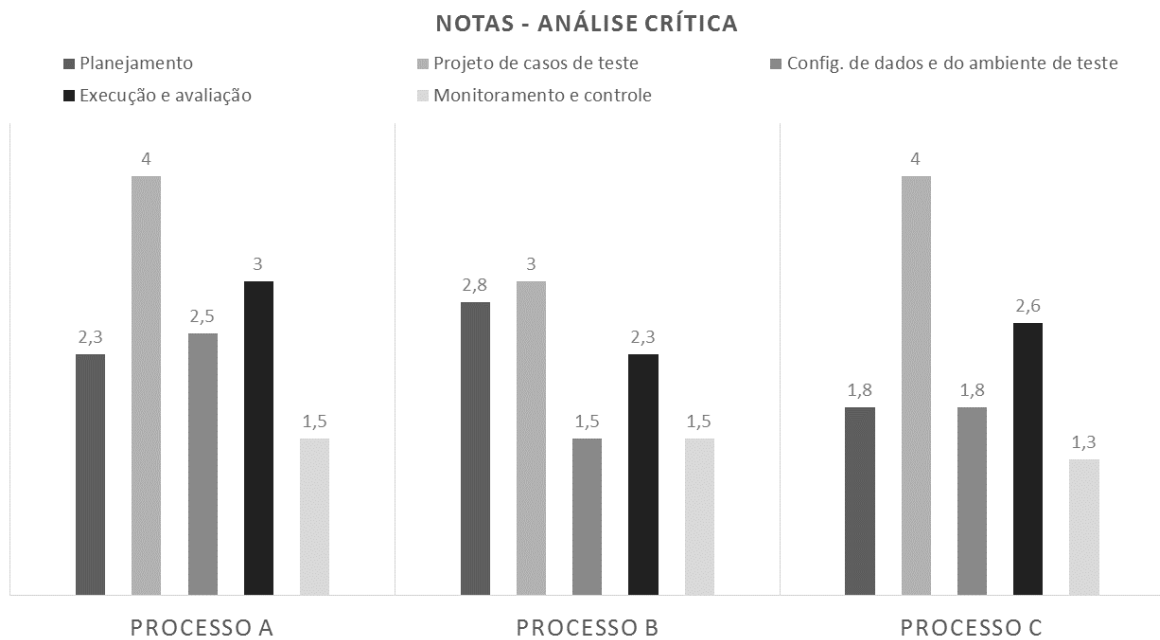


Figura 3.8: Análise Crítica Agrupada por Fase do Processo de Teste

A seguir serão comentados os resultados, agrupados de acordo com as fases do processo de teste (Figura 3.8).

Planejamento

O maior grupo de práticas é representado pela fase *Planejamento*, que contempla 10 das 33 práticas do processo de teste. Dentre as atividades realizadas, estão: análise de riscos, definição de cronogramas e definição de abordagem de teste. Ou seja, esta fase atua principalmente na área gerencial. As notas obtidas pelos *processos* na fase *Planejamento* variaram meio ponto entre si.

Com relação à fase *Planejamento*, os *processos* foram mais bem avaliados para as seguintes práticas:

- *Processos A e C*:
 - SP4.2 Planejar a equipe de teste
- *Processo B*:
 - SP1.2 Identificar riscos do produto
 - SP2.1 Identificar elementos e características a serem testados

- SP2.2 Definir a abordagem de teste
- SP4.2 Planejar a equipe de teste

O *Processo B* destacou-se entre os outros, por ser de cunho gerencial, ou seja, como é baseado em documentos *lessons learned*, o *processo* armazena informações que são relevantes às decisões gerenciais de um projeto.

Os *Processos A* e *C* receberam nota 4 (*Extremamente Aplicável*) apenas para a prática **Planejar a equipe de teste (SP4.2)**, o que mostra o caráter não gerencial destes *processos*. Esta prática está relacionada às Páginas Amarelas, técnica de Gestão do Conhecimento presente em todos os *processos* avaliados.

Para identificar quais são as deficiências, foram consideradas as práticas para aos quais os *processos* receberam notas 1 ou 2 (*Não Aplicável* e *Pouco Aplicável*, respectivamente). São elas:

- *Processo A:*

- SP1.3 Analisar riscos do produto (nota 2)
- SP2.4 Definir critérios de parada (nota 2)
- SP4.1 Estabelecer o cronograma de teste (nota 1)
- SP4.5 Estabelecer o plano de teste (nota 1)
- SP1.3 Analisar os requisitos do ambiente de teste (nota 1)

- *Processo B:*

- SP4.1 Estabelecer o cronograma de teste (nota 1)
- SP4.5 Estabelecer o plano de teste (nota 1)
- SP1.3 Analisar os requisitos do ambiente de teste (nota 1)

- *Processo C:*

- SP1.2 Identificar riscos do produto (nota 2)
- SP1.3 Analisar riscos do produto (nota 2)
- SP2.1 Identificar elementos e características a serem testados (nota 2)
- SP2.2 Definir a abordagem de teste (nota 2)

- SP2.4 Definir critérios de parada (nota 1)
- SP4.1 Estabelecer o cronograma de teste (nota 1)
- SP4.5 Estabelecer o plano de teste (nota 1)
- SP1.1 Obter (eliciar) necessidades do ambiente de teste (nota 2)
- SP1.3 Analisar os requisitos do ambiente de teste (nota 1)

Os *processos A, B e C* obtiveram nota 1, ou seja, foram considerados não aplicáveis, em um mesmo conjunto de práticas: *Estabelecer o cronograma de teste (SP4.1)*, *Estabelecer o plano de teste (SP4.5)*, *Analisar os requisitos do ambiente de teste (SP1.3)*. Percebe-se que estas são falhas gerais aos *processos* investigados, e as técnicas de Gestão do Conhecimento aplicadas não conseguem extrair a experiência dos envolvidos para auxiliar a realização de tais atividades.

Dentre os três *processos*, o melhor avaliado para a fase de *Planejamento* foi o *B*.

Projeto de casos de teste

Composta por apenas duas práticas, esta fase foi aquela que mais se destacou na Análise Crítica, pois os *processos* obtiveram boas notas em ambas as práticas.

Diferentemente da fase anterior (*Planejamento*), nesta fase os *Processos A e C* foram os mais bem avaliados, em razão de se adequarem mais às necessidades das atividades exercidas (*Projeto de casos de teste*). O *Processo B*, por abranger informações gerenciais (*lessons learned*), não atende por completo os objetivos da fase.

As práticas nas quais os *processos* foram mais bem avaliados para a fase de *Projeto de casos de teste* foram:

- *Processos A e C*:
 - SP1.2 Identificar e priorizar casos de teste
 - SP1.3 Identificar dados de teste específicos necessários

O *Processo B* recebeu nota 3 (*Aplicável*) em todas as suas práticas.

Configuração de dados e do ambiente de teste

Esta fase resultou em uma das maiores variações entre as notas dos *processos*. Para esta fase, não se observou um padrão de avaliação entre as práticas, ou seja, os pontos fortes e as deficiências são diferentes entre os *processos*.

Pelas notas obtidas, fica evidente que os *processos* pouco podem auxiliar nas atividades realizadas na fase *Configuração de dados e do ambiente de teste*. Apenas uma nota 4 (*Extremamente Aplicável*) foi atribuída, considerando todos os *processos*. Em particular, para a prática ***Desenvolver cronograma de execução do teste (SP2.4)***, ao *Processo A* foi atribuída essa nota.

Os *processos* receberam avaliação baixa (notas 1 - *Não Aplicável* e 2 - *Pouco Aplicável*) para as seguintes práticas:

- *Processo A*:
 - SP2.1 Implementar o ambiente de teste (nota 1)
 - SP2.4 Realizar o teste *intake* (pré-teste) do ambiente de teste (nota 2)

- *Processo B*:
 - SP2.1 Desenvolver e priorizar os procedimentos de teste (nota 1)
 - SP2.1 Implementar o ambiente de teste (nota 1)
 - SP2.4 Realizar o teste *intake* (pré-teste) do ambiente de teste (nota 1)

- *Processo C*:
 - SP2.4 Desenvolver cronograma de execução do teste (nota 2)
 - SP2.1 Implementar o ambiente de teste (nota 1)
 - SP2.4 Realizar o teste *intake* (pré-teste) do ambiente de teste (nota 1)

As práticas ***Implementar o ambiente de teste (SP2.1)*** e ***Realizar o teste intake (pré-teste) do ambiente de teste (SP2.4)*** foram as menos atendidas pelos *processos*. Salienta-se assim a inaptidão destas técnicas de Gestão do Conhecimento em atender as atividades de montagem e validação do ambiente de testes.

Dentre os três *processos*, o melhor avaliado para a fase de *Configuração de dados e do ambiente de teste* foi o *Processo A*.

Execução e avaliação do teste

Com o segundo maior grupo de práticas, esta fase obteve o maior número de notas máximas. Entretanto, devido à quantidade de práticas que a compõe, as médias calculadas para esta fase não estiveram entre as mais altas (os valores *Pouco Aplicável* e *Aplicável*), pois os *processos* também receberam diversas notas baixas.

As práticas nas quais os *processos* foram mais bem avaliados (receberam nota 4) na fase *Execução e avaliação do teste* foram:

- *Processo A*:
 - SP3.2 Executar casos de teste
 - SP4.1 Decidir sobre incidentes com o grupo de controle de configuração
 - SP4.2 Executar ações apropriadas para corrigir os incidentes de teste
 - SP5.1 Executar casos de teste não funcional

- *Processo C*:
 - SP3.2 Executar casos de teste
 - SP4.2 Executar ações apropriadas para corrigir os incidentes de teste
 - SP5.1 Executar casos de teste não funcional

Observa-se que os *Processos A* e *C* possuem, entre as mais bem atendidas, uma lista de três práticas em comum: *Executar casos de teste (SP3.2)*, *Executar ações apropriadas para corrigir os incidentes de teste (SP4.2)* e *Executar casos de teste não funcional (SP5.1)*. As três práticas estão diretamente ligadas à execução do teste de software. Já o *Processo B* não foi considerado *Extremamente Aplicável* para nenhuma das práticas.

Este fato contribui para a hipótese de que o *Processo B*, por armazenar informações mais gerenciais, atenderá melhor aos objetivos das práticas cuja fase também tenham cunho gerencial como, por exemplo, a fase *Planejamento*. Os *Processos A* e *C*, por sua vez, tornam-se mais aplicáveis em fases que estão relacionadas à execução do teste de software como, por exemplo, as fases *Projeto de casos de teste* e *Execução e avaliação do teste*.

Dentre as práticas desta fase, aquelas cujos *processos* obtiveram notas baixas (notas 1 - *Não Aplicável* e 2 - *Pouco Aplicável*) foram:

- *Processo A*:

- SP3.4 Escrever *log* de teste (nota 2)
- SP4.3 Acompanhar o *status* dos incidentes de teste (nota 1)
- SP5.3 Escrever *log* de teste não funcional (nota 2)

- *Processo B:*
 - SP3.4 Escrever *log* de teste (nota 1)
 - SP4.2 Executar ações apropriadas para corrigir os incidentes de teste (nota 2)
 - SP4.3 Acompanhar o *status* dos incidentes de teste (nota 1)
 - SP5.3 Escrever *log* de teste não funcional (nota 1)

- *Processo C:*
 - SP3.4 Escrever *log* de teste (nota 1)
 - SP4.1 Decidir sobre incidentes com o grupo de controle de configuração (nota 1)
 - SP4.3 Acompanhar o *status* dos incidentes de teste (nota 1)
 - SP5.3 Escrever *log* de teste não funcional (nota 1)

A quantidade de notas baixas recebidas nesta fase reduziu a média calculada para os *processos*. Os principais problemas relatados estão relacionados ao registro de *log* de teste (seja funcional, ou não funcional), e ao *status* dos incidentes de teste. Para a prática **Acompanhar o status dos incidentes de teste (SP4.3)**, todos os *processos* receberam a avaliação de *Não Aplicável*.

Dentre os três *processos*, o melhor avaliado para a fase de *Execução e avaliação do teste* foi o *Processo A*.

Monitoramento e controle

Esta fase resultou na pior avaliação dos *processos*. De acordo com a avaliação realizada, nenhum *processo* recebeu nota 4 *Extremamente Aplicável*. As médias calculadas estiveram entre os valores *Não Aplicável* e *Pouco Aplicável*, indicando que nenhuma das técnicas de Gestão do Conhecimento utilizadas nos *processos* são aplicáveis às atividades de *Monitoramento e controle*.

Levando em consideração que os *processos* receberam notas baixas para quase a totalidade das práticas desta fase, serão listadas a seguir apenas as práticas nas quais os *processos* foram considerados *Aplicáveis* (receberam nota 3):

- *Processo A*:
 - SP1.6 Conduzir revisões do progresso do teste

- *Processo B*:
 - SP3.1 Analisar problemas
 - SP3.2 Tomar ação corretiva

Para todas as outras práticas desta fase, os *processos* obtiveram notas 1 - *Não Aplicável* e 2 - *Pouco Aplicável*.

Constata-se novamente para esta fase, a qual possui cunho gerencial, que o *processo B* se destacou, especialmente entre as práticas que exigem tomada de decisão.

3.3.3 Avaliação Geral

Pontos Fortes e Fracos dos *Processos* de Gestão do Conhecimento

Ao comparar os resultados da Análise Crítica, identifica-se que para determinadas práticas um *processo* se destaca perante os outros. Sendo assim, é importante evidenciar quais são estas características que podem ser consideradas como pontos fortes e fracos² de cada *processo*.

O *Processo A* se sobressai perante os *Processos B* e *C* nas seguintes práticas: ***Desenvolver cronograma de execução do teste (SP2.4)***, ***Escrever log de teste (SP3.4)*** e ***Escrever log de teste não funcional (SP5.3)***, todas pertencentes a fase *Execução e avaliação do teste*. Além disso, o mesmo *processo* se sobressai para a prática ***Conduzir revisões do progresso do teste (SP1.6)***, da fase *Monitoramento e controle*. Para tais práticas, o *Processo A* recebeu nota 3 (*Aplicável*) enquanto os *Processos B* e *C* foram avaliados com nota 1 (*Não Aplicável*). Isso se dá devido ao apoio à execução, característica pertencente apenas ao *Processo A*.

A prática na qual o *Processo B* se evidencia em relação aos outros *processos* é ***Definir critérios de parada (SP2.4)***, da fase *Planejamento*. Nesta prática, a avaliação do *Processo B* foi ***Extremamente Aplicável*** (nota 4), enquanto que o *Processo A* foi avaliado como *Pouco Aplicável* (nota 2), e o *Processo C* como *Não Aplicável* (nota 1). O *Processo B* é beneficiado devido ao escopo bem definido das *lessons learned*, em comparação com as informações armazenadas sobre os projetos nos outros dois *processos*.

²Na análise apresentada, os pontos fracos estão relacionados às notas baixas atribuídas aos *processos* para as práticas de teste mencionadas.

Não houve uma prática na qual o *Processo C* se destacasse ante os *Processos A e B*, pois para todas as práticas que recebeu notas 3 e 4 (*Aplicável e Extremamente Aplicável*, respectivamente), as notas dos outros *processos* foram semelhantes, impossibilitando a definição de pontos fortes exclusivos deste *processo*.

Dependências Entre as Práticas

Algumas práticas pertencentes ao modelo de processo de teste de software utilizado nesta dissertação são interdependentes, conforme apresentado no trabalho de Höhn (2011). Estas dependências indicam que certas práticas devem preceder outras em sua ordem de implementação.

A estrutura de dependência é mostrada na Figura 3.9, na qual as setas caracterizam o relacionamento entre as práticas. O sentido da seta aponta qual é a prática que possui a dependência. Por exemplo, para implementar *Estabelecer o cronograma de teste (SP4.1)*, é necessário que se tenha implementado *Estabelecer o plano de teste (SP4.5)* (parte superior da Figura 3.9).

Para cada prática também são mostradas as notas recebidas pelos *processos*, possibilitando assim identificar quaisquer omissões quanto às dependências, pois é esperado que práticas interdependentes sejam (ou não) atendidas pelo *processo* de maneira semelhante.

Percebe-se que para o primeiro conjunto de práticas, mostrado na parte superior da Figura 3.9 (*Estabelecer o cronograma de teste - SP4.1, Planejar a equipe de teste - SP4.2 e Estabelecer o plano de teste - SP4.5*), a Análise Crítica não avaliou os *processos* conforme o previsto, pois as notas atribuídas estão nos extremos opostos da escala. A análise revela a necessidade de estabelecer o plano de teste e o cronograma de teste, para que seja possível planejar a equipe de teste. No entanto, a Análise Crítica indica que todos os *processos* auxiliam na realização da atividade de planejar a equipe de teste, sem necessariamente atender às atividades que a precedem.

Este fato evidencia que outras fontes devem ser consultadas para a realização desta atividade, pois apesar de serem *Extremamente Aplicáveis* à prática *Planejar a equipe de teste (SP4.2)*, informações relevantes (por exemplo, quanto a escopo e prazo) não serão encontradas nos *processos*.

A avaliação do segundo grupo de práticas, mostrada na parte inferior da Figura 3.9, está de acordo com o esperado. Nenhuma variação significativa quanto às notas obtidas pelos *processos* foi observada.

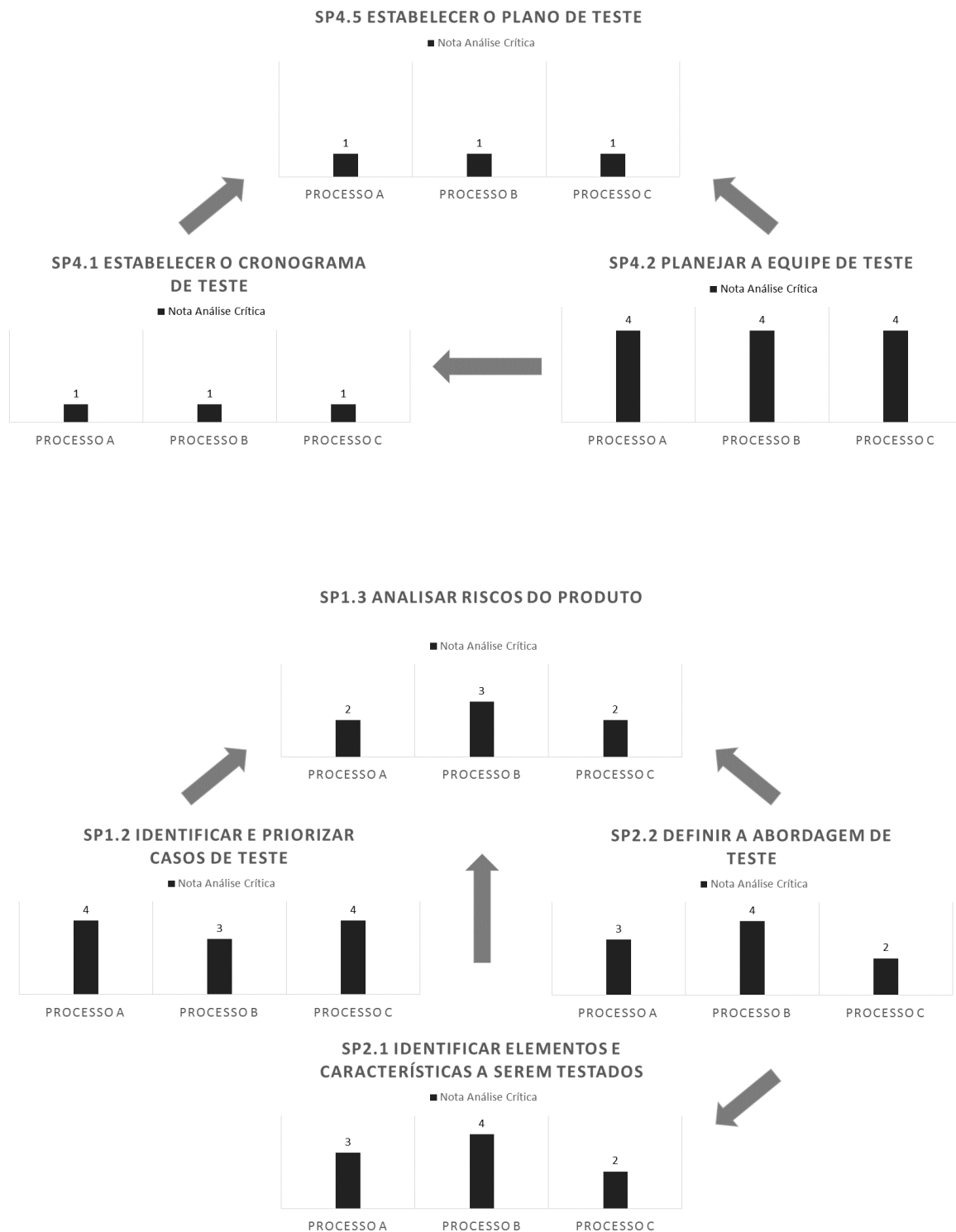


Figura 3.9: Dependências Entre as Práticas

Avaliação Geral

Conforme os resultados apresentados na seção anterior, em cada fase do processo de teste um *processo* de Gestão do Conhecimento se mostrou mais aplicável. Contudo, é necessário identificar qual deles se adequa melhor ao processo de teste como um todo, sendo possível utilizá-lo em todas as práticas implementadas, em qualquer uma das fases.

Desta forma, as notas obtidas pelos *processos* em cada fase do processo de teste foram utilizadas para calcular suas notas finais. O método adotado para determinar tais notas foi a média aritmética. Os resultados são ilustrados na Figura 3.10.

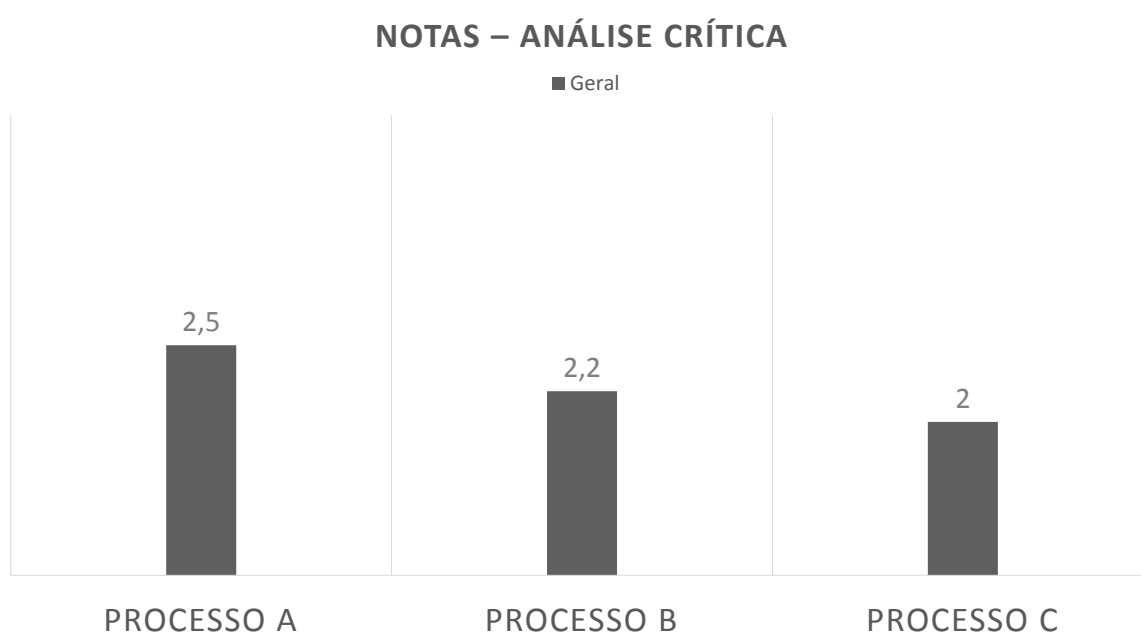


Figura 3.10: Análise Crítica Agrupada por Processo

De acordo com esta Análise Crítica, o *Processo A* foi aquele que, no geral, mostrou-se mais aplicável às práticas do processo de teste, seguido pelos *Processos B* e *C*, nesta ordem.

3.4 Considerações Finais

Este capítulo apresentou as etapas percorridas para a realização da Análise Crítica dos *processos* de Gestão do Conhecimento feita pela autora desta dissertação. Partindo da necessidade de padronização dos trabalhos para que uma comparação inicial fosse possível, a Análise Crítica se baseou em questões relevantes para a indústria. Os fluxogramas na notação BPMN foram apresentados em conjunto com a descrição de suas atividades. Uma comparação entre os *processos* de Gestão do Conhecimento elaborados foi brevemente apresentada e foram discutidos

a metodologia utilizada na Análise Crítica.

Constatou-se que a ordem de aplicabilidade dos *processos* de Gestão do Conhecimento partindo do mais aplicável às práticas do processo de teste foi: *Processo A*, *Processo B* e *Processo C*.

No próximo capítulo será apresentado um estudo realizado com profissionais da área de teste de software, que possui os mesmos moldes utilizados nesta Análise Crítica. A intenção é complementar os resultados desta Análise Crítica com a opinião de profissionais da área de teste de software.

Capítulo 4

A VISÃO DE PROFISSIONAIS: GESTÃO DO CONHECIMENTO EM PROCESSO DE TESTE DE SOFTWARE

4.1 Considerações Iniciais

Com a finalidade de complementar a Análise Crítica realizada e relatada no Capítulo 3 desta dissertação, uma investigação foi conduzida, na qual três abordagens de Gestão do Conhecimento foram analisadas por profissionais da área de teste de software. O objetivo desta investigação foi identificar o grau de aplicabilidade de tais abordagens em um processo de teste de software, de acordo com a experiência desses profissionais.

As três abordagens e o processo de teste utilizados na investigação foram os mesmos da Análise Crítica. Os profissionais convidados a participar possuem experiência na área de teste de software (na indústria e/ou na academia), e receberam a orientação teórica necessária ao longo de sua participação.

A investigação concretizou-se como um *Survey*, que se utiliza de questionários para extrair informações de uma população (WOHLIN et al., 2000).

Nas próximas seções serão discutidos a metodologia utilizada para a aplicação do *Survey*, a organização dos dados, o critério de seleção de respostas e os procedimentos adotados para a análise dos resultados.

É importante ressaltar que o termo *processo* refere-se aos processos de Gestão do Conhecimento derivados dos *frameworks* selecionados neste trabalho.

4.2 Planejamento e Projeto do Survey

Para a realização do *Survey* foi utilizada a ferramenta LimeSurvey¹. A escolha da ferramenta levou em consideração características como a distribuição e acesso das questões via *web*, a possibilidade de pausar o questionário para posterior retorno e de inserir figuras e arquivos ao longo das questões. A ferramenta foi disponibilizada em um servidor interno do Departamento de Computação da Universidade Federal de São Carlos.

Foram criadas três versões do *Survey*, cada uma abordando um *framework* de Gestão do Conhecimento específico. As versões foram denominadas A², B³ e C⁴. A versão A corresponde ao trabalho de Colomo-Palacios et al. (2014), a versão B ao trabalho de Andrade et al. (2013), e a versão C ao trabalho de Xue-Mei et al. (2009).

Os participantes do estudo foram orientados a escolher uma versão do *Survey* ao acaso e responder apenas a versão escolhida. Após o encerramento da pesquisa, constatou-se que somente um conjunto de respostas foi obtido para cada participante.

Os convites para a participação no *Survey* foram disponibilizados em grupos e comunidades de teste de software. Exemplos desses grupos e comunidades são o GUTS – Grupo de Usuários de Testes de Software – SUCESU-RS⁵ e o Teste de Software Ceará⁶. Alguns participantes foram selecionados individualmente para responder ao *Survey*, por possuírem experiência em teste de software na indústria.

Cada versão do *Survey* foi dividida em três etapas, cada uma agrupando questões e informações relacionadas. A primeira etapa continha questões de caracterização do perfil do participante. A segunda etapa apresentava informações sobre o processo de teste e sobre o *processo* de Gestão do Conhecimento. A terceira, por sua vez, continha as questões nas quais o participante deveria avaliar a aplicabilidade do *processo* de Gestão do Conhecimento ao processo de teste apresentado.

¹<https://www.limesurvey.org/> – Acessado em 10/04/2015.

²<http://amon.dc.ufscar.br/limesurvey/index.php?sid=64742&lang=pt-BR> - Acessado em 10/04/2015.

³<http://amon.dc.ufscar.br/limesurvey/index.php?sid=55296&lang=pt-BR> - Acessado em 10/04/2015.

⁴<http://amon.dc.ufscar.br/limesurvey/index.php?sid=26614&lang=pt-BR> - Acessado em 10/04/2015.

⁵<https://groups.google.com/forum/?hl=pt-BR\#!forum/guts-rs-sucesu> - Acessado em 10/04/2015.

⁶<https://groups.google.com/forum/?hl=pt-BR\#!forum/teste-software-ce> - Acessado em 10/04/2015.

A primeira etapa foi igual nas três versões do *Survey*, e consistia na identificação dos participantes, informações sobre formação acadêmica e grau de experiência na área de teste de software. Um trecho da tela desta primeira etapa pode ser visualizado na Figura 4.1.

A imagem mostra uma interface web com o título "Caracterização dos participantes" em um cabeçalho verde. Abaixo, há quatro seções de formulário:

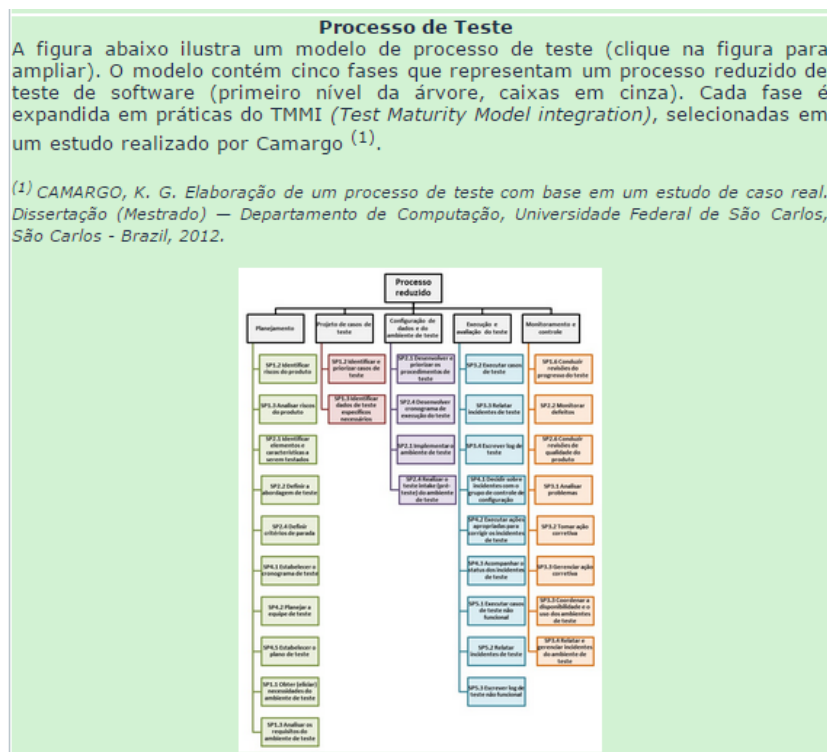
- Nome:** Um campo de texto vazio.
- * E-mail:** Um campo de texto vazio.
- * Grau de instrução:** Um campo com o texto "Escolha uma das seguintes respostas:" e quatro opções de radio buttons: "Superior", "Especialização", "Mestrado" e "Doutorado".
- * Área de formação:** Um campo com o texto "Escolha uma das seguintes respostas:" e quatro opções de radio buttons: "Administração", "Computação", "Engenharia" e "Outros:". Abaixo da opção "Outros:" há um campo de texto vazio.

Figura 4.1: Página de Caracterização dos Participantes

A segunda etapa foi composta pela apresentação das teorias envolvidas, sendo dividida em duas partes: processo de teste e *processo* de Gestão do Conhecimento. Essa segunda etapa foi de fundamental importância para o *Survey*. Era imprescindível que o participante compreendesse o processo de teste e o *processo* de Gestão do Conhecimento apresentados. Nenhuma questão compôs esta etapa. Trechos das telas desta segunda etapa podem ser visualizados na Figura 4.2.

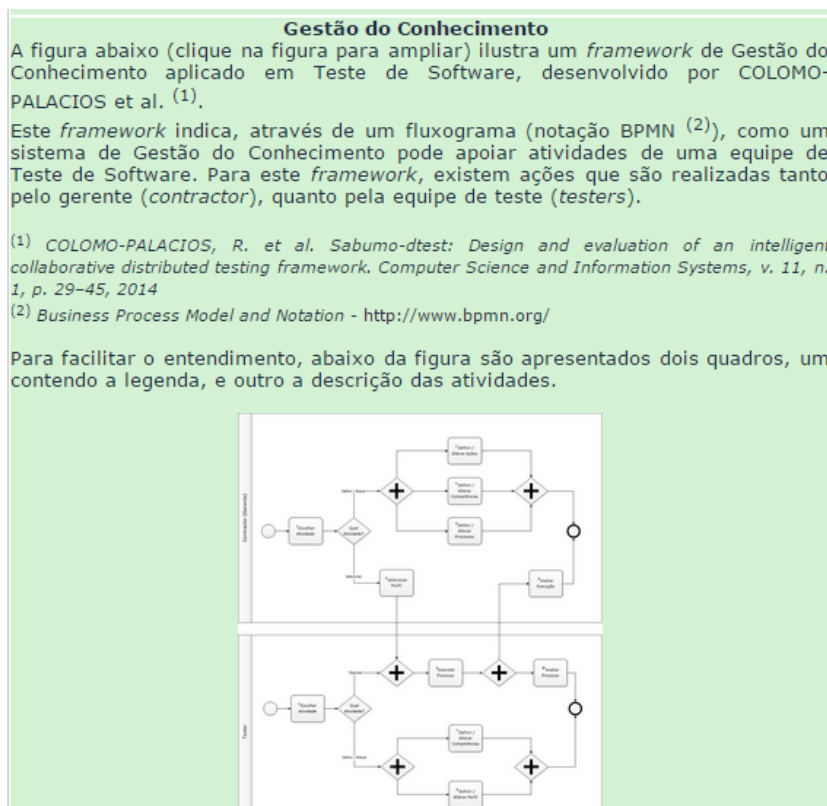
Para a apresentação do processo de teste, foi disponibilizado um arquivo PDF, o qual poderia ser salvo na máquina do participante para posterior consulta, contendo as cinco fases do processo de teste e as 33 práticas do TMMi relacionadas a cada uma das fases. Uma tabela também foi disponibilizada contendo as informações detalhadas das práticas do TMMi como, por exemplo, os níveis, as fases, e a descrição de cada uma. A apresentação do processo de teste foi replicada nas três versões do *Survey*, pois todas se basearam no mesmo processo de teste. O processo de teste presente no arquivo disponibilizado é visualizado na Seção 2.3.1, Figura 2.5, e a tabela é encontrada no Anexo A, Tabela A.3.

A parte variante da segunda etapa corresponde à apresentação do *processo* de Gestão do Conhecimento, pois cada uma das versões do *Survey* tratou de um *processo* diferente.



A seguir, as fases do processo são detalhadas, e na tabela que segue, as particularidades de cada uma das práticas são informadas, como por exemplo, o

(a) Processo de Teste de Software



(b) Processo de Gestão do Conhecimento

Figura 4.2: Página de Apresentação das Teorias

Nessa etapa, também foi disponibilizado um arquivo PDF contendo o fluxograma da abordagem de Gestão do Conhecimento correspondente à versão do *Survey* (A, B ou C), uma legenda para melhor entendimento da notação BPMN, e a descrição breve de cada uma das atividades presentes no fluxograma. Os fluxogramas, que deste ponto em diante são chamados de A, B e C, são vistos na Seção 3.2, Figuras 3.2, 3.3 e 3.4, respectivamente.

A terceira etapa foi composta pelas questões de avaliação do *processo*. O participante analisou, de acordo com sua experiência, a aplicabilidade do *processo* a cada uma das práticas do TMMi que compõem o processo de teste. O intuito foi examinar as práticas individualmente, tornando possível determinar para quais práticas o *processo* em questão poderia fornecer maior apoio à sua realização.

As práticas foram agrupadas em suas respectivas fases do processo de teste. Para cada grupo de práticas, a questão de avaliação utilizada foi: “Qual a aplicabilidade do processo de Gestão do Conhecimento em cada uma das práticas relacionadas à fase X?”, na qual X corresponde a cada uma das fases do processo de teste: *Planejamento*; *Projeto de Casos de Teste*; *Configuração de Dados e do Ambiente de Teste*; *Execução e Avaliação do Teste*; e *Monitoramento e Controle*. Um trecho da tela da terceira etapa pode ser visualizado na Figura 4.3.

	Extremamente aplicável	Aplicável	Pouco aplicável	Não aplicável
SP1.2 Identificar e priorizar casos de teste	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SP1.3 Identificar dados de teste específicos necessários	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Comentários

Figura 4.3: Página das Questões de Avaliação

Como resposta à questão definida, o participante deveria escolher um dos itens da escala apresentada. Para isso, uma escala *Likert* de quatro graus (composta por: *Extremamente Aplicável*, *Aplicável*, *Pouco Aplicável* e *Não Aplicável*) foi utilizada (a mesma escala descrita na Seção 3.3.1, Análise Crítica). Em cada grupo de práticas, o participante poderia, ainda,

adicionar comentários livres sobre as questões.

Na Figura 4.4 é possível ver o esquema que representa as etapas de cada versão criada do *Survey*.

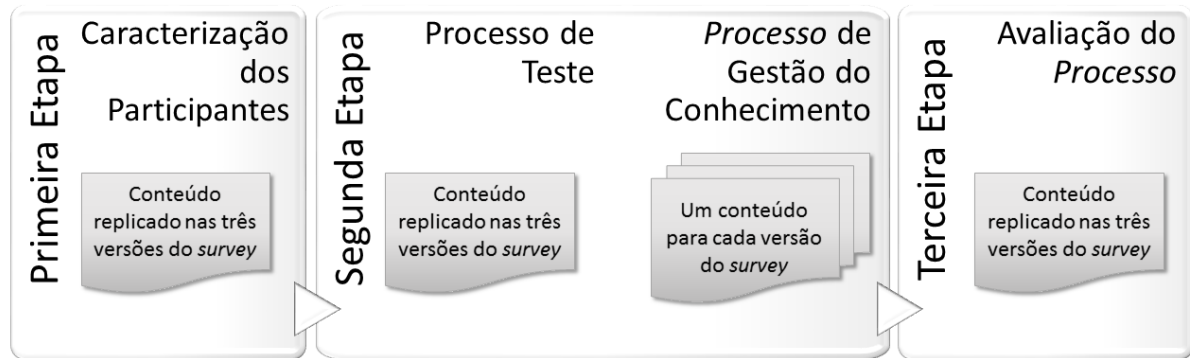


Figura 4.4: Representação das Etapas do *Survey*

4.3 Análise de Dados

Nesta seção serão apresentados os resultados do *Survey*. A seção é dividida em dois estágios. O primeiro consiste na caracterização dos participantes e na identificação dos perfis existentes. O segundo estágio seleciona a amostra que será analisada, conforme a determinação dos perfis, e explora seus resultados.

4.3.1 Amostra e Caracterização dos Perfis

Os *links* dos questionários foram distribuídos entre grupos de discussão brasileiros sobre teste de software. Além disso, os *links* foram enviados diretamente a alguns profissionais da área de teste. Os participantes foram solicitados a divulgar o estudo em sua rede de contatos. No total foram obtidas 43 respostas, das quais 16 relacionadas à versão A, 14 à versão B, e 13 à versão C.

De acordo com as questões de caracterização pertencentes aos questionários, a distribuição de respostas é mostrada nos gráficos da Figura 4.5. Pode-se observar alto grau de participação de pessoas provenientes da indústria; da totalidade de respondentes, apenas 3 não possuem experiência com teste de software em empresas (Figura 4.5(C)).

Devido a este fato, é natural que dos 43 participantes, 28 possuam no máximo a titulação

de nível superior (65% aproximadamente), e que apenas 6 dos respondentes (14% aproximadamente) possuam título de Mestre ou Doutor (Figura 4.5(A)).

Isto corrobora o baixo envolvimento com a pesquisa, sendo que 72% dos participantes (31 no total) nunca estiveram envolvidos com pesquisa acadêmica (Figura 4.5(E)). O número de docentes participantes também foi reduzido; apenas 5 possuem alguma experiência em docência (11% aproximadamente), conforme ilustrado na Figura 4.5(F). Tal cenário era esperado, devido aos locais de divulgação da pesquisa e ao interesse desta pesquisa em focar na opinião de profissionais da indústria.

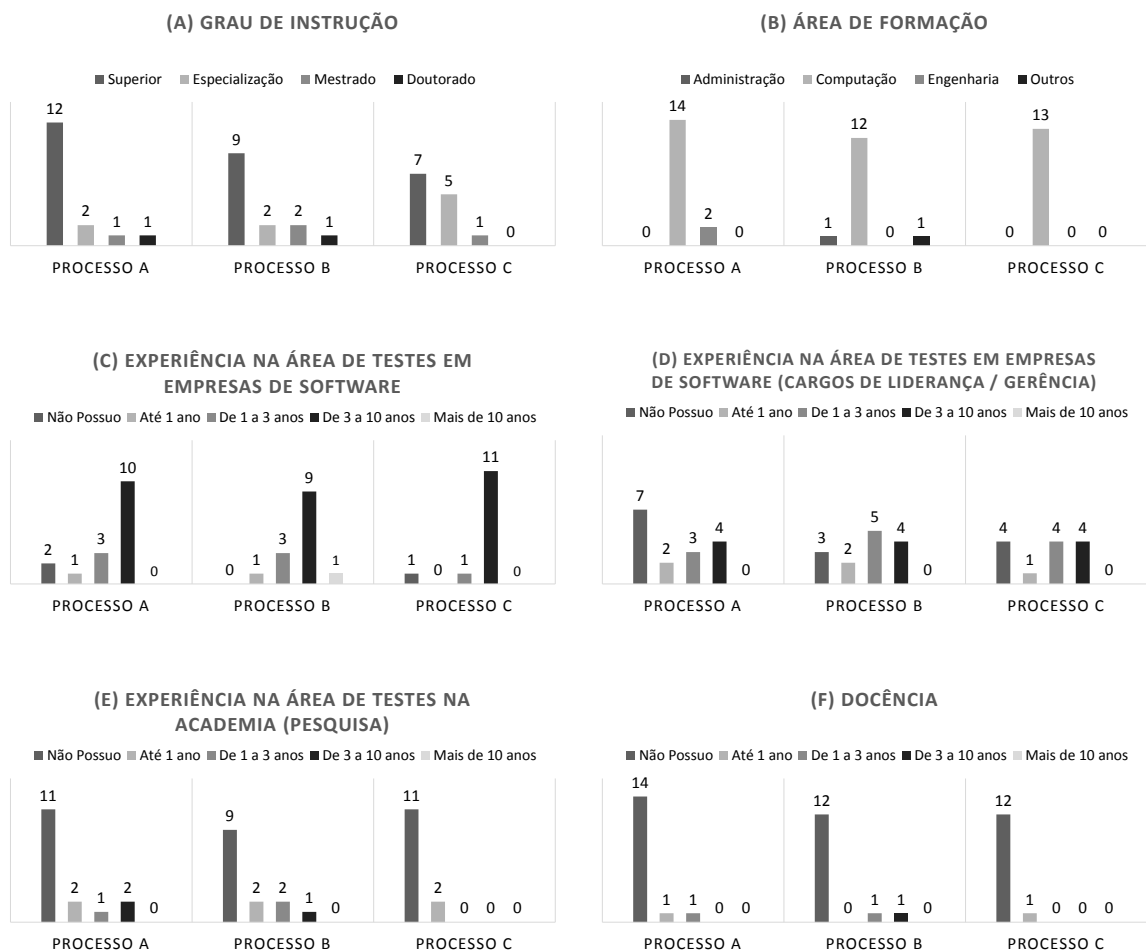


Figura 4.5: Caracterização dos Participantes

De acordo com a Figura 4.5(B), a área de formação em Computação foi a que mais teve colaboradores: 39 participantes, o que representa aproximadamente 90% do total. Dentre as demais áreas, houve 2 Engenheiros, 1 Administrador, e 1 opção marcada como *outros*, que através do questionário pôde-se determinar que se tratava do curso Sistemas de Informação.

Pela amostra caracterizada, pode-se afirmar que a maioria dos participantes que trabalham na indústria, na área de teste de software, possuem formação de nível superior em Computação.

Existe grande interesse por parte desta pesquisa em saber a opinião de pessoas que atuam em nível gerencial. Devido a isso, também foi explicitado no questionário se cada participante possuía experiência em gestão na área de teste de software (cargos de liderança/gerência). Como resultado, 68% dos participantes (29 no total) informaram ter alguma experiência em cargos de liderança e gerência (Figura 4.5(D)). Dentre estes, 12 (28% do total de participantes) relataram possuir mais de três anos de atuação como líderes ou gerentes. Configura-se, dessa forma, uma amostra relevante para o propósito de captar a opinião de gestores.

Além da caracterização apresentada na Figura 4.5, gráficos de Venn foram criados para agrupar os participantes conforme seu perfil de experiência. É importante ressaltar que nesse tipo de gráfico não há distinção entre o nível de experiência. Ou seja, caso dois participantes declarem ter experiência em docência, porém com faixas de tempo diferentes, ambos serão contabilizados de maneira igual na esfera da docência. Os gráficos de Venn são exibidos nas Figuras 4.6, 4.7, 4.8, e 4.9. O primeiro gráfico mostra todos os respondentes (independentemente da versão do questionário respondido). Os gráficos subsequentes mostram a divisão por versão de questionário: *Processo A* (COLOMO-PALACIOS et al., 2014), *Processo B* (ANDRADE et al., 2013), e *Processo C* (XUE-MEI et al., 2009).

Os gráficos de Venn são compostos por quatro esferas: *Empresas*, *Gestão*, *Pesquisa*, e *Docência*. Cada esfera corresponde a uma pergunta de caracterização de perfil presente no questionário.

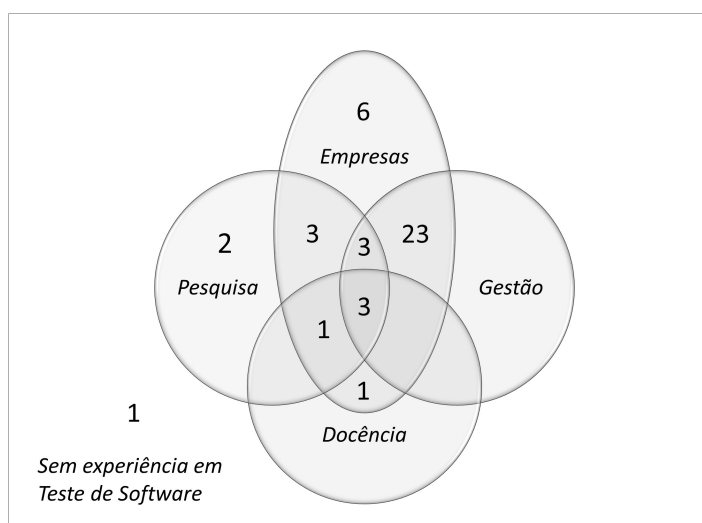


Figura 4.6: Participantes - GERAL

A esfera *Empresas* está relacionada à questão *Experiência na área de testes em empresas*

que desenvolvem software. Estão incluídos nessa esfera aqueles que declararam possuir alguma experiência nessa questão. A esfera *Gestão* corresponde à pergunta *Experiência na área de testes em empresas que desenvolvem software (cargos de liderança/gerência)*. A esfera *Pesquisa e Docência* representa a pergunta *Experiência na área de testes na academia (pesquisa)*, e a esfera *Docência* retrata as respostas obtidas pela questão *Docência*.

Com a apresentação dos participantes em um gráfico de Venn é possível identificar perfis que surgem observando-se a intersecção das esferas. Percebe-se que houve um respondente no questionário C (Figura 4.9) que não possui qualquer experiência em Teste de Software.

Devido ao interesse em obter a opinião de profissionais com experiência em teste de software na indústria, com especial atenção à gestão nessa área, foram selecionadas para análise as respostas dos perfis que se encaixaram na seguinte intersecção: *Empresas e Gerência*. *Empresas*, pois é desejável que o profissional tenha trabalhado na área de testes em empresas que desenvolvem software; e *Gestão*, pois é importante que o profissional tenha atuado em cargos de liderança e gestão. A indicação de tais perfis pode ser visualizada nas Figuras 4.7, 4.8, e 4.9, na área mais escura dos gráficos.

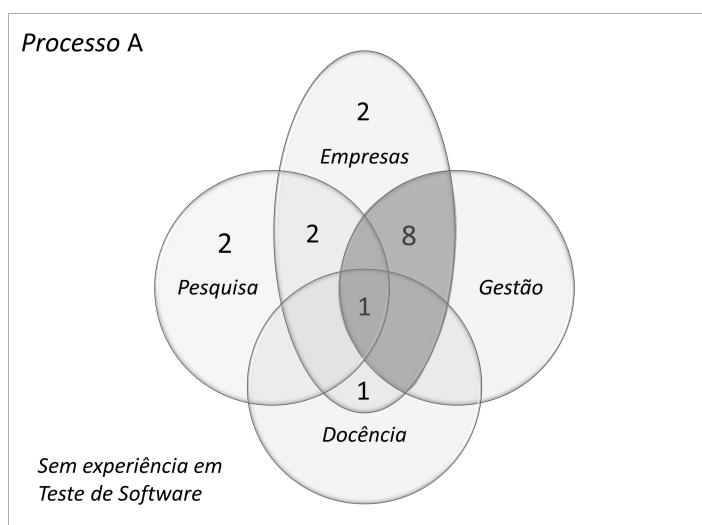


Figura 4.7: Participantes - Processo A

Na próxima seção serão discutidos os resultados obtidos a partir da seleção de respostas dos usuários que se encaixam no perfil de *Empresas e Gestão*.

4.3.2 Análise dos Resultados

As Figuras 4.10, 4.11 e 4.12 apresentam a frequência de respostas obtidas em cada uma das três versões do questionário, considerando o perfil de *Empresas e Gestão* mencionado na

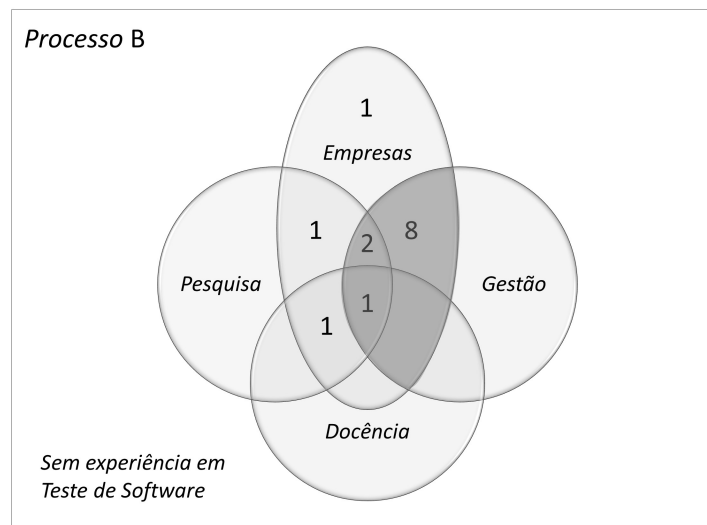


Figura 4.8: Participantes - Processo B

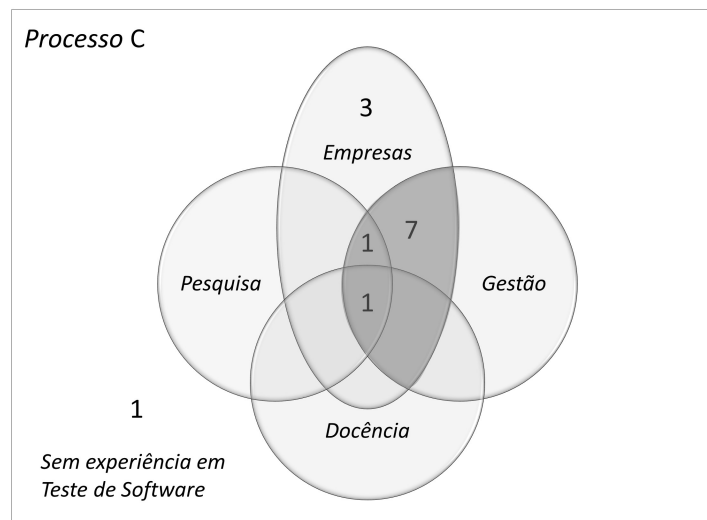


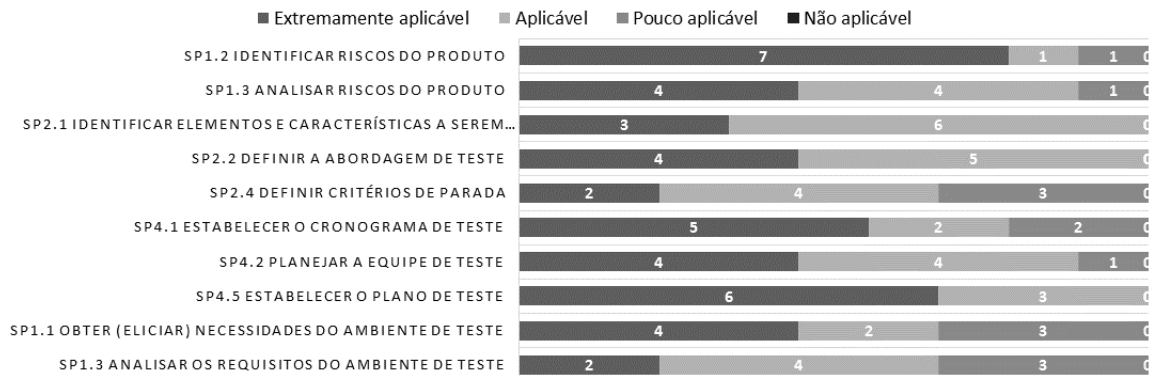
Figura 4.9: Participantes - Processo C

seção anterior. As práticas foram agrupadas em suas respectivas fases do processo para melhor visualização. As tabelas de frequência com as respostas de todos os participantes podem ser visualizadas no anexo B (Tabelas B.1, B.2 e B.3).

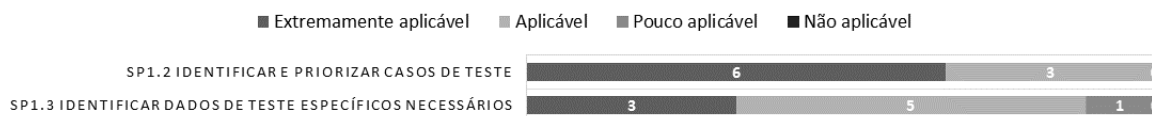
A soma dos valores anotados nas barras correspondem à totalidade de respostas em cada prática: 9 para os *Processos A e C*, e 11 para o *Processo B*. Cada item da escala é indicado por um tom de cinza diferente, e seu tamanho é proporcional à quantidade de respostas obtidas. As barras também possuem a frequência das respostas. Quando em uma determinada prática um item da escala não obteve nenhuma resposta, a barra correspondente não é exibida, e o número zero é indicado entre as barras vizinhas.

PROCESSO A

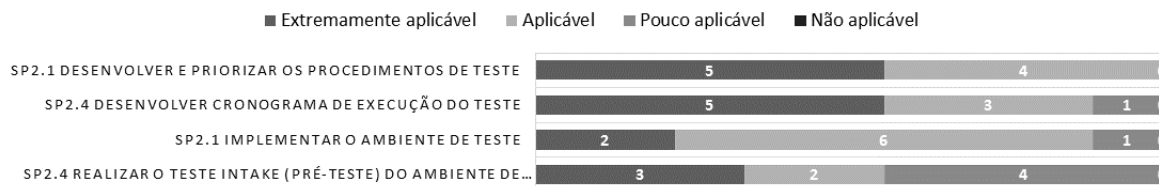
PLANEJAMENTO



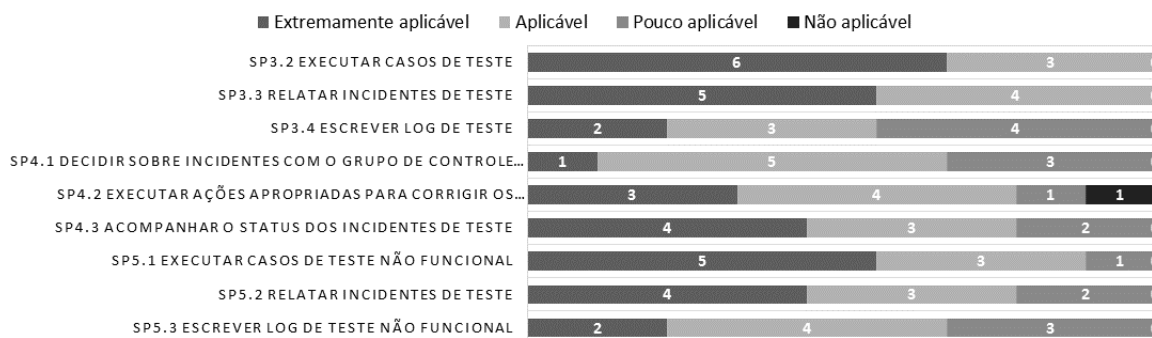
PROJETO DE CASOS DE TESTE



CONFIGURAÇÃO DE DADOS E DO AMBIENTE DE TESTE



EXECUÇÃO E AVALIAÇÃO DO TESTE



MONITORAMENTO E CONTROLE

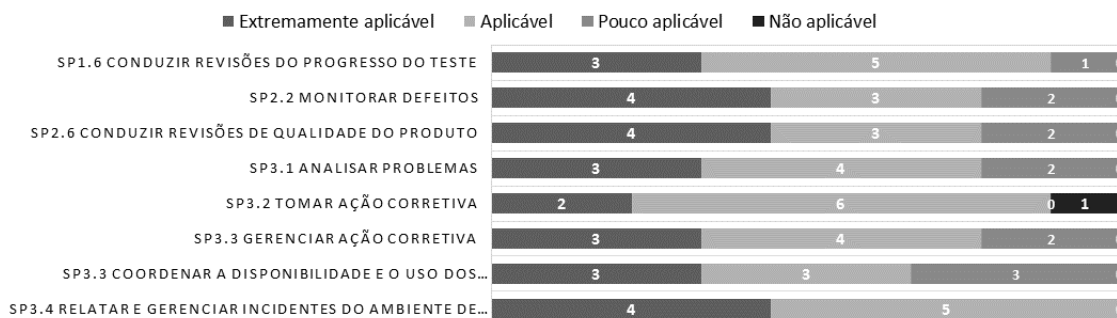
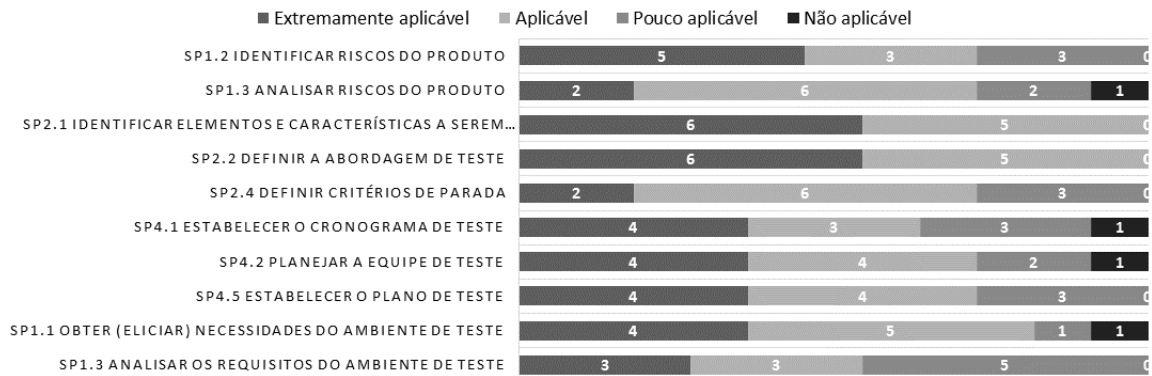


Figura 4.10: Processo A

PROCESSO B

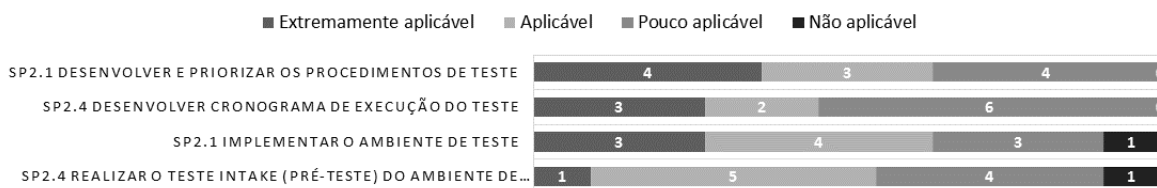
PLANEJAMENTO



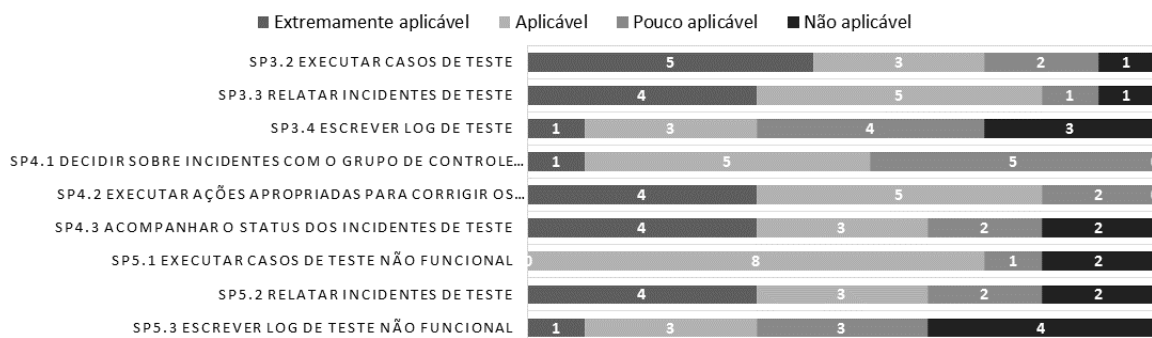
PROJETO DE CASOS DE TESTE



CONFIGURAÇÃO DE DADOS E DO AMBIENTE DE TESTE



EXECUÇÃO E AVALIAÇÃO DO TESTE



MONITORAMENTO E CONTROLE

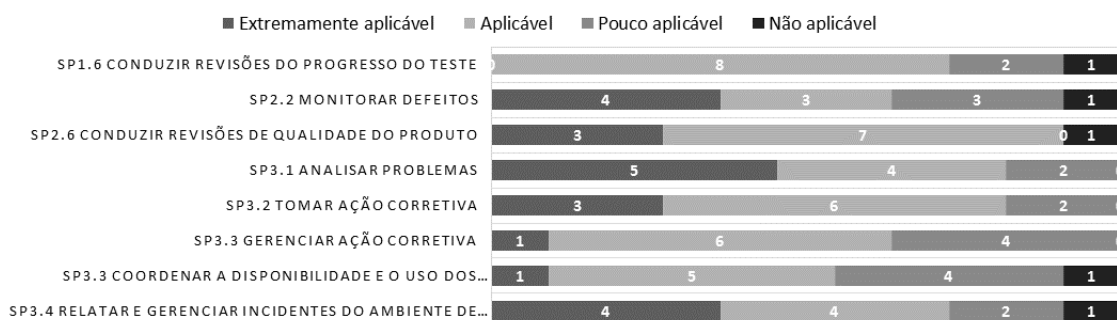
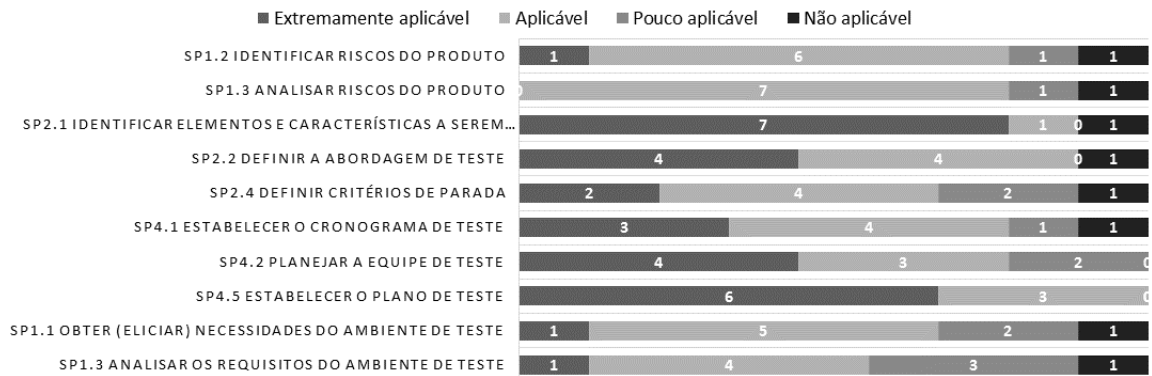


Figura 4.11: Processo B

PROCESSO C

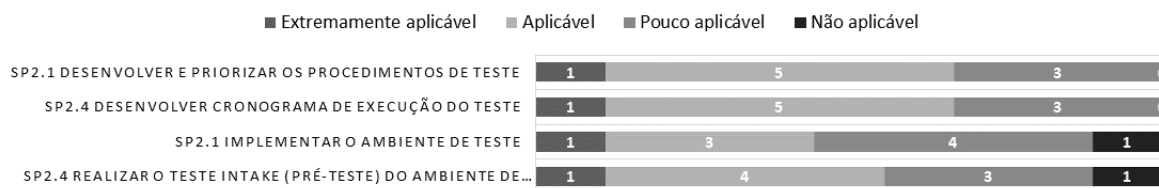
PLANEJAMENTO



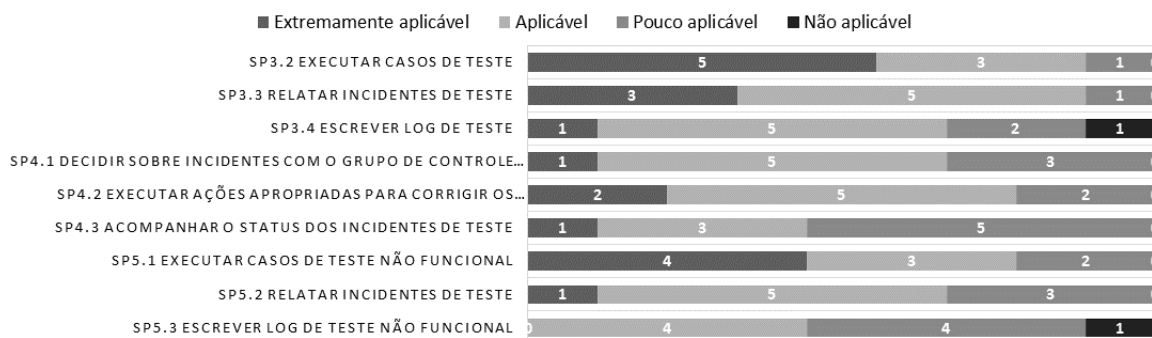
PROJETO DE CASOS DE TESTE



CONFIGURAÇÃO DE DADOS E DO AMBIENTE DE TESTE



EXECUÇÃO E AVALIAÇÃO DO TESTE



MONITORAMENTO E CONTROLE

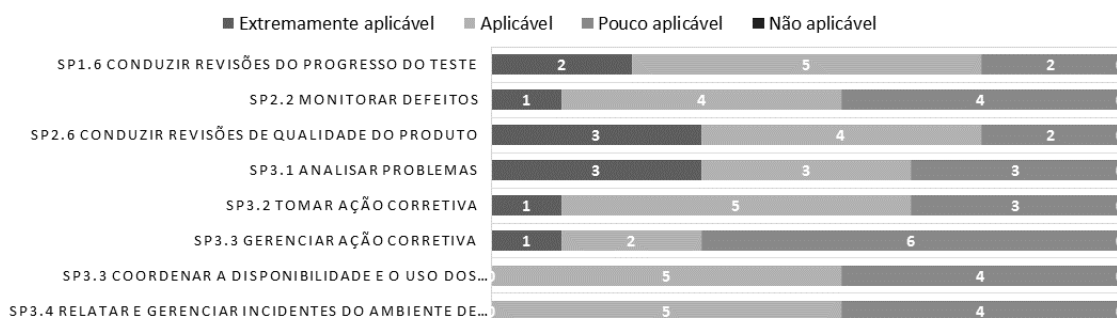


Figura 4.12: Processo C

Para a análise dos dados obtidos, alguns testes estatísticos não-paramétricos foram experimentados, pois a amostra não seguia uma distribuição de probabilidade conhecida. Contudo, em razão do pequeno número de dados considerados neste *Survey*, nenhum teste estatístico se mostrou adequado a aplicação. Portanto, optou-se pelo uso da mediana como uma medida que representasse os dados da pesquisa.

A mediana é uma medida de tendência central, que tem como característica ser robusta, pois depende apenas da posição, não sendo influenciada pelos valores dos dados. A mediana divide um grupo ao meio, separando-o em duas partes iguais, com base na ordenação dos elementos. Desta forma, 50% dos elementos terão valor igual ou maior que a mediana (COSTA NETO, 1977).

Assim como na Análise Crítica apresentada no Capítulo 3, houve a necessidade de transformar a escala nominal utilizada em numeral. Cada grau da escala recebeu um valor de 1 (um) a 4 (quatro), no qual o valor 1 corresponde a *Não aplicável*, 2 corresponde a *Pouco aplicável*, 3 corresponde a *Aplicável* e 4 corresponde a *Extremamente aplicável*.

Os dados de cada uma das práticas foram ordenados, e a mediana calculada. As Figuras 4.13, 4.14 e 4.15 exibem os resultados obtidos com o cálculo da mediana para cada *processo*.

As raias são numeradas de 1 a 33, correspondendo às práticas do processo de teste considerado (CAMARGO, 2012; CAMARGO; FERRARI; FABBRI, 2013). Os círculos representam as notas obtidas por cada prática (de 1 a 4, conforme a escala apresentada). O centro corresponde à nota 1, e o círculo mais externo à nota 4.

Observa-se que o *Processo A* obteve uma avaliação melhor em comparação aos *Processos B* e *C*, com nove notas atingindo o valor máximo, ou seja, para 27% das práticas o *processo* foi considerado *Extremamente aplicável* (nota 4), e nenhuma nota abaixo do valor 3 (correspondente a *Aplicável* na escala).

A avaliação do *Processo B* esteve concentrada no valor 3 com pouca variação. Esse *processo* foi avaliado com essa nota (isto é, *Aplicável*) para aproximadamente 85% das práticas do processo de teste abordado; somente para 3 práticas o *processo* foi avaliado como *Extremamente aplicável*, enquanto que para 2 práticas ele foi avaliado como *Pouco aplicável*.

Por fim, a avaliação do *Processo C* apresentou maior variação de notas, particularmente quando comparado com os outros *processos*. Das 33 práticas, o *processo* foi considerado *Aplicável* em 24 delas (aproximadamente 73% das práticas). Também obteve 5 notas máximas e 4 notas abaixo do valor 3.

A relação das práticas é encontrada na Tabela A.2.

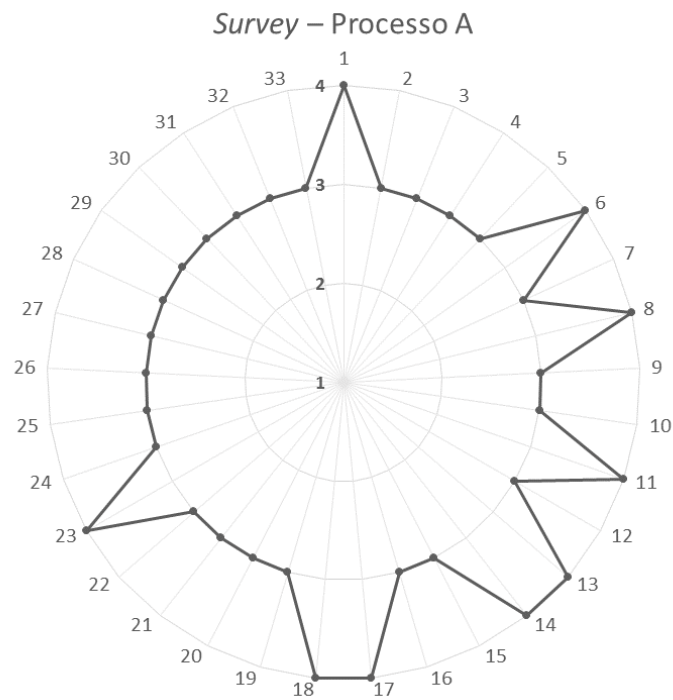


Figura 4.13: Mediana *Processo A*

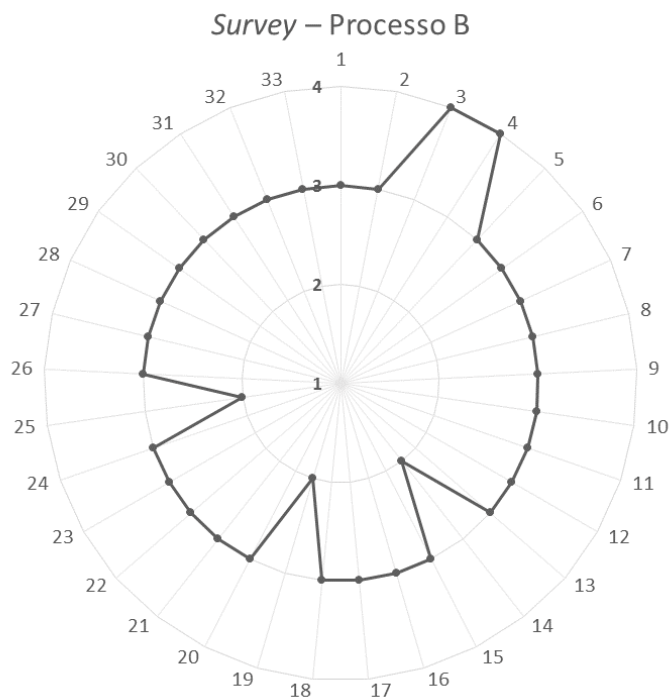


Figura 4.14: Mediana *Processo B*

Além da comparação das medianas calculadas para cada uma das práticas, os *processos* também foram investigados quanto às suas fases. Para isso, as práticas foram agrupadas de acordo com as fases do processo de teste, e conforme este agrupamento, foi calculada a média aritmética das medianas. Este processo foi realizado para estar de acordo com a metodologia

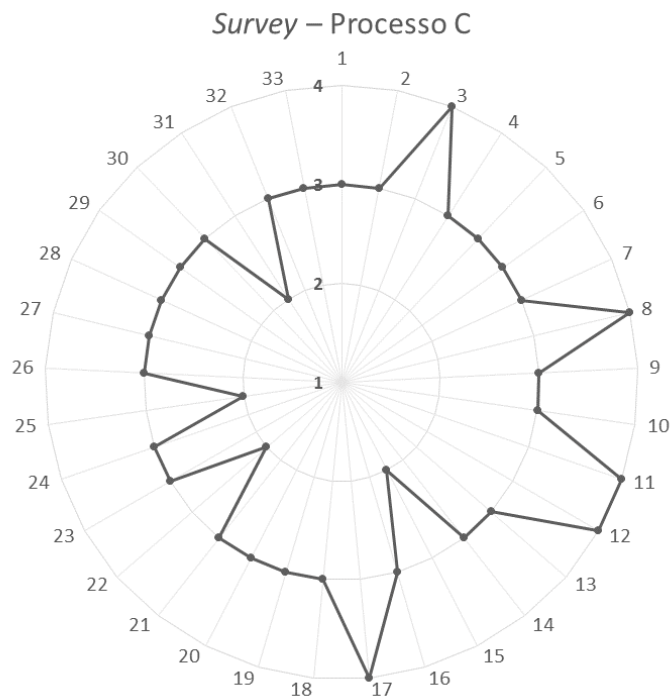


Figura 4.15: Mediana *Processo C*

utilizada na Análise Crítica, que também fez uso da média aritmética para calcular as notas das fases. O resultado é apresentado na Figura 4.16.

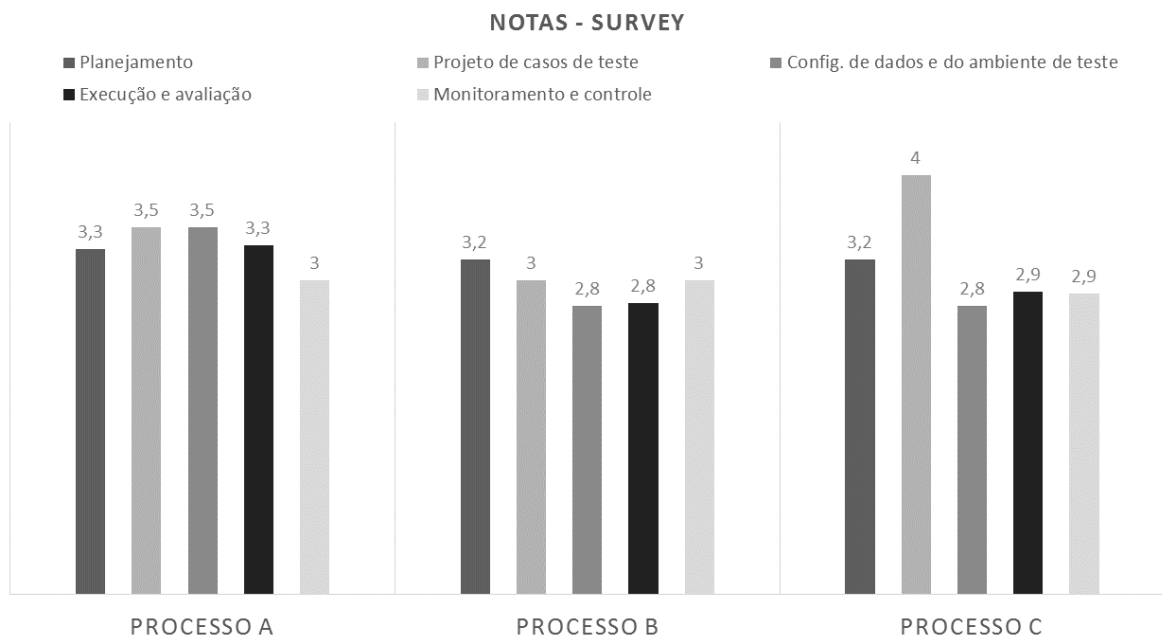


Figura 4.16: Nota *Survey* Agrupada por Fase do Processo de Teste

Planejamento

Os resultados da avaliação dos *processos* com relação às práticas da fase de *Planejamento* foram semelhantes, com uma ligeira vantagem do *Processo A*. As notas ficaram entre os valores *Extremamente Aplicável* (nota 4) e *Aplicável* (nota 3). Na seção de comentários do *Survey*, um dos participantes reforçou a importância da aplicação de Gestão do Conhecimento, dizendo que melhora as atividades de planejamento, devido à necessidade de elaborar estratégias consistentes para os projetos. Ressalta-se ainda que nenhum *processo* foi avaliado com nota abaixo de 3 (isto é, *Aplicável*) nas práticas da fase de *Planejamento*.

Com relação a essa fase, os *processos* foram avaliados com nota máxima para as seguintes práticas:

- *Processo A*:
 - SP1.2 Identificar riscos do produto
 - SP4.1 Estabelecer o cronograma de teste
 - SP4.5 Estabelecer o plano de teste

- *Processo B*:
 - SP2.1 Identificar elementos e características a serem testados
 - SP2.2 Definir a abordagem de teste

- *Processo C*:
 - SP2.1 Identificar elementos e características a serem testados
 - SP4.5 Estabelecer o plano de teste

Percebe-se que os *Processos A* e *B* atendem muito bem, na opinião dos participantes, a atividades diferentes. No entanto, as duas práticas nas quais o *Processo C* recebeu as maiores notas estavam presentes também nas listas de A e B. São elas: ***Identificar elementos e características a serem testados (SP2.1)*** (*Processos B* e *C*), atividade que indica o que deve e o que não se deve ser testado; e ***Estabelecer o plano de teste (SP4.5)*** (*Processos A* e *C*), que está relacionada a definir o *baseline* do plano de teste.

O *Processo A* foi o único considerado *Extremamente Aplicável* à atividade de análise de risco. Esse fato pode ser um diferencial, pois trata-se de uma atividade que pode antecipar os possíveis problemas em um projeto.

Projeto de casos de teste

Os *processos* obtiveram uma boa avaliação na fase *Projeto de casos de teste*, pois nenhuma nota menor que 3 (*Aplicável*) foi atribuída com relação às práticas dessa fase. As notas ficaram entre 3 e 4, com diferença exata de meio ponto entre os *processos*. Ressalta-se que a fase *Projeto de casos de teste* possui apenas duas práticas, e para todos os *processos* a avaliação resultou nas maiores notas.

As práticas para as quais os *processos* foram avaliados com nota máxima para a fase *Projeto de casos de teste* foram:

- *Processo A*:
 - SP1.2 Identificar e priorizar casos de teste

- *Processo C*:
 - SP1.2 Identificar e priorizar casos de teste
 - SP1.3 Identificar dados de teste específicos necessários

Os *Processos A* e *C* obtiveram nota máxima em uma prática em comum: ***Identificar e priorizar casos de teste (SP1.2)***. Ao passo que o *Processo C* recebeu nota máxima para as duas práticas dessa fase, o *Processo B* recebeu nota 3 (*Aplicável*) para ambas.

Dentre os três *processos*, o *C* foi o mais bem avaliado para a fase *Projeto de casos de teste*.

Configuração de dados e do ambiente de teste

Configuração de dados e do ambiente de teste é a primeira fase na qual há distância de mais de meio ponto entre os *processos*. Trata-se de uma fase com apenas quatro práticas e, similarmente à fase *Projeto de casos de teste*, sensível ao cálculo da média, ou seja, valores extremos possuirão maior influência no resultado final.

As práticas nas quais os *processos* foram avaliados com nota máxima para a fase *Configuração de dados e do ambiente de teste* foram:

- *Processo A*:
 - SP2.1 Desenvolver e priorizar os procedimentos de teste
 - SP2.4 Desenvolver cronograma de execução do teste

Nenhum *processo* foi avaliado com nota 1 (*Não Aplicável*) nas práticas dessa fase. Esta é a primeira fase na qual os participantes do *Survey* consideraram que um *processo* é *Pouco Aplicável* com relação às práticas, ou seja, recebeu nota 2.

As práticas da fase de *Configuração de dados e do ambiente de teste* nas quais os *processos* foram considerados pouco aplicáveis foram:

- *Processo B*:
 - SP2.4 Desenvolver cronograma de execução do teste
- *Processo C*:
 - SP2.1 Implementar o ambiente de teste

Um ponto a se considerar para essa fase do processo de teste é a prática ***Desenvolver cronograma de execução do teste (SP2.4)***, para a qual o *Processo A* recebeu nota máxima, porém com nota 2 no caso do *Processo B*. Pode-se inferir, neste caso, que para a definição do cronograma de execução, a utilização do *Processo B* não trará benefícios, o contrário do que se espera do *Processo A*. Ainda assim, com a diferença de mais de meio ponto entre os *processos*, todos estão com as médias adjacentes ao valor *Aplicável* da escala.

Dentre os três *processos*, o mais bem avaliado para a fase *Configuração de dados e do ambiente de teste* foi o *A*.

Execução e avaliação do teste

A fase *Execução e avaliação do teste* revela uma disparidade entre a avaliação dos *processos*. Enquanto o *Processo A* possui três práticas para as quais é *Extremamente Aplicável* e para todas as demais esse *processo* foi considerado *Aplicável*, cada um dos *Processos B* e *C* foi considerado *Pouco Aplicável* para duas práticas dessa fase.

Apesar dessa disparidade, a fase contempla o segundo maior grupo de práticas (nove, no total), deixando-a menos sensível a estas diferenças em relação a média final, ou seja, a maior quantidade de práticas faz com que a diferença de notas seja mais equilibrada.

Com relação à fase de *Execução e avaliação do teste*, os *processos* foram avaliados com nota máxima para as seguintes práticas:

- *Processo A*:

- SP3.2 Executar casos de teste
 - SP3.3 Relatar incidentes de teste
 - SP5.1 Executar casos de teste não funcional
- *Processo C:*
 - SP3.2 Executar casos de teste

Nenhum *processo* foi avaliado com nota 1 (*Não Aplicável*) nas práticas dessa fase, portanto serão considerados como mal avaliados os *processos* cujas práticas obtiveram nota 2 (*Pouco Aplicável*). Essa avaliação ocorreu para os *processos* nas seguintes práticas:

- *Processo B:*
 - SP3.4 Escrever log de teste
 - SP5.3 Escrever log de teste não funcional
- *Processo C:*
 - SP4.3 Acompanhar o status dos incidentes de teste
 - SP5.3 Escrever log de teste não funcional

A avaliação dos *processos* dessa fase se mostra, para alguns casos, coerente, porém alguns pontos de divergência. Nessa fase de *Execução e avaliação do teste* há a replicação de um grupo práticas, que possuem nomes parecidos, mas objetivos diferentes: umas relacionadas à execução de testes funcionais, e outras à execução de testes não funcionais. São elas: ***Executar casos de teste (SP3.2)*** e ***Executar casos de teste não funcional (SP5.1)***; ***Escrever log de teste (SP3.4)*** e ***Escrever log de teste não funcional (SP5.3)***.

À vista disso, o mais sensato seria avaliar os *processos* de maneira semelhante para tais práticas, cujas atividades são similares sob seus pontos de vista. Isso foi observado no caso das práticas ***Executar casos de teste (SP3.2)*** e ***Executar casos de teste não funcional (SP5.1)***, para as quais o *Processo A* foi considerado *Extremamente Aplicável*, e no caso das práticas ***Escrever log de teste (SP3.4)*** e ***Escrever log de teste não funcional (SP5.3)***, para as quais o *Processo B* foi considerado *Pouco Aplicável*. No entanto, não foi o caso do *Processo C*, que não obteve notas semelhantes nessas duas duplas de práticas.

Dentre os três *processos*, o mais bem avaliado para a fase de *Execução e avaliação do teste* foi o *Processo A*.

Monitoramento e controle

A fase do processo *Monitoramento e controle* foi a que obteve uma avaliação mais equilibrada dentre todas as fases. Os *processos* receberam nota 3 (*Aplicável*) para todas as práticas do processo de teste, com exceção de uma avaliação do *Processo C*, referente à prática **Gerenciar ação corretiva (SP3.3)**, tendo recebido nota 2 (*Pouco Aplicável*).

Portanto, é natural que as médias dos *processos* estejam tão próximas, com apenas um décimo de diferença.

4.3.3 Avaliação Geral

Nesta seção primeiramente serão averiguados os resultados obtidos no *Survey* de acordo com a relação de dependências entre as práticas. Mais adiante, as notas gerais agrupadas por *processo* serão apresentadas, além de uma breve comparação dos resultados gerais do *Survey*, com e sem a seleção dos perfis *Empresas e Gestão*, com o intuito de analisar as diferenças obtidas.

Dependências Entre as Práticas

Assim como na Análise Crítica, uma das preocupações com os resultados da pesquisa esteve relacionada às dependências que ocorrem entre as práticas. Tais dependências recomendam que certas práticas devem ser implementadas antes de outras (HÖHN, 2011). A indicação das dependências pode ser vista na Figura 4.17.

Na Figura 4.17 também são exibidas as frequências de respostas de cada um dos *processos*, assim como suas medianas. Assim, é possível determinar se há coerência quanto à aplicabilidade do mesmo, pois se existe alguma dependência, é esperado que as práticas sejam (ou não) atendidas pelo *processo* de maneira semelhante.

Não houve variação significativa quanto as medianas calculadas para cada um dos *processos*. Por exemplo, para todas as práticas indicadas na Figura 4.17 a mediana do *Processo A* esteve entre os valores 3 e 4. Isso pode também ser observado nos outros dois *processos*.

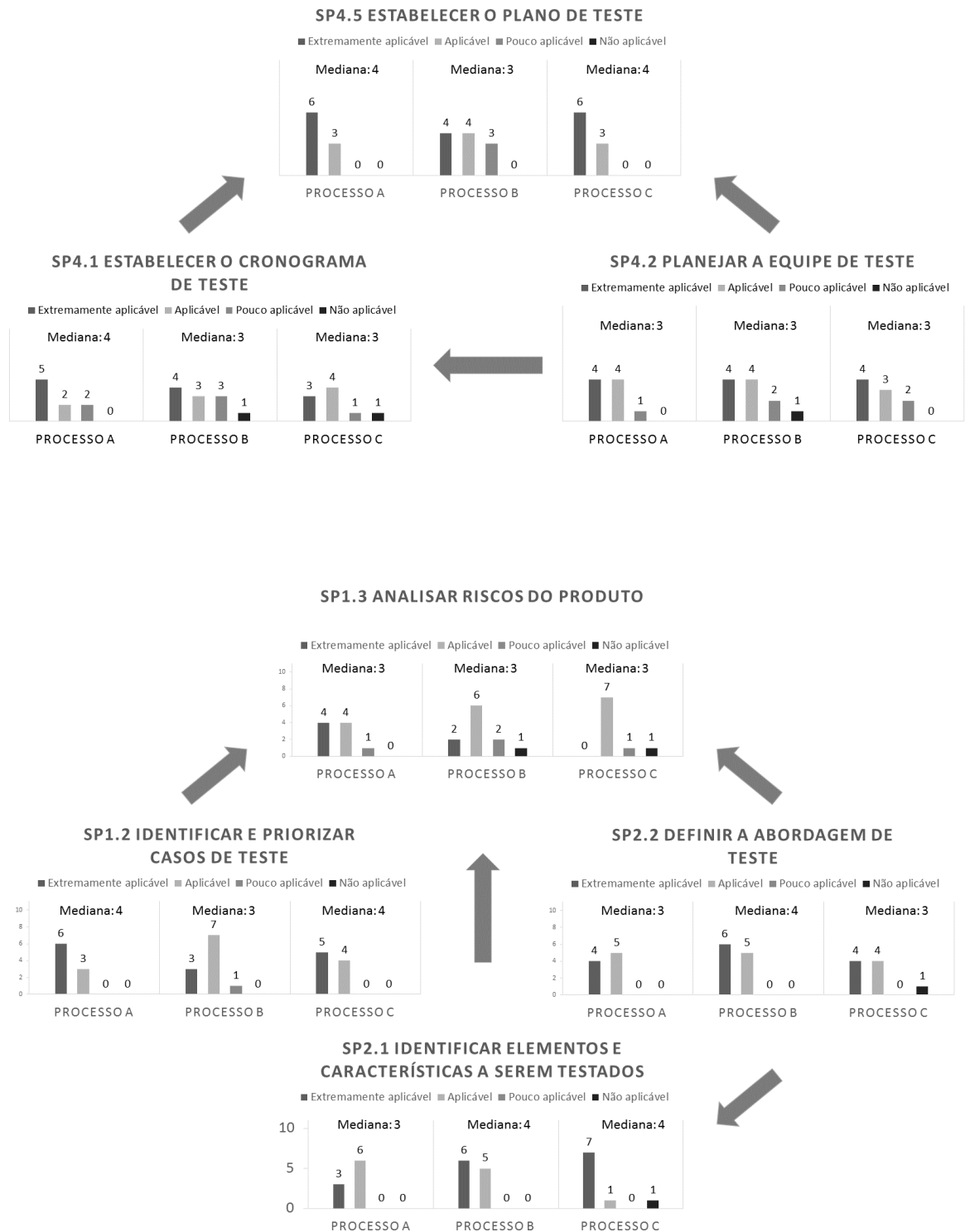


Figura 4.17: Dependências Entre as Práticas

Avaliação Geral

Se a avaliação dos *processos* for levada em consideração nas fases do processo de teste, distintamente, seria necessário adotar um *processo* diferente em cada uma das fases. Por isso, é necessário verificar qual deles se adapta melhor ao processo de teste como um todo. Ou seja, deve-se buscar um *processo* que, no geral, possa ser útil em todas as práticas implementadas, em qualquer uma das fases.

Sendo assim, as notas obtidas pelos *processos* em cada fase do processo de teste foram utilizadas para calcular, também com base na média aritmética, suas notas finais. Os resultados são ilustrados na Figura 4.18.

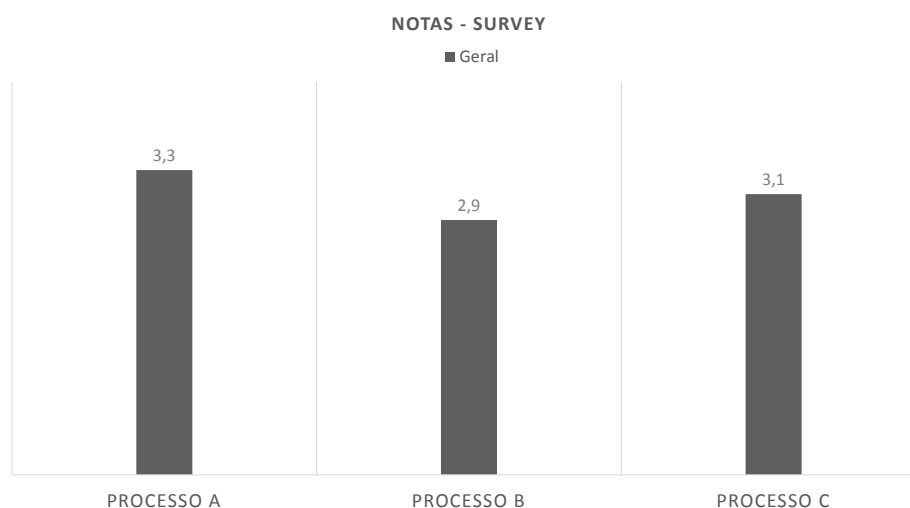


Figura 4.18: Nota do *Survey* Agrupada por *Processo*

O resultado final do *Survey* determina que o *Processo A* foi aquele que, no geral, esteve mais aplicável às práticas do processo de teste. Ressalta-se que esse resultado está alinhado com o obtido na Análise Crítica apresentada no Capítulo 3.

Como mencionado na Seção 4.3.1, os dados foram analisados conforme a seleção das respostas que se enquadraram nos perfis *Empresas* e *Gestão*. Por motivo de comparação, a seguir apresentam-se os resultados obtidos quando todos os participantes do *Survey* são considerados. Os resultados são sumarizados na Figura 4.19, incluindo todas as práticas do processo de teste de software.

Ao confrontar os gráficos da Figura 4.19 com os resultados apresentados ao longo deste capítulo, é possível notar melhora na avaliação de todos os *processos*, principalmente quanto à quantidade de notas baixas.

Nas Figuras 4.20 e 4.21 apresenta-se a avaliação dos *processos* com as práticas agrupadas

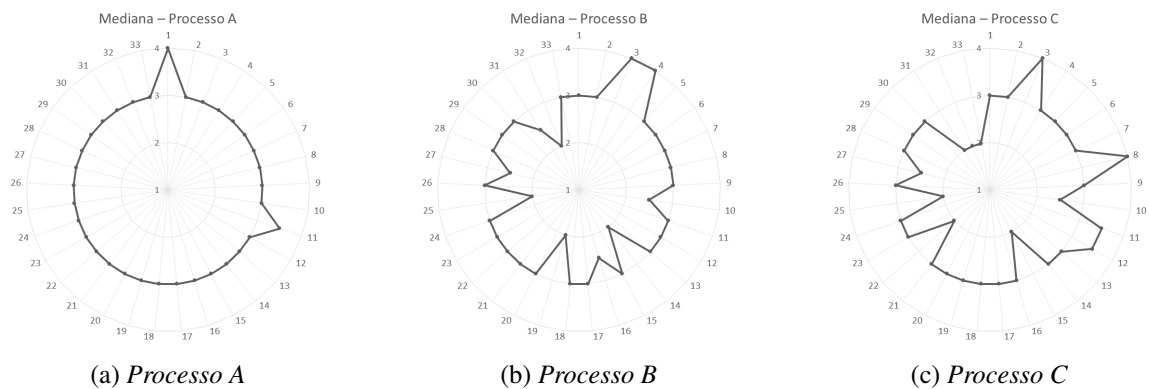


Figura 4.19: Notas Obtidas Pelos processos Sem a Seleção dos Perfis

por fase e por *processo*, respectivamente. Assim como os resultados obtidos na Análise Crítica e no *Survey*, o *Processo A* foi aquele com melhor avaliação. A diferença de resultado se dá nos *Processos B* e *C*, que para este caso, possuem notas finais mais próximas quando as fases são consideradas individualmente, e média geral idêntica, com o valor de 2,88.

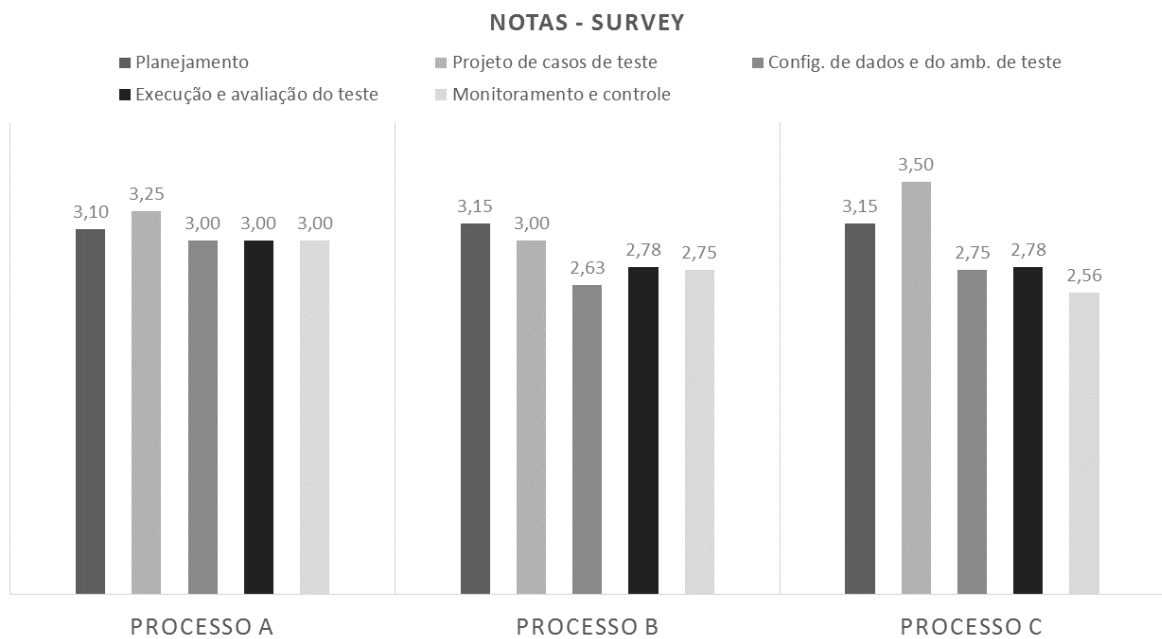


Figura 4.20: Nota Survey Agrupada por Fase

É interessante que essa avaliação também esteja presente neste trabalho, para reforçar a escolha do *Processo A* como o mais aplicável ao processo de teste de software considerado.

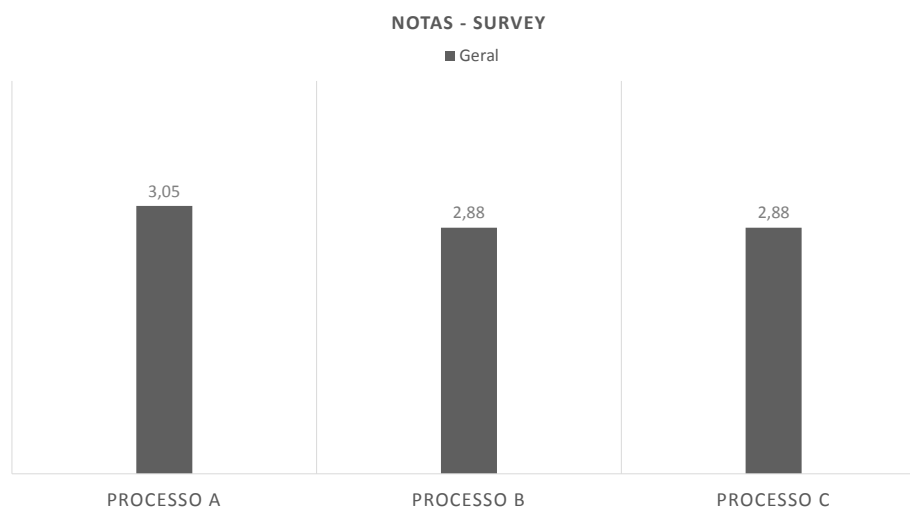


Figura 4.21: Nota *Survey* Agrupada por *Processo*

4.4 Considerações Finais

Este capítulo relatou o *Survey* realizado com a intenção de complementar os resultados obtidos na Análise Crítica. O objetivo deste *Survey* foi captar a opinião de profissionais da área de teste de software com relação ao grau de aplicabilidade de abordagens de Gestão do Conhecimento aplicadas em um processo de teste de software.

Foram apresentadas a metodologia utilizada para a criação e aplicação do *Survey*, o embasamento teórico para entendimento das questões, o critério de seleção de profissionais e a estratégia adotada para a análise de suas respostas.

A análise das respostas dos participantes indicou que o *Processo A* foi considerado o mais aplicável às práticas do processo de teste de software enxuto, seguido dos *Processos C* e *B*. Ressalta-se que o *Processo A* também foi o mais aplicável na Análise Crítica, diferentemente das segunda e terceira posições, ocupadas pelos *Processos B* e *C*.

No próximo capítulo será apresentada uma comparação detalhada dos resultados obtidos nos dois estudos realizados: a Análise Crítica e o *Survey*. Também será explorada a integração dos conceitos de Gestão do Conhecimento investigados em um único processo, e a sua avaliação.

Capítulo 5

COMPARAÇÃO DE RESULTADOS E INTEGRAÇÃO DE CONCEITOS-CHAVE

5.1 Considerações Iniciais

Neste capítulo aborda-se, inicialmente, a comparação dos resultados da Análise Crítica (descritos no Capítulo 3) com os resultados do *Survey* (descritos no Capítulo 4), realizado com o intuito de complementar o resultado da Análise Crítica. Desta forma, fez-se necessário que estes resultados fossem contrastados, identificando-se as semelhanças e divergências dos dois estudos até então descritos neste trabalho.

Além da comparação, neste capítulo também é analisada a combinação das três técnicas de Gestão do Conhecimento investigadas neste trabalho (*feedback*, *lesson learned* e documentos do conhecimento) em um único processo de Gestão do Conhecimento. Estas técnicas foram incorporadas ao *Processo A*, identificado como o mais bem aplicável a um processo de teste de software pela Análise Crítica (Capítulo 3) e pelo *Survey* (Capítulo 4). O *processo* resultante é chamado de *Processo Combinado*. Para avaliar este *Processo Combinado*, solicitou-se aos participantes do *Survey* que respondessem a uma nova pesquisa com o intuito de verificar se houve aperfeiçoamento sobre a abordagem da qual se originou.

No decorrer do capítulo também serão abordados os trabalhos relacionados a esta dissertação. É importante ressaltar que o termo *processo*, refere-se aos processos de Gestão do Conhecimento derivados dos *frameworks* selecionados neste trabalho.

5.2 Comparação da Análise Crítica com os Resultados do Survey

Nos Capítulos 3 e 4 apresentaram-se dois estudos que compõem este trabalho, uma Análise Crítica e um *Survey*, ambos investigando o mesmo tema: a aplicabilidade de *processos* de Gestão do Conhecimento em um processo de teste de software. Para a realização destes estudos, três abordagens de Gestão do Conhecimento foram selecionadas (cada uma representando um *processo*). Estas abordagens foram descritas na Seção 3.2.

A intenção dos estudos foi identificar quais destas três abordagens melhor atende uma equipe de teste de software com relação à centralização das informações relevantes, o que permite o compartilhamento do conhecimento, a definição dos melhores profissionais e principalmente a melhoria contínua das atividades realizadas. Como meio de atingir o objetivo almejado, os estudos contaram com a participação de profissionais da área de teste de software, que deixaram sua opinião, segundo suas experiências na indústria. Caracterizou-se então os dois estudos em questão: a Análise Crítica, que refletiu a opinião da autora desta dissertação, e o *Survey*, que refletiu a experiência dos profissionais participantes.

O material utilizado, tanto para a Análise Crítica quanto para o *Survey*, foi o mesmo: um conjunto de práticas do TMMi, que caracteriza um modelo de processo de teste de software (CAMARGO; FERRARI; FABBRI, 2013), e três *processos* de Gestão do Conhecimento (COLOMOPALACIOS et al., 2014; ANDRADE et al., 2013; XUE-MEI et al., 2009). Entretanto, a apresentação e utilização deste material se deu de maneiras diferentes nos dois estudos, principalmente devido ao engajamento da autora desta dissertação com a análise minuciosa do material, visando a atingir os objetivos propostos para o trabalho aqui relatado. Desta forma, na Análise Crítica, o contato com o conteúdo foi maior e os passos que compuseram a análise foram mais aprofundados (a Análise Crítica completa é encontrada no Anexo A), pois a própria avaliação consistiu em uma análise aprofundada de cada *processo*. Em contrapartida, o contato dos participantes do *Survey* com o conteúdo foi breve, apenas no momento da participação, e as informações foram apresentadas de forma condensada. Este breve contato pode ter influenciado a interpretação das práticas e o entendimento dos *processos*. Nesse contexto, diferenças nos resultados indicam o caráter complementar das avaliações, já que refletem não só a opinião aprofundada de uma pessoa, que pode ser tendenciosa, mas também a opinião diversificada de profissionais da área.

Para que fosse possível comparar os resultados, a escala utilizada como resposta foi a mesma em ambos os estudos (a descrição da escala é encontrada na Seção 3.3.1). O agrupamento das práticas em fases, representando um processo de teste de software, também foi

empregado nos dois estudos.

No *Survey*, dentre todas as respostas obtidas, foram analisadas apenas aquelas pertencentes a um grupo de profissionais que declararam possuir um perfil gerencial em equipes de teste de software. Tal decisão garantiu um nível equiparável de experiência dos participantes selecionados. A Análise Crítica também dispôs de um profissional com experiência em gestão de equipe de teste de software, no caso, a autora desta dissertação.

Devido ao número reduzido de respostas no *Survey*, não foi possível a aplicação de nenhum teste estatístico para determinar a nota dos *processos* em cada prática. A estratégia escolhida foi utilizar a mediana como forma de medir as respostas obtidas, ou seja, em todas as práticas pertencentes ao modelo, foi extraída a mediana das respostas de cada um dos *processos*. Este passo não foi necessário durante a Análise Crítica, pois contemplou a opinião de apenas um profissional.

5.2.1 Disparidade entre os Resultados

Ao comparar os resultados da Análise Crítica com os resultados do *Survey*, uma disparidade foi identificada, a qual pode ser atribuída a diferentes interpretações das práticas que foram investigadas. Como exemplo, considera-se o caso da prática *Estabelecer o plano de teste (SP4.5)* da fase de *Planejamento*. Na Análise Crítica, todos os *processos* obtiveram nota 1 (*Não Aplicável*) para esta prática, porém no *Survey* os *Processos A* e *C* obtiveram nota 4 (*Extremamente Aplicável*), e o *Processo B* obteve nota 3 (*Aplicável*). Isso significa que as notas estiveram nos extremos opostos da escala utilizada.

No material contido no *Survey*, a descrição desta prática – extraída do guia de referência do TMMi (TMMi Foundation, 2010) – era a seguinte: “*O plano de teste é estabelecido e mantido como base para o gerenciamento de testes. Os resultados das práticas anteriores são documentados em um plano global de teste, ligando as informações de uma maneira lógica*”.

De acordo com a interpretação da Análise Crítica, essa atividade seria uma síntese de outras atividades anteriores, uma formalização das decisões, um estabelecimento de *baseline*. Por este motivo, os *processos* não foram aplicáveis a ela, uma vez que nenhuma informação nova é necessária para a realização desta atividade. No entanto, pode não ter sido esta a interpretação dos participantes do *Survey*, ao atribuírem notas tão altas à prática em questão.

Assim como no exemplo da prática mencionada, há outras práticas que receberam notas divergentes na Análise Crítica e no *Survey*, e que serão abordadas a seguir, considerando cada um dos *processos* avaliados. Para auxiliar a identificação das práticas, a relação com seus

respectivos números é encontrada na Tabela A.2.

Processo A

Desenvolvido por Colomo-Palacios et al. (2014), este *framework* tem como principal característica a utilização de *feedback*. Os detalhes do *processo* de Gestão do Conhecimento elaborado a partir deste *framework* foram apresentados na Seção 3.2.1.

Na Figura 5.1 indica-se quais foram as notas recebidas pelo *Processo A* em cada uma das práticas. Observa-se que no gráfico da Análise Crítica houve grande variação de notas, e mesmo considerando as fases do processo de teste de forma independente, não há um padrão que possa ser identificado. Diferentemente, no gráfico que sumariza as notas do *Survey*, observa-se que para a maioria das práticas o *processo* recebeu notas que variaram entre 3 e 4. Ao comparar as duas avaliações, identifica-se que para 9 práticas, tanto na Análise Crítica quanto no *Survey*, esse *processo* recebeu as mesmas notas. São elas:

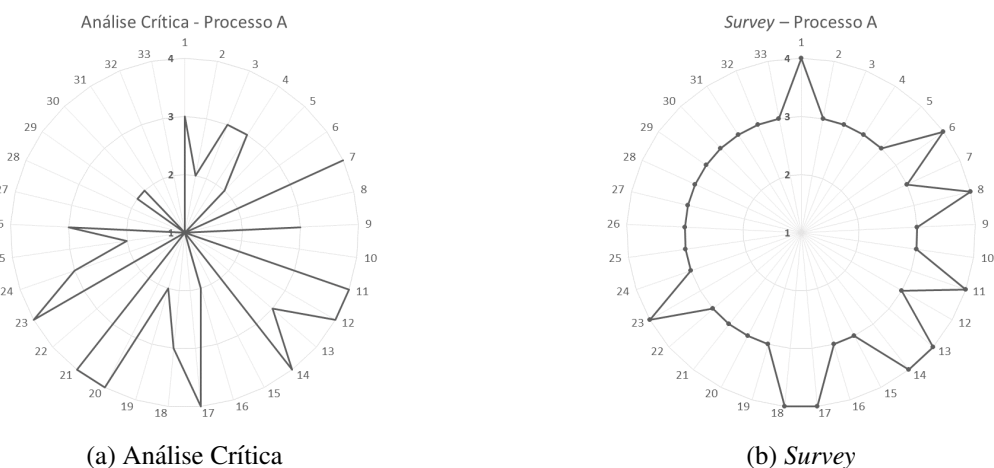


Figura 5.1: Comparação Análise Crítica Versus Survey por Prática - Processo A

- *Planejamento:*

- SP2.1 Identificar elementos e características a serem testados (nota 3 - *Aplicável*)
- SP2.2 Definir a abordagem de teste (nota 3 - *Aplicável*)
- SP1.1 Obter (eliciar) necessidades do ambiente de teste (nota 3 - *Aplicável*)

- *Projeto de casos de teste:*

- SP1.2 Identificar e priorizar casos de teste (nota 4 - *Extremamente Aplicável*)

- *Configuração de dados e do ambiente de teste:*
 - SP2.4 Desenvolver cronograma de execução do teste (nota 4 - *Extremamente Aplicável*)

- *Execução e avaliação do teste:*
 - SP3.2 Executar casos de teste (nota 4 - *Extremamente Aplicável*)
 - SP5.1 Executar casos de teste não funcional (nota 4 - *Extremamente Aplicável*)
 - SP5.2 Relatar incidentes de teste não funcional (nota 3 - *Aplicável*)

- *Monitoramento e controle:*
 - SP1.6 Conduzir revisões do progresso do teste (nota 3 - *Aplicável*)

As 9 práticas nas quais o *Processo A* recebeu a mesma avaliação estão distribuídas ao longo das fases do processo de teste, e receberam notas 3 (*Aplicável*) ou 4 (*Extremamente Aplicável*). Isso significa que os pontos de convergência nos dois estudos são aqueles que o *Processo A* é mais relevante para o processo de teste, e todos estão intimamente relacionados à atividade de execução dos testes.

Em 4 práticas, o *Processo A* foi mais bem avaliado na Análise Crítica. As notas obtidas foram 4 (*Extremamente Aplicável*) em todas estas práticas da Análise Crítica e 3 (*Aplicável*) em todas estas práticas do *Survey*. Portanto, ainda que a Análise Crítica tenha sido mais bem avaliada, a diferença entre as notas do *Survey* foi a mínima possível, indicando que, para estas práticas, existe conformidade entre os estudos. São elas: ***Planejar a equipe de teste (SP4.2)***, da fase *Planejamento*, ***identificar dados de teste específicos necessários (SP1.3)***, da fase *Projeto de casos de teste*, ***Decidir sobre incidentes com o grupo de controle de configuração (SP4.1)*** e ***executar ações apropriadas para corrigir os incidentes de teste (SP4.2)***, da fase *Execução e avaliação do teste*.

Não fizeram parte deste conjunto de práticas aquelas pertencentes às fases *Configuração de dados e do ambiente de teste*, e *Monitoramento e controle*. Para as 20 práticas restantes, o *Processo A* foi mais bem avaliado no *Survey*, e houve pouca convergência entre as notas obtidas com a Análise Crítica, principalmente com relação àquelas em que o *processo* recebeu nota 1 (*Não Aplicável*) na Análise Crítica e notas 3 (*Aplicável*) ou 4 (*Extremamente Aplicável*) no *Survey*. Estas se concentram principalmente na fase *Monitoramento e Controle*. São elas: ***Estabelecer o cronograma de teste (SP4.1)***, ***Estabelecer o plano de teste (SP4.5)*** e ***Analisar os requisitos***

do ambiente de teste (SP1.3) da fase Planejamento, Implementar o ambiente de teste (SP2.1), da fase Configuração de Dados e do Ambiente de Teste, Acompanhar o status dos incidentes de teste (SP4.3), da fase Execução e avaliação do teste, Monitorar defeitos (SP2.2), Conduzir revisões de qualidade do produto (SP2.6), Gerenciar ação corretiva (SP3.3), Coordenar a disponibilidade e o uso dos ambientes de teste (SP3.3) e Relatar e gerenciar incidentes do ambiente de teste (SP3.4) da fase Monitoramento e Controle.

Portanto, com relação ao *Processo A*, os resultados dos estudos coincidem que é mais bem aplicável às atividades relacionadas ao planejamento da equipe e à execução do teste.

Processo B

Desenvolvido por Andrade et al. (2013), este *processo* tem como base os documentos chamados *Lessons Learned*. Os detalhes do *processo* de Gestão do Conhecimento elaborado a partir deste *framework* foram apresentados na Seção 3.2.2.

Na Figura 5.2 indicam-se quais foram as notas recebidas pelo *Processo B* em cada uma das práticas. Assim como observado no *Processo A*, diferenças entre a avaliação da Análise Crítica e do Survey ocorrem no *Processo B*. As notas obtidas na Análise Crítica oscilaram entre todos os valores da escala, sem a possibilidade de encontrar um padrão. Porém, as notas do Survey se comportaram de maneira estável no valor 3 (*Aplicável*) da escala, com poucas variações para mais e para menos. Mesmo assim, é possível encontrar diversos pontos coincidentes entre as avaliações. São eles:

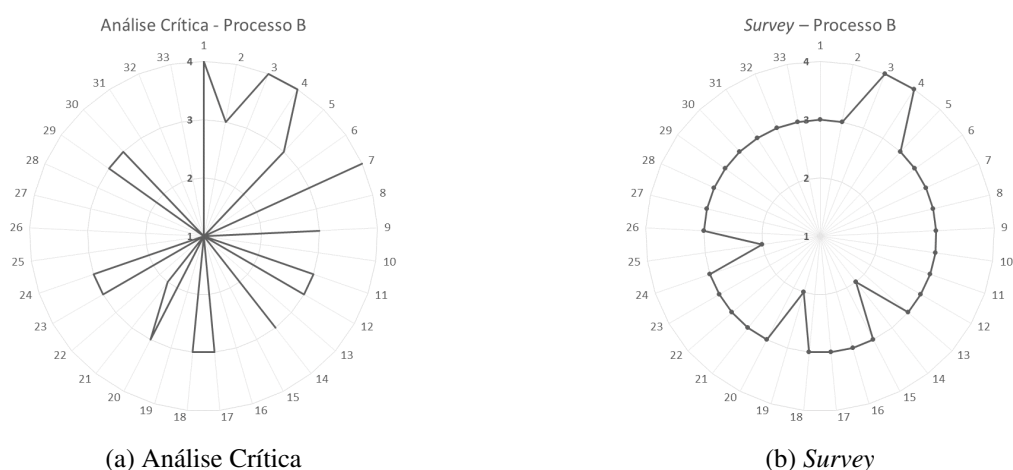


Figura 5.2: Comparação Análise Crítica Versus Survey por Prática - Processo B

- *Planejamento:*

- SP1.3 Analisar riscos do produto (nota 3 - *Aplicável*)

- SP2.1 Identificar elementos e características a serem testados (nota 4 - *Extremamente Aplicável*)
- SP2.2 Definir a abordagem de teste (nota 4 - *Extremamente Aplicável*)
- SP2.4 Definir critérios de parada (nota 3 - *Aplicável*)
- SP1.1 Obter (eliciar) necessidades do ambiente de teste (nota 3 - *Aplicável*)
- *Projeto de casos de teste:*
 - SP1.2 Identificar e priorizar casos de teste (nota 3 - *Aplicável*)
 - SP1.3 Identificar dados de teste específicos necessários (nota 3 - *Aplicável*)
- *Execução e avaliação do teste:*
 - SP3.2 Executar casos de teste (nota 3 - *Aplicável*)
 - SP3.3 Relatar incidentes de teste (nota 3 - *Aplicável*)
 - SP4.1 Decidir sobre incidentes com o grupo de controle de configuração (nota 3 - *Aplicável*)
 - SP5.1 Executar casos de teste não funcional (nota 3 - *Aplicável*)
 - SP5.2 Relatar incidentes de teste não funcional (nota 3 - *Aplicável*)
- *Monitoramento e controle:*
 - SP3.1 Analisar problemas (nota 3 - *Aplicável*)
 - SP3.2 Tomar ação corretiva (nota 3 - *Aplicável*)

Nota-se que para 14 práticas, as avaliações da Análise Crítica e do *Survey* coincidiram, sendo que para 11 dessas práticas o *Processo B* recebeu nota 3 (*Aplicável*), e para as demais recebeu nota 4 (*Extremamente Aplicável*). É importante destacar que para as duas práticas da fase *Projeto de casos de teste*, o *processo* foi avaliado com as mesmas notas. Porém, para a fase *Configuração de dados e do ambiente de teste*, não se obteve notas iguais para o *Processo B*. Percebe-se que os pontos em que as avaliações se igualaram estão intimamente relacionados à tomada de decisões, reforçando a característica gerencial deste *processo*.

Em 3 práticas, o *Processo B* recebeu notas mais altas na Análise Crítica. Assim como o ocorrido no *Processo A*, a diferença entre as notas da Análise Crítica e do *Survey* foi de apenas um ponto na escala. Tais práticas são listadas a seguir, com as respectivas notas obtidas

pelo *Processo B* na Análise Crítica: **Identificar riscos do produto (SP1.2)** (nota 4) e **Planejar a equipe de teste (SP4.2)** (nota 4), da fase *Planejamento*; e **Desenvolver cronograma de execução do teste (SP2.4)** (nota 3), da fase *Configuração de dados e do ambiente de teste*. É notável a relação entre essas três práticas e atividades gerenciais. Nas duas avaliações obtidas, o *Processo B* recebeu notas semelhantes (sempre entre os valores 3 e 4 da escala) para as práticas com estas características.

As 12 práticas que mais divergem entre as duas análises, nas quais o *Processo B* recebeu nota 1 (*Não Aplicável*) na Análise Crítica e nota 3 (*Aplicável*) no *Survey* estão, assim como no *Processo A*, concentradas na fase de *Monitoramento e Controle*. São elas: **Estabelecer o cronograma de teste (SP4.1)**, **Estabelecer o plano de teste (SP4.5)** e **Analisar os requisitos do ambiente de teste (SP1.3)** da fase *Planejamento*, **Implementar o ambiente de teste (SP2.1)** e **Realizar o teste intake (pré-teste) do ambiente de teste (SP2.4)** da fase *Configuração de Dados e do Ambiente de Teste*, **Acompanhar o status dos incidentes de teste (SP4.3)** da fase *Execução e Avaliação do Teste*, **Conduzir revisões do progresso do teste (SP1.6)**, **Monitorar defeitos (SP2.2)**, **Conduzir revisões de qualidade do produto (SP2.6)**, **Gerenciar ação corretiva (SP3.3)**, **Coordenar a disponibilidade e o uso dos ambientes de teste (SP3.3)** e **Relatar e gerenciar incidentes do ambiente de teste (SP3.4)** da fase *Monitoramento e Controle*.

Processo C

Desenvolvido por Xue-Mei et al. (2009), os elementos principais deste *processo* são os Documentos de Conhecimento. Os detalhes do *processo* de Gestão do Conhecimento elaborado a partir deste *framework* foram apresentados na Seção 3.2.3.

Na Figura 5.3 indicam-se quais foram as notas recebidas pelo *Processo C* em cada uma das práticas. Este *processo* também apresenta grande diferença entre as avaliações da Análise Crítica e do *Survey*. Na Análise Crítica, a concentração de notas está em torno dos valores 1 (*Não Aplicável*) e 2 (*Pouco Aplicável*) da escala. No *Survey*, por sua vez, as notas estão distribuídas entre os valores 2 (*Pouco Aplicável*), 3 (*Aplicável*) e 4 (*Extremamente Aplicável*) da escala.

Dentre os três *processos*, o C foi aquele que obteve menor quantidade de notas equivalentes nas duas avaliações, recebendo notas iguais para apenas seis práticas. São elas:

- *Projeto de casos de teste*:
 - SP1.2 Identificar e priorizar casos de teste (nota 4 - *Extremamente Aplicável*)

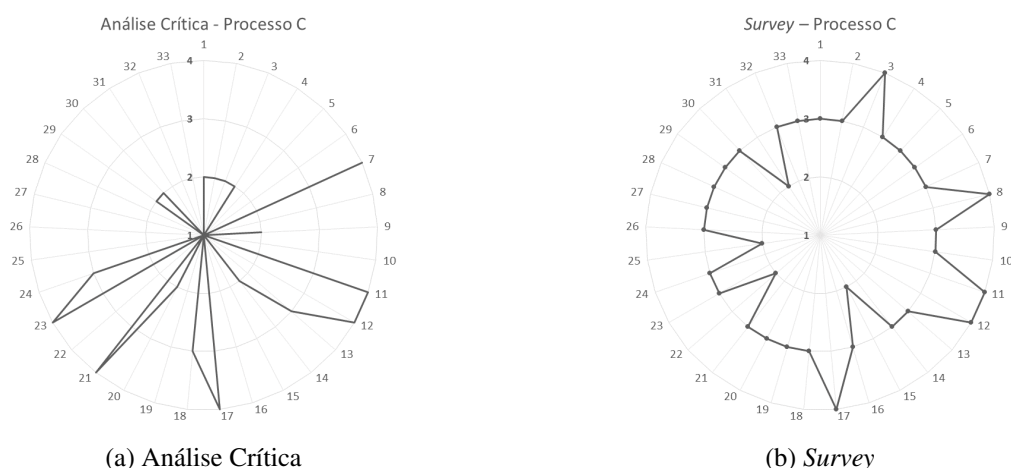


Figura 5.3: Comparação Análise Crítica Versus Survey por Prática - Processo C

- SP1.3 Identificar dados de teste específicos necessários (nota 4 - *Extremamente Aplicável*)
- *Configuração de dados e do ambiente de teste:*
 - SP2.1 Desenvolver e priorizar os procedimentos de teste (nota 3 - *Aplicável*)
- *Execução e avaliação do teste:*
 - SP3.2 Executar casos de teste (nota 4 - *Extremamente Aplicável*)
 - SP3.3 Relatar incidentes de teste (nota 3 - *Aplicável*)
 - SP5.2 Relatar incidentes de teste não funcional (nota 3 - *Aplicável*)

Para as seis práticas em que os estudos receberam notas iguais há variação entre os valores 3 (*Aplicável*) e 4 (*Extremamente Aplicável*) da escala. Nota-se entre estas práticas a recorrência de dois temas principais: execução dos testes e relato de incidentes. Neste *processo*, a fase *Projeto de casos de teste* também possui todas as suas práticas com as mesmas notas, nas duas avaliações. No entanto, nenhuma das práticas pertencente as fases de *Planejamento e Monitoramento e controle* recebeu notas iguais.

Em três práticas, o *Processo C* foi mais bem avaliado na Análise Crítica. Assim como o ocorrido nos outros *processos*, a diferença entre as notas foi de apenas 1 ponto na escala, sendo que as notas obtidas foram 4 (*Extremamente Aplicável*) na Análise Crítica e 3 (*Aplicável*) no *Survey*. Estas práticas são as listadas a seguir: *Planejar a equipe de teste (SP4.2)*, da fase *Planejamento*; e *Executar ações apropriadas para corrigir os incidentes de teste (SP4.2)* e *Executar casos de teste não funcional (SP5.1)*, da fase *Execução e avaliação do teste*.

Não houve convergência entre um grupo de 11 práticas, nas quais o *Processo C* recebeu nota 1 (*Não Aplicável*) na Análise Crítica e notas 3 (*Aplicável*) ou 4 (*Extremamente Aplicável*) no *Survey*. Assim como nos outros *processos* a fase *Monitoramento e Controle* concentrou maior número destas práticas. São elas: ***Definir critérios de parada (SP2.4)***, ***Estabelecer o cronograma de teste (SP4.1)***, ***Estabelecer o plano de teste (SP4.5)*** e ***Analisar os requisitos do ambiente de teste (SP1.3)***, da fase *Planejamento*, ***Realizar o teste intake (pré-teste) do ambiente de teste (SP2.4)*** da fase *Configuração de Dados e do Ambiente de Teste*, ***Escrever log de teste (SP3.4)*** da fase *Execução e Avaliação do Teste*, ***Conduzir revisões do progresso do teste (SP1.6)***, ***Monitorar defeitos (SP2.2)***, ***Conduzir revisões de qualidade do produto (SP2.6)***, ***Coordenar a disponibilidade e o uso dos ambientes de teste (SP3.3)*** e ***Relatar e gerenciar incidentes do ambiente de teste (SP3.4)***, da fase *Monitoramento e Controle*.

5.2.2 Avaliação Geral

Dependências Entre as Práticas

Conforme o discutido nas Seções 3.3.3 e 4.3.3, há um relacionamento de dependência entre algumas práticas, assim como identificado no trabalho de Höhn (2011). Na Figura 5.4, resumiram-se as dependências das práticas averiguadas nesta dissertação¹. Nesta figura, indicam-se as notas obtidas pelos *processos* em cada um dos estudos (Análise Crítica à esquerda e *Survey* à direita).

É esperado que as práticas envolvidas em um relacionamento de dependência sejam atendidas pelos *processos* da mesma maneira, ou seja, que os *processos* sejam avaliados de forma semelhante para tais práticas. Segundo esta observação, verifica-se que o conjunto de práticas encontrado na Figura 5.4a é o único que destoa do padrão. As práticas (***Estabelecer o cronograma de teste - SP4.1*** e ***Estabelecer o plano de teste - SP4.5***) foram avaliadas com notas 1 (*Não Aplicável*), enquanto a prática ***Planejar a equipe de teste - SP4.2***, que depende das anteriores, foi avaliado com nota 4 *Extremamente Aplicável*, estando no extremo oposto da escala de notas.

De acordo com a Análise Crítica, fica evidente que existe a necessidade de verificar outras fontes para a realização da atividade de planejamento da equipe de testes. Exemplos seriam um documento que contenha as datas em que os membros da equipe encontram-se disponíveis para alocação, ou um documento que apresente a lista de projetos futuros que devem ser planejados. Todavia, os conjuntos de práticas restantes atendem ao quesito da dependência, e são considerados suficientes para a realização de suas respectivas atividades.

¹Visões ampliadas dos gráficos apresentados podem ser encontradas nas Seções 3.3.3 e 4.3.3

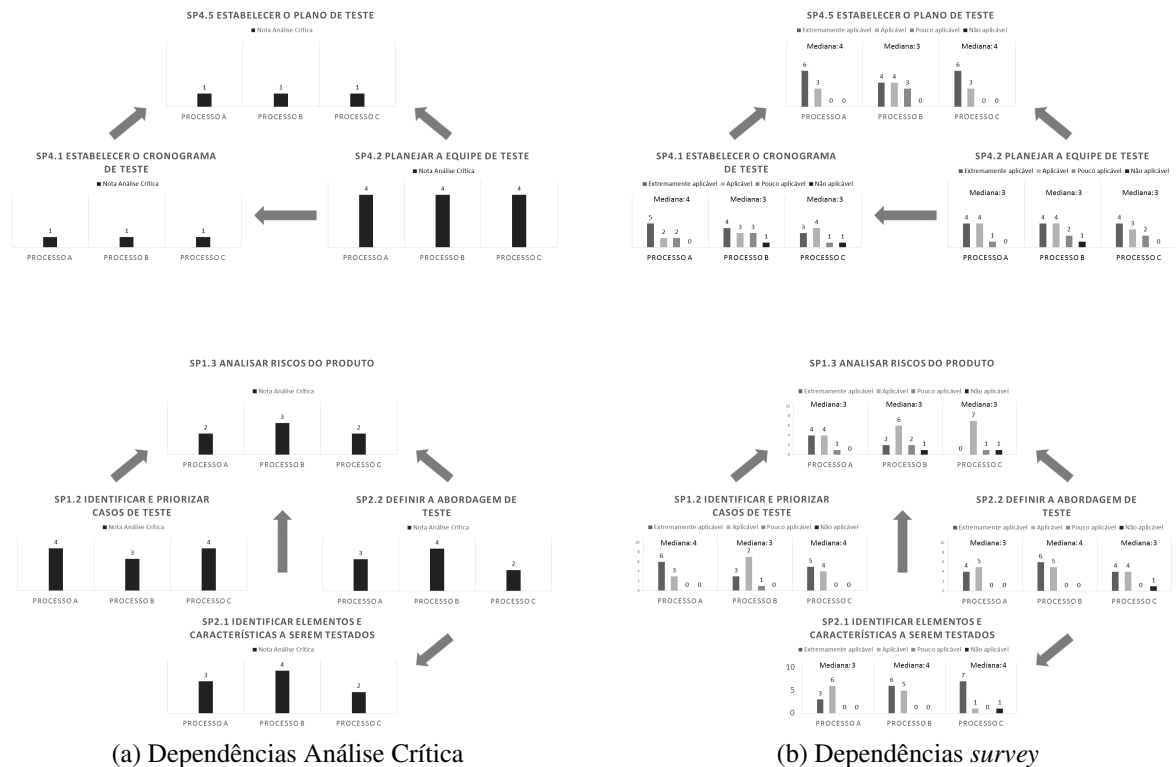


Figura 5.4: Dependências Entre as Práticas

Avaliação Geral

De acordo com as características das avaliações de cada *processo* discutidas nas seções anteriores deste capítulo, pode-se observar alguns pontos em comum aos dois estudos:

- Para todos os *processos* que receberam notas iguais na Análise Crítica e no Survey, as notas obtidas estiveram entre os valores 3 e 4 da escala (*Aplicável* e *Extremamente Aplicável*, respectivamente).
- No cenário em que os *processos* receberam notas maiores na Análise Crítica, as notas obtidas também estiveram entre os valores 3 e 4 (*Aplicável* e *Extremamente Aplicável*). A diferença entre as notas recebidas na Análise Crítica e no Survey, para estes casos, nunca foi maior que 1 ponto na escala.

Para a avaliação das fases do processo de teste, foi calculada a média aritmética das notas dos *processos* para cada uma das práticas, agrupadas em suas respectivas fases. A comparação dos resultados pode ser visualizada na Figura 5.5. É importante ressaltar que no Survey, as notas individuais dos processos com relação às suas práticas foram obtidas através do cálculo da mediana das respostas dos participantes.

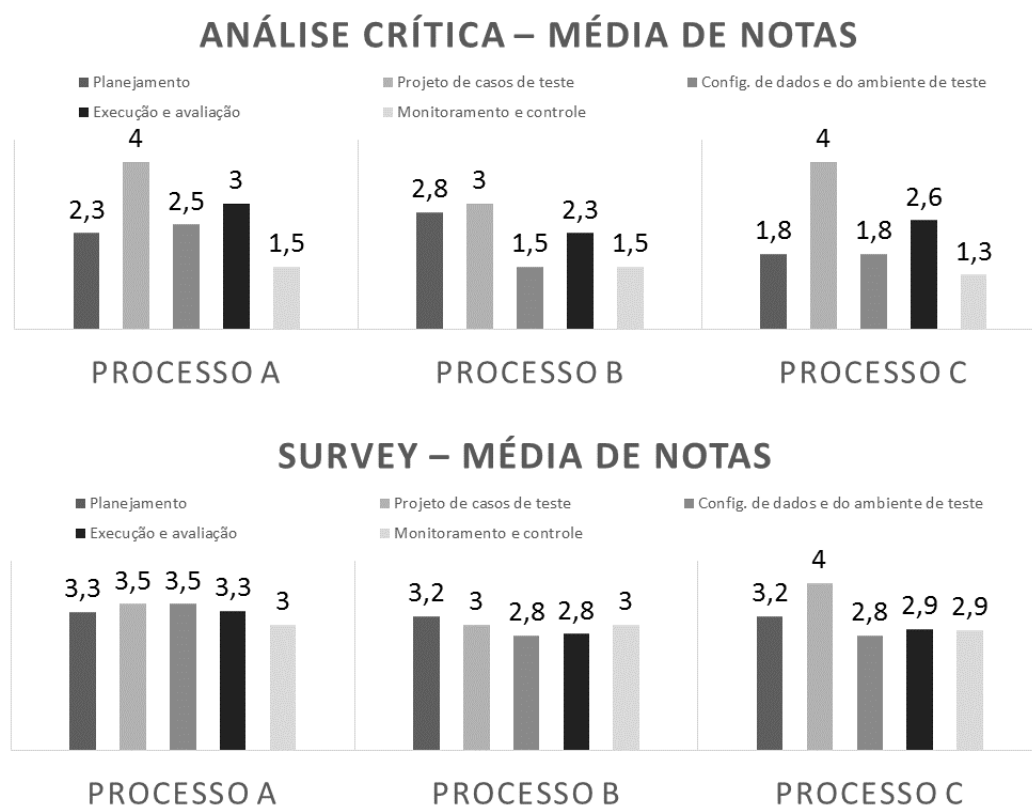


Figura 5.5: Comparação Análise Crítica Versus Survey por Fase

A forte variação de notas observada na Análise Crítica é refletida nas fases do processo de teste. Há casos em que um mesmo *processo* recebeu uma nota muito próxima do valor 1 (*Não Aplicável*) em uma determinada fase, e recebeu nota 4 (*Extremamente Aplicável*) em outra fase, ou seja, tais avaliações encontram-se nos extremos opostos da escala. Diferenças desse tipo não são tão evidentes nas notas do *Survey*.

As notas gerais dos *processos* também foram calculadas através da média aritmética das notas recebidas pelas práticas. Na Figura 5.6 mostra-se a comparação das notas obtidas nos dois estudos.

Os resultados finais determinam que o *Processo A* foi aquele que, no geral, esteve mais aplicável às práticas do processo de teste. A diferença está na inversão da posição obtida pelos *Processos B* e *C*, sendo que no *Survey* o *Processo C* foi avaliado como o segundo mais aplicável ao processo de teste considerado, enquanto a Análise Crítica avaliou esse *processo* como o menos aplicável dentre os três. É nítida a proximidade das notas, cuja diferença não passa de 0,5 na Análise Crítica e 0,4 no *Survey*.

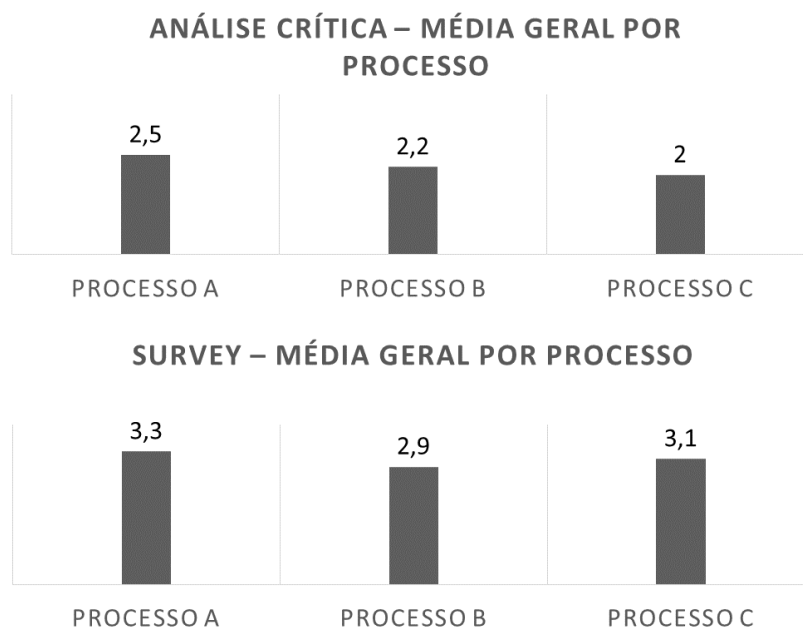


Figura 5.6: Comparação Análise Crítica Versus Survey por Processo

5.3 Integração de Conceitos-Chave dos Processos de Gestão do Conhecimento Avaliados

Em vista dos resultados dos estudos realizados e apresentados até agora nesta dissertação, uma nova investigação foi realizada com o intuito de mesclar os conceitos-chave de Gestão do Conhecimento discutidos nesta dissertação em um único *processo*, que nesta seção será chamado de *Processo Combinado*. O objetivo foi investigar possíveis benefícios oriundos da integração das abordagens avaliadas. Esta investigação foi conduzida de maneira exploratória e não tem intenção de substituir os resultados já encontrados.

Para compor o *Processo Combinado*, o *Processo A* foi tomado como referência, pois nas duas avaliações anteriores, esse *processo* foi identificado como sendo o mais aplicável às atividades de uma equipe de testes. A ele foram adicionados os conceitos principais dos outros dois *processos*: *Lessons Learned* e *Documentos do Conhecimento*.

Para avaliar essa possível integração que culminaria em um *processo a priori* mais completo, um novo *Survey* foi realizado. Os participantes do primeiro *Survey* foram convidados a respondê-lo, e suas opiniões foram analisadas.

Nas próximas seções serão apresentados o fluxograma do *Processo Combinado*, bem como os detalhes desta nova pesquisa.

5.3.1 Processo Combinado

Para incorporar os conceitos de *Lessons Learned* e *Documentos do Conhecimento* no *Processo Combinado*, alguns ajustes foram necessários na estrutura do *Processo A*. Em primeiro lugar, dois atores foram adicionados ao fluxograma, para que a distribuição de tarefas estivesse mais adequada aos papéis e responsabilidades de uma equipe de teste de software. A figura do Líder de Equipe foi incluída, pois o Analista de Teste e o Gerente, atores provenientes do *Processo A*, ocupam lugares distantes na estrutura hierárquica de uma equipe devidamente estruturada. Portanto, o Líder de Equipe ocupará esta posição intermediária entre os dois atores que já participavam do fluxograma.

Outro ator adicionado ao fluxograma foi o Analista de Teste Experiente. Em uma equipe de teste, é comum que os Líderes e Gerentes estejam sobrecarregados e deleguem atividades de menor importância. Neste contexto, surge a figura do Analista de Teste Experiente, que é responsável por substituir seus superiores em determinadas atividades nas quais a experiência é relevante.

Devido à inclusão de novos atores, uma raia foi adicionada ao fluxograma, de modo a dispor as atividades de acordo com a responsabilidade do cargo daqueles que as exercem. Os atores foram divididos em três grupos, cada um correspondendo a uma raia do fluxograma. Detalhes desses grupos são fornecidos a seguir:

- Gerente / Líder / Analista de Teste Experiente: Para este grupo, estão incluídas as atividades que originalmente eram de responsabilidade gerencial, mas que podem ser delegadas a um Analista de Teste Experiente.
- Gerente / Líder: Este grupo engloba atividades exclusivamente gerenciais.
- Analista de Teste: Estão envolvidas neste grupo as atividades realizadas pelos Analistas de Teste, com todos os níveis de experiência.

Algumas atividades tiveram seus rótulos alterados, de modo a referenciarem com maior exatidão as ações que representam. Por exemplo, a atividade *Selecionar perfil* foi substituída por *Selecionar equipe*. Houve também a divisão da atividade *Avaliar processo* nas atividades *Avaliar caso de teste* e *Avaliar plano de teste*.

O *Processo Combinado* é apresentado na Figura 5.7. Apenas os elementos adicionados e modificados, que não estavam presentes no *Processo A*, são descritos a seguir:

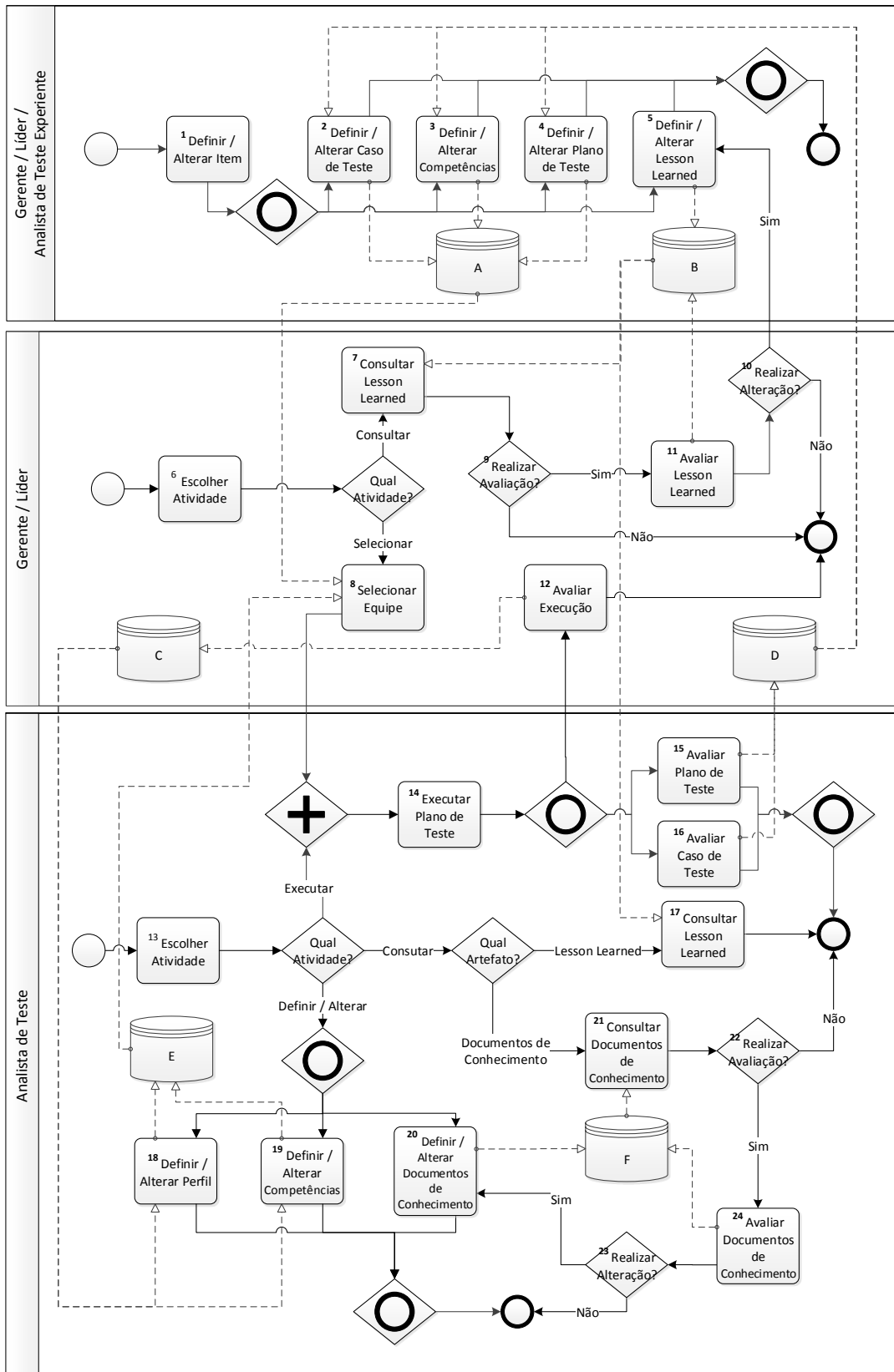


Figura 5.7: Processo Combinado

2. **Definir / Alterar Caso de Teste:** Especificação, realizada pelo Gerente / Líder / Analista de Teste Experiente, dos casos de teste que serão executados pelo Analista de Teste em um processo.
3. **Definir / Alterar Competências Requeridas:** Especificação, realizada pelo Gerente / Líder / Analista de Teste Experiente, das competências (experiências) necessárias em um Analista de Teste para a realização de um processo.
4. **Definir / Alterar Plano de Teste:** Especificação, realizada pelo Gerente / Líder / Analista de Teste Experiente, das atividades de teste (plano de teste) que serão realizadas pelo Analista de Teste.
5. **Definir / Alterar *Lesson Learned*:** Preenchimento inicial (ou alteração) dos dados de uma *lesson learned* pelo Gerente / Líder / Analista de Teste Experiente,.
7. **Consultar *Lesson Learned*:** Consulta, realizada pelo Gerente / Líder, a uma *lesson learned*.
8. **Selecionar Equipe:** Escolha, realizada pelo Gerente / Líder / Analista de Teste Experiente, de qual Analista de Teste possui as competências necessárias para a execução do processo.
9. **Realizar Avaliação:** Verificação da necessidade de avaliar uma *lesson learned*.
10. **Realizar Alteração:** Verificação da necessidade de alterar uma *lesson learned*.
11. **Avaliar *Lesson Learned*:** Avaliação do conteúdo (conteúdo relevante) de uma *lesson learned*.
14. **Executar Plano de Teste:** Após ser selecionado, o Analista de Teste seguirá instruções contidas no plano de teste para a realização do teste.
15. **Avaliar Caso de Teste:** Avaliação, realizada pelo Analista de Teste, dos casos de teste contidos no plano de teste.
16. **Avaliar Plano de Teste:** Avaliação, realizada pelo Analista de Teste, dos planos de teste.
17. **Consultar *Lesson Learned*:** Consulta, realizada pelo Analista de Teste, a uma *lesson learned*.
20. **Definir / Alterar Documentos de Conhecimento:** Preenchimento inicial (ou alteração), realizada pelo Analista de Teste, de um documento que contém informações relevantes para a realização do teste, que podem ser reaproveitadas no futuro.

21. **Consultar Documento de Conhecimento:** Consulta, realizada pelo Analista de Teste, aos documentos de conhecimento que contém informações relevantes para a realização do teste.
22. **Realizar Avaliação:** Verificação, realizada pelo Analista de Teste, da necessidade de avaliar um documento de conhecimento.
23. **Realizar Alteração:** Verificação, realizada pelo Analista de Teste, da necessidade de alterar um documento de conhecimento.
24. **Avaliar Documento de Conhecimento:** Avaliação, realizada pelo Analista de Teste, da relevância das informações contidas nos documentos de conhecimento.

Descrição dos elementos de Repositório de Dados:

- A **Armazenamento das Definições:** Todas as informações definidas ao longo do processo são armazenadas em banco de dados para serem utilizadas na atividade 8. *Selecionar Equipe*.
- B **Armazenamento de uma *Lesson Learned*:** As informações contidas em uma *Lesson Learned* são armazenadas em base de dados e também são utilizadas nas atividades 7. e 17. *Consultar Lessons Learned*.
- C **Armazenamento da Avaliação da Execução:** O *Gerente / Líder* avaliará o desempenho do Analista de Teste após a execução do processo (atividade 12. *Avaliar Execução*). Essas informações serão utilizadas no aprimoramento das competências e perfil do Analista de Teste (atividades 18. *Definir / Alterar Perfil* e 19. *Definir / Alterar Competências*).
- D **Armazenamento da Avaliação do Plano de Teste:** As informações constatadas pelo Analista de Teste nas atividades 15. *Avaliar Plano de Teste* e 16. *Avaliar Caso de Teste*, devem ser armazenadas em banco de dados para permitir a melhoria de suas descrições.
- E **Armazenamento das Características do Analista de Teste:** As informações profissionais do Analista de Teste definidas ao longo do processo são armazenadas em banco de dados para serem utilizadas na atividade 8. *Selecionar Equipe*.
- F **Armazenamento de um Documento de Conhecimento:** O documento de conhecimento é armazenado em base de dados para ser utilizado na atividade 21. *Consultar Documentos de Conhecimento*.

5.3.2 Avaliação

Um novo *Survey*² foi criado, com o objetivo de verificar a aceitação do *Processo Combinado* entre os participantes dos surveys anteriores. A intenção era de que os respondentes indicassem, de forma justificada, qual a aplicabilidade do *Processo Combinado* com relação ao processo de teste de software já apresentado anteriormente (*Processo A*).

O novo *Survey* foi dividido em duas etapas: (i) avaliação com relação a um subgrupo de práticas de teste; e (ii) a avaliação com relação às fases do processo de teste. Ambas as etapas são ilustradas na Figura 5.8 (partes (a) e (b), respectivamente). Estas etapas foram apoiadas pela apresentação das teorias envolvidas, ou seja, o processo de teste (CAMARGO; FERRARI; FABBRI, 2013), o *Processo A* e o *Processo Combinado*.

O subgrupo de práticas foi composto por aquelas nas quais o *Processo A* obteve mais de 25% das notas menores ou iguais a 2 (*Pouco Aplicável*) no *Survey*. A expectativa era entender se um processo de Gestão do Conhecimento mais completo poderia ser mais bem avaliado. Este subgrupo abrigou 16 práticas no total. São elas (agrupadas por fase do processo de teste):

- *Planejamento*
 - SP2.4 Definir critérios de parada
 - SP4.1 Estabelecer o cronograma de teste
 - SP1.1 Obter (eliciar) necessidades do ambiente de teste
 - SP1.3 Analisar os requisitos do ambiente de teste
- *Configuração de dados e do ambiente de teste*
 - SP2.4 Realizar o teste *intake* (pré-teste) do ambiente de teste
- *Execução e avaliação do teste*
 - SP3.4 Escrever log de teste
 - SP4.1 Decidir sobre incidentes com o grupo de controle de configuração
 - SP4.2 Executar ações apropriadas para corrigir os incidentes de teste
 - SP4.3 Acompanhar o status dos incidentes de teste
 - SP5.2 Relatar incidentes de teste não funcional

²<http://amon.dc.ufscar.br/limesurvey/index.php?sid=46987&lang=pt-BR> – Acessado em 10/04/2015.

*** O novo framework é aplicável à prática SP2.4 Definir critérios de parada, contida na fase Planejamento do processo de teste? É possível que as atividades relacionadas à prática sejam suportadas pelo framework? Escolha uma das seguintes respostas:**

Extremamente aplicável
 Aplicável
 Pouco aplicável
 Não aplicável

Por favor, coloque aqui o seu comentário:

?

Prática: SP2.4 Definir critérios de parada
Descrição: Um conjunto de critérios de saída relacionados com o processo e com a qualidade do produto são definidos para saber quando o teste deverá ser concluído.

*** O novo framework é aplicável à prática SP4.1 Estabelecer o cronograma de teste, contida na fase Planejamento do processo de teste? É possível que as atividades relacionadas à prática sejam suportadas pelo framework? Escolha uma das seguintes respostas:**

Extremamente aplicável
 Aplicável
 Pouco aplicável
 Não aplicável

Por favor, coloque aqui o seu comentário:

?

Prática: SP4.1 Estabelecer o cronograma de teste
Descrição: O cronograma de testes, com os estágios pré-definidos e de tamanhos gerenciáveis, é estabelecido e mantido com base na estimativa e no ciclo de vida de teste definidos. Informações como duração, dependências, recursos e entradas são identificadas. Também são estabelecidos critérios para determinar o que constitui um desvio significativo do plano de teste e pode indicar a necessidade de alterar o cronograma.

(a) Avaliação das Práticas

*** A inclusão de novas técnicas de Gestão do Conhecimento ao framework (lessons learned e documentos do conhecimento), faz com que a fase de Planejamento seja melhor atendida? Escolha uma das seguintes respostas:**

Concordo
 Discordo

?

Práticas presentes na fase de planejamento:

- SP1.2 Identificar riscos do produto
- SP1.3 Analisar riscos do produto
- SP2.1 Identificar elementos e características a serem testados
- SP2.2 Definir a abordagem de teste
- SP2.4 Definir critérios de parada
- SP4.1 Estabelecer o cronograma de teste
- SP4.2 Planejar a equipe de teste
- SP4.5 Estabelecer o plano de teste
- SP1.1 Obter (eliciar) necessidades do ambiente de teste
- SP1.3 Analisar os requisitos do ambiente de teste

*** Com relação à fase de Planejamento, quais melhorias deveriam ser propostas para que o novo framework estivesse mais adequado às necessidades de uma equipe de testes?**

(b) Avaliação das Fases do Processo de Teste

Figura 5.8: Página de Avaliação do Novo Survey

- SP5.3 Escrever log de teste não funcional
- *Monitoramento e controle*
 - SP2.2 Monitorar defeitos
 - SP2.6 Conduzir revisões de qualidade do produto
 - SP3.1 Analisar problemas
 - SP3.3 Gerenciar ação corretiva
 - SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste

Os participantes do primeiro *Survey* foram novamente convidados a deixarem sua opinião. Dos 42 participantes (sem considerar o *outlier*), apenas 4 atenderam à solicitação. Houve a participação de um profissional que não esteve envolvido no primeiro *Survey*, e sua contribuição ainda assim foi considerada, pois declarou ter experiência de quatro anos na indústria, na área de teste de software, o que certifica uma contribuição relevante para a pesquisa. No total, foram obtidas 5 respostas.

Dois perfis foram identificados neste novo *Survey*: (i) profissionais com experiência em teste de software e gestão de equipes de teste (3 participantes); e (ii) profissionais apenas com experiência em teste de software (2 participantes). Devido ao pequeno índice de participação, não houve distinção entre as respostas dos perfis identificados.

Análise dos Resultados

A primeira etapa do *Survey* foi composta pela avaliação do *Processo Combinado* com relação ao subgrupo de práticas. Assim como no primeiro *Survey*, a mediana foi utilizada como medida que representasse os dados da pesquisa. Os resultados da avaliação por práticas são exibidos na Figura 5.9.

A relação das práticas é encontrada na Tabela A.2.

Na segunda etapa do novo *Survey*, os participantes responderam a seguinte pergunta “*A inclusão de novas técnicas de Gestão do Conhecimento ao processo (lessons learned e documentos do conhecimento) faz com que a fase X seja mais bem atendida?*”, na qual X corresponde a cada uma das fases do processo de teste: *Planejamento, Projeto de casos de teste, Configuração de dados e do ambiente de teste, Execução e avaliação e Monitoramento e controle*. O percentual de participantes que concordou com esta pergunta é exibido na Figura 5.10.

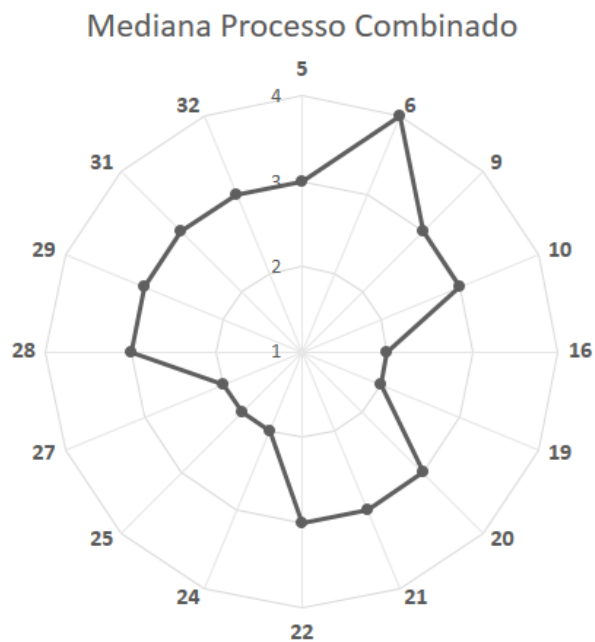


Figura 5.9: Avaliação do *Processo Combinado* por Práticas

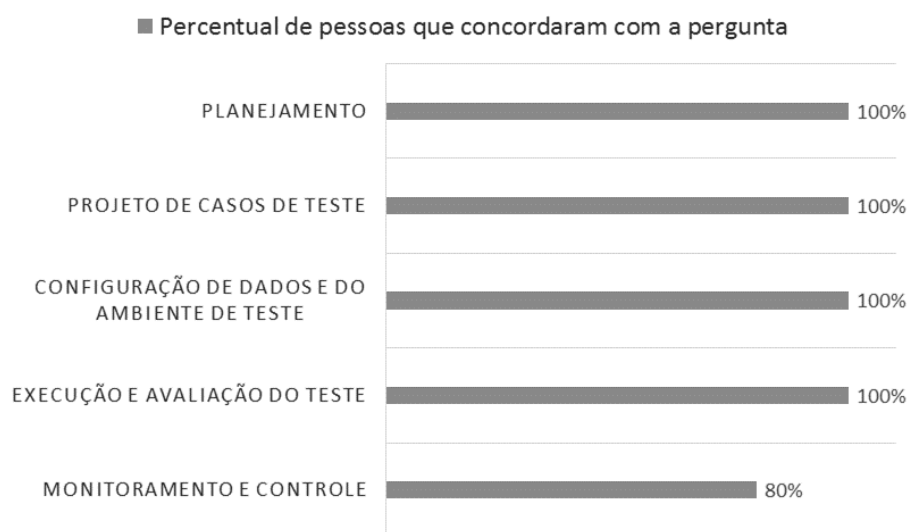


Figura 5.10: Avaliação do *Processo Combinado* por Fases do Processo de Teste

Para as quatro primeiras fases do processo de teste, todos os participantes indicaram que o *Processo Combinado* é mais eficaz quanto à sua aplicabilidade, se comparado com o *Processo A*. Para a última fase, somente um participante não concordou com a pergunta, ou seja o *Processo Combinado* não traz qualquer benefício à fase *Monitoramento e controle*.

Ao examinar as duas etapas deste novo *Survey*, nota-se que não houve melhor aceitação do *Processo Combinado* com relação à aplicabilidade das práticas do *Processo A*, ou seja, ao avaliarem o *Processo Combinado* com relação ao subconjunto de práticas, individualmente, a avaliação do *Processo A* foi superior. Por outro lado, os participantes avaliaram o *Processo Combinado* como mais bem aplicável do que o *Processo A* com relação às fases do processo de teste. Esta pode ser uma indicação de que ao adicionar novas técnicas de Gestão do Conhecimento ao *Processo A* o processo resultante foi tão ineficaz quanto o seu original para o subconjunto de práticas em questão. No entanto esta inclusão trouxe benefícios gerais ao *Processo Combinado* quando comparado com o *Processo A*.

5.4 Trabalhos Relacionados

Nesta dissertação, foram detalhados e amplamente discutidos três *processos* de Gestão do Conhecimento, derivados dos *frameworks* descritos nos trabalhos de Colomo-Palacios et al. (2014), Andrade et al. (2013) e Xue-Mei et al. (2009). A principal diferença entre estes três trabalhos e o trabalho descrito nesta dissertação é que os autores dos *frameworks* avaliados tiveram como objetivo desenvolver ferramentas que automatizassem um determinado processo de Gestão do Conhecimento. O objetivo do trabalho descrito nesta dissertação, por outro lado foi avaliar a aplicabilidade de processos de Gestão do Conhecimento em um processo de teste de software, em especial o processo de teste definido por Camargo, Ferrari e Fabbri (2013).

O trabalho de Camargo, Ferrari e Fabbri (2013) também é descrito nesta dissertação (Seção 2.3.1), pois foi selecionado como processo de teste base, sobre o qual os estudos realizados (Análise Crítica e *Survey*) foram elaborados. Seu objetivo foi estabelecer um processo de teste, a partir de um grupo de práticas do TMMi (TMMi Foundation, 2010).

Além dos trabalhos supracitados, há um estudo recente, realizado por Souza (2014), que desenvolveu um *framework* de Gestão do Conhecimento para teste de software, chamado *Testing-Knowledge Management* (T-KM). O *framework* T-KM foi construído a partir de uma ontologia que representa os conceitos do domínio de teste de software, diferentemente desta dissertação, que derivou *processos* de Gestão do Conhecimento de *frameworks* existentes. Um *Survey* também foi realizado, com profissionais da área de teste e desenvolvimento de software. O

objetivo do *Survey* de Souza (2014) foi definir o escopo de um Portal de Gestão do Conhecimento para teste de software, chamado de *Testing KM Portal*.

Diferentemente do *Survey* realizado por Souza (2014), os *Surveys* apresentados no Capítulo 4 e na Seção 5.3 tiveram participação exclusiva de profissionais da área de teste de software e tiveram como objetivo avaliar *processos* de Gestão do Conhecimento.

Em relação ao *Testing KM Portal*, esse foi desenvolvido como base no *framework* TK-M e nos resultados do *Survey*. As principais técnicas de Gestão do Conhecimento contidas no portal são *Lessons Learned*, *Páginas Amarelas* e *Fóruns de Discussão*. Com exceção de *Lessons Learned*, o *Testing KM Portal* se baseou em técnicas diferentes, se comparado com as abordadas majoritariamente nesta dissertação. Por meio de demonstrações do *Testing KM Portal*, líderes de projetos avaliaram o sistema. As avaliações do *Testing KM Portal* foram positivas, ressaltando a importância da aplicação de Gestão do Conhecimento em um processo de teste de software. No entanto, nesta dissertação, a proposta de um *Processo Combinado* não foi tão bem recebida por profissionais da área de teste, que mantiveram um *processo* que emprega apenas uma técnica de Gestão do Conhecimento – neste caso, centrada em *Feedback* – como mais bem aplicável a um processo de teste de software.

5.5 Considerações Finais

Neste capítulo foram comparados os resultados dos dois estudos realizados nesta dissertação: a Análise Crítica e o *Survey*. Os principais pontos em comum obtidos foram associados, assim como as suas diferenças.

Ainda neste capítulo foi explorada a combinação dos conceitos-chave das três abordagens de Gestão do Conhecimento investigadas nesta dissertação. Um *processo* único foi proposto, o *Processo Combinado*, unindo estes conceitos e tomando como base o *Processo A*. Uma nova investigação foi conduzida com os participantes do primeiro *Survey*, com a intenção de avaliar a aplicabilidade deste *processo*, em comparação com o *processo* a partir do qual o *Processo Combinado* foi originado, ou seja, o *Processo A*.

O último assunto apresentado foi uma síntese de trabalhos relacionados, na qual foram mencionadas as principais semelhanças e diferenças entre estes trabalhos e o trabalho descrito nesta dissertação.

O próximo capítulo apresenta as conclusões desta dissertação em conjunto com as contribuições obtidas e as respectivas limitações.

Capítulo 6

CONCLUSÃO

Conforme discutido nos capítulos anteriores, a realização da atividade de teste de software resulta na melhoria da qualidade dos artefatos produzidos, desde que aplicada de maneira sistemática e apoiada por um processo bem estruturado (METTE; HASS, 2008). Modelos de maturidade podem ser utilizados como um guia para a definição do processo de teste, pois fornecem diretrizes sobre as melhores práticas utilizadas na área, permitindo a evolução gradual do processo de uma organização.

Como a atividade de teste de software é dependente do conhecimento daqueles que a realizam (XUE-MEI et al., 2009), espera-se obter benefícios com a aplicação de Gestão do Conhecimento em um processo de teste de software. Por exemplo, redução de custos, melhoria de qualidade, compartilhamento do conhecimento, entre outras.

Diversos estudos discutem o tema Gestão do Conhecimento no contexto de teste de software. Neste trabalho, foram investigados três destes estudos (COLOMO-PALACIOS et al., 2014; ANDRADE et al., 2013; XUE-MEI et al., 2009), com o objetivo de identificar o grau de aplicabilidade de abordagens de Gestão do Conhecimento aplicadas em um processo de teste de software, segundo o ponto de vista de profissionais com experiência em liderança e gestão de equipes de teste.

Cada estudo investigado descreve um *framework* de Gestão do Conhecimento aplicado em teste de software, que originou um processo subjacente representando um *Processo de Gestão do Conhecimento*. Estes processos foram avaliados em duas fases: uma Análise Crítica, realizada com base na experiência prática e teórica da autora desta dissertação; e um *Survey*, com a participação de profissionais com ampla experiência na área de teste de software. Além disso, também foi explorada a possibilidade de reunir, em um único processo, os conceitos-chave de Gestão do Conhecimento extraídos dos três *frameworks* investigados. Mais uma vez, profissio-

nais com experiência em teste de software apoiaram este último *Survey*.

Os resultados decorrentes dos dois estudos realizados, a Análise Crítica e o *Survey*, indicam que um processo de Gestão do Conhecimento com base em *feedback* - *Processo A*, extraído do *framework* proposto por Colomo-Palacios et al. (2014) - é a melhor opção para gerenciar o conhecimento de uma equipe de teste de software e apoiar suas atividades, seja na escolha do profissional mais indicado para cada tarefa, no acompanhamento dos testes realizados, ou na avaliação dos artefatos produzidos. Uma investigação adicional revelou que a combinação de conceitos de Gestão do Conhecimento em um único processo, mais complexo, pode não ser uma opção adequada para aprimorar os pontos ineficazes já existentes. Um subconjunto dos participantes do primeiro *Survey* avaliou o chamado *Processo Combinado* e os resultados obtidos, se comparados com os da Análise Crítica e com os do *Survey*, ainda apontam para o *Processo A* como mais aplicável a um processo de teste de software.

6.1 Contribuições

As principais contribuições deste trabalho são:

- A identificação e elaboração de *processos* de Gestão do Conhecimento aplicados em teste de software, a partir de *frameworks* de Gestão do Conhecimento existentes.
- A avaliação da aplicabilidade dos processos de Gestão do Conhecimento elaborados em relação a um processo de teste de software genérico, com base na opinião de profissionais com experiência em liderança e gestão de equipes de teste de software.
- Uma análise de viabilidade da combinação dos principais conceitos de Gestão do Conhecimento investigados em um único *processo*.

6.2 Limitações e Ameaças à Validade

Com relação às limitações desta pesquisa, além da análise estatística simples realizada, principalmente devido à pequena amostra obtida, foram identificados alguns problemas que podem ameaçar os procedimentos adotados e os resultados obtidos.

O primeiro diz respeito à correção e integridade dos processos de Gestão do Conhecimento concebidos. Como explicitado na Seção 3.2, os fluxogramas foram obtidos a partir da interpretação das principais características presentes nos três *frameworks* originais, conforme as

descrições de seus autores (COLOMO-PALACIOS et al., 2014; ANDRADE et al., 2013; XUE-MEI et al., 2009). No entanto, não há nenhuma garantia de que a documentação disponível fosse precisa e fornecia detalhes suficientes para a concepção dos processos de Gestão do Conhecimento subjacentes. Para reduzir essa ameaça, os fluxogramas elaborados foram enviados para os autores originais. O autor de um dos *frameworks* analisados (COLOMO-PALACIOS et al., 2014) enviou um *feedback* positivo sobre o processo originado de seu trabalho, o que foi considerado como um indicativo da correta interpretação a respeito dos processos subjacentes dos *frameworks* analisados. Ainda é importante ressaltar que diferentes interpretações levariam a resultados distintos dos observados nesta dissertação.

O segundo está relacionado com a experiência pessoal diversificada dos participantes da pesquisa. Isso pode ter influenciado os resultados dos dois *Surveys*. Para reduzir essa ameaça, para o primeiro *Survey* analisaram-se as respostas de um subgrupo de participantes, considerando apenas aqueles que declararam ter experiência na área de testes em empresas e também em gestão de equipes de teste, como explicado na Seção 4.3.1. No entanto, para o segundo *Survey*, devido ao pequeno número de respostas, não foi feita distinção entre os perfis dos participantes.

Outra limitação diz respeito às respostas obtidas nos *surveys*. Para o primeiro, houve o risco de maior concentração de respostas em um *Survey* em detrimento dos outros, pois os participantes foram convidados a escolher, ao acaso, apenas uma versão do *Survey* para responder. Contudo, as respostas estiveram distribuídas proporcionalmente entre as três versões. Para o segundo *Survey*, a limitação esteve relacionada às poucas respostas obtidas. Para reduzir o impacto desse pequeno número, foi aceita a resposta de um profissional que não respondeu o primeiro *Survey*, mas que possui vasta experiência em teste de software. Além disso, dada a natureza exploratória dessa terceira avaliação (descrita na Seção 5.3.2), a opinião de especialistas possibilitou delinear algumas conclusões em relação à viabilidade da união conceitos-chave de Gestão do Conhecimento em um único processo.

6.3 Trabalhos Futuros

Mais pesquisas são necessárias de maneira a contribuir para a evolução do trabalho apresentado nesta dissertação. São elas:

- Explorar *frameworks* e processos de Gestão do Conhecimento mais abrangentes, com respeito ao processo de desenvolvimento de software como um todo, e não somente relacionados a área de Teste de Software.

- Investigar outras combinações de técnicas de Gestão do Conhecimento, a fim de permitir maiores avanços em um processo de teste de software. Exemplos seriam a incorporação das técnicas presentes nos *Processos A e C (Feedback e Documentos do Conhecimento)* na estrutura do *Processo B* formando um único *processo*, e a incorporação das técnicas dos *Processos A e B* na estrutura do *Processo C*.
- Envolver desenvolvedores de software para a criação de uma estrutura extensível que permita experimentar a aplicação dos conceitos e práticas de Gestão do Conhecimento em equipes de teste do mundo real.
- Empreender pesquisas com profissionais que avaliem todas as práticas pertencentes ao processo de teste utilizado como base, e não somente um subgrupo de práticas. Assim, as conclusões poderão ser relacionadas ao processo de teste como um todo.

REFERÊNCIAS

- ABDULLAH, R.; ERI, Z.; TALIB, A. A model of knowledge management system in managing knowledge of software testing environment. In: *5th Malaysian Conference in Software Engineering (MySEC), 2011*. [S.l.: s.n.], 2011. p. 229–233.
- ACKOFF, R. L. From data to wisdom. *Journal of Applied Systems Analysis*, v. 16, p. 3–9, 1989.
- ANDRADE, J. et al. An architectural model for software testing lesson learned systems. *Information and Software Technology*, v. 55, n. 1, p. 18 – 34, 2013. ISSN 0950-5849. Special section: Best papers from the 2nd International Symposium on Search Based Software Engineering 2010.
- BEIZER, B. *Software Testing Techniques*. 2nd. ed. New York, NY, USA: Van Nostrand Reinhold Co., 1990. ISBN 0-442-20672-0.
- BJØRNSON, F. O.; DINGSØYR, T. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, Butterworth-Heinemann, Newton, MA, USA, v. 50, n. 11, p. 1055 – 1068, 2008. ISSN 0950-5849.
- CAMARGO, K. G. *Elaboração de um Processo de Teste com Base em um Estudo de Caso real*. Dissertação (Mestrado) — Departamento de Computação, Universidade Federal de São Carlos, São Carlos - Brazil, 2012.
- CAMARGO, K. G.; FERRARI, F. C.; FABBRI, S. C. P. F. Identifying a subset of TMMi practices to establish a streamlined software testing process. In: *Proceedings of the 27th Brazilian Symposium on Software Engineering (SBES)*. Brasilia/DF - Brazil: IEEE Computer Society, 2013. p. 137–146. ISBN 978-0-7695-5165-4.
- COLOMO-PALACIOS, R. et al. Sabumo-dtest: Design and evaluation of an intelligent collaborative distributed testing framework. *Computer Science and Information Systems*, v. 11, n. 1, p. 29–45, 2014.
- COSTA NETO, P. L. O. *Estatística*. 2nd. ed. [S.l.]: Editora E. Blücher, 1977.
- DAVENPORT, T. H.; PRUSAK, L.; PRUSAK, L. *Working Knowledge: How Organizations Manage What They Know*. Boston, Mass.: Harvard Business School Press, 2000. 240 p. ISBN 1-57851-301-4.
- DELAMARO, M. D.; MALDONADO, J. C.; JINO, M. *Introdução ao Teste de Software*. 1nd. ed. São Carlos, Brazil: Editora Campus, 2007. ISBN 978-85-352-2634-8.

- DINGSØYR, T.; CONRADI, R. Knowledge management systems as a feedback mechanism in software development processes: A search for success criteria. In: *Feedback and Evolution in the Software Process*. [S.l.: s.n.], 2002.
- DUSTIN, E.; GARRETT, T.; GAUF, B. *Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2009. ISBN 0321580516, 9780321580511.
- FURTADO, A. P. C. C. et al. Mpt. br: A brazilian maturity model for testing. In: *Proceedings - International Conference on Quality Software*. [S.l.: s.n.], 2012. p. 220–229.
- GARLAND, R. The Mid-Point on a Rating Scale: Is it Desirable. *Marketing Bulletin*, n. 2, p. 66–70, 1991.
- HANSEN, M. T.; NOHRIA, N.; TIERNEY, T. What's your strategy for managing knowledge. *Harvard Business Review*, v. 77, n. 2, p. 106 – 116, 1999.
- HÖHN, E. N. *KITest: Um Arcabouço de Conhecimento e Melhoria de Processo de Teste*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP - Brazil, jun 2011.
- KARHU, K.; TAIPALE, O.; SMOLANDER, K. Investigating the relationship between schedules and knowledge transfer in software testing. *Information and Software Technology*, Butterworth-Heinemann, Newton, MA, USA, v. 51, n. 3, p. 663–677, 2009. ISSN 0950-5849.
- KASURINEN, J. et al. A self-assessment framework for finding improvement objectives with iso/iec 29119 test standard. In: *Proceedings of 18th European Conference EuroSPI 2011*. [S.l.]: Springer Berlin Heidelberg, 2011. p. 27–29.
- LIAO, S. hsien. Knowledge management technologies and applications - literature review from 1995 to 2002. *Expert Systems with Applications*, v. 25, n. 2, p. 155 – 164, 2003. ISSN 0957-4174.
- MATHUR, A. P. *Foundations of Software Testing*. 1st. ed. Canada: Addison-Wesley Professional, 2008.
- METTE, A.; HASS, J. Testing processes. In: *IEEE International Conference on Software Testing Verification and Validation Workshop, 2008. ICSTW '08*. [S.l.: s.n.], 2008. p. 321–327.
- MICHAELIS. 2014. Online. <http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=conhecimento> – Acessado em 10/04/2015.
- MORELL, L. A theory of fault-based testing. *IEEE Transactions on Software Engineering*, v. 16, n. 8, p. 844–857, Aug 1990. ISSN 0098-5589.
- MPT.BR. *Melhoria do Processo de Teste Brasileiro*. 2011.
- MYERS, G. J.; SANDLER, C. *The Art of Software Testing*. 2nd. ed. Hoboken/NJ - USA: John Wiley & Sons, 2004. ISBN 0471469122.
- NONAKA, I.; TAKEUCHI, H. *The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation*. [S.l.]: Oxford University Press, 1995. ISBN 9780195092691.

O'LEARY, D. Enterprise knowledge management. *Computer*, v. 31, n. 3, p. 54–61, Mar 1998. ISSN 0018-9162.

OMG. *Business Process Model and Notation*. 2014. Online. <http://www.bpmn.org/> – Acessado em 10/04/2015.

POLANYI, M. *The Tacit Dimension*. [S.l.]: University Of Chicago Press, 1966. ISBN 9780226672984.

ROWLEY, J. The wisdom hierarchy: Representations of the dikw hierarchy. *Journal of Information Science*, Sage Publications, Inc., v. 33, n. 2, p. 163–180, 2007.

RUS, I.; LINDVALL, M. Guest editors' introduction: Knowledge management in software engineering. *IEEE Software*, IEEE Computer Society, Los Alamitos, CA, USA, v. 19, n. 3, p. 26–38, 2002. ISSN 0740-7459.

SECCHI, P.; CIASCHI, R.; SPENCE, D. *A Concept for an ESA Lessons Learned System*. Noordwijk: The Netherlands: ESTEC, 1999.

Software Engineering Institute. *CMMI for Development, Version 1.3*. Bedford/MA - USA, 2010. <http://www.sei.cmu.edu/reports/10tr033.pdf> - accessed on 18/05/2015.

SOUZA, E. F. *Knowledge Management Applied to Software Testing: an Ontology Based Framework*. Tese (Doutorado) — National Institute for Space Research, INPE, 2014.

TAKEUCHI, H.; NONAKA, I. *Hitotsubashi on Knowledge Management*. [S.l.]: Wiley, 2004. ISBN 0470820748.

TIAN, J.; NAKAMORI, Y.; WIERZBICKI, A. P. Knowledge management and knowledge creation in academia: A study based on surveys in a japanese research university. *Journal of Knowledge Management*, v. 13, n. 2, p. 76–92, 2009.

TMMi Foundation. *Test Maturity Model integration (TMMi) (Version 3.1)*. 2010. 1-181 p.

WOHLIN, C. et al. *Experimentation in Software Engineering: An Introduction*. 1. ed. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN 0-7923-8682-5.

XU-XIANG, L.; WEN-NING, Z. The pdca-based software testing improvement framework. In: *International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA), 2010*. [S.l.: s.n.], 2010. p. 490–494.

XUE-MEI, L. et al. Research and implementation of knowledge management methods in software testing process. In: *World Congress on Computer Science and Information Engineering, 2009 WRI*. [S.l.: s.n.], 2009. v. 7, p. 739–743.

GLOSSÁRIO

BPMN – *Business Process Model and Notation*

CMMI – *Capability Maturity Model Integration*

DIKW – *Data, Information, Knowledge, Wisdom*

MPT.Br – *Melhoria do Processo de Teste de Software Brasileiro*

TMMi – *Test Maturity Model integration*

Anexo A

ANÁLISE CRÍTICA DAS ABORDAGENS DE GESTÃO DO CONHECIMENTO

Este anexo apresenta a Análise Crítica completa, complementando o material apresentado no Capítulo 3. Mais adiante, também será apresentada a estrutura do TMMi que é utilizada no trabalho de Camargo, Ferrari e Fabbri (2013), compondo o processo de teste de software enxuto.

A.1 Análise Crítica

A Figura A.1 apresenta as questões formuladas para a realização da investigação das práticas.

Questionamentos	
a	Atividade envolve tomada de decisão?
b	Atividade exige um ator com conhecimento especializado?
c	Atividade pode ser automatizada?
d	Atividade requer a consulta a um repositório de dados (base histórica dos projetos)?
e	Atividade requer o envolvimento de outras pessoas/equipes?
f	Atividade pode influenciar outras atividades (futuras)?

Figura A.1: Questões Para Análise Crítica

A seguir são apresentadas as tabelas que contemplam a Análise Crítica (dividida em 13 partes).

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP1.2 Identificar riscos do produto <i>Riscos do produto (específicos de teste) são identificados e documentados</i>	a	Não neste momento. Esta etapa é relacionada apenas com o estudo e análise do projeto e a identificação dos riscos existentes.	O processo pode ser utilizado como páginas amarelas, para encontrar especialistas no domínio (consulta de competências).	O processo não é de grande valia para a realização desta atividade, pois concentra-se principalmente nos documentos de conhecimento, documentos estes, que armazenam conhecimento operacional para apoiar as atividades dos testers.
	b	Sim. Fundamentalmente pessoas com conhecimento no domínio. O conhecimento técnico também é importante. Pois caso contrário, riscos relevantes podem não ser identificados.	Projetos anteriores também podem ser consultados, para que os seus processos sejam averiguados e identificados riscos que existiram no passado e que ainda podem ser pertinentes ao novo projeto.	Entretanto, existe a possibilidade de consulta às informações pessoais da equipe, o que funciona como páginas amarelas, para que seja possível encontrar especialistas no domínio.
	c	Não. Como envolve análise do projeto e a interpretação do mesmo, não é possível automatizar a etapa.		
	d	Sim, é importante que exista um repositório de dados que forneça um histórico dos riscos identificados em projetos passados. Projetos semelhantes podem ter riscos semelhantes.		
	e	Sim. Requer o envolvimento de especialistas no domínio, principalmente os envolvidos com <i>design</i> e desenvolvimento, também a equipe gerencial, que detém a visão completa do projeto.		
	f	Sim. Os resultados obtidos com a identificação dos riscos podem mudar características do projeto. Uma identificação ineficiente também pode trazer prejuízos futuros relevantes ao projeto.	Aplicável	Extremamente aplicável
SP1.3 Analisar riscos do produto <i>Riscos do produto (específicos de teste) são avaliados, categorizados e priorizados utilizando categorias e parâmetros predefinidos.</i>	a	Sim. É uma atividade que pode antecipar problemas, e também soluções a serem adotadas. Esta atividade pode também determinar alterações no escopo do projeto.		
	b	Na análise, o conhecimento do domínio pode ser menos relevante, pois os riscos já foram identificados pelos especialistas. É necessário a averiguação das informações levantadas durante a identificação dos riscos.		Devido às características das <i>lessons learned</i> , e das informações sobre o projeto, o processo pode ser relevante na análise dos riscos. Pois os problemas que ocorrerem em projetos anteriores, e as soluções que foram adotadas para resolvê-los podem auxiliar na criação dos planos de mitigação. O processo também oferece consulta ao mapa do projeto, que contém informações relevantes, principalmente sobre especialistas.
	c	Entretanto por envolver um plano de mitigação, o especialista pode auxiliar.		
	d	Não. A análise dos riscos identificados é uma tarefa subjetiva, e depende de fatores que nem sempre terão os mesmos pesos em todos os projetos. Além disso, é necessário um plano de mitigação para cada risco identificado. Sendo fundamental a intervenção humana, o que impede a automatização da atividade.	Pensando na prática isoladamente, o processo oferece suporte à consulta de competências e informações de projetos anteriores. Entretanto, nesta atividade, o processo não fornecerá informações adicionais (além daquelas levantadas na atividade anterior) para a análise dos riscos e criação do plano de mitigação.	Pensando na prática isoladamente, o processo oferece suporte à consulta de informações pessoais (dos membros das equipes) e à algumas informações de projetos anteriores. Entretanto, nesta atividade, o processo não fornecerá informações adicionais (além daquelas levantadas na atividade anterior) para a análise dos riscos e criação do plano de mitigação.
	e	Sim. Assim como na atividade anterior, a análise dos riscos também requer consulta aos projetos anteriores, pois exemplos podem ser levados em consideração, para justificar a tomada de decisão.		
	f	Nesta atividade além da equipe gerencial, os clientes podem ser envolvidos devido a necessidade de alteração de escopo de projeto.	Pouco aplicável	Aplicável
	Sim. Riscos analisados incorretamente, e planos de mitigação que não contemplam as ações corretas podem acarretar em problemas futuros no projeto.			

Tabela A.1: Análise Crítica

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
<p>SP2.1 Identificar elementos e características a serem testados</p> <p>Os itens e características a serem testados, e a não serem testados, são identificados com base nos riscos do produto (específicos de teste).</p>	<p>a Sim. Nesta atividade o teste será moldado de acordo com as funcionalidades: o que será e o que não será testado, a partir dos riscos que foram analisados.</p>	<p>Para a identificação realizada nesta atividade, o processo não oferece nenhum tipo de informação relevante. Entretanto, o histórico de projetos anteriores pode ser consultado, assim como o feedback dos processos e execuções. Fornecendo informações relevantes sobre os resultados obtidos. O processo pode ser utilizado também como páginas amarelas.</p>	<p>Este processo fornece informações relevantes de projetos anteriores, principalmente de seus resultados (informações contidas nas lessons learned). Auxiliando a tomada de decisão que a atividade requer. Também é possível a utilização do processo como páginas amarelas.</p>	<p>O processo oferece informações de projetos anteriores, que podem ser consultados, entretanto estas informações podem não contemplar acertos e erros do projeto. Apesar do pouco auxílio na atividade de identificação, o processo possibilita a busca de especialistas, através das páginas amarelas.</p>
	<p>b É importante um ator com conhecimento (pelo menos mediano) no domínio.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>c Não. Novamente uma atividade que envolve ponderações que podem vir a ser diferentes das encontradas em projetos anteriores. Sendo necessária a interação humana.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>d Sim. Pode existir um relacionamento "de-para" que informa a partir de riscos mais conhecidos quais são os itens que devem ser testados, e quais não. Entretanto deve-se estudar muito bem este relacionamento, pois o projeto pode ter um escopo bastante específico, que se diferencia de projetos anteriores.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>e Não. Nesta atividade não faz-se necessário o envolvimento de outras pessoas e/ou equipes.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>f Sim. A decisão de testar ou não testar determinado item poderá acarretar em problemas de projeto no futuro.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
<p>SP2.2 Definir a abordagem de teste</p> <p>A abordagem de teste é definida para mitigar os riscos do produto (específicos de teste) que foram identificados e priorizados.</p>	<p>a Sim. Esta atividade consiste em uma série de tomadas de decisão envolvendo custo, prazo, equipe, entre outras decisões de projeto. Além identificação das técnicas que serão utilizadas para a realização do teste.</p>	<p>Quanto à abordagem de teste, o processo não fornece nenhuma informação relevante. Mas informações relacionadas aos processos de projetos anteriores podem ser consultadas, assim como o feedback dos processos também podem ser recuperados. Tais informações podem servir como base para o projeto atual. O processo pode também ser utilizado como páginas amarelas para a busca de especialistas no domínio.</p>	<p>Devido ao suporte às <i>lessons learned</i>, informações relevantes quanto ao sucesso das abordagens utilizadas em projetos anteriores podem ser consultadas e levadas em consideração na tomada de decisão para o projeto atual. Especialistas no domínio também podem ser encontrados através das páginas amarelas.</p>	<p>Assim como na atividade anterior, este processo permite a consulta de informações de projetos passados. Entretanto tais informações não ser suficientes para a tomada de decisão necessária nesta atividade. Apesar da pouca utilidade na identificação da abordagem de teste, o processo possibilita a busca de especialistas, através das páginas amarelas.</p>
	<p>b Sim. Atores com conhecimento no domínio podem contribuir com detalhes que influenciarão positivamente as decisões de projeto. Além disso, é fundamental um ator com experiência gerencial para a realização desta atividade.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>c Não. Devido à características que são intrínsecas a cada projeto, a definição da abordagem de teste não pode ser automatizada.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>d Sim. Pode haver necessidade de consultar informações de projetos anteriores (custo, prazo, equipe, entre outras), para que as decisões referentes ao projeto atual sejam balizadas.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>e Sim. É uma atividade que além de tomadas de decisão envolve diversas negociações com outras equipes.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>
	<p>f Sim. Esta atividade pode influenciar positiva ou negativamente praticamente todos os aspectos futuros do projeto.</p>	<p>Aplicável</p>	<p>Extremamente aplicável</p>	<p>Pouco aplicável</p>

Análise Crítica (Parte 1)

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
<p>a</p> <p>Sim. Nesta atividade será definido o momento adequado para que a finalização do teste ocorra.</p> <p>Não. O conhecimento do domínio não terá relevância nesta atividade. Uma vez que todas as informações necessárias já foram coletadas, é necessário somente um ator com experiência gerencial.</p>	<p>O processo não armazena informações relevantes sobre cobertura de testes, e acompanhamento de riscos dos projetos anteriores (apesar de armazenar o feedback das execuções, e os resultados obtidos). Desta forma, mesmo com a possibilidade de verificar informações de projetos passados, não se pode afirmar que haverá grande contribuição para esta atividade.</p>	<p>Critérios de parada e outras informações que são relevantes para esta atividade fazem parte das lessons learned, que compõem este processo. Portanto, projetos anteriores podem ser consultados e tais informações verificadas.</p>	<p>O processo não traz informações relevantes para a realização desta atividade.</p>	
<p>b</p> <p>Não. Esta atividade depende de particularidades do projeto, que podem variar de acordo com a criticidade e complexidade. Exigindo que cada caso seja avaliado individualmente.</p> <p>Não. Da mesma maneira que a questão anterior, esta atividade depende de particularidades do projeto, que podem variar de acordo com a criticidade e complexidade. Portanto, cada projeto deve ser avaliado individualmente.</p> <p>Sim. Como se trata de tomada de decisões gerenciais, outras equipes com participação no projeto, e até mesmo o cliente devem ser envolvidos.</p> <p>Sim. Uma falha nesta atividade pode deixar de fora do teste elementos importantes do sistema.</p>	<p>Pouco aplicável</p>	<p>Aplicável</p>	<p>Não aplicável</p>	
<p>c</p> <p>Sim, a definição do cronograma está diretamente ligada à tomada de decisão.</p> <p>Não. Esta é uma atividade que se baseará em informações já coletadas. Portanto o conhecimento do domínio não é relevante. Um ator com perfil gerencial é necessário.</p> <p>Não. Apesar de ter acesso às informações necessárias para a realização desta atividade, é fundamental o alinhamento de prazos entre os recursos disponíveis. Ajustes de cronograma podem ocorrer devido a fatores externos.</p> <p>Não. O cronograma é um dado bem restrito a um determinado projeto. Não faz sentido consultar uma base histórica para recuperar informações de projetos anteriores para este quesito.</p> <p>Sim. Esta atividade novamente está ligada à tomada de decisões gerenciais. Desta forma, outras equipes com participação no projeto, e principalmente o cliente devem ser envolvidos.</p>	<p>O processo, e as informações armazenadas por ele não apoiam a realização desta atividade.</p>	<p>O processo, e as informações armazenadas por ele não apoiam a realização desta atividade.</p>	<p>O processo, e as informações armazenadas por ele não apoiam a realização desta atividade.</p>	
<p>d</p> <p>Sim. Um cronograma que não seja o ideal (para mais ou para menos), traz impactos negativos ao projeto como um todo.</p>	<p>Não aplicável</p>	<p>Não aplicável</p>	<p>Não aplicável</p>	

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP4.2 Planejar a equipe de teste <i>Plano para a disponibilidade dos recursos humanos necessários que têm o conhecimento e as habilidades necessárias para realizar o teste.</i>	a Sim. Nesta atividade serão decididos quais são os recursos humanos mais adequados. Além de verificar quais serão as necessidades destes recursos (como por exemplo, treinamentos).	<p>O processo possui acesso às páginas amarelas (seleção de competências), que é utilizada eficientemente para esta atividade.</p>	<p>O processo possui acesso às páginas amarelas (consulta ao mapa do projeto), que é utilizada eficientemente para esta atividade.</p>	<p>O processo possui acesso às páginas amarelas (consulta às informações pessoais), que é utilizada eficientemente para esta atividade.</p>
	b Não. Conhecimento no domínio não é relevante para esta atividade.			
	c Não. Talvez a escolha dos recursos quanto ao conhecimento pode ser automatizada, mas existem outros fatores como disponibilidade, que influenciam no palejamento da equipe, e que nem sempre podem ser previstos de maneira automatizada.			
	d Sim. É preciso consultar um repositório para coletar informações sobre o conhecimento que os recursos possuem.			
	e Não. A decisão sobre quais recursos irão compor o projeto é interna à equipe de testes.			
	f Sim. A escolha de recursos inadequados pode interferir na qualidade e no andamento do projeto.			
SP4.5 Estabelecer o plano de teste <i>O plano de teste é estabelecido e mantido como base para o gerenciamento de testes. Os resultados das práticas anteriores são documentados em um plano global de teste, ligando as informações de uma maneira lógica.</i>	a Não. Esta atividade é apenas uma formalização de todas as atividades realizadas anteriormente. As decisões já foram tomadas previamente.	<p>Extremamente aplicável</p>	<p>Extremamente aplicável</p>	<p>Extremamente aplicável</p>
	b Não. Nenhum ator com conhecimento ou experiência específica faz-se necessário.			
	c Sim. Uma vez que todas as decisões foram tomadas, e as informações já estão coletadas, esta atividade seria "criar um baseline" do projeto.			
	d Não. Nenhuma consulta a base histórica faz-se necessária.			
	e Não. As informações que são relevantes para outras equipes já foram distribuídas.			
	f Não. Esta atividade não gera nenhuma informação nova.			
SP1.1 Obter (eliciar) necessidades do ambiente de teste <i>Obter o ambiente de teste, incluindo dados genéricos de teste, necessidades expectativas e restrições.</i>	a Sim. A atividade descreverá quais são os requisitos de ambiente necessários para o teste.	<p>Extremamente aplicável</p>	<p>Extremamente aplicável</p>	<p>Extremamente aplicável</p>
	b Sim, nesta atividade um especialista no domínio é necessário.			
	c Não. Devido à individualidade dos requisitos do ambiente, não é possível automatizar esta atividade.			
	d Sim. É relevante averiguar quais foram as necessidades de ambiente de projetos anteriores, para antecipar possíveis problemas de custo, prazo, ou até mesmo incompatibilidade de recursos de ambiente.			
	e Não. Estas informações são coletadas e decididas internamente à equipe de testes.			
	f Sim. A determinação incorreta do ambiente pode influenciar no custo e prazo do projeto.			

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP.1.3 Analisar os requisitos do ambiente de teste <i>Analisar os requisitos para garantir que eles são necessários, suficientes e viáveis.</i>	a Sim. A atividade decidirá se o ambiente existente é suficiente para suprir as necessidades de ambiente do projeto em questão.	O processo não dá suporte a realização desta atividade.	O processo não dá suporte a realização desta atividade.	O processo não dá suporte a realização desta atividade.
	b Sim, nesta atividade um especialista no domínio é necessário, para identificar corretamente se as necessidades de ambiente foram atingidas.			
	c Não. Esta atividade pode exigir algumas análises quanto a integração dos ambientes, além da averiguação de possíveis riscos ligados ao ambiente, não podendo ser automatizada.			
	d Não. A compatibilidade de ambiente com os requisitos é uma característica que não é influenciada pelo repositório de dados.			
	e Sim. Envolvimento da equipe que controla o ambiente para negociações sobre mudanças, e possivelmente alguma interação com o cliente.			
	f Sim. A escolha do ambiente errado influencia. E também a identificação da necessidade de alterações no ambiente pode influenciar no custo e prazo do projeto.			
SP.1.2 Identificar e priorizar casos de teste <i>Caso de teste são identificados e priorizados utilizando técnicas de design de teste</i>	a Sim. Nesta atividade são definidas quais serão as entradas necessárias e as saídas esperadas. Além disso existe a tomada de decisão quanto à priorização dos casos de teste.	Este processo é preparado para receber os planos de teste. Oferece apoio para consulta dos planos de projetos anteriores, e também é possível encontrar pessoas com competências necessárias para a realização da atividade.	O processo não apoia a criação de planos de teste diretamente, pois somente são armazenadas informações do projeto. Mas oferece meios de encontrar lessons learned relacionadas a projetos anteriores, que podem auxiliar em algumas questões. Também é possível encontrar especialistas através das páginas amarelas.	O processo não realiza a criação de planos de teste diretamente, entretanto, informações relevantes podem ser encontradas nos documentos de conhecimento. O processo também oferece meios de encontrar pessoas que possuem conhecimento necessário para realização da atividade.
	b Sim, conhecimento no domínio é fundamental.			
	c Não. A criação dos casos de teste pode ser automatizada, mas a priorização necessita de intervenção humana.			
	d Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade.			
	e Não. A identificação dos casos de teste é uma atividade interna à equipe de testes.			
	f Sim. Casos de teste insuficientes podem influenciar todo o projeto.			
SP.1.3 Identificar dados de teste específicos necessários <i>São identificados os dados de testes específicos necessários para suportar as condições de teste e execução de casos de teste.</i>	a Sim, decisão sobre a criação dos casos de teste são tomadas nesta atividade.	Extremamente aplicável	Aplicável	Extremamente aplicável
	b Sim, conhecimento no domínio é fundamental para a correta identificação dos dados de teste.			
	c Não. A seleção dos dados de teste específicos não pode ser automatizada, pois é uma atividade restrita ao escopo do projeto.	Este processo é preparado para receber os planos de teste. Oferece apoio para consulta dos planos de projetos anteriores, e também é possível encontrar pessoas com competências necessárias para a realização da atividade.	O processo não apoia a criação de planos de teste diretamente, pois somente são armazenadas informações do projeto. Mas oferece meios de encontrar lessons learned relacionadas a projetos anteriores, que podem auxiliar em algumas questões. Também é possível encontrar especialistas através das páginas amarelas.	O processo não realiza a criação de planos de teste diretamente, entretanto, informações relevantes podem ser encontradas nos documentos de conhecimento. O processo também oferece meios de encontrar pessoas que possuem conhecimento necessário para realização da atividade.
	d Sim. A consulta (e reaproveitamento) a casos de teste de projetos anteriores pode facilitar a realização da atividade. Entretanto há a necessidade de adequar tais casos de teste para os requisitos individuais deste projeto.			
	e Não. A identificação dos dados de teste é uma atividade interna à equipe de testes.			
	f Sim. Casos de teste insuficientes podem influenciar todo o projeto.	Extremamente aplicável	Aplicável	Extremamente aplicável

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
<p>a</p> <p>Não. Esta atividade está relacionada à criação de scripts automatizados de teste. Decisões e prioridades (relacionadas à criação dos casos de teste) já foram realizadas nas etapas anteriores.</p>	<p>Não. Este processo armazena informações sobre os planos de teste de projetos anteriores (processos de teste). Estas informações podem ser consultadas, para reaproveitamento dos scripts de teste automatizados já criados.</p>	<p>Não aplicável</p>	<p>Aplicável</p>	<p>Podem existir scripts automatizados de projetos anteriores presentes nos documentos de conhecimento. Tais scripts podem facilmente ser recuperados de maneira a auxiliar a realização desta atividade.</p>
<p>b</p> <p>SP2.1 Desenvolver e priorizar os procedimentos de teste</p> <p>Os procedimentos de teste são desenvolvidos e priorizados.</p>	<p>Não. Os casos de teste e dados dos casos de teste (que exigiam atores com conhecimento) já foram criados. Esta atividade visa ordenar os casos de teste formando os scripts automatizados, portanto não é necessário atores com conhecimento específico, uma vez que as informações já foram coletadas.</p> <p>Sim. Como se trata de coleta de informações levantadas em estágios anteriores, existe a possibilidade de automatização.</p> <p>A geração dos scripts automatizados pode requerer uma consulta aos projetos anteriores, para reutilizar o que já foi criado no passado.</p> <p>Não. Esta é uma atividade interna à equipe de testes.</p> <p>Não. Nenhum dado novo é criado neste momento.</p>	<p>Aplicável</p>	<p>Não aplicável</p>	<p>Aplicável</p>
<p>c</p> <p>SP2.4 Desenvolver cronograma de execução do teste</p> <p>É desenvolvido um cronograma de execução do teste que descreve a sequência em que os procedimentos de teste serão executados.</p>	<p>Sim. Uma vez que o cronograma geral do teste já está definido (datas inicial e final), esta atividade define quais recursos humanos serão alocados em cada fase do teste, e define a ordem em que cada caso de teste será executado, respeitando as datas inicial e final.</p> <p>Não. Uma vez que os dados de priorização já foram coletados, e os planos de teste já estão criados, cabe a esta atividade criar o cronograma e alocar recursos aos casos de teste, para a realização da atividade. Conhecimento no domínio não é necessário, mas é importante o conhecimento das experiências dos recursos envolvidos.</p> <p>Não. Em projetos pequenos, nos quais há apenas a participação de um tester, podemos considerar que esta atividade pode ser automática, entretanto, se generalizarmos o caso para projetos em que mais de um tester precisa ser alocado, existe a necessidade de um estudo das experiências detidas pelos testers, para que possam ser alocados corretamente aos casos de teste.</p> <p>Sim. Pode ser necessário a verificação de projetos anteriores, quanto ao cronograma e quanto aos recursos envolvidos.</p>	<p>Este processo é preparado para receber os planos de teste, e também a associar os recursos aos processos (planos de teste). A partir deste processo é possível encontrar profissionais com as competências necessárias, através da busca nas páginas amarelas.</p>	<p>O processo não apoia a criação de planos de teste diretamente, pois são armazenadas somente as informações do projeto. Mas oferece meios de encontrar lessons learned relacionada a projetos anteriores, que podem conter informações sobre os resultados obtidos, acertos e erros relacionados aos cronogramas. Também é possível encontrar especialistas através das páginas amarelas.</p>	<p>O processo auxilia na escolha dos melhores recursos para a realização desta atividade, através da busca nas páginas amarelas. Apesar disso, os documentos de conhecimento não auxiliarão o desenvolvimento do cronograma, por não conter informações relevantes para decisões gerenciais. Pois são específicos para armazenar conhecimento operacional.</p>
<p>d</p>	<p>Sim. Equipes de desenvolvimento e o cliente podem ser envolvidos nesta atividade, pois é importante que ofereçam suporte adequado de acordo com o cronograma previsto, principalmente com relação à ordem dos casos de teste que serão executados (é necessário suporte da equipe de desenvolvimento para correções, e do cliente para esclarecimento de dúvidas funcionais).</p>	<p>Extremamente aplicável</p>	<p>Aplicável</p>	<p>Pouco aplicável</p>
<p>e</p>	<p>Sim. Escolha incorreta de recursos pode influenciar na qualidade do teste realizado. Assim como o cronograma inadequado pode fazer com que a atividade não seja realizada por completo.</p>	<p>Aplicável</p>	<p>Aplicável</p>	<p>Pouco aplicável</p>
<p>f</p>	<p>Sim. Escolha incorreta de recursos pode influenciar na qualidade do teste realizado. Assim como o cronograma inadequado pode fazer com que a atividade não seja realizada por completo.</p>	<p>Extremamente aplicável</p>	<p>Aplicável</p>	<p>Pouco aplicável</p>

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP2.1 Implementar o ambiente de teste <i>Implementar o ambiente de teste, conforme descrito na especificação de requisitos do ambiente de teste e de acordo com o plano definido.</i>	a Não. Esta é uma atividade operacional, que tem como objetivo a montagem do ambiente de teste			
	b Não. As informações necessárias para a implementação do ambiente devem estar presentes no documento de instalação, portanto nenhum conhecimento especializado é necessário.			
	c Sim, pode ser desenvolvido um script de instalação (como os existentes em <i>softwares</i> comerciais para uso doméstico), mas usualmente no mundo corporativo, esta atividade é manual, pois envolve diversas configurações específicas ao ambiente da empresa em questão.	O processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	d Não. Dados de projetos anteriores não influenciarão a realização desta atividade.			
	e Não. Como a instalação ocorrerá no ambiente de testes, apenas a equipe de teste deve ser envolvida.			
	f Instalação incorreta, ou utilização de ambiente incorreto pode influenciar todas as próximas etapas do projeto.	Não aplicável	Não aplicável	Não aplicável
SP2.4 Realizar o teste intake (pré-teste) do ambiente de teste <i>Realizar o pré-teste de ambiente (teste de confiança) para decidir se o ambiente de teste está pronto para ser utilizado.</i>	a Sim. É necessário realizar a análise dos resultados obtidos para definir se realmente existe um problema, se o problema encontrado está no ambiente, ou nos arquivos de instalação.			
	b Não necessariamente. Aqui não é realizado nenhum teste de domínio, mas usualmente apenas pessoas com experiência realizam esta atividade, pois alterações no ambiente são realizadas.			
	c Sim. Existem diversos comandos que podem ser incluídos em um script para avaliação dos resultados da manipulação do ambiente.	No caso deste pré-teste de ambiente estar detalhado no plano de teste (o que normalmente não ocorre), o processo pode ser útil. Mas no geral, o processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	d Não. Esta atividade é específica para o projeto corrente. Não havendo necessidade de olhar os dados dos projetos passados.			
	e Sim. Equipes responsáveis pelo ambiente, e pelo desenvolvimento dos scripts de instalação devem ser envolvidas nos resultados.			
	f Sim. No caso de alterações a serem realizadas nos scripts devido a erros contidos nas instruções de instalação. Problemas nesta atividade podem prejudicar todas as atividades futuras.	Pouco aplicável	Não aplicável	Não aplicável

Análise Crítica (Parte 6)

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP3.2 Executar casos de teste <i>Os casos de teste são executados manualmente, utilizando procedimentos de teste documentados e/ou automatizados, através de scripts de teste.</i>	a Sim. Os procedimentos devem ser seguidos de acordo com o descrito nos casos de teste, mas existe a comparação e análise dos resultados, para averiguar se as condições necessárias foram satisfeitas.	<p>Os casos de testes são executados e acompanhados diretamente pelo processo. Também é possível encontrar especialistas no domínio, através da busca nas páginas amarelas.</p> <p>Extremamente aplicável</p>	<p>O processo não apoia a execução de planos de teste diretamente, pois somente são armazenadas informações do projeto. Acreditado que as lessons learned podem auxiliar (embora não contenham informações muito relevantes para a execução) a solução de dúvidas, e especialistas podem ser encontrados através das páginas amarelas.</p> <p>Aplicável</p>	<p>Informações relevantes para apoiar a execução podem ser encontradas nos documentos de conhecimento que são encontrados no processo. Também é possível realizar a busca por especialistas nas páginas amarelas.</p> <p>Extremamente aplicável</p>
	b Sim. Conhecimento do domínio é relevante na atividade de execução.			
	c Sim. Existem diversos projetos em que a execução dos casos de teste é totalmente automatizada.			
	d Não. Resultados de execuções de projetos anteriores não são consultadas durante a execução do projeto atual.			
	e Não. A atividade de execução é interna à equipe de teste. O envolvimento com outras equipes ocorrerá em etapas posteriores à execução.			
	f Sim. Caso a atividade não seja realizada corretamente, erros podem não ser revelados, impactando a confiabilidade do sistema.			
SP3.3 Relatar incidentes de teste <i>As discrepâncias são relatadas como incidentes de teste, quando há diferenças entre os resultados reais e esperados.</i>	a Sim. Decisões são tomadas para classificar os incidentes quanto à sua severidade e prioridade.	<p>O processo não oferece apoio para esta atividade (de relatar), mas por armazenar os casos de teste, e dar suporte à execução dos mesmos, deve também armazenar informações sobre o incidente, permitindo que o teste seja refeito, após a correção do incidente. Além da identificação de especialistas nas páginas amarelas.</p> <p>Aplicável</p>	<p>As lessons learned contém informações generalizadas sobre os incidentes encontrados, mas não fornece o relato de cada uma deles individualmente. O processo pode ser utilizado nesta atividade para encontrar especialistas no domínio.</p> <p>Aplicável</p>	<p>O processo não apoia o relato dos incidentes, pois as informações armazenadas estão contidas nos documentos de conhecimento. Entretanto, especialistas podem ser encontrados a partir das páginas amarelas.</p> <p>Aplicável</p>
	b Sim. Para classificar corretamente os dados dos incidentes é necessário um especialista no domínio.			
	c Não. Apesar do relato dos incidentes poder ser automatizado, para a classificação dos mesmos é necessário o envolvimento humano.			
	d Não. O relato dos incidentes, e sua classificação não exige que uma consulta a projetos antigos seja realizada.			
	e Sim, é necessária a interação com outras equipes para o relato dos incidentes.			
	f Sim. Relatos incorretos podem dificultar a sua correção.			
SP3.4 Escrever log de teste <i>Logs de teste são escritos para fornecer um registro cronológico com detalhes relevantes sobre a execução dos testes.</i>	a Não. Esta atividade envolve a documentação dos logs que foram coletados ao longo da execução dos casos de teste.	<p>Este processo possui local específico para a execução dos casos de teste, entretanto não fica claro se a documentação dos incidentes é armazenada na ferramenta.</p> <p>Pouco aplicável</p>	<p>O processo não dá suporte à atividade de documentação de incidentes de teste.</p> <p>Não aplicável</p>	<p>O processo não dá suporte à atividade de documentação de incidentes de teste.</p> <p>Não aplicável</p>
	b Não. É necessário apenas a transcrição dos incidentes ocorridos.			
	c Sim. Resultados dos incidentes podem ser coletados e documentados automaticamente.			
	d Não. Nenhuma consulta a base histórica faz-se necessária.			
	e Não. Outras equipes não necessitam estar envolvidas nesta atividade.			
	f Não. Nenhuma interferência em atividades futuras é esperada como resultado desta atividade.			

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)	
SPA.1 Decidir sobre incidentes com o grupo de controle de configuração <i>Ações apropriadas sobre os incidentes de teste são decididas por um conselho de controle de configuração (CCB).</i>	a	Sim. O acompanhamento do projeto deve ser realizado com a equipe de CCB, que tomará decisões sobre o futuro do projeto.	Histórico de incidentes reportados em projetos anteriores podem estar presentes na "avaliação" da execução realizada pelo gerente. Além do mais, informações sobre os incidentes podem fazer parte de observações contidas no plano de teste. O processo pode auxiliar na identificação de pessoas com conhecimento do domínio, através da busca nas páginas amarelas.	Pouco aplicável	
	b	Sim. Os incidentes devem ser revisados, é importante que o revisor tenha conhecimento do domínio.		Aplicável	
	c	Não. Esta atividade depende da interação entre as pessoas envolvidas, não podendo ser realizada automaticamente.			
	d	Não. Os incidentes são específicos aos projetos que está sendo testado e às características de ambiente. Portanto, incidentes de outros projetos podem ter causas diferentes.			
	e	Sim, há o envolvimento com a equipe de CCB.			
	f	Sim. As decisões tomadas nesta atividade podem influenciar no custo e prazo da atividade de teste. Também pode ser necessário a alteração do escopo, e o envolvimento do cliente de acordo com os problemas encontrados.			
SPA.2 Executar ações apropriadas para corrigir os incidentes de teste <i>São tomadas medidas adequadas para corrigir, retestar e encerrar os incidentes de teste.</i>	a	Sim. Apesar da correção dos incidentes não ser realizada internamente, na equipe de testes, esta atividade envolve a realização dos re-testes, e em alguns casos até mesmo testes de regressão. Neste caso, assim como durante os testes, decisões operacionais são tomadas com relação aos casos de teste.	No processo são armazenadas informações sobre a execução do plano de teste, desta forma é possível inferir que informações sobre o re-teste podem ser armazenadas também. O processo também pode ser utilizado para encontrar especialistas no domínio.	Aplicável	
	b	Sim. Assim como a atividade de execução, é necessário um especialista no domínio.			
	c	Sim. No caso de testes automatizados, com a utilização de <i>scripts</i> automáticos. Entretanto, pode ser que seja necessário corrigir os <i>scripts</i> automáticos para o re-teste.			
	d	Não. Como cada projeto possui suas próprias características e particularidades, a realização desta atividade não necessita de consulta à projetos anteriores.			
	e	Não. Esta é uma atividade interna à equipe de testes.			
	f	Sim. Assim como a atividade de execução, as ações realizadas nesta atividade podem não ser as corretas, deixando de evidenciar um erro.			
SPA.3 Acompanhar o status dos incidentes de teste <i>O status dos incidentes de teste são monitorados e as ações apropriadas são tomadas, se necessário.</i>	a	Não. O propósito desta atividade é a geração de relatórios.	Extremamente aplicável	Extremamente aplicável	
	b	Não. Como as informações já foram coletadas, neste momento é realizada a consolidação das mesmas em um documento.			
	c	Sim. A geração do relatório pode ser automatizada.			
	d	Não. Informações de outros projetos não influenciam o resultado desta atividade.			
	e	Não. É uma atividade interna à equipe de testes.			
	f	Sim. Apesar de ser uma tarefa de consolidação de informações em um único lugar, a ausência desta atividade pode fazer com que informações relevantes não sejam divulgadas, influenciando decisões de projeto que podem deixar de serem tomadas.	O processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)	
SP5.1 Executar casos de teste não funcional Os casos de teste não funcional são executados manualmente, utilizando procedimentos de teste automatizados e/ou scripts de teste.	a	Sim. Os procedimentos devem ser seguidos de acordo com o descrito nos casos de teste, mas existe a comparação e análise dos resultados, para averiguar se as condições necessárias foram satisfeitas.	Os casos de testes são executados e acompanhados diretamente pelo processo. Também é possível encontrar especialistas no domínio, através da busca nas páginas amarelas.	O processo não apoia a execução de planos de teste diretamente, pois somente são armazenadas informações do projeto. Acreditado que as lessons learned podem auxiliar (embora não contenham informações muito relevantes para a execução) a solução de dúvidas, e especialistas podem ser encontrados através das páginas amarelas.	Informações relevantes para apoiar a execução podem ser encontradas nos documentos de conhecimento que são encontrados no processo. Também é possível realizar a busca por especialistas nas páginas amarelas.
	b	Sim. Conhecimento do domínio é relevante na atividade de execução.	Extremamente aplicável	Aplicável	Extremamente aplicável
	c	Sim. Existem diversos projetos em que a execução dos casos de teste é totalmente automatizada.			
	d	Não. Resultados de execuções de projetos anteriores não são consultadas durante a execução do projeto atual.			
	e	Não. A atividade de execução é interna à equipe de teste. O envolvimento com outras equipes ocorrerá em etapas posteriores à execução.			
	f	Sim. Caso a atividade não seja realizada corretamente, erros podem não ser revelados, impactando a confiabilidade do sistema.			
SP5.2 Relatar incidentes de teste As diferenças entre os resultados reais e esperados são relatadas como incidentes de testes não funcional.	a	Sim. Decisões são tomadas para classificar os incidentes quanto à sua severidade e prioridade.	O processo não oferece apoio para esta atividade (de relatar), mas por armazenar os casos de teste, e dar suporte à execução dos mesmos, deve também armazenar informações sobre o incidente, permitindo que o teste seja refeito, após a correção do incidente. Além da identificação de especialistas nas páginas amarelas.	Aplicável	O processo não apoia o relato dos incidentes, pois as informações armazenadas estão contidas nos documentos de conhecimento. Entretanto, especialistas podem ser encontrados a partir das páginas amarelas.
	b	Sim. Para classificar corretamente os dados dos incidentes é necessário um especialista no domínio.			
	c	Não. Apesar do relato dos incidentes poder ser automatizado, para a classificação dos mesmos é necessário o envolvimento humano.			
	d	Não. O relato dos incidentes, e sua classificação não exige que uma consulta a projetos antigos seja realizada.			
	e	Sim. É necessário a interação com outras equipes para o relato dos incidentes.			
	f	Sim. Relatos incorretos podem dificultar a sua correção.			
SP5.3 Escrever log de teste não funcional Logs de teste são escritos para fornecer um registro cronológico com detalhes relevantes sobre a execução dos testes não funcional.	a	Não. Esta atividade envolve a documentação dos logs que foram coletados ao longo da execução dos casos de teste.	Aplicável	Aplicável	Aplicável
	b	Não. É necessário apenas a transcrição dos incidentes ocorridos.	Este processo possui local específico para a execução dos casos de teste, entretanto não fica claro se a documentação dos incidentes é armazenada na ferramenta.		O processo não dá suporte à atividade de documentação de incidentes de teste.
	c	Sim. Resultados dos incidentes podem ser coletados e documentados automaticamente.			
	d	Não. Nenhuma consulta a base histórica faz-se necessária.			
	e	Não. Outras equipes não necessitam estar envolvidas nesta atividade.			
	f	Não. Nenhuma interferência em atividades futuras é esperada como resultado desta atividade.	Pouco aplicável	Não aplicável	Não aplicável

Análise Crítica (Parte 9)

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP1.6 Conduzir revisões do progresso do teste <i>Revisar periodicamente o progresso do teste, o desempenho e os problemas. Revisões de progresso são realizadas para manter os envolvidos informados. Estas revisões são tipicamente informais realizadas regularmente, por exemplo, semanal, bi-semanal ou mensal.</i>	Sim. Durante as revisões do progresso, o andamento do teste e os incidentes são analisados. Decisões devem ser tomadas para que o custo e prazo estimados sejam atendidos.	O progresso do teste pode ser acompanhado através deste processo, sendo possível gerar o relatório de status à partir de suas informações.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	Não são necessários especialistas no domínio. Nesta atividade devem ser envolvidas pessoas da equipe gerencial.			
	Não. A coleta e consolidação dos dados em um relatório de status pode ser realizada automaticamente. Entretanto a análise depende do envolvimento humano.			
	Não. Informações de outros projetos não influenciam o resultado desta atividade.			
	Sim. A equipe gerencial, e até mesmo o cliente devem estar envolvidos nesta atividade.	Aplicável	Não aplicável	Não aplicável
	Sim. Impactos em atividades e no planejamento podem acontecer, caso o progresso não esteja de acordo com o esperado.			
SP2.2 Monitorar defeitos <i>Monitorar as medidas de defeitos encontradas durante os testes que não atendem às expectativas.</i>	Não. O propósito desta atividade é o acompanhamento dos relatórios gerados.			
	Não. Nenhum conhecimento específico é exigido nesta atividade.			
	Sim. O acompanhamento das métricas pode ser automatizado, criando alarmes que irão informar os envolvidos conforme as métricas mudam de valor.	O processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	Não. Informações de outros projetos não influenciam o resultado desta atividade.			
	Não. É uma atividade interna à equipe de testes.			
	Sim. Apesar de ser uma tarefa de acompanhamento de métricas, a ausência desta atividade pode fazer com que informações relevantes não sejam divulgadas, influenciando decisões de projeto que podem deixar de serem tomadas.	Não aplicável	Não aplicável	Não aplicável
SP2.6 Conduzir revisões de qualidade do produto <i>Revisar periodicamente a qualidade do produto. Revisões de qualidade do produto são realizadas para manter os envolvidos informados. Estas revisões são tipicamente informais realizadas regularmente, por exemplo, semanal, bi-semanal ou mensal.</i>	Sim. Durante as revisões de qualidade, os resultados do teste e a criticidade dos incidentes são analisados. Decisões devem ser tomadas para que o custo e prazo estimados sejam atendidos.	O processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	Não são necessários especialistas no domínio. Nesta atividade devem ser envolvidas pessoas da equipe gerencial.			
	Não. A coleta e consolidação dos dados em um relatório de status pode ser realizada automaticamente. Entretanto a análise depende do envolvimento humano.			
	Não. Informações de outros projetos não influenciam o resultado desta atividade.			
	Sim. A equipe gerencial, e até mesmo o cliente devem estar envolvidos nesta atividade.	Não aplicável	Não aplicável	Não aplicável
	Sim. Impactos em atividades e no planejamento podem acontecer, caso o progresso não esteja de acordo com o esperado.			

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)	
SP3.1 Analisar problemas <i>Coletar e analisar os problemas e determinar as ações corretivas necessárias para resolvê-los.</i>	a Sim. A análise dos problemas oferecerá material para a tomada de decisão sobre as ações corretivas.	O processo não oferece apoio para a atividade de análise de problemas, mas oferece meios de contatar pessoas especializadas no domínio através da busca nas páginas amarelas.	O processo permite meios de consultar projetos anteriores para análise de lessons learned que possam ajudar a analisar os problemas. O processo também pode indicar especialistas no domínio, utilizando-se das páginas amarelas.	O processo não oferece apoio para a atividade de análise de problemas, mas oferece meios de contatar pessoas especializadas no domínio através da busca nas páginas amarelas.	
	b Sim, pois problemas encontrados podem interferir no <i>design</i> do projeto, o que envolve um esforço de retrabalho em todas as frentes. Neste caso, é importante conhecimento do domínio.				
	c Não. Nesta atividade é realizada a análise dos relatórios gerados em atividades anteriores. E como toda atividade de análise, esta não pode ser automatizada.				
	d Sim. A consulta dos problemas encontrados nos projetos anteriores pode ser realizada, para orientar a tomada de decisão.				
	e Sim. Todas as equipes envolvidas no projeto devem participar desta atividade, em alguns casos, até mesmo o cliente.				
	f Sim, as ações corretivas podem influenciar outras atividades futuras. E até mesmo envolver retrabalho de atividades passadas.		Pouco aplicável	Aplicável	Pouco aplicável
SP3.2 Tomar ação corretiva <i>Tomar ações corretivas sobre os problemas identificados.</i>	a Sim. De acordo com a análise realizada na atividade anterior, os envolvidos decidem quais são as ações necessárias para atender o prazo e o custo do projeto.	O processo não oferece apoio para esta atividade. Entretanto, oferece meios de buscar pessoas especializadas no domínio.	Informações coletadas ao longo do projeto e armazenadas como lessons learned podem auxiliar a tomada de decisão sobre quais ações devem ser realizadas. O processo também oferece meios de buscar pessoas especializadas no domínio.	O processo não oferece apoio para esta atividade. Entretanto, oferece meios de buscar pessoas especializadas no domínio.	
	b Sim, as ações corretivas exigem que um especialista no domínio esteja envolvido na atividade.				
	c Não. Esta atividade depende das decisões que serão tomadas por pessoas.				
	d Não. A ação a ser realizada depende das características do projeto atual. Informações de outros projetos podem ser irrelevantes nesta atividade.				
	e Sim, pessoas de outras equipes podem ser envolvidas para o caso de mudanças no projeto.				
	f Sim. Nesta atividade serão definidas quais alterações devem ser realizadas, muitas vezes alterando o curso das atividades que estão em andamento.		Pouco aplicável	Aplicável	Pouco aplicável
SP3.3 Gerenciar ação corretiva <i>Gerenciar ações corretivas até seu encerramento.</i>	a Não. O propósito desta atividade é o acompanhamento das ações a serem realizadas.				
	b Não. Nenhum conhecimento específico é exigido nesta atividade.				
	c Não. Monitorar ações corretivas vai além de acompanhar métricas em um relatório, a realização de atividades adicionais podem estar envolvidas, e também existe retrabalho de atividades já encerradas. Portanto o monitoramento não pode ser automatizado.				
	d Não. Informações de outros projetos não influenciam o resultado desta atividade.				
	e Não. É uma atividade interna à equipe de testes.				
	f Sim. Apesar de ser uma tarefa de acompanhamento, a ausência desta atividade pode fazer com que informações relevantes não sejam divulgadas, influenciando decisões de projeto que podem deixar de serem tomadas.				

Prática (TMMi)	Análise Crítica (Prática)	Análise Crítica (Processo A)	Análise Crítica (Processo B)	Análise Crítica (Processo C)
SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste <i>A disponibilidade eo uso dos ambientes de teste por vários grupos são coordenados.</i>	a Sim. Envolve decisão quanto a disponibilidade e liberação de ambientes de teste.			
	b Não. Nenhuma atividade que necessita de conhecimento especializado é realizada.			
	c Não. Esta atividade está envolvida com planejamento e liberação de recursos, portanto, não pode ser automatizada.	O processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	d Não. Dados de projetos anteriores não influenciarão a realização desta atividade.			
	e Sim. no caso de problemas com prazos de projetos que impactam a liberação de ambientes para novos projetos, os membros de outras equipes devem ser envolvidos.			
	f Sim. Uma gestão ruim desta atividade pode impactar no ambiente dos projetos em andamento.	Não aplicável	Não aplicável	Não aplicável
SP3.4 Relatar e gerenciar incidentes do ambiente de teste <i>Os problemas que ocorrem quando se utiliza o ambiente de teste são formalmente relatados como incidentes.</i>	a Sim. São tomadas decisões sobre a resolução dos problemas encontrados nos ambientes existentes.			
	b Não. Nenhuma atividade que necessita de conhecimento especializado é realizada.			
	c Não. Esta atividade está envolvida com planejamento e liberação de recursos, portanto, não pode ser automatizada.	O processo não apoia esta atividade.	O processo não apoia esta atividade.	O processo não apoia esta atividade.
	d Não. Dados de projetos anteriores não influenciarão a realização desta atividade.			
	e Sim. Todos os envolvidos com o processo de teste, membros do CCB, e até mesmo os envolvidos de desenvolvimento e cliente podem ser envolvidos na atividade.			
	f Sim. Alterações de ambiente e até mesmo de características do projeto podem ser resultados desta atividade.	Não aplicável	Não aplicável	Não aplicável

A.2 Relação das Práticas do TMMi

A relação das práticas é encontrada na Tabela A.2.

Tabela A.2: Relação das Práticas

Planejamento		Execução e avaliação do teste	
1	SP1.2 Identificar riscos do produto	17	SP3.2 Executar casos de teste
2	SP1.3 Analisar riscos do produto	18	SP3.3 Relatar incidentes de teste
3	SP2.1 Identificar elementos e características a serem testados	19	SP3.4 Escrever log de teste
4	SP2.2 Definir a abordagem de teste	20	SP4.1 Decidir sobre incidentes com o grupo de controle de configuração
5	SP2.4 Definir critérios de parada	21	SP4.2 Executar ações apropriadas para corrigir os incidentes de teste
6	SP4.1 Estabelecer o cronograma de teste	22	SP4.3 Acompanhar o status dos incidentes de teste
7	SP4.2 Planejar a equipe de teste	23	SP5.1 Executar casos de teste não funcional
8	SP4.5 Estabelecer o plano de teste	24	SP5.2 Relatar incidentes de teste não funcional
9	SP1.1 Obter (eliciar) necessidades do ambiente de teste	25	SP5.3 Escrever log de teste não funcional
10	SP1.3 Analisar os requisitos do ambiente de teste		
Projeto de casos de teste		Monitoramento e controle	
11	SP1.2 Identificar e priorizar casos de teste	26	SP1.6 Conduzir revisões do progresso do teste
12	SP1.3 Identificar dados de teste específicos necessários	27	SP2.2 Monitorar defeitos
		28	SP2.6 Conduzir revisões de qualidade do produto
Configuração de dados e do ambiente de teste		29	SP3.1 Analisar problemas
13	SP2.1 Desenvolver e priorizar os procedimentos de teste	30	SP3.2 Tomar ação corretiva
14	SP2.4 Desenvolver cronograma de execução do teste	31	SP3.3 Gerenciar ação corretiva
15	SP2.1 Implementar o ambiente de teste	32	SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste
16	SP2.4 Realizar o teste intake (pré-teste) do ambiente de teste	33	SP3.4 Relatar e gerenciar incidentes do ambiente de teste

A.3 Práticas do TMMi presentes no Processo de Teste Enxuto

A tabela a seguir contempla o processo de teste enxuto utilizado como base para a análise das abordagens de Gestão do Conhecimento apresentadas nesta dissertação.

Fase do processo	Nível (TMMi)	Área de Processo (TMMi)	Objetivo Específico (TMMi)	Prática (TMMi)	Descrição
Planejamento	Nível 2	PA 2.2 Planejamento de Teste	SG 1 Realizar avaliação de risco do produto	1 SP1.2 Identificar riscos do produto	Riscos do produto (específicos de teste) são identificados e documentados
				2 SP1.3 Analisar riscos do produto	Riscos do produto (específicos de teste) são avaliados, categorizados e priorizados utilizando categorias e parâmetros predefinidos.
			SG 2 Estabelecer uma abordagem de teste	3 SP2.1 Identificar elementos e características a serem testados	Os itens e características a serem testados, e a não serem testados, são identificados com base nos riscos do produto (específicos de teste).
				4 SP2.2 Definir a abordagem de teste	A abordagem de teste é definida para mitigar os riscos do produto (específicos de teste) que foram identificados e priorizados.
				5 SP2.4 Definir critérios de parada	Os critérios de saída para o teste são definidos para saber quando o teste deverá ser concluído.
			SG 4 Desenvolver um plano de teste	6 SP4.1 Estabelecer o cronograma de teste	O cronograma de testes, com os estágios pré-definidos e de tamanhos gerenciáveis, é estabelecido e mantido com base na estimativa de teste e no ciclo de vida de teste definidos.
				7 SP4.2 Planejar a equipe de teste	Plano para a disponibilidade dos recursos humanos necessários que têm o conhecimento e as habilidades necessárias para realizar o teste.
			8 SP4.5 Estabelecer o plano de teste	8 SP4.5 Estabelecer o plano de teste	O plano de teste é estabelecido e mantido como base para o gerenciamento de testes. Os resultados das práticas anteriores são documentados em um plano global de teste, ligando as informações de uma maneira lógica.
		PA 2.5 Ambiente de Teste		9 SG 1 Desenvolver os requisitos do ambiente de teste	9 SP1.1 Obter (eliciar) necessidades do ambiente de teste
			10 SP1.3 Analisar os requisitos do ambiente de teste	10 SP1.3 Analisar os requisitos do ambiente de teste	Analisar os requisitos para garantir que eles são necessários, suficientes e viáveis.
Projeto de casos de teste	Nível 2	PA 2.4 Projeto e Execução de Teste	SG 1 Realizar análise e projeto de teste usando técnicas de projeto de teste	11 SP1.2 Identificar e priorizar casos de teste	Caso de teste são identificados e priorizados utilizando técnicas de design de teste.
				12 SP1.3 Identificar dados de teste específicos necessários	São identificados os dados de testes específicos necessários para suportar as condições de teste e execução de casos de teste.
Configuração de dados e do ambiente de teste	Nível 2	PA 2.4 Projeto e Execução de Teste	SG 2 Executar implementação de teste	13 SP2.1 Desenvolver e priorizar os procedimentos de teste	Os procedimentos de teste são desenvolvidos e priorizados.
				14 SP2.4 Desenvolver cronograma de execução do teste	É desenvolvido um cronograma de execução do teste que descreve a sequência em que os procedimentos de teste serão executados.
		PA 2.5 Ambiente de Teste	SG 2 Realizar implementação do ambiente de teste	15 SP2.1 Implementar o ambiente de teste	Implementar o ambiente de teste, conforme descrito na especificação de requisitos do ambiente de teste e de acordo com o plano definido.
				16 SP2.4 Realizar o teste intake (pré-teste) do ambiente de teste	Realizar o pré-teste de ambiente (teste de confiança) para decidir se o ambiente de teste está pronto para ser utilizado.
Execução e avaliação do teste	Nível 2	PA 2.4 Projeto e Execução de Teste	SG 3 Realizar execução de teste	17 SP3.2 Executar casos de teste	Os casos de teste são executados manualmente, utilizando procedimentos de teste documentados e/ou automatizados, através de scripts de teste.
				18 SP3.3 Relatar incidentes de teste	As discrepâncias são relatadas como incidentes de teste, quando há diferenças entre os resultados reais e esperados.
				19 SP3.4 Escrever log de teste	Logs de teste são escritos para fornecer um registro cronológico com detalhes relevantes sobre a execução dos testes.
			SG 4 Gerenciar incidentes de teste para encerramento	20 SP4.1 Decidir sobre incidentes com o grupo de controle de configuração	Ações apropriadas sobre os incidentes de teste são decididos por um conselho de controle de configuração (CCB).
				21 SP4.2 Executar ações apropriadas para corrigir os incidentes de teste	São tomadas medidas adequadas para corrigir, retestar e encerrar os incidentes de teste.
				22 SP4.3 Acompanhar o status dos incidentes de teste	O status dos incidentes de teste são monitorados e as ações apropriadas são tomadas, se necessário.
	Nível 3	PA 3.4 Teste Não Funcional	SG 5 Realizar execução de teste não funcional	23 SP5.1 Executar casos de teste não funcional	Os casos de teste não funcional são executados manualmente, utilizando procedimentos de teste documentados e/ou automatizados, através de scripts de teste.
				24 SP5.2 Relatar incidentes de teste não funcional	As diferenças entre os resultados reais e esperados são relatadas como incidentes de testes não funcional.
				25 SP5.3 Escrever log de teste não funcional	Logs de teste são escritos para fornecer um registro cronológico com detalhes relevantes sobre a execução dos testes não funcional.
				Nível 2	PA 2.3 Monitoramento e Controle de Teste
27 SP2.2 Monitorar defeitos	Monitorar as medidas de defeitos encontradas durante os testes que não atendem às expectativas.				
SG 2 Monitorar a qualidade do produto em relação ao plano e expectativas	28 SP2.6 Conduzir revisões de qualidade do produto	Revisar periodicamente a qualidade do produto. Revisões de qualidade do produto são realizadas para manter os envolvidos informados. Estas revisões são tipicamente informais realizadas regularmente, por exemplo, semanal, bi-semanal ou mensal.			
	SG 3 Gerenciar ações corretivas para encerramento	29 SP3.1 Analisar problemas	Coletar e analisar os problemas e determinar as ações corretivas necessárias para resolvê-los.		
30 SP3.2 Tomar ação corretiva		Tomar ações corretivas sobre os problemas identificados.			
PA 2.5 Ambiente de Teste	SG 3 Gerenciar e controlar os ambientes de teste	31 SP3.3 Gerenciar ação corretiva	Gerenciar ações corretivas até seu encerramento.		
		32 SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste	A disponibilidade eo uso dos ambientes de teste por vários grupos são coordenados.		
		33 SP3.4 Relatar e gerenciar incidentes do ambiente de teste	Os problemas que ocorrem quando se utiliza o ambiente de teste são formalmente relatados como incidentes.		

Tabela A.3: Estrutura TMMi Utilizada no Processo de Teste Enxuto Desenvolvido por Camargo, Ferrari e Fabbri (2013)

Anexo B

RESULTADOS DO SURVEY

Este anexo apresenta as tabelas de frequência que sumarizam as respostas dos participantes obtidas no primeiro *Survey* (Capítulo 4), sem considerar a seleção dos perfis indicada na Seção 4.3.2.

Tabela B.1: Tabela de Frequência - Processo A

Processo A					
	Extremamente aplicável	Aplicável	Pouco aplicável	Não aplicável	
SP1.2 Identificar riscos do produto	9	5	1	1	1
SP1.3 Analisar riscos do produto	6	8	1	1	1
SP2.1 Identificar elementos e características a serem testados	5	9	1	1	1
SP2.2 Definir a abordagem de teste	6	9	0	1	1
SP2.4 Definir critérios de parada	4	6	5	1	1
SP4.1 Estabelecer o cronograma de teste	6	4	5	1	1
SP4.2 Planejar a equipe de teste	6	8	2	0	0
SP4.5 Estabelecer o plano de teste	7	8	0	1	1
SP1.1 Obter (eliciar) necessidades do ambiente de teste	5	5	5	1	1
SP1.3 Analisar os requisitos do ambiente de teste	3	8	4	1	1
SP1.2 Identificar e priorizar casos de teste	8	6	2	0	0
SP1.3 Identificar dados de teste específicos necessários	4	9	3	0	0
SP2.1 Desenvolver e priorizar os procedimentos de teste	5	11	0	0	0
SP2.4 Desenvolver cronograma de execução do teste	6	9	1	0	0
SP2.1 Implementar o ambiente de teste	2	10	3	1	1
SP2.4 Realizar o teste intake (pré-teste) do ambiente de teste	3	6	6	1	1
SP3.2 Executar casos de teste	7	8	1	0	0
SP3.3 Relatar incidentes de teste	6	9	0	1	1
SP3.4 Escrever log de teste	2	8	5	1	1
SP4.1 Decidir sobre incidentes com o grupo de controle de configuração	1	9	6	0	0
SP4.2 Executar ações apropriadas para corrigir os incidentes de teste	4	9	2	1	1
SP4.3 Acompanhar o status dos incidentes de teste	5	8	3	0	0
SP5.1 Executar casos de teste não funcional	5	9	2	0	0
SP5.2 Relatar incidentes de teste	6	6	4	0	0
SP5.3 Escrever log de teste não funcional	2	10	4	0	0
SP1.6 Conduzir revisões do progresso do teste	4	9	3	0	0
SP2.2 Monitorar defeitos	6	5	5	0	0
SP2.6 Conduzir revisões de qualidade do produto	5	7	4	0	0
SP3.1 Analisar problemas	4	9	3	0	0
SP3.2 Tomar ação corretiva	3	12	0	1	1
SP3.3 Gerenciar ação corretiva	5	7	4	0	0
SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste	4	9	3	0	0
SP3.4 Relatar e gerenciar incidentes do ambiente de teste	5	9	1	1	1

Tabela B.2: Tabela de Frequência - Processo B

Processo B					
	Extremamente aplicável	Aplicável	Pouco aplicável	Não aplicável	
SP1.2 Identificar riscos do produto	6	4	4	0	0
SP1.3 Analisar riscos do produto	4	7	2	1	1
SP2.1 Identificar elementos e características a serem testados	8	5	1	0	0
SP2.2 Definir a abordagem de teste	8	6	0	0	0
SP2.4 Definir critérios de parada	3	8	3	0	0
SP4.1 Estabelecer o cronograma de teste	5	4	4	1	1
SP4.2 Planejar a equipe de teste	6	4	3	1	1
SP4.5 Estabelecer o plano de teste	5	5	3	1	1
SP1.1 Obter (eliciar) necessidades do ambiente de teste	5	6	2	1	1
SP1.3 Analisar os requisitos do ambiente de teste	4	3	7	0	0
SP1.2 Identificar e priorizar casos de teste	4	8	2	0	0
SP1.3 Identificar dados de teste específicos necessários	5	3	6	0	0
SP2.1 Desenvolver e priorizar os procedimentos de teste	4	5	5	0	0
SP2.4 Desenvolver cronograma de execução do teste	3	3	8	0	0
SP2.1 Implementar o ambiente de teste	3	5	4	2	2
SP2.4 Realizar o teste intake (pré-teste) do ambiente de teste	1	6	5	2	2
SP3.2 Executar casos de teste	5	4	3	2	2
SP3.3 Relatar incidentes de teste	4	6	2	2	2
SP3.4 Escrever log de teste	1	4	5	4	4
SP4.1 Decidir sobre incidentes com o grupo de controle de configuração	1	7	5	1	1
SP4.2 Executar ações apropriadas para corrigir os incidentes de teste	5	5	2	2	2
SP4.3 Acompanhar o status dos incidentes de teste	4	4	3	3	3
SP5.1 Executar casos de teste não funcional	0	9	3	2	2
SP5.2 Relatar incidentes de teste	4	4	3	3	3
SP5.3 Escrever log de teste não funcional	1	4	4	5	5
SP1.6 Conduzir revisões do progresso do teste	0	9	4	1	1
SP2.2 Monitorar defeitos	4	3	4	3	3
SP2.6 Conduzir revisões de qualidade do produto	3	7	3	1	1
SP3.1 Analisar problemas	6	4	3	1	1
SP3.2 Tomar ação corretiva	3	8	3	0	0
SP3.3 Gerenciar ação corretiva	1	6	6	1	1
SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste	1	5	5	3	3
SP3.4 Relatar e gerenciar incidentes do ambiente de teste	4	4	4	2	2

Tabela B.3: Tabela de Frequência - Processo C

Processo C					
	Extremamente aplicável	Aplicável	Pouco aplicável	Não aplicável	
SP1.2 Identificar riscos do produto	2	8	1	1	1
SP1.3 Analisar riscos do produto	0	10	1	1	1
SP2.1 Identificar elementos e características a serem testados	8	3	0	1	1
SP2.2 Definir a abordagem de teste	5	5	1	1	1
SP2.4 Definir critérios de parada	3	5	2	2	2
SP4.1 Estabelecer o cronograma de teste	5	5	1	1	1
SP4.2 Planejar a equipe de teste	5	5	2	0	0
SP4.5 Estabelecer o plano de teste	7	5	0	0	0
SP1.1 Obter (eliciar) necessidades do ambiente de teste	1	7	2	2	2
SP1.3 Analisar os requisitos do ambiente de teste	1	5	4	2	2
SP1.2 Identificar e priorizar casos de teste	6	6	0	0	0
SP1.3 Identificar dados de teste específicos necessários	6	5	1	0	0
SP2.1 Desenvolver e priorizar os procedimentos de teste	1	7	4	0	0
SP2.4 Desenvolver cronograma de execução do teste	1	7	4	0	0
SP2.1 Implementar o ambiente de teste	1	4	4	3	3
SP2.4 Realizar o teste intake (pré-teste) do ambiente de teste	1	6	4	1	1
SP3.2 Executar casos de teste	5	4	3	0	0
SP3.3 Relatar incidentes de teste	3	6	3	0	0
SP3.4 Escrever log de teste	1	6	4	1	1
SP4.1 Decidir sobre incidentes com o grupo de controle de configuração	1	7	4	0	0
SP4.2 Executar ações apropriadas para corrigir os incidentes de teste	2	6	4	0	0
SP4.3 Acompanhar o status dos incidentes de teste	1	4	7	0	0
SP5.1 Executar casos de teste não funcional	4	4	3	1	1
SP5.2 Relatar incidentes de teste	1	6	5	0	0
SP5.3 Escrever log de teste não funcional	0	4	7	1	1
SP1.6 Conduzir revisões do progresso do teste	2	7	3	0	0
SP2.2 Monitorar defeitos	1	5	6	0	0
SP2.6 Conduzir revisões de qualidade do produto	3	5	4	0	0
SP3.1 Analisar problemas	3	4	4	1	1
SP3.2 Tomar ação corretiva	1	6	4	1	1
SP3.3 Gerenciar ação corretiva	1	4	7	0	0
SP3.3 Coordenar a disponibilidade e o uso dos ambientes de teste	0	5	6	1	1
SP3.4 Relatar e gerenciar incidentes do ambiente de teste	0	5	6	1	1