



ODYSSEY-PROCESSCASE: UMA ABORDAGEM DE LINHA DE PROCESSOS DE SOFTWARE BASEADA EM CASOS

Diogo Matheus Costa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadoras: Cláudia Maria Lima Werner
Eldânae Nogueira Teixeira

Rio de Janeiro
Março de 2019

ODYSSEY-PROCESSCASE: UMA ABORDAGEM DE LINHA DE PROCESSOS DE
SOFTWARE BASEADA EM CASOS

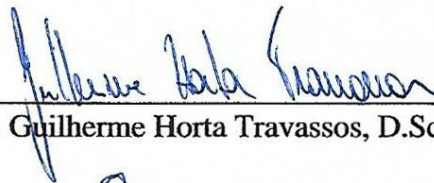
Diogo Matheus Costa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:



Prof.^a. Cláudia Maria Lima Werner, D.Sc.



Prof. Guilherme Horta Travassos, D.Sc.



Prof. Edson Alves de Oliveira Junior, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2019

Costa, Diogo Matheus

Odyssey-ProcessCase: Uma Abordagem de Linha de Processos de Software Baseada em Casos / Diogo Matheus Costa. – Rio de Janeiro: UFRJ/COPPE, 2019.

XV, 161 p.: il.; 29,7 cm.

Orientadoras: Cláudia Maria Lima Werner

Eldânae Nogueira Teixeira

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 128-138.

1. Linha de Processos de Software. 2. Definição de Processo de Software. 3. Raciocínio Baseado em Casos. 4. Reutilização. I. Werner, Cláudia Maria Lima *et al.*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus pais Cláudio e Maria de Fátima.

AGRADECIMENTOS

Aos meus familiares, pela confiança e compreensão, em especial ao meu irmão Thiago e meus pais Cláudio e Maria de Fátima. Obrigado pelos ensinamentos e carinho.

À minha noiva Daiana Freitas, pelo companheirismo, apoio e motivação desde o início desta jornada. Obrigado pela calma e compreensão durante este período.

À minha orientadora, Cláudia Werner, pela oportunidade e direcionamento que recebi no mestrado. Obrigado pelos conselhos, confiança depositada ao longo dos trabalhos e ensinamentos que guiaram essa pesquisa. Espero ter contribuído face a confiança em mim depositada.

À minha coorientadora, Eldânae Teixeira, pelo empenho e esforço aplicado em seus trabalhos, que possibilitaram essa continuidade. Obrigado pela disponibilidade, discussões e questionamentos que direcionaram os desafios desta pesquisa.

Aos colegas do Programa de Engenharia de Sistemas e Computação (PESC), em especial ao Victor e André, e todo o pessoal do grupo de Reutilização de Software (Sergio, Thaiana, Filipe, Eduardo, Gabriella, Benno, Marcelo, Susie, Rodrigo, etc.), pela convivência e suporte sempre que necessário.

À Fundação COPPETEC, por possibilitar o exercício da função de forma equilibrada com as atividades de pesquisa durante o período do mestrado. Aos colegas da CAPGov e Lemobs, em especial ao Evandro, Luiz, Yuri e Jacson.

Aos professores Guilherme Travassos e Edson Junior, por aceitarem participar da banca de defesa. A todos os professores da COPPE/UFRJ que tive o prazer de conhecer, e toda equipe técnico-administrativa pelo suporte necessário durante todo o mestrado.

Agradeço, por fim, à CAPES e FAPERJ pelo financiamento.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ODYSSEY-PROCESSCASE: UMA ABORDAGEM DE LINHA DE PROCESSOS DE SOFTWARE BASEADA EM CASOS

Diogo Matheus Costa

Março/2019

Orientadoras: Cláudia Maria Lima Werner

Eldânae Nogueira Teixeira

Programa: Engenharia de Sistemas e Computação

Processos de software têm sido foco de discussão na comunidade de Engenharia de Software, porém, definir um processo que atenda às necessidades específicas de um projeto continua sendo um desafio. Linha de Processos de Software (LPrS) consiste em uma sistemática para identificar similaridades e variabilidades entre processos para apoiar a reutilização de processos de software. Com base em uma análise de revisão da literatura, observou-se uma concentração de abordagens de LPrS que utilizam técnicas baseadas em mapeamento e regras para apoiar a definição de processo de software específico de projeto. Porém, o processo de aquisição de conhecimento exigido por este tipo de técnica não é trivial, por causa da indisponibilidade de especialistas e sobrecarga na compreensão do domínio. O objetivo desta pesquisa é estabelecer uma abordagem para LPrS, denominada *Odyssey-ProcessCase*, focada no apoio a tomada de decisão visando à resolução de variabilidades durante a definição de processo de software específico de projeto. Esta abordagem aplica técnicas como Raciocínio Baseado em Casos e Sistema Baseado em Regras para oferecer mecanismos complementares para apoiar a tomada de decisão. Para ofertar meios para utilização prática da abordagem, um ferramental de suporte foi implementado e, posteriormente, avaliado para caracterizar sua aceitação, com base no modelo de aceitação tecnológica (TAM). Os resultados apresentaram indícios de aceitação do ferramental de suporte visando a definição de processo de software específico de projeto, bem como possibilidades de melhoria.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ODYSSEY-PROCESSCASE: A CASE-BASED SOFTWARE PROCESS LINE
APPROACH

Diogo Matheus Costa

March/2019

Advisors: Cláudia Maria Lima Werner

Eldânae Nogueira Teixeira

Department: Computer and Systems Engineering

Software processes have been the focus of discussion in the Software Engineering community, but defining a software process that meets project-specific needs remains a challenge. Software Process Line (SPrL) consists of a systematic to identify similarities and variabilities among processes to support software process reuse. Based on a literature review analysis, a concentration of SPrL approaches that use mapping and rules techniques to support project-specific software process definition was observed. However, the knowledge acquisition process required by this kind of technique is not trivial, because of the unavailability of experts and overhead in domain understanding. The objective of this research is to establish an approach for SPrL, called *Odyssey-ProcessCase*, focused on the decision-making support to solve variabilities during the project-specific software process definition. This approach applies techniques such as Case Based Reasoning and Rule Based Systems to provide complementary mechanisms to support decision-making. To support the practical use of the approach, a supporting tool was implemented and then evaluated to characterize its acceptance based on the Technology Acceptance Model (TAM). The results indicated acceptance of the supporting tool to project-specific software process definition, as well as improvement possibilities.

ÍNDICE

CAPÍTULO 1 - INTRODUÇÃO	1
1.1. Contextualização	1
1.2. Motivação	1
1.3. Caracterização do Problema	3
1.4. Objetivo de Pesquisa	4
1.5. Metodologia de Pesquisa	6
1.6. Organização da Dissertação.....	8
CAPÍTULO 2 - LINHA DE PROCESSOS DE SOFTWARE.....	10
2.1. Revisão da Literatura - LPrS	12
2.1.1. Planejamento e Execução	14
2.1.2. Resultados da Revisão	16
2.1.2.1. Distribuição de Documentos por Ano.....	16
2.1.2.2. Abordagens Identificadas.....	17
2.1.2.3. Representação de Contexto	18
2.1.2.4. Técnicas Aplicadas na Fase de EPEP	20
2.1.2.5. Suporte Ferramental.....	23
2.1.2.6. Avaliação	28
2.1.3. Ameaças à Validade	30
2.1.4. Discussões	30
2.2. Considerações Finais	32
CAPÍTULO 3 - SISTEMA ESPECIALISTA	33
3.1. Sistema Baseado em Regras (SBR).....	34
3.2. Raciocínio Baseado em Casos (RBC)	37

3.2.1. Métrica de Similaridade	41
3.2.2. Revisão da Literatura - RBC	42
3.2.2.1. Planejamento e Execução.....	43
3.2.2.2. Resultados da Revisão	44
3.2.2.2.1. Distribuição de Documentos por Ano	44
3.2.2.2.2. Abordagens Identificadas	45
3.2.2.2.3. Representação dos Casos.....	47
3.2.2.2.4. Descrição dos Casos	47
3.2.2.2.5. Suporte para Tomada de Decisão	49
3.2.2.2.6. Avaliação	49
3.2.2.3. Ameaças à Validade.....	50
3.2.2.4. Discussões	50
3.3. Considerações Finais	51
CAPÍTULO 4 - ABORDAGEM ODYSSEY-PROCESSCASE.....	53
4.1. Introdução.....	53
4.2. Contexto de Pesquisa.....	54
4.2.1. Metamodelo e Notação <i>OdysseyProcess-FEX</i>	54
4.2.2. Abordagem <i>UbiFEX</i>	57
4.2.3. Metodologia <i>OdysseyProcessReuse</i>	59
4.2.4. Modelo de Contexto para Adaptação de Processo de Software	61
4.3. Requisitos da Abordagem <i>Odyssey-ProcessCase</i>	62
4.4. Visão Geral - <i>Odyssey-ProcessCase</i>	63
4.4.1. Pré-requisitos	64
4.4.2. Engenharia de Domínio de Processo de Software (EDPS)	65

4.4.3. Engenharia de Processo Específico de Projeto (EPEP).....	67
4.4.4. Avaliação de Reusabilidade	73
4.5. Considerações Finais	74
CAPÍTULO 5 - IMPLEMENTAÇÃO.....	76
5.1. Introdução.....	76
5.2. Requisitos do Ferramental de Suporte.....	76
5.3. Visão Geral do Ferramental de Suporte	77
5.3.1. Ferramenta Odyssey Estendida	79
5.3.1.1. Extensão da ferramenta Odyssey	80
5.3.2. Ferramenta Odyssey Repository.....	82
5.3.2.1. RESTful API.....	86
5.3.2.2. <i>Framework</i> jCOLIBRI.....	87
5.3.2.3. Modelo de Entidade Relacionamento (MER).....	87
5.4. Cenário de Exemplo	89
5.5. Considerações Finais	100
CAPÍTULO 6 - ESTUDO DE OBSERVAÇÃO	102
6.1. Introdução	102
6.2. Planejamento	103
6.2.1. Objeto de Estudo	103
6.2.2. Objetivo Global	103
6.2.3. Objetivos, Questões e Métricas	104
6.2.4. Modelo de Interpretação.....	105
6.2.5. Expectativa	107
6.2.6. Seleção dos Participantes	107

6.2.7. Seleção de Variáveis.....	107
6.2.8. Instrumentação	107
6.2.9. Treinamento.....	108
6.2.10. Recursos	109
6.2.11. Avaliação do Plano de Pesquisa	109
6.3. Execução do Estudo.....	110
6.3.1. Piloto.....	110
6.3.1.1. Caracterização de Perfil do Participante	110
6.3.1.2. Execução	110
6.3.2. Estudo de Observação	111
6.3.2.1. Caracterização de Perfil dos Participantes	111
6.3.2.2. Execução	112
6.4. Análise dos Resultados	113
6.4.1. Interpretação dos Dados Obtidos.....	113
6.4.2. Identificação de Melhorias	118
6.5. Ameaças à Validade	119
6.5.1. Validade Interna	120
6.5.2. Validade Externa	120
6.5.3. Validade de <i>Constructo</i>	120
6.5.4. Validade de Construção.....	120
6.6. Considerações Finais	121
CAPÍTULO 7 - CONCLUSÕES	122
7.1. Epílogo.....	122
7.2. Contribuições.....	123

7.3. Limitações	125
7.4. Publicações	126
7.5. Trabalhos Futuros	127
REFERÊNCIAS BIBLIOGRÁFICAS	128
APÊNDICE A - PROTOCOLO DE REVISÃO DA LITERATURA (LPRS)	139
APÊNDICE B - PROTOCOLO DE REVISÃO DA LITERATURA (RBC)	142
APÊNDICE C - ENTIDADES E INFORMAÇÕES DE CONTEXTO PARA ADAPTAÇÃO DE PROCESSO DE SOFTWARE	145
APÊNDICE D - INSTRUMENTOS UTILIZADOS NO ESTUDO DE OBSERVAÇÃO	154

ÍNDICE DE FIGURAS

Figura 1.1 – Metodologia de pesquisa.....	6
Figura 2.1 – Distribuição de documentos por ano (LPrS).....	17
Figura 2.2 – Frequência de uso das técnicas identificadas (LPrS).....	21
Figura 2.3 – Porcentagem de suporte ferramental na fase de EPEP (LPrS).....	24
Figura 2.4 – Frequência dos tipos de avaliação (LPrS).....	28
Figura 2.5 – Porcentagem do envolvimento da indústria nas avaliações (LPrS).....	29
Figura 3.1 – Visão geral de SBRs, adaptado de WOO <i>et al.</i> (2014).....	36
Figura 3.2 – Ciclo RBC, adaptado de AAMODT & PLAZA (1994).....	39
Figura 3.3 – <i>Weighted Nearest Neighbors</i> (WANGENHEIM & WANGENHEIM, 2008)	42
Figura 3.4 – Distribuição de documentos por ano (RBC).....	45
Figura 4.1 – Modelo de características de exemplo - <i>OdysseyProcess-FEX</i>	55
Figura 4.2 – Modelo de contexto de exemplo - <i>UbiFEX</i> (FERNANDES, 2009).....	58
Figura 4.3 – Entidades e informações de contexto, adaptado de LEITE (2011).....	61
Figura 4.4 – Visão geral da abordagem <i>Odyssey-ProcessCase</i>	64
Figura 4.5 – Mecanismo de reconhecimento de contexto.....	69
Figura 4.6 – Visão geral da etapa de EPEP da abordagem <i>Odyssey-ProcessCase</i>	72
Figura 5.1 – Visão geral do ferramental de suporte.....	77
Figura 5.2 – Diagrama de caso de uso do ferramental de suporte.....	78
Figura 5.3 – Infraestrutura da ferramenta Odyssey, adaptado de TEIXEIRA (2011)....	79
Figura 5.4 – <i>Dashboard</i> da ferramenta Odyssey Repository.....	84
Figura 5.5 – Interface pelo Swagger UI da RESTful API (Odyssey Repository).....	86
Figura 5.6 – Modelo de Entidade Relacionamento (Odyssey Repository).....	88

Figura 5.7 – Interface de cadastro de organização (Odyssey Repository)	90
Figura 5.8 – Interface de cadastro de projeto (Odyssey Repository)	90
Figura 5.9 – Interface de modelagem de domínio de processo (Odyssey).....	91
Figura 5.10 – Interface de sincronização de LPrS (Odyssey)	92
Figura 5.11 – Interface de visualização de LPrS (Odyssey Repository).....	92
Figura 5.12 – Modelo de características da organização XPTO	93
Figura 5.13 – Caracterização do projeto de software Orion.....	94
Figura 5.14 – Interface do mecanismo de recorte (Odyssey).....	95
Figura 5.15 – Interface do preenchimento de contexto (Odyssey).....	95
Figura 5.16 – Interfaces do mecanismo de busca de similaridade (Odyssey).....	96
Figura 5.17 – Interface do mecanismo de busca de similaridade durante o recorte (Odyssey).....	97
Figura 5.18 – Interface de sincronização de processo de software (Odyssey).....	97
Figura 5.19 – Interface de visualização de processo de software (Odyssey Repository).....	98
Figura 5.20 – Interface de submissão de <i>feedback</i> (Odyssey).....	99
Figura 5.21 – Interface de controle de reusabilidade (Odyssey Repository).....	99
Figura 6.1 – Número de participantes X opção selecionada (G1).....	114
Figura 6.2 – Número de participantes X opção selecionada (G2).....	115
Figura 6.3 – Número de participantes X opção selecionada (G3).....	116
Figura 6.4 – Número de participantes X opção selecionada (G4).....	117
Figura 6.5 – Número de participantes X opção selecionada (G5).....	118

ÍNDICE DE TABELAS

Tabela 2.1 – Revisões da literatura de reutilização de processos de software	13
Tabela 2.2 – Documentos selecionados para extração de informações (LPrS).....	15
Tabela 2.3 – Abordagens identificadas (LPrS).....	17
Tabela 2.4 – Técnicas aplicadas para apoiar na resolução de variabilidades (LPrS)	20
Tabela 2.5 – Ferramental de suporte por abordagem (LPrS)	23
Tabela 2.6 – Tipos de avaliação identificados (LPrS).....	28
Tabela 3.1 – Números da etapa de execução nas bases eletrônicas (RBC).....	43
Tabela 3.2 – Documentos selecionados para extração de informações (RBC).....	44
Tabela 3.3 – Abordagens identificadas (RBC).....	45
Tabela 3.4 – Representação dos casos nas abordagens (RBC).....	47
Tabela 3.5 – Descrição dos casos nas abordagens (RBC).....	48
Tabela 5.1 – Ficha técnica da ferramenta Odyssey Estendida	80
Tabela 5.2 – Casos de uso da ferramenta Odyssey Estendida.....	81
Tabela 5.3 – Ficha técnica da ferramenta Odyssey Repository.....	85
Tabela 5.4 – Casos de uso da ferramenta Odyssey Repository	85
Tabela 6.1 – Objetivos específicos do estudo de observação	104
Tabela 6.2 – Questões utilizadas para analisar os objetivos específicos	104
Tabela 6.3 – Métricas utilizadas para avaliar as questões	105
Tabela 6.4 – Modelo de interpretação dos dados	106
Tabela 6.5 – Caracterização do participante do piloto	110
Tabela 6.6 – Caracterização dos participantes do estudo de observação	111
Tabela 6.7 – Melhorias identificadas no estudo de observação	119

CAPÍTULO 1 - INTRODUÇÃO

1.1. Contextualização

Diante da alta competitividade da indústria de software, que exige qualidade, mas estabelece orçamentos e prazos desafiadores, produtores de software estão em constante busca por tecnologias de desenvolvimento visando o aumento de produtividade e flexibilidade para fornecer serviços e produtos (TEIXEIRA, 2016). Neste cenário, considerando que o processo pelo qual um software é desenvolvido influencia na qualidade do produto (OSTERWEIL, 1987, FUGGETTA, 2000, DAI & LI, 2007, CHEMUTURI & CAGLEY, 2010), organizações produtoras de software estão investindo para melhoria de seus processos de software.

De acordo com FUGGETTA (2000), processo de software é um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que são necessários para planejar, desenvolver, implementar e manter um produto de software. Segundo PAULK (2009), processo de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e seus produtos associados.

Nesta dissertação de mestrado, o termo processo de software é utilizado para representar o grão de processos de desenvolvimento de software, que propõem-se a oferecer modelos de organização para o ciclo de vida de desenvolvimento e manutenção de um sistema de software. Tais modelos de organização tem como objetivo orientar equipes de desenvolvimento, para que estas consigam atingir os seus objetivos no menor tempo e com a maior qualidade possível (MALIK & CHOUDHARY, 2009).

Definir um processo de software do zero envolve riscos e exige tempo e esforço (WASHIZAKI, 2006, XU & RAMESH, 2008). Além disso, a definição de processo de software é considerada uma atividade complexa, que exige experiência e conhecimento de uma variedade de disciplinas de Engenharia de Software (ALEIXO *et al.*, 2010, BARRETO *et al.*, 2010).

1.2. Motivação

Ciente da relação entre o processo de software utilizado e a qualidade do produto desenvolvido, bem como da complexidade envolvida na atividade de definição de

processos de software, pesquisas da área de Reutilização de Software estão se tornando cada vez mais relevantes para apoiar a reutilização de processos de software.

A área de Reutilização de Software representa o uso de software existente ou conhecimento para construção de um novo software, com o objetivo de colaborar para a redução do esforço e melhoria da qualidade no desenvolvimento de software (FRAKES & KANG, 2005). De acordo com SOMMERVILLE (2011), um dos benefícios da Reutilização de Software é o uso eficaz de especialistas, que ao invés de repetirem o mesmo trabalho, podem desenvolver artefatos reutilizáveis que encapsulem seu conhecimento.

No entanto, para alcançar estes objetivos, é necessário reutilizar diversos artefatos e conhecimento envolvidos no ciclo de vida do software (TEIXEIRA, 2011). O que exige expandir as práticas de reutilização, muitas vezes limitadas ao reuso de código-fonte, para diferentes artefatos (*e.g.*, processo de software).

Nesta dissertação, destaca-se o uso da técnica de Linha de Processos de Software (LPrS), que tem seu foco na reutilização de processos de software. LPrS consiste em uma sistemática para reutilização de processos de software, focada na identificação de similaridades e variabilidades entre processos. De acordo com WASHIZAKI (2006), LPrS representa um conjunto de processos dentro de um domínio particular, ou para um propósito específico, que compartilham características, sendo estes construídos com base em ativos de processo reutilizáveis.

Entre os benefícios dessa técnica, podemos citar: I. Aumento do potencial de reutilização; II. Redução dos riscos e esforço; e III. Aumento da qualidade e adequação dos processos de software gerados (ROMBACH, 2005, JAUFMAN & MÜNCH, 2005, BARRETO *et al.*, 2009).

Segundo ROMBACH (2005), as similaridades de processo devem ser capturadas em nível organizacional e as variabilidades controladas devem ser especificadas com diretrizes para adaptação. Neste cenário, destaca-se o conceito de Engenharia de Linha de Processos de Software (ELPrS) (ROMBACH, 2005), que divide a sistemática de LPrS em duas fases principais, neste trabalho referenciadas como: I. Engenharia de Domínio de Processo de Software (EDPS); e II. Engenharia de Processo Específico de Projeto (EPEP).

Na fase de EDPS, desenvolvimento *para* reutilização, os engenheiros de processo de software definem os artefatos reutilizáveis de LPrS. Neste ponto, o foco é compreender o domínio de processo e gerar um conjunto de modelos para representá-lo, especificando as similaridades e variabilidades entre processos. Por outro lado, na fase de EPEP, desenvolvimento *com* reutilização, elementos de processo reutilizáveis são selecionados para definir um processo de software específico de projeto considerando o contexto de definição de processo de software, ou seja, informações de contexto que caracterizam o projeto de software (*e.g.*, informações relacionadas ao cliente, organização, produto, etc.).

Neste ponto, a motivação pessoal do pesquisador para trabalhar com os conceitos da técnica de LPrS está relacionada à sua vivência profissional em fábricas de software, onde envolveu-se com múltiplos projetos, diferentes clientes e perfis diversos de liderança. Neste cenário, foi observado que o processo padrão existente na organização, que direcionava os profissionais e liderança, não oferecia apoio para compreensão da necessidade de adaptação do processo de software padrão visando atender as necessidades específicas dos diferentes projetos de software.

Nesta dissertação de mestrado, supõe-se que o conceito de gerência de variabilidades, explorado pela técnica de LPrS, bem como a visão de engenharia desta técnica, pode colaborar para o cenário observado, seja para explicitar que diferentes projetos de software exigem diferentes conjuntos de elementos de processo, ou para disseminar boas práticas, *e.g.*, representar como variantes as diferentes técnicas disponíveis na literatura para realizar a atividade de inspeção de requisitos.

1.3. Caracterização do Problema

Na técnica de Linha de Processos de Software, ao adotar o conceito de gerência de variabilidades, diferentes desafios são ressaltados nas atividades das fases de EDPS e EPEP. Na fase de EDPS, o desafio está relacionado à identificação e representação da variabilidades entre processos por meio de pontos de variação, ou seja, pontos de configuração no processo em que mais de um elemento de processo (*i.e.*, variante) está disponível. Por outro lado, na fase de EPEP, o gerente de projetos tem o desafio de compreender o contexto de definição do processo de software para resolver as variabilidades de domínio de processo de software, ou seja, decidir quais variantes, dentre as disponíveis, serão escolhidas para compor o processo de software

específico de projeto (MAGDALENO *et al.*, 2012). Neste ponto, compreender o contexto de definição do processo de software para resolver as variabilidades de domínio de processo de software, visando selecionar elementos de processo reutilizáveis para compor um processo de software específico de projeto, representa um problema de tomada de decisão.

Neste cenário, BARRETO *et al.* (2009), na época em que sua pesquisa foi conduzida, indicavam que o tópico de LPrS era recente e ainda não estava consolidado. Em 2014, DE CARVALHO *et al.* (2014) ressaltam que o tópico de LPrS ainda apresentava indícios de imaturidade, com a necessidade de evoluir em termos de avaliação. Por fim, PILLAT *et al.* (2015), destacam que não há uma abordagem consensual para realizar a adaptação de processo de software de maneira controlada e consistente.

Dessa forma, com base na problemática descrita, esta dissertação de mestrado foca no problema de tomada de decisão enfrentado pelo gerente de projetos durante a resolução de variabilidades de domínios de processos de software em LPrS, visando apoiar a definição de um processo de software específico de projeto (*i.e.*, fase de EPEP). Neste ponto, surge a questão de pesquisa desta dissertação de mestrado: *Como apoiar a tomada de decisão para resolução de variabilidades visando definir um processo de software específico de projeto a partir de artefatos reutilizáveis de Linha de Processos de Software?*

1.4. Objetivo de Pesquisa

Esta dissertação de mestrado busca solucionar a problemática apresentada, principalmente, relacionada à dependência de especialistas para a tomada de decisão durante a definição de processo de software específico de projeto usando LPrS.

Esse trabalho foi executado no contexto do grupo de Reutilização de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ. Neste cenário, diferentes estudos conduzidos pelo grupo, bem como pesquisas externas, foram utilizados como base para o desenvolvimento desta dissertação de mestrado. Neste ponto, destacam-se: I. Ferramenta Odyssey, que oferta uma infraestrutura de reutilização baseada em modelos de domínio; II. Metodologia *OdysseyProcessReuse*, que descreve uma sistemática para reutilização de processos de software no tópico de LPrS focada na fase de EDPS; e III. Modelo de Contexto para Adaptação de Processo

de Software, proposto por LEITE (2011), sendo este composto por 31 informações de contexto distribuídas em seis entidades de contexto. O contexto de pesquisa desta dissertação de mestrado é descrito em detalhes no Capítulo 4.

Considerando a problemática apresentada, duas suposições de pesquisa foram elaboradas após conduzir uma revisão da literatura focada na definição de processo de software específico de projeto usando LPrS (Seção 2.1):

1. *A similaridade do contexto de definição de processo de software, ou seja, semelhança entre as informações que caracterizam os projetos de software (e.g., criticidade, tamanho da equipe), representa uma oportunidade de reutilização de processos de software específicos de projeto;*
2. *Ofertar mecanismos complementares para definição de processo de software específico de projeto usando LPrS, ou seja, diferentes meios para apoiar na tomada de decisão, poderia permitir o apoio a diferentes perfis de profissionais e organizações.*

O objetivo desta dissertação é estabelecer uma abordagem para LPrS, denominada *Odyssey-ProcessCase*. O foco desta abordagem é apoiar na definição de processo de software específico de projeto a partir de artefatos reutilizáveis de LPrS, permitindo a tomada de decisão durante a resolução de variabilidades de domínios de processos de software.

De modo a alcançar o objetivo principal desta dissertação, foram estabelecidos os seguintes objetivos específicos:

- Identificar as técnicas utilizadas para apoiar na tomada de decisão visando a resolução de variabilidades durante a definição de processo de software específico de projeto usando LPrS;
- Especificar uma abordagem para LPrS focada no suporte para tomada de decisão visando a resolução de variabilidades de domínios de processos de software;
- Especificar e implementar um ferramental de suporte integrado que ofereça apoio para as atividades de ambas as fases de LPrS;
- Avaliar o ferramental de suporte desenvolvido, visando caracterizar sua aceitação no cenário de definição de processo de software específico de projeto.

Para ofertar meios para utilização prática da sistemática da abordagem *Odyssey-ProcessCase* (Capítulo 4), um ferramental de suporte integrado foi implementado (Capítulo 5), oferecendo suporte integrado para as atividades das fases de EDPS (TEIXEIRA, 2011, TEIXEIRA, 2016) e EPEP. Este ferramental de suporte é composto pelos componentes: I. Ferramenta Odyssey Estendida (ODYSSEY, 2019), um ambiente de reutilização baseado em modelos; e II. Ferramenta Odyssey Repository, um repositório de processos de software. Neste ponto, um estudo de observação foi realizado para caracterizar a aceitação do ferramental de suporte com base no modelo de aceitação tecnológica (TAM), no contexto de especialistas da área de processo de software definindo processos de software específicos de projeto (Capítulo 6).

1.5. Metodologia de Pesquisa

A metodologia de pesquisa aplicada no desenvolvimento deste trabalho é apresentada na Figura 1.1, composta por três fases: I. Concepção, focada na condução de duas revisões da literatura (LPrS e RBC), bem como na definição da abordagem *Odyssey-ProcessCase*; II. Implementação, que visa o desenvolvimento do ferramental de suporte por meio da implementação de um repositório de processos de software e extensão da ferramenta Odyssey (ODYSSEY, 2019); e III. Avaliação, focada na realização de um estudo de observação para caracterizar a aceitação do ferramental de suporte implementado com base no modelo de aceitação tecnológica (TAM). No total, seis etapas foram realizadas, conforme descrito a seguir.

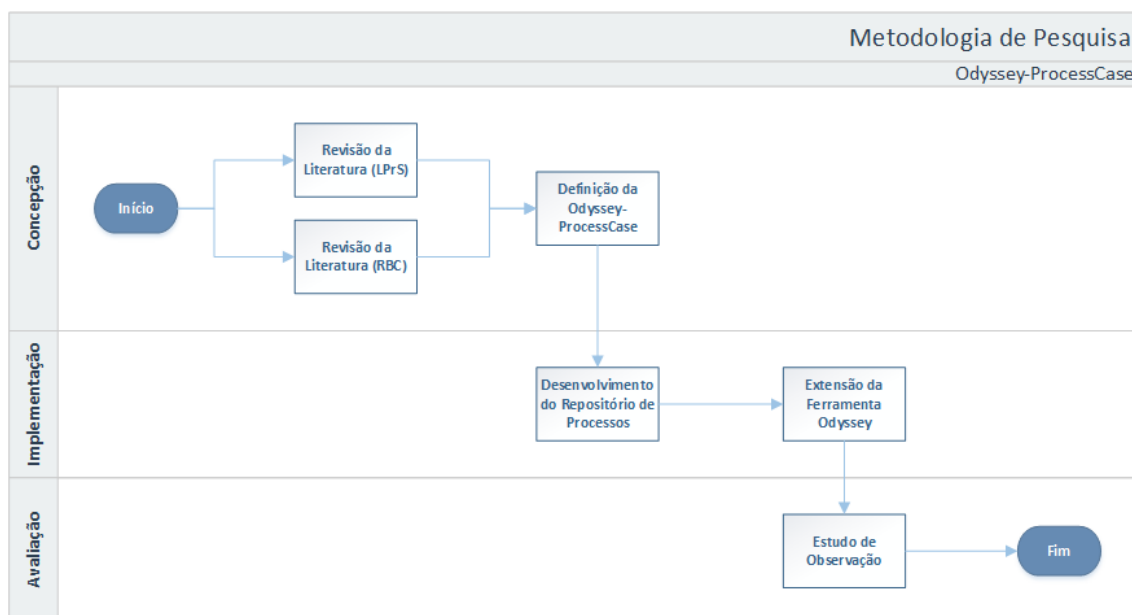


Figura 1.1 – Metodologia de pesquisa

Revisão da Literatura (LPrS): Com base na questão de pesquisa que direciona essa dissertação, uma *Quasi-Revisão Sistemática da Literatura*, onde não se tem pretensão de comparar alternativas de solução (TRAVASSOS *et al.*, 2008), foi conduzida para caracterizar as abordagens de LPrS com suporte para definição de processo de software específico de projeto. Nesta etapa, embora tenha sido definido um protocolo de pesquisa, os documentos selecionados na segunda execução da revisão da literatura de TEIXEIRA (2016), realizada em 2016, foram adotados como ponto de partida para aplicação dos critérios de seleção do protocolo elaborado. A revisão da literatura de TEIXEIRA (2016), focada no tópico de LPrS, cobre abordagens para o desenvolvimento, utilização, gerenciamento e evolução de LPrS, porém, questões específicas da fase de EPEP não foram tratadas. Os resultados obtidos nesta etapa possibilitaram a definição do objetivo e suposições de pesquisa (Seção 1.4).

Revisão da Literatura (RBC): Nesta etapa, motivado pela indicação da literatura de que os conceitos da técnica de RBC representam uma alternativa aos SBRs (*i.e.*, técnica de suporte utilizada por diferentes abordagens de LPrS para resolução de variabilidades). Uma *Quasi-Revisão Sistemática da Literatura* (TRAVASSOS *et al.*, 2008) foi conduzida para caracterizar o suporte ofertado por abordagens que aplicam os conceitos da técnica de RBC para definição de processo de software. Os resultados obtidos pela revisão influenciaram na sistemática da abordagem *Odyssey-ProcessCase*, bem como na implementação do seu ferramental de suporte.

Definição da *Odyssey-ProcessCase*: A partir dos resultados obtidos das revisões da literatura, surgiu o arcabouço da abordagem *Odyssey-ProcessCase*. Trata-se de uma abordagem para LPrS, fundamentada no modo de funcionamento da competência humana especializada, que recupera e modifica as soluções de problemas previamente resolvidos e armazenados na memória (SMYTH & CUNNINGHAM, 1992), conceito este diretamente ligado à técnica de Raciocínio Baseado em Casos (RBC). A abordagem *Odyssey-ProcessCase* foca no suporte para tomada de decisão visando apoiar na resolução de variabilidades de LPrS utilizando mecanismos complementares de definição de processo de software.

Desenvolvimento do Repositório de Processos: Com base no conhecimento adquirido a partir das revisões da literatura e definição da abordagem *Odyssey-ProcessCase*, um repositório de processo foi especificado e implementado para atender ao cenário de LPrS. O repositório implementado, *Odyssey Repository*, é

composto por dois componentes: I. Aplicação Web, focado na visualização e gerenciamento de artefatos e informações; e II. RESTful API, que visa possibilitar a integração com diferentes ferramentas de modelagem de processo de software, porém, nesta dissertação de mestrado a ferramenta Odyssey foi utilizada.

Extensão da Ferramenta Odyssey: Nesta etapa, a ferramenta Odyssey (ODYSSEY, 2019), que oferta uma infraestrutura baseada em modelos de domínio, foi estendida para suportar a sistemática definida neste trabalho, utilizando o repositório de processos desenvolvido na etapa anterior para oferecer um ferramental de suporte integrado, que disponibilize recursos para apoiar nas atividades das fases de EDPS, previamente implementado (TEIXEIRA, 2011, TEIXEIRA, 2016), e EPEP, contribuição deste trabalho de pesquisa.

Estudo de Observação: Como última etapa desta dissertação, foi planejado e conduzido um estudo de observação, focado na caracterização da aceitação do ferramental de suporte implementado com base no modelo de aceitação tecnológica (TAM). Os resultados apresentaram indícios de aceitação visando a definição de processo de software específico de projeto, bem como possibilidades de melhoria.

1.6. Organização da Dissertação

Esta dissertação está organizada em sete capítulos. O Capítulo 1, de Introdução, apresentou a motivação para o desenvolvimento deste trabalho, bem como o problema, questão de pesquisa, objetivo e a organização da dissertação.

O Capítulo 2 apresenta uma fundamentação sobre o tópico de LPrS, bem como uma revisão da literatura focada na definição de processo de software usando LPrS.

No Capítulo 3, uma fundamentação sobre o tópico de Sistema Especialista (SE) é descrita, bem como uma revisão da literatura focada em abordagens que aplicam os conceitos da técnica de Raciocínio Baseado em Casos (RBC) para definição de processo de software.

No Capítulo 4, o contexto de pesquisa no qual está dissertação de mestrado foi realizada é detalhado. Em seguida, a abordagem *Odyssey-ProcessCase* é descrita, uma abordagem para LPrS focada no apoio para o processo de tomada de decisão da atividade de definição de processo de software específico de projeto.

O Capítulo 5 descreve os detalhes da implementação do ferramental de suporte. As melhorias realizadas na ferramenta Odyssey (ODYSSEY, 2019) são apresentadas, bem como as etapas de desenvolvimento e integração relacionadas ao Odyssey Repository. Além disso, um cenário de exemplo é descrito para ilustrar o funcionamento da abordagem por meio do ferramental de suporte.

No Capítulo 6, o planejamento do estudo de observação realizado neste trabalho é detalhado e os resultados alcançados são discutidos.

No Capítulo 7, as conclusões e contribuições desta dissertação são apresentadas. Além disso, as limitações identificadas são relatadas, bem como as propostas de trabalhos futuros são enumeradas.

O Apêndice A detalha o protocolo da *Quasi-Revisão Sistemática da Literatura* abordada na Seção 2.1, focada na definição de processo de software específico de projeto usando LPrS.

O Apêndice B detalha o protocolo da *Quasi-Revisão Sistemática da Literatura* abordada na Seção 3.2.2, focada na definição de processo de software utilizando os conceitos da técnica de Raciocínio Baseado em Casos.

O Apêndice C apresenta as entidades e informações de contexto do modelo de contexto para adaptação de processo de software (LEITE, 2011), adotadas nesta dissertação para representar o contexto de definição de processo de software.

O Apêndice D descreve os instrumentos desenvolvidos durante o planejamento do estudo de observação apresentado no Capítulo 6.

CAPÍTULO 2 - LINHA DE PROCESSOS DE SOFTWARE

A técnica de Linha de Processos de Software (LPrS) pode ser definida como “um conjunto de processos de software que compartilham um conjunto de características comuns e variáveis, e são desenvolvidas a partir de artefatos (*core assets*), que podem ser reutilizados e combinados entre si, segundo regras de composição e recorte, para compor e adaptar processos de software” (NUNES *et al.*, 2010, MAGDALENO, 2010, TEIXEIRA, 2011). Essa técnica consiste em uma sistemática para reutilização de processos de software, focada na especificação de processo(s) genérico(s) que captura(m) as similaridades e variabilidades de um domínio (ROMBACH, 2005, MAGDALENO *et al.*, 2015).

O uso de abordagens de LPrS é considerado uma iniciativa proativa de reutilização, onde uma organização prepara um processo de software para um conjunto de necessidades esperadas (ARMBRUST *et al.*, 2008). Dentre os benefícios desta técnica, podemos citar: I. Aumento do potencial de reutilização; II. Redução dos riscos e esforço; e III. Aumento da qualidade e adequação dos processos de software gerados (ROMBACH, 2005, JAUFMAN & MÜNCH, 2005, BARRETO *et al.*, 2009).

Diferentes abordagens de LPrS têm sido propostas na literatura (ROMBACH, 2005, JAUFMAN & MÜNCH, 2005, WASHIZAKI, 2006, ARMBRUST *et al.*, 2009). Estas abordagens aplicam os princípios da técnica de Linha de Produtos de Software (LPS) no domínio de processo de software.

LPS consiste em um conjunto de sistemas de software que compartilham características comuns e controladas, elaborado a partir de artefatos reutilizáveis pré-definidos, que satisfazem as necessidades de um segmento em particular (NORTHROP, 2002). De acordo com NORTHROP (2002), três atividades são consideradas essenciais dentre os conceitos da técnica de LPS, sendo essas: I. Desenvolvimento do núcleo de artefatos, ou seja, arquitetura de referência, componentes reutilizáveis, modelos de domínio, etc; II. Desenvolvimento de produto, realizado com base nos artefatos do núcleo; e III. Gerenciamento, que tem como objetivo coordenar e documentar as atividades anteriores.

Segundo OSTERWEIL (1987), processos de software também são software e, assim como este, poderiam ter seus requisitos especificados, poderiam ser modelados,

desenvolvidos, testados e reutilizados. Neste cenário, ROMBACH (2005) destaca que os processos de software também são variáveis e precisam ser gerenciados de forma sistemática.

Na visão de engenharia da técnica de LPrS, Engenharia de Linha de Processos de Software (ELPrS), duas fases principais são destacadas (ROMBACH, 2005): I. Engenharia de Domínio de Processo de Software (EDPS); e II. Engenharia de Processo Específico de Projeto (EPEP). Na fase de EDPS, desenvolvimento *para* reutilização, os engenheiros de processo de software definem os artefatos reutilizáveis da LPrS. Na fase de EPEP, desenvolvimento *com* reutilização, elementos de processo reutilizáveis são selecionados para compor um processo de software específico de projeto. De acordo com WASHIZAKI (2006), ELPrS representa um sistema de abordagens estratégicas e sistemáticas inter-relacionadas para a construção, aplicação e gerenciamento de LPrS.

As atividades da fase de EDPS são focadas na compreensão do domínio. Neste momento, modelos são gerados para explicitar o conhecimento adquirido. Neste ponto, embora não exista um consenso sobre os modelos gerados na fase de EDPS, ao longo desta dissertação de mestrado alguns modelos são citados, dentre estes: I. Modelo de características, usado para identificar as similaridades e variabilidades do domínio; II. Modelo de contexto, usado para caracterizar entidades, informações e definições de contexto para especificação de regras de contexto, com o objetivo de apoiar na seleção dos elementos de processo reutilizáveis; e III. Modelo de componentes, utilizado em determinadas abordagens (*e.g.*, TEIXEIRA *et al.*, 2018), para oferecer uma visão modular com blocos de construção que combinam elementos de processo fortemente relacionados. Nesta fase, MAGDALENO *et al.* (2012) destacam que o conceito de gerência de variabilidades em processos de software torna-se importante para lidar com a diversidade de projetos de software (*i.e.*, necessidades específicas).

De acordo com LU & SADIQ (2007), o conceito de gerência de variabilidades permite a propagação de boas práticas. Neste ponto, ROUILLÉ *et al.* (2012) destacam que as variabilidades inerentes aos requisitos de projetos de software implicam em variabilidades nos processos de software, que ainda são complexos de capturar.

Por outro lado, as atividades da fase de EPEP são focadas na definição do processo de software específico de projeto, onde ao invés de definir um processo do zero os elementos de processo reutilizáveis identificados na fase de EDPS são utilizados. Dentre as atividades da fase de EPEP, estão: I. Caracterização do projeto de software, identificando suas necessidades específicas; II. Seleção de elementos de processo reutilizáveis, solucionando pontos de configuração; e III. Adaptação do processo de software, para atender necessidades específicas que podem não ser contempladas no domínio de processo. De acordo com MAGDALENO *et al.* (2012), neste cenário, o gerente de projetos toma decisões associadas aos pontos de configuração, ou seja, pontos de variação no processo em que se pode decidir quais variantes, dentre as disponíveis, serão escolhidas para compor o processo de software. O resultado desta fase é um processo de software específico de projeto, definido com todos os elementos que devem compor sua execução.

Segundo PRIETO-DÍAZ (1991), um programa de reutilização de software requer uma estrutura organizacional e ferramentas para promover, gerenciar e manter a prática de reutilização. No cenário de LPrS não é diferente, sendo necessário oferecer suporte para as atividades das fases de EDPS e EPEP.

Este capítulo está organizado de forma que, além desta introdução ao conceito de LPrS, a Seção 2.1 apresenta uma revisão da literatura focada na definição de processo de software específico de projeto usando LPrS. Por fim, na Seção 2.2, as considerações finais do capítulo são apresentadas.

2.1. Revisão da Literatura - LPrS

Diferentes estudos têm investigado o uso da técnica de LPrS para definição de processo de software específico de projeto em busca de benefícios como a redução de risco e esforço, e aumento da qualidade e adequação dos processos de software gerados (ROMBACH, 2005, JAUFMAN & MÜNCH, 2005, BARRETO *et al.*, 2009).

Neste cenário, algumas revisões da literatura foram realizadas para identificar abordagens de reutilização de processos de software existentes na literatura e suas principais características, conforme apresentado na Tabela 2.1. Sete revisões foram identificadas, sendo quatro artigos e três estudos realizados como parte de pesquisa de doutorado. Neste ponto, apenas duas revisões foram focadas no tópico de LPrS (DE

CARVALHO *et al.*, 2014 e TEIXEIRA, 2016) e, principalmente, discutiram sobre os desafios da fase de EDPS.

Tabela 2.1 – Revisões da literatura de reutilização de processos de software

Autor	Tipo	Nº de documentos	Área	Ano
PEDREIRA <i>et al.</i>	Artigo	28	Adaptação de Processo	2007
BARRETO	Seção de tese	57	Técnicas de Reutilização de Processos	2011
MARTÍNEZ-RUIZ <i>et al.</i>	Artigo	32	Adaptação de Processo	2012
ALEIXO	Seção de tese	40	Gerência de Variabilidades de Processo	2013
ROCHA & FANTINATO	Artigo	63	LPS aplicada na área de BPM	2013
DE CARVALHO <i>et al.</i>	Artigo	40	LPrS	2014
TEIXEIRA	Seção de tese	25	LPrS	2016

DE CARVALHO *et al.* (2014) investigaram a fase de EDPS em abordagens de LPrS, incluindo pontos como representação de variabilidade. Os resultados destacaram o uso de modelo de características, bem como da notação SPEM e derivadas para representar LPrS. Além disso, os autores destacaram o aumento do interesse de pesquisa sobre LPrS, mesmo relatando a imaturidade da área.

TEIXEIRA (2016) participou de uma revisão da literatura sobre o tópico de LPrS, tendo como objetivo identificar abordagens para o desenvolvimento, utilização, gerenciamento e evolução de LPrS. Neste ponto, uma equipe de pesquisa composta por seis colaboradores de quatro universidades brasileiras conduziu o estudo, sendo um mestrando e cinco doutores. Os resultados da primeira execução (2014) dessa revisão indicaram que todos os estudos analisados priorizaram as atividades da fase de EDPS. Por outro lado, apenas 40% dos estudos (10 dos 25) apresentaram suporte para ambas as fases de LPrS (*i.e.*, EDPS e EPEP).

Após analisar as revisões da literatura focadas no tópico de LPrS, observou-se oportunidades de investigação relacionadas às atividades da fase de EPEP, que inclui identificar o suporte ferramental ofertado para apoiar as atividades desta fase, bem como mapear quais técnicas são aplicadas para apoiar na tomada de decisão para resolução de variabilidades de LPrS durante a definição de processo de software específico de projeto.

Com o objetivo de explorar estas oportunidades, considerando a importância das atividades da fase de EPEP, esta seção apresenta os principais resultados de uma

Quasi-Revisão Sistemática da Literatura focada na fase de EPEP em LPrS. Essa revisão da literatura está organizada de forma que, além desta introdução, na Seção 2.1.1, as etapas de planejamento e execução são detalhadas; na Seção 2.1.2, os resultados são apresentados; na Seção 2.1.3, as ameaças à validade são analisadas; e por fim, na Seção 2.1.4, uma discussão é introduzida.

2.1.1. Planejamento e Execução

O objetivo principal desta revisão da literatura envolve a identificação de publicações científicas sobre LPrS com o propósito de caracterizar seu suporte à fase de EPEP do ponto de vista do pesquisador. Para atingir este objetivo geral, um protocolo de pesquisa foi elaborado (Apêndice A).

Neste cenário, considerando o escopo da pesquisa realizada por TEIXEIRA (2016), que engloba o escopo específico desta revisão da literatura focada na fase de EPEP, bem como pela proximidade da data de execução da pesquisa (*i.e.*, segunda execução, conforme descrito a seguir), optou-se pela adoção dos documentos selecionados pela autora como ponto de partida para aplicação do protocolo de pesquisa elaborado para esta revisão da literatura (Apêndice A).

A primeira execução da revisão de TEIXEIRA (2016) foi realizada em novembro de 2014, porém, em junho de 2016 os mesmos colaboradores atualizaram os documentos selecionados, por meio de uma segunda execução. Os números resultantes das duas execuções correspondem a 861 artigos retornados, 134 documentos duplicados removidos; 669 documentos eliminados em um primeiro filtro (análise de título e resumo); 3 documentos excluídos, devido a indisponibilidade de recuperação de seu conteúdo completo; e 13 documentos removidos após a fase de leitura (análise do texto), por estar fora do escopo. Resumindo, um total de 42 artigos foi identificado como estudos relevantes. Neste momento, os resultados da análise da segunda execução da revisão da literatura ainda não foram publicados.

Para esta revisão da literatura, focada nas questões específicas da fase de EPEP em LPrS, os 42 documentos selecionados por TEIXEIRA (2016) foram utilizados como ponto de partida. Neste ponto, ao aplicar os critérios de seleção especificados no protocolo, 16 documentos foram retirados por não apresentarem suporte para as atividades da fase de EPEP. Como resultado desta etapa, 26 documentos foram selecionados para análise e extração de informações (Tabela 2.2).

Tabela 2.2 – Documentos selecionados para extração de informações (LPoS)

Nº	Publicação	Ano
1	JAUFMAN, I., MÜNCH, J. "Acquisition of a Project-Specific Process". In: <i>Proceedings of the International Conference on Product Focused Software Process Improvement (PROFES)</i> , pp 328-342 (June), Oulu, Finland. 2005.	2005
2	ROMBACH, D. "Integrated Software Process and Product Lines". In: <i>Unifying the Software Process Spectrum</i> , v. 3840, Lecture Notes in Computer Science, Springer, pp. 83-90, 2005.	2005
3	THRÄNERT, M., WERNER, A. "A Process Family Approach for the reuse of development processes". In: <i>Innovations and Advanced Techniques in Computer and Information Sciences and Engineering</i> , Springer, pp. 309-313, 2007.	2007
4	MARTÍNEZ-RUIZ, T., GARCÍA, F., PIATTINI, M., <i>et al.</i> "Process Institutionalization using Software Process Lines". In: <i>Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS)</i> , pp. 359-362, Milan, Italy, May. 2009.	2009
5	TERNITÉ, T. "Process lines: A product line approach designed for process model development". In: <i>Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)</i> , pp. 173-180, Patras, Greece, Dec. 2009.	2009
6	ALEIXO, F. A., FREIRE, M. A. SANTOS, W. C., <i>et al.</i> "A Model-driven Approach to Managing and Customizing Software Process Variabilities". In: <i>Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS)</i> , v. 3, pp. 92-100, Madeira, Portugal, June, 2010.	2010
7	ALEIXO, F. A., FREIRE, M. A., SANTOS, W. C., <i>et al.</i> "An approach to manage and customize variability in software processes". In: <i>Proceedings of the Brazilian Symposium on Software Engineering (SBES)</i> , pp. 118-127, Salvador, Bahia, Brazil, Sept. 2010.	2010
8	BARRETO, A., DUARTE, E., ROCHA, A. R., <i>et al.</i> "Supporting the definition of software processes at consulting organizations via software process lines". In: <i>Proceedings of the International Conference on the Quality of Information and Communications Technology (QUATIC)</i> , pp. 15-24, Porto, Portugal. 2010.	2010
9	BARRETO, A., MURTA, L. G. P., DA ROCHA, A. R. C., 2011, "Software Process Definition: a Reuse-based Approach", <i>Journal of Universal Computer Science</i> , v. 17, n. 13 (Jan), pp. 1765-1799.	2011
10	JAFARINEZHAD, O., RAMSIN, R. "Development of situational requirements engineering processes: A process factory approach". In: <i>Proceedings of the IEEE 36th Annual Computer Software and Applications Conference (COMPSAC)</i> , pp. 279-288, Izmir, Turkey, July. 2012.	2012
11	MAGDALENO, A. M., DE ARAUJO, R. M., WERNER, C. M. L. "COMPOOTIM: An Approach to Software Processes Composition and Optimization". In: <i>Proceedings of the Iberoamerican Conference on Software Engineering (CIBSE)</i> , pp. 1-14, Buenos Aires, Argentina, April. 2012.	2012
12	ROUILLÉ, E., COMBEMALE, B., BARAIS, O., <i>et al.</i> "Leveraging CVL to manage variability in software process lines". In: <i>Proceedings of the 19th Asia-Pacific Software Engineering Conference (APSEC)</i> , pp. 148-157, Hong Kong, China, Dec. 2012.	2012
13	HURTADO, J. A., BASTARRICA, M. C., OCHOA, S. F., <i>et al.</i> , 2013, "MDE software process lines in small companies", <i>Journal of Systems and Software</i> , v. 86, n. 5 (May), pp. 1153-1171.	2013
14	BASTARRICA, M. C., SIMMONDS, J., SILVESTRE, L. "Using megamodeling to improve industrial adoption of complex MDE solutions". In: <i>Proceedings of the 6th International Workshop on Modeling in Software Engineering (MiSE)</i> , pp. 31-36, Hyderabad, India, June. 2014.	2014
15	KUHRMANN, M., FERNÁNDEZ, D. M., TERNITÉ, T. "Realizing software process lines: Insights and experiences". In: <i>Proceedings of the 2014 International Conference on Software and System Process (ICSSP)</i> , pp. 99-108, Nanjing, China, May. 2014.	2014

Nº	Publicação	Ano
16	LORENZ, W. G., BRASIL, M. B., FONTOURA, L. M., <i>et al.</i> , 2014, "Activity-based software process lines tailoring", <i>International Journal of Software Engineering and Knowledge Engineering (SEKE)</i> , v. 24, n. 09 (May), pp. 1357-1381.	2014
17	SILVESTRE, L. "A Domain Specific Transformation Language to Support the Interactive Definition of Model Transformation Rules". In: <i>Doctoral Symposium at International Conference on Model Driven Engineering Languages and Systems (MoDELS)</i> , v. 1321, 2014.	2014
18	SILVESTRE, L., BASTARRICA, M. C., OCHOA, S. F. "A model-based tool for generating software process model tailoring transformations". In: <i>Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)</i> , pp. 533-540, Lisbon, Portugal, Jan. 2014.	2014
19	SILVESTRE, L., BASTARRICA, M. C., OCHOA, S. F. "Reducing Complexity of Process Tailoring Transformations Generation" In: <i>Model-Driven Engineering and Software Development</i> , v. 506, Communications in Computer and Information Science, Springer, pp. 171-182, 2014.	2014
20	BRONDANI, C., BERTOUL, G., FONTOURA, L. M. "Quality Evaluation of Artifacts in Tailored Software Process Lines". In: <i>Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering (SEKE)</i> , pp. 223-226, July. 2015.	2015
21	GARCIA, C., PALUDO, M., MALUCELLI, A., <i>et al.</i> "A software process line for service-oriented applications". In: <i>Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC)</i> , pp. 1680-1687, Salamanca, Spain, April. 2015.	2015
22	MAGDALENO, A. M., BARROS, M. O., WERNER, C. M. L., <i>et al.</i> , 2015, "Collaboration optimization in software process composition", <i>Journal of Systems and Software</i> , v. 103, pp. 452-466.	2015
23	SCHRAMM, J., DOHRMANN, P., KUHRMANN, M. "Development of flexible software process lines with variability operations: a longitudinal case study". In: <i>Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE)</i> , Nanjing, China, April. 2015.	2015
24	SIMMONDS, J., PEROVICH, D., BASTARRICA, M. C., <i>et al.</i> "A megamodel for Software Process Line modeling and evolution". In: <i>Proceedings of the ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)</i> , pp. 406-415, Ottawa, ON, Canada, Sept. 2015.	2015
25	SILVESTRE, L., BASTARRICA, M. C., OCHOA, S. F. "A Usable MDE-based Tool for Software Process Tailoring". In: <i>Posters and Demos at International Conference on Model Driven Engineering Languages and Systems (MoDELS)</i> , pp.36-39. 2015.	2015
26	KUHRMANN, M., MÉNDEZ FERNÁNDEZ, D., TERNITÉ, T., 2016, "On the use of variability operations in the V-Modell XT software process line", <i>Journal of Software: Evolution and Process</i> , v. 28, n. 4 (Sept), pp. 241-253.	2016

2.1.2. Resultados da Revisão

2.1.2.1. Distribuição de Documentos por Ano

Pela distribuição de documentos por ano (Figura 2.1), é possível observar o crescimento do suporte à fase de EPEP em abordagens de LPrS desde 2005, acompanhando o crescimento do tópico ao longo dos anos. Na Figura 2.1, observa-se que as atividades da fase de EPEP foram mais discutidas nos anos de 2014 e 2015, com seis publicações cada ano.

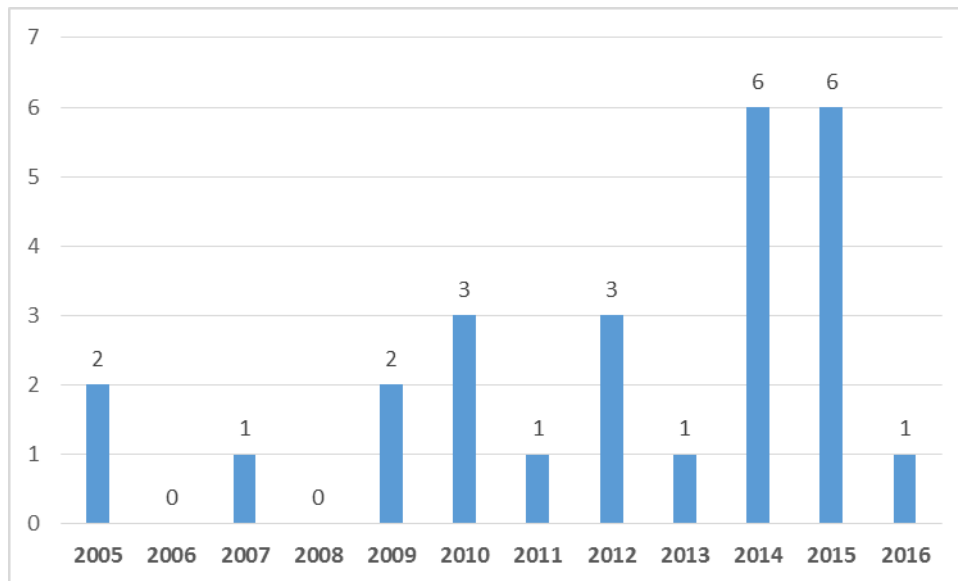


Figura 2.1 – Distribuição de documentos por ano (LPrS)

Neste ponto, nenhum nicho entre os locais de publicação (*e.g.*, conferências) foi observado com base nos documentos selecionados. A conferência MoDELS (*International Conference on Model Driven Engineering Languages and Systems*) foi o local de publicação com maior número de documentos selecionados, três publicações.

2.1.2.2. Abordagens Identificadas

Após realizar a análise dos 26 documentos selecionados, 13 abordagens foram identificadas (Tabela 2.3). Neste ponto, HURTADO *et al.* destacam-se como o grupo com maior número de publicações selecionadas nesta revisão da literatura, sete no total.

Tabela 2.3 – Abordagens identificadas (LPrS)

Abordagem	Nº de Pesquisadores	Nº de Documentos	Documento (Tabela 2.2)
ALEIXO <i>et al.</i>	4	2	[6][7]
BARRETO <i>et al.</i>	4	2	[8][9]
GARCÍA <i>et al.</i>	4	1	[21]
HURTADO <i>et al.</i>	6	7	[13][14][17][18][19][24][25]
JAFARINEZHAD & RAMSIN	2	1	[10]
JAUFMAN & MÜNCH	2	1	[1]
LORENZ <i>et al.</i>	6	2	[16][20]
MAGDALENO <i>et al.</i>	5	2	[11][22]
MARTÍNEZ-RUIZ <i>et al.</i>	3	1	[4]
ROMBACH	1	1	[2]
ROUILLÉ <i>et al.</i>	5	1	[12]
TERNITÉ <i>et al.</i>	5	4	[5][15][23][26]
THRÄNERT & WERNER	2	1	[3]

Neste cenário, cada documento selecionado foi analisado para extração de informações de acordo com o formulário definido no protocolo de pesquisa (Apêndice A). Abordagens com múltiplos documentos selecionados (6 de 13) tiveram seus documentos analisados de forma conjunta, para agrupar as informações e derivar uma análise agregada. Os resultados e discussões desta revisão da literatura, focada na fase de EPEP em LPrS, são descritos considerando as 13 abordagens identificadas.

2.1.2.3. Representação de Contexto

De acordo com ALEGRIA *et al.* (2011), cada projeto possui características próprias que requerem um conjunto particular de atividades, técnicas e estratégias, organizadas de forma ordenada e coesa para serem contidas no processo de software. Durante a definição do processo de software, essas características podem ser capturadas por meio de informações de contexto. Segundo DEY *et al.* (2001), um contexto corresponde a qualquer informação usada para caracterizar a situação de uma entidade considerada relevante (*e.g.*, pessoa, lugar, ação, objeto). Neste cenário, entidades de contexto (*e.g.*, dimensões de software) são descritas por um conjunto específico de informações de contexto que devem ser monitoradas.

Todas as abordagens identificadas indicaram o uso de informações de contexto para apoiar na definição de processos de software específicos de projeto. No entanto, das 13 abordagens, 7 abordagens reportaram a influência de características de projeto na atividade de definição de processo de software, porém, não apresentaram entidades ou informações de contexto (ALEIXO *et al.*, GARCÍA *et al.*, JAUFMAN & MÜNCH, MARTÍNEZ-RUIZ *et al.*, ROUILLÉ *et al.*, TERNITÉ *et al.*, THRÄNERT & WERNER).

BARRETO *et al.* relataram o uso de informações de projeto como um aspecto, qualidade, ou característica com a qual o processo deve estar em conformidade. Essas informações são aplicadas para orientar a seleção de elementos de processo, restringindo as opções disponíveis durante a derivação do processo de software. Neste ponto, estas informações são aplicadas como um mecanismo de alto-nível para apoiar na definição de processo de software.

HURTADO *et al.* indicaram que o contexto de projeto pode variar de acordo com diferentes variáveis (*i.e.*, informações de contexto como tamanho do projeto, conhecimento, complexidade, tipo) dentro de entidades específicas, tais como:

I. Produto; II. Projeto; e III. Equipe de desenvolvimento. Neste ponto, os autores definiram um metamodelo, nomeado de *Software Process Context Metamodel* (SPCM), para representar o modelo de contexto de cada projeto de software. Não há relatos de um conjunto predefinido de entidades ou informações de contexto.

JAFARINEZHAD & RAMSIN utilizam um modelo de situação para descrever o contexto do projeto de software. Este modelo define uma situação de projeto em termos de fatores de situação e critérios de processo. Neste ponto, os autores indicam que as informações podem ser representadas informalmente ou definidas por meio de uma lista de característica-valor usando uma notação específica (*e.g.*, linguagem de modelagem i*). O modelo de situação permite realizar uma análise de conflitos, além de influenciar na resolução do modelo de características. Um conjunto de 23 informações de contexto foi proposto pelos autores: Tipo de Projeto, Domínio de Aplicação, Tamanho do Projeto, Complexidade, Comprometimento de Gestão, Grau de Resistência, Volatilidade de Requisitos, Nível de Criticidade, Escassez de Pessoas e Recursos, Tamanho da Equipe, Familiaridade com o Domínio, Conhecimento da Equipe de Engenharia de Requisitos, Grau de Conhecimento sobre os Requisitos, Disponibilidade de Facilitadores Qualificados, Potencial de Conflito, Nível de Inovação do Projeto, Disponibilidade do Cliente, Grau de Reusabilidade, Grau de Conhecimento Implícito, Grau de Terceirização Necessário, Nível de Maturidade da Capacidade, Impacto Organizacional e Importância Estratégica.

LORENZ *et al.* contextualizam os projetos de software por meio de características que devem ser consideradas na definição de processo de software. Neste ponto, o contexto é definido por fatores com valores associados. Os autores indicam três conjuntos de informações de contexto como sugestão, sendo estes: I. BOEHM & TURNER (2003); III. COCKBURN (2004); e II. *Octopus Model* (KRUCHTEN, 2013). Somando os três conjuntos, 16 informações de contexto são sugeridas. Porém, no estudo de caso realizado, apenas o *Octopus Model* foi utilizado.

MAGDALENO *et al.* utilizam um modelo de contexto composto por entidades e informações de contexto. Neste cenário, regras de contexto são definidas para direcionar a seleção de elementos de processo com base em definições de contexto (*i.e.*, situações caracterizadas por um conjunto de informações de contexto com valores específicos). Três entidades de contexto são relatadas, sendo estas: I. Organizacional; II. Projeto; e III. Equipe. Além disso, 17 informações de contexto são sugeridas, sendo estas:

Estrutura Organizacional, Cultura Organizacional, Gestão do Conhecimento, Objetivo de Negócio, Relacionamento com o Cliente, Tamanho, Complexidade, Inovação, Criticidade, Duração, Estabilidade de Requisitos, Tamanho da Equipe, Experiência no Domínio, Experiência Técnica, Experiência de Trabalho Juntos, Proximidade e Estabilidade. Tais entidades e informações de contexto fazem parte do modelo proposto por LEITE (2011), elaborado com base nos resultados de uma revisão da literatura conduzida para identificar informações relevantes para adaptação de processo de software. No escopo do estudo realizado por LEITE (2011), o modelo foi avaliado por meio de duas etapas: I. Avaliação com especialistas, onde uma versão inicial foi avaliada por um conjunto de especialistas; e II. Estudo exploratório, onde o modelo adaptado com base na avaliação com especialistas foi aplicado em uma organização para verificar sua viabilidade.

Por fim, ROMBACH indica três entidades para representar o contexto de definição de processo de software em sua pesquisa: I. Projeto; II. Produto; e III. Processo.

2.1.2.4. Técnicas Aplicadas na Fase de EPEP

Cinco técnicas aplicadas para apoiar na resolução de variabilidades de LPrS foram identificadas nas abordagens analisadas, sendo estas apresentadas na Tabela 2.4.

Tabela 2.4 – Técnicas aplicadas para apoiar na resolução de variabilidades (LPrS)

Técnica	Abordagem (Tabela 2.3)
Mapeamento de Características	ALEIXO <i>et al.</i> ; BARRETO <i>et al.</i> ; GARCÍA <i>et al.</i> ; ROUILLE <i>et al.</i>
Sistema Baseado em Regras	HURTADO <i>et al.</i> ; THRÄNERT & WERNER; MAGDALENO <i>et al.</i>
Sistema de Inferência Difusa	JAFARINEZHAD & RAMSIN
AHP - <i>Analytic Hierarchy Process</i>	LORENZ <i>et al.</i>
Algoritmo Genético	MAGDALENO <i>et al.</i>
N/A	JAUFMAN & MÜNCH; MARTÍNEZ-RUIZ <i>et al.</i> ; ROMBACH; TERNITÉ <i>et al.</i>

A frequência de uso de cada técnica identificada (Figura 2.2) enfatiza o uso das técnicas Mapeamento de Características e Sistema Baseado em Regras. Neste ponto, 4 abordagens não relataram o uso de técnicas para apoiar na resolução de variabilidades de LPrS, sendo estas representadas pelo item N/A.

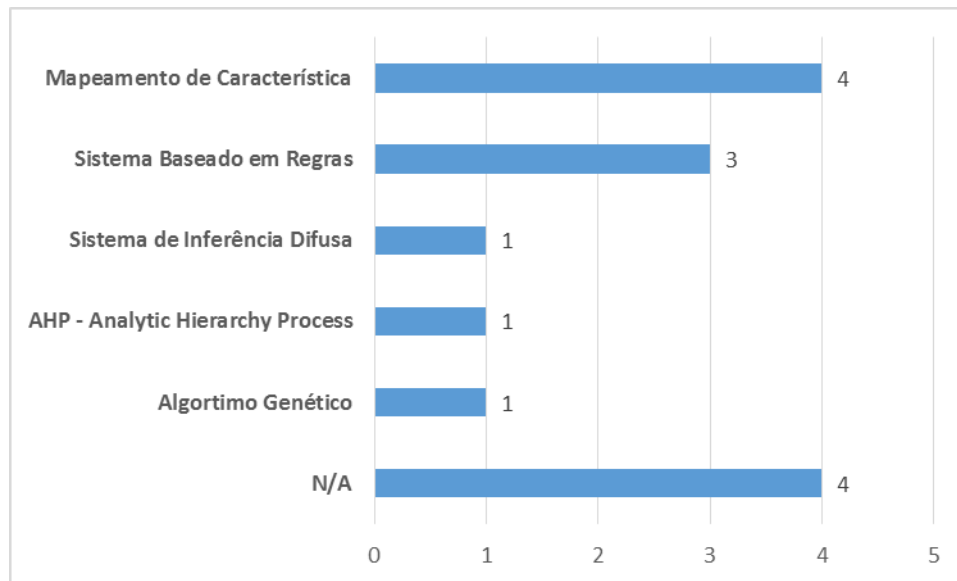


Figura 2.2 – Frequência de uso das técnicas identificadas (LPrS)

A técnica de Mapeamento de Características, nesta revisão, representa o mapeamento direto de um fator (*e.g.*, informação de contexto, característica ou meta) com elementos de processo. Neste ponto, ao selecionar um fator na definição de processo de software, um ou mais elementos podem ser incluídos ou excluídos como consequência desta seleção. Conforme observado na Figura 2.2, esta foi a técnica com maior frequência nas abordagens analisadas. No entanto, especificar situações de contexto complexas (*e.g.*, múltiplas informações de contexto impactando múltiplos elementos de processo) não parece ser trivial neste tipo de técnica. Este ponto pode representar uma limitação para apoiar na resolução de variabilidades de LPrS.

Sistema Baseado em Regras é uma das técnicas aplicadas para construção de Sistemas Especialistas (CARRICO *et al.*, 1989). Nesta técnica, as regras são separadas do mecanismo de processamento, sendo essas armazenadas em uma base de conhecimento, também referenciada como base de regras. Cada regra disponível na base de conhecimento representa uma instrução condicional. Essas regras são utilizadas para realizar operações de inferir, com o objetivo de chegar a uma conclusão, resposta ou sugestão apropriada (HAYES-ROTH, 1985, LIAO, 2005). Nas abordagens, esta técnica foi aplicada para definição de regras para resolução de variabilidades de LPrS com base em informações de contexto (*i.e.*, fatos).

Sistemas de Inferência Difusa (PEDRYCZ, 1993, HÁJEK, 1998) são utilizados para formular o mapeamento de uma entrada específica para uma saída usando lógica

difusa. O mapeamento, então, fornece uma base a partir da qual decisões podem ser tomadas. Esta técnica é comumente usada para lidar com a incerteza e a imprecisão de julgamento em processos de tomada de decisão com múltiplos objetivos. Na abordagem de JAFARINEZHAD & RAMSIN, esta técnica foi aplicada para apoiar a análise dos elementos de processo durante a definição de processo de software específico de projeto. Neste ponto, os valores dos fatores do modelo de situação proposto pela abordagem (*i.e.*, informações de contexto) são utilizados como entrada, representando objetivos. Dessa forma, considerando estes valores, um mecanismo de regras difusas é utilizado para indicar um valor de saída, que representa a aptidão de cada elemento de processo. Por fim, o valor de aptidão é utilizado pelo gerente de projetos para apoiar no processo de tomada de decisão para determinar se o elemento irá compor o processo de software específico de projeto.

AHP (*Analytic Hierarchy Process*) é um método aplicado para apoiar o processo de tomada de decisão em cenários complexos. Esta técnica trabalha com alternativas e um objetivo geral, onde se calcula a probabilidade numérica de cada alternativa e quanto maior sua probabilidade, melhores são as chances da alternativa satisfazer o objetivo geral (VARGAS, 2010). AHP pode ser aplicada em diferentes situações de decisão (FORMAN & GASS, 2001), tais como: I. Escolha, seleção de uma alternativa dentre um determinado conjunto de alternativas, geralmente, com múltiplos critérios de decisão envolvidos; II. Ranking, onde um conjunto de alternativas é colocado em ordem de acordo com sua relevância; e III. Priorização, determinando o mérito relativo de membros de um conjunto de alternativas, ao invés de selecionar uma alternativa ou simplesmente classificá-las. Na abordagem de LORENZ *et al.* esta técnica é aplicada para priorizar as atividades recuperadas do repositório, considerando os valores das informações de contexto para atender critérios de adaptação de processo de software (*e.g.*, critérios de prevenção de riscos). Neste cenário, as atividades são ordenadas de forma decrescente, ou seja, as primeiras posições representam as atividades mais adequadas para atender os requisitos do projeto. Por fim, o gerente de projetos fica responsável pelo processo de tomada de decisão para determinar quais atividades deverão compor o processo de software específico de projeto.

Algoritmo Genético é um algoritmo de busca baseado no mecanismo de seleção natural e genética natural (GOLDBERG, 1989). Trata-se de uma meta-heurística que pertence à classe de algoritmos evolutivos, que combina a sobrevivência do mais apto,

reprodução orientada por aptidão e mutação aleatória para desenvolver uma população de soluções candidatas. No entanto, essa técnica não garante uma solução globalmente otimizada. Além disso, o uso desta técnica exige uma representação genética do domínio da solução (*e.g.*, *array de bits*) e uma função de aptidão para avaliar as soluções candidatas. Na abordagem de MAGDALENO *et al.* essa técnica é aplicada para otimizar a colaboração em processos de software, de forma que a colaboração é utilizada como uma função de aptidão mono-objetiva. Antes disso, os conceitos da técnica de Sistema Baseado em Regras são aplicados para apoiar na resolução de variabilidades de LPrS.

2.1.2.5. Suporte Ferramental

As ferramentas de suporte relatadas nas abordagens analisadas são apresentadas na Tabela 2.5, de forma que a segunda coluna destaca todas as ferramentas de suporte indicadas nas abordagens e a terceira coluna apresentada as ferramentas de suporte focada no apoio para as atividades da fase de EPEP.

Tabela 2.5 – Ferramental de suporte por abordagem (LPrS)

Abordagem (Tabela 2.3)	Ferramentas de suporte	Ferramentas de suporte (EPEP)
ALEIXO <i>et al.</i>	EPF Composer, GenArch e jBPM	GenArch
BARRETO <i>et al.</i>	Protótipo	Protótipo
GARCÍA <i>et al.</i>	EPF Composer, GenArch, UMA2BPMN, Yaoqiang BPMN e Activiti	GenArch
HURTADO <i>et al.</i>	EPF Composer, Exeed, ATR, Protótipo e AM3	ATR, Protótipo e AM3
JAFARINEZHAD & RAMSIN	MATLAB	MATLAB
JAUFMAN & MÜNCH	N/A	N/A
LORENZ <i>et al.</i>	MfPTt	MfPTt
MAGDALENO <i>et al.</i>	Odyssey e COMPOOTIM	COMPOOTIM
MARTÍNEZ-RUIZ <i>et al.</i>	N/A	N/A
ROMBACH	N/A	N/A
ROUILLÉ <i>et al.</i>	Ferramental CVL	Ferramental CVL
TERNITÉ <i>et al.</i>	V-Modell XT VMEd e V-Modell XT VMPA	V-Modell XT VMPA
THRÄNERT & WERNER	N/A	N/A

Conforme observado na Figura 2.3, a maioria das abordagens (69%) apresentou ferramental de suporte para as atividades da fase EPEP. No entanto, ferramentas existentes foram reutilizadas (*e.g.*, ferramentas de LPS). Embora não seja um problema, esse tipo de reutilização pode representar uma limitação para necessidades e desafios

específicos de LPrS. Porém, nenhum tipo de relato ou debate neste sentido foi apresentado.

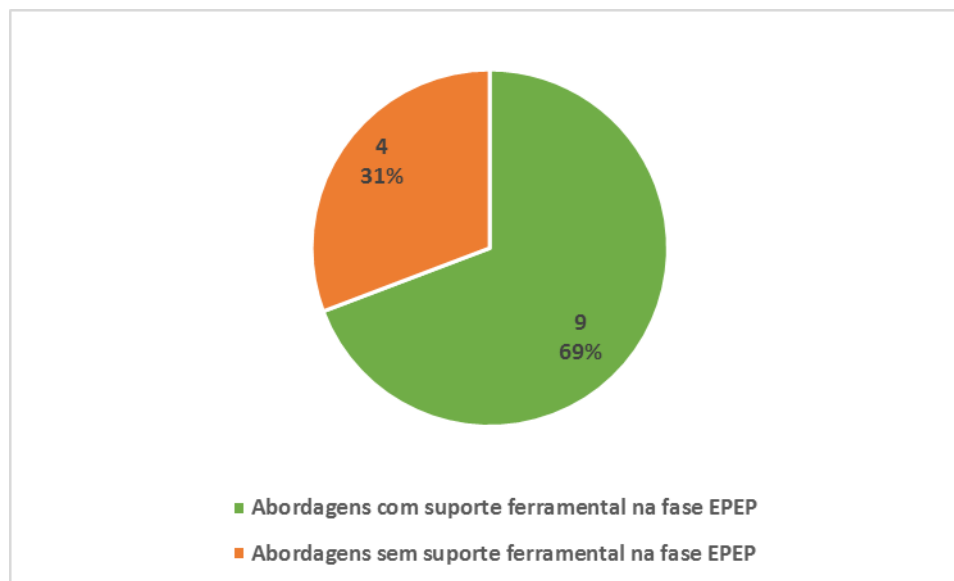


Figura 2.3 – Porcentagem de suporte ferramental na fase de EPEP (LPrS)

ALEIXO *et al.* indicaram o uso de três ferramentas (EPF Composer, GenArch e jBPM). O *Eclipse Process Framework* (EPF) foi utilizado para especificar o domínio de processo de software com base na notação UMA (*Unified Method Architecture*), por meio da ferramenta EPF Composer (*Eclipse Process Framework Composer*). Neste ponto, o gerência de variabilidades, bem como a definição de processo de software específico de projeto são executados por meio da ferramenta GenArch, uma ferramenta de LPS existente, estendida para suportar processos de software. Por fim, após finalizar as atividades da fase de EPEP, o processo de software específico de projeto é processado por meio de transformações usando as linguagens ATL e Acceleo. Essas transformações geram uma especificação de fluxo de trabalho JPDL que pode ser implementada e executada no mecanismo de fluxo de trabalho do jBPM (*JBoss Business Process Management*).

BARRETO *et al.* apresentam um protótipo desenvolvido para oferecer suporte para organizações implementadoras de processos de software. Este tipo de organização oferece consultoria para empresas que pretendem definir, implantar ou melhorar seus processos de software. O protótipo apresentado pelos autores oferece suporte para as atividades das fases de EDPS e EPEP. Inicialmente, os componentes de processo são especificados e armazenados no repositório. Em seguida, informações de contexto são

identificadas, bem como um mapeamento entre as informações de contexto e componentes de processo é realizado. Na fase de EPEP, este mapeamento é interpretado pelo protótipo para apoiar na resolução de variabilidades de LPrS. Porém, nos documentos analisados não fica claro se o ferramental de suporte foi finalizado para apoiar na definição do processo de software específico de projeto.

GARCÍA *et al.* com base no estudo de ALEIXO *et al.*, relataram um conjunto de ferramentas bastante semelhante ao original. Neste cenário, as ferramentas EPF Composer e GenArch são utilizadas para apoiar na especificação do modelo de características de LPrS, bem como gerenciar as variabilidades de LPrS e apoiar na definição de processo de software específico de projeto. GARCÍA *et al.* se diferenciam ao utilizar BPMN. Para isso, foi necessário o desenvolvimento da ferramenta UMA2BPMN, utilizada para realizar a conversão da notação UMA para BPMN. Além disso, os autores indicam o uso das ferramentas Yaoqiang BPMN (*i.e.*, ferramenta de modelagem BPMN) e plataforma Activiti (*i.e.*, plataforma BPM usada para execução, monitoramento e controle de processos de software).

HURTADO *et al.* indicam o uso de cinco ferramentas (EPF Composer, Exeed, ATR, Protótipo e AM3). Esta cadeia de ferramentas foi construída sobre o *Eclipse Modeling Framework* (EMF) e a linguagem de transformação ATL.

Na fase de EDPS, o domínio de processo de software da organização é especificado por meio da ferramenta EPF Composer. Neste ponto, os modelos especificados são editados por meio da ferramenta Exeed (*Extended EMF Editor*), um editor EMF reflexivo. Em seguida, o protótipo implementado para apoiar na sistemática da abordagem é utilizado para definir o modelo de contexto da organização, especificando as informações de contexto e seus possíveis valores com base no *Software Process Context Metamodel* (SPCM), um metamodelo proposto na abordagem. Finalizando as atividades da fase de EDPS, a ferramenta ATR (*Architect of Tailoring Rules*) é aplicada para definir regras, produzindo um modelo de decisão de variação com base no modelo de contexto e domínio de processo de software da organização. O modelo de decisão de variação representa formalmente as regras de transformação aplicadas para apoiar na definição do processo de software específico de projeto.

Na fase de EPEP, o protótipo desenvolvido pelos autores é utilizado para delimitar as informações de contexto, ou seja, atribuir valores para as informações especificadas no modelo de contexto. Em seguida, com base nos valores atribuídos às informações de contexto, o modelo de decisão de variação é interpretado e o processo de software específico de projeto é definido com base nas regras.

Além disso, HURTADO *et al.* adotam o conceito de *megamodeling* da *Global Model Management* (GMM) por meio da ferramenta AM3, um *plug-in open-source* do Eclipse. O conceito de *megamodeling* é aplicado para executar uma série de transformações de modelo para modelo (M2M), texto para modelo (T2M) e modelo para texto (M2T) para automatizar o ferramental de suporte, ocultando parte da complexidade envolvida. Porém, neste ponto, os autores relataram a ferramenta AM3 como instável.

JAFARINEZHAD & RAMSIN relatam o uso do *Fuzzy Logic Toolbox*, parte da ferramenta MATLAB. Esta ferramenta foi aplicada para representar um Sistema de Inferência Difusa, que utiliza internamente um mecanismo de regras difusas para realizar o mapeamento da entrada (*e.g.*, informações de contexto) para obter uma saída (*e.g.*, valor de aptidão de cada elemento de processo). O suporte ofertado pela ferramenta se limita as atividades da fase de EPEP. Neste ponto, os valores de aptidão de cada elemento de processo são considerados pelo gerente de projetos para determinar quais elementos de processo irão compor o processo de software específico de projeto. Os autores indicam que este processo de tomada de decisão pode ser automatizado, para isso, seria necessário definir um valor de aptidão mínimo para determinar quando um elemento de processo deve ser selecionado.

LORENZ *et al.* indicam o uso da ferramenta de suporte MfPTt, desenvolvida para oferecer suporte para definição de processo de software usando LPrS. A ferramenta MfPTt oferece suporte para registrar elementos de processo (*e.g.*, artefatos, tarefas, papéis, atividades), especificar arquiteturas de LPrS e definir processos de software específicos de projeto. Para apoiar as atividades da fase de EPEP, o módulo de adaptação da ferramenta MfPTt oferece cinco etapas, sendo estas: I. Definição do contexto de projeto; II. Seleção dos critérios de adaptação; III. Seleção da arquitetura de LPrS; IV. Priorização de atividades; e V. Definição de processo de software específico de projeto. Na etapa de priorização de atividades, o ferramental de suporte utiliza os

conceitos da técnica de AHP (*Analytic Hierarchy Process*) para apoiar na tomada de decisão realizada pelo gerente de projetos.

MAGDALENO *et al.* relataram o uso de duas ferramentas para apoiar nas atividades da abordagem, sendo essas: I. Odyssey, um ambiente de reutilização, utilizado para modelar e manter os modelos de LPrS (*e.g.*, características, contexto); e II. COMPOOTIM, uma ferramenta baseada no JMetal Framework, desenvolvida para apoiar na definição e otimização de processos de software específicos de projeto. Nas atividades da fase de EPEP, a ferramenta COMPOOTIM oferece apoio ao gerente de projetos na seleção de elementos de processo, bem como na otimização do processo de software considerando o aspecto de colaboração. Para isso, os modelos de LPrS são utilizados como entrada, onde os elementos de processo, regras de composição, regras de contexto e informações de contexto são interpretados para apoiar na definição do processo de software específico de projeto (*i.e.*, Sistema Baseado em Regras). Por fim, um conjunto de processos alternativos que maximizam o aspecto de colaboração é sugerido usando os conceitos da técnica de Algoritmo Genético.

ROUILLÉ *et al.* exploram o uso do ferramental CVL, baseado na linguagem *Common Variability Language* (CVL), uma linguagem independente de domínio utilizada para especificar e resolver variabilidades. Porém, poucos detalhes sobre o ferramental são disponibilizados, dificultando seu entendimento. De acordo com os autores, o ferramental é utilizado para apoiar na especificação da LPrS, bem como para apoiar a derivação automática de processos de software específicos de projeto. Para isso, três modelos são apresentados, sendo estes: I. Modelo de processo de software; II. VAM (*Variability Abstraction Model*), que captura as variabilidades de LPrS; e III. VRM (*Variability Realization Model*), que realiza o mapeamento entre as informações de contexto e os elementos de processo. Segundo ROUILLÉ *et al.*, os modelos utilizados pela abordagem contêm informações o suficiente para ofertar um modelo de processo de software sem variabilidades, ou seja, processo de software específico de projeto. Por fim, os autores indicam que o mecanismo de derivação depende da API do *Eclipse Modeling Framework* (EMF) para carregar, gerenciar e salvar os modelos.

TERNITÉ *et al.* indicaram o uso do V-Modell XT SPrL Framework, um framework de processo de software, que inclui metamodelos, ferramentas, implementações de referência e diretrizes). Neste cenário, duas ferramentas são

utilizadas para suportar as atividades da abordagem, sendo essas: I. V-Modell XT VMEd, um editor que oferece apoio para tarefas de criação, implementação e gerenciamento de processos; e II. V-Modell XT VMPPA, um assistente que apoia na definição de processo de software específico de projeto. Embora os autores indiquem outras ferramentas utilizadas para realizar o estudo de avaliação, estas não foram classificadas como ferramentas de suporte nesta revisão da literatura.

2.1.2.6. Avaliação

Na análise das abordagens, quatro tipos de avaliação foram identificados, sendo esses apresentados na Tabela 2.6.

Tabela 2.6 – Tipos de avaliação identificados (LPrS)

Tipo de Avaliação	Abordagem (Tabela 2.3)
Estudo de caso	ALEIXO <i>et al.</i> ; HURTADO <i>et al.</i> ; LORENZ <i>et al.</i> ; TERNITÉ <i>et al.</i>
Experimento controlado	JAUFMAN & MÜNCH; MAGDALENO <i>et al.</i> ; ROUILLÉ <i>et al.</i>
Avaliação com especialistas	GARCÍA <i>et al.</i> ; MAGDALENO <i>et al.</i>
Survey	BARRETO <i>et al.</i>
N/A	JAFARINEZHAD & RAMSIN; MARTÍNEZ-RUIZ <i>et al.</i> ; ROMBACH; THRÄNERT & WERNER

Neste ponto, embora algumas abordagens forneçam um exemplo para facilitar o entendimento do leitor, este tipo de relato não foi considerado um estudo de avaliação.

A Figura 2.4 apresenta a frequência de uso dos tipos de avaliação nas abordagens. Esta figura enfatiza o maior uso de estudos de caso (4) e experimentos controlados (3). Quatro abordagens não relataram nenhum tipo de avaliação realizada.

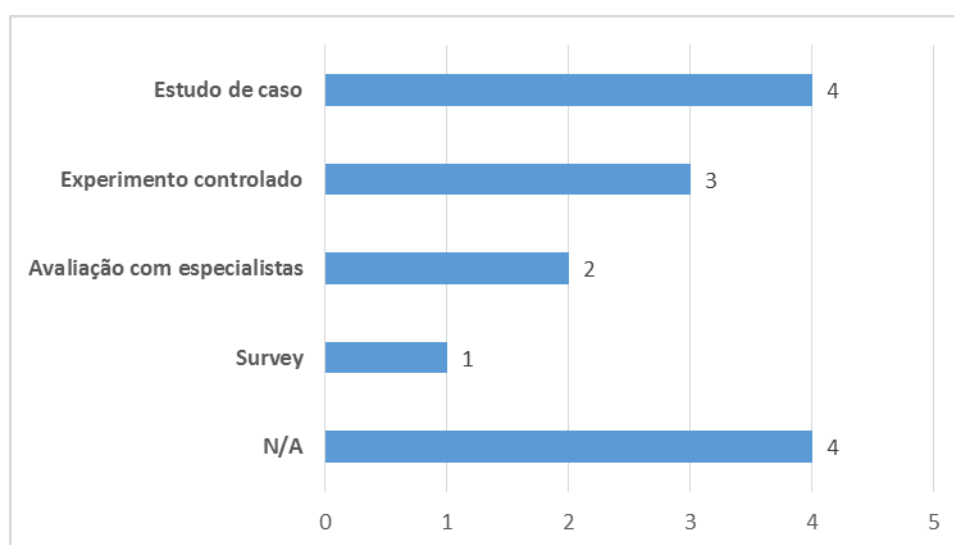


Figura 2.4 – Frequência dos tipos de avaliação (LPrS)

MAGDALENO *et al.* apresentaram mais de um tipo de avaliação (avaliação com especialistas e experimento controlado). Porém, o envolvimento da indústria foi relatado apenas na avaliação com especialistas. No experimento controlado, os resultados da técnica de Algoritmo Genético, aplicada para composição de processo de software, foram comparados com outras técnicas de busca (*e.g.*, *Hill Climbing*).

Neste cenário, quatro abordagens se destacam pelo envolvimento da indústria em suas avaliações: I. HURTADO *et al.*, por conduzirem múltiplos estudos de caso com empresas chilenas de desenvolvimento de software; II. MAGDALENO *et al.*, por realizarem uma avaliação com especialistas envolvendo uma grande empresa brasileira de petróleo e gás; III. ROUILLÉ *et al.*, por executarem um experimento controlado envolvendo uma empresa de serviços de software e computação; e IV. TERNITÉ *et al.*, por conduzirem um estudo de caso de longo prazo (2 anos) envolvendo múltiplas organizações. As abordagens de BARRETO *et al.* e LORENZ *et al.* não relataram envolvimento da indústria, no entanto, projetos em ambiente acadêmico foram utilizados.

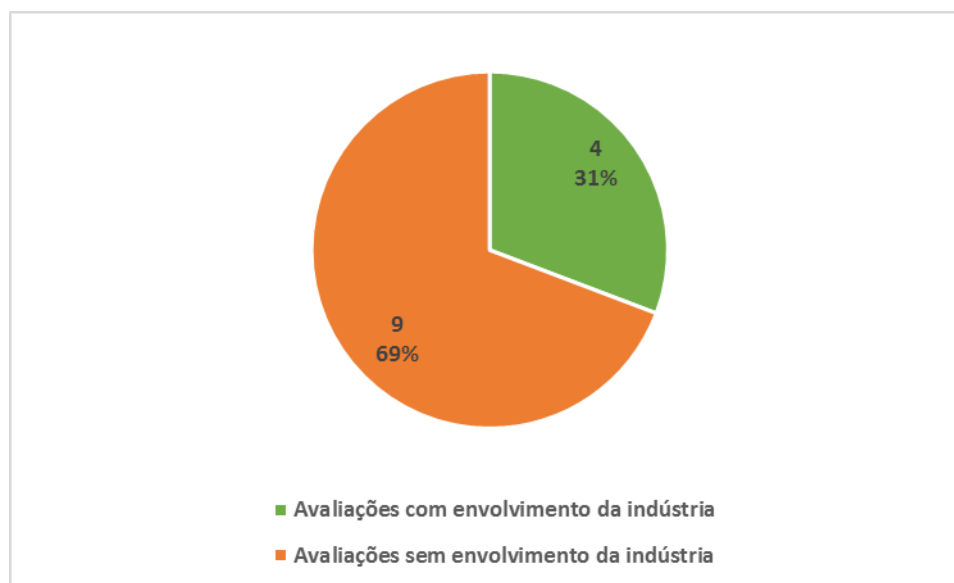


Figura 2.5 – Porcentagem do envolvimento da indústria nas avaliações (LPrS)

A porcentagem de abordagens que relataram o envolvimento da indústria em suas avaliações (Figura 2.5) indica que essa prática ainda é pouco explorada. No entanto, é necessário levar em consideração que LPrS é um tópico de pesquisa emergente.

2.1.3. Ameaças à Validade

Uma ameaça à validade desta revisão da literatura diz respeito a possível seleção incompleta ou inadequada de estudos primários. Para mitigar esta ameaça, os documentos selecionados pelo estudo de TEIXEIRA (2016) foram adotados, pois trata-se de uma revisão no tópico de LPrS que envolveu múltiplos pesquisadores de diferentes universidades brasileiras. Além disso, outra ameaça presente neste estudo está relacionada à etapa de extração de informações dos documentos. Neste ponto, para mitigar essa ameaça, além da definição do protocolo de pesquisa que envolve um formulário de extração, múltiplos pesquisadores analisaram os documentos de maneira isolada, realizando debates quando necessário para solucionar divergências.

2.1.4. Discussões

Conforme apresentado na seção de resultados, no geral, as abordagens relataram o uso de informações de contexto, bem como sua influência para definição de processo de software específico de projeto. Dentre as informações de contexto mais citadas pelas abordagens analisadas, estão: Criticidade (5), Tamanho do Projeto (4), Tamanho da Equipe (3), Complexidade (3), Tipo de Projeto (2), Experiência no Domínio (2), Cultura Organizacional (2), Estabilidade de Requisitos (2), Distribuição de Equipe (2) e Experiência da Equipe (2).

Embora ferramentas integradas tenham sido encontradas, um baixo nível de integração ferramental foi observado, principalmente, por conta da fragmentação de ferramentas que compõem o ferramental de suporte das abordagens identificadas. Neste ponto, ciente da importância de ferramentas integradas, assim como de interfaces e representações padronizadas para apoiar nas atividades de Engenharia de Software (OSSHER *et al.*, 2000), propostas de ferramentas integradas com suporte para as atividades das fases de EDPS e EPEP continuam sendo necessárias.

Conforme ressaltado nos resultados desta revisão, cinco técnicas de apoio para resolução de variabilidades de LPrS foram identificadas ao analisar as abordagens. Neste ponto, uma concentração de abordagens que utilizam técnicas baseadas em mapeamento e regras foi observada, o que pode representar uma limitação, pois o processo de aquisição de conhecimento exigido por este tipo de técnica não é trivial, principalmente, por conta da indisponibilidade de especialistas e sobrecarga na fase de EDPS. Esta limitação pode ser acentuada caso não exista nenhuma alternativa para

apoiar na resolução de variabilidades de LPrS durante a definição de processo de software específico de projeto.

Por fim, três problemas que afetam as atividades da fase de EPEP foram observados e podem representar oportunidades de pesquisa futura:

- I. Baixa compreensão sobre complexidade em modelos de LPrS e seu(s) impacto(s): Nenhuma das abordagens abordou os possíveis impactos do crescimento descontrolado do número de variabilidades de domínios de processos de software e sua relação com o esforço para definir um processo de software específico de projeto.
- II. Falta de compreensão sobre os impactos do processo de software definido: Nenhum dos estudos analisados apresentou suporte ou debate para analisar os possíveis impactos do processo de software definido, tanto na execução do projeto de software, quanto no produto de software resultante.
- III. Dependência de conhecimento de especialistas na tomada de decisão para resolução de variabilidades: O apoio à tomada de decisão observado nos estudos concentrou-se em explicitar o conhecimento de especialistas na fase de EDPS (*e.g.*, mapeamento e regras), o que pode representar uma possível sobrecarga nesta fase. Por outro lado, na fase de EPEP, a compreensão dos possíveis impactos gerados pela tomada de decisão para resolução de variabilidades de LPrS na definição de processo de software específico de projeto depende do conhecimento do responsável pela execução da atividade.

Esta revisão da literatura teve como objetivo caracterizar o estado da arte da fase de EPEP em LPrS. Neste ponto, destaca-se a importância de desenvolver ferramentas de suporte integradas, que ofereçam apoio para as atividades das fases de EDPS e EPEP, pois, no geral, as abordagens identificadas nesta revisão apresentaram um baixo nível de integração ferramental. Além disso, o problema de tomada de decisão para resolução de variabilidades de LPrS precisa ser melhor explorado, diversificando as técnicas aplicadas para apoiar o gerente de projetos ao tomar decisões durante a definição de processo de software específico de projeto.

Nesta dissertação de mestrado, focada nas atividades da fase de EPEP em LPrS, o terceiro problema é parcialmente explorado, ou seja, dependência de conhecimento de

especialistas na tomada de decisão para resolução de variabilidades de LPrS. Neste ponto, os conceitos da técnica de RBC são introduzidos no tópico de LPrS com o objetivo de mitigar a sobrecarga observada na fase de EDPS. Além disso, os conceitos da técnica de RBC são combinados com os conceitos da técnica SBR para ofertar mecanismos complementares de definição de processo de software. Conforme ressaltado no Capítulo 7, de Conclusão, espera-se que os demais problemas observados nesta revisão da literatura sejam explorados em trabalhos futuros.

2.2. Considerações Finais

Esse capítulo apresentou os conceitos da técnica de Linha de Processos de Software (LPrS), neste ponto, o tópico de LPrS é apresentado como uma técnica utilizada para representar um conjunto de processos dentro de um domínio particular ou para um propósito específico, que compartilham características, sendo estes construídos com base em ativos de processo reutilizáveis. Neste cenário, destaca-se o conceito de Engenharia de Linha de Processos de Software (ELPrS), que organiza as atividades e desafios desta técnica em duas fases principais: I. Engenharia de Domínio de Processo de Software (EDPS); e II. Engenharia de Processo Específico de Projeto (EPEP).

Além disso, trabalhos relacionados são analisados por meio de uma *Quasi-Revisão Sistemática da Literatura*, conduzida no escopo desta pesquisa, sendo esta revisão focada na definição de processo de software específico de projeto usando LPrS. Neste ponto, observou-se uma concentração de abordagens que utilizam técnicas baseadas em mapeamento e regras (*e.g.*, SBRs) para oferecer suporte na resolução de variabilidades de LPrS, durante a definição de processo de software específico de projeto. Por fim, foi observado que o ferramental de suporte ofertado pelas abordagens analisadas apresentou um baixo nível de integração, no que diz respeito ao suporte para as atividades das fases de EDPS e EPEP.

CAPÍTULO 3 - SISTEMA ESPECIALISTA

Sistema Especialista (SE) é um programa de computador que oferece soluções ou sugestões para resolução de problemas que exigem alto nível de especialização, ou seja, conhecimento sobre um domínio específico (BUCHANAN & DUDA, 1983). Originado na área de Inteligência Artificial (IA), os SEs são utilizados para executar tarefas específicas, onde se espera obter um desempenho similar ao produzido por especialistas humanos (*i.e.*, simular o comportamento de especialistas).

Na área de IA, os SEs representam um dos tipos de Sistema Baseado em Conhecimento (SBC). SBCs são utilizados para explicitar o conhecimento existente em determinado domínio, separando este do sistema, com o objetivo de resolver problemas com base no conhecimento mapeado. Neste cenário, um SBC é classificado como SE quando aplicado em cenários nos quais o conhecimento manipulado restringe-se ao conhecimento de alto nível de especificação para resolução de problemas ordinariamente resolvidos por humanos.

De acordo com YANG & YAU (2000), um SE é formado por duas fases: I. Desenvolvimento; e II. Resolução de problemas. Na fase de desenvolvimento, um engenheiro de conhecimento extrai conhecimento útil do(s) especialista(s) de domínio e o coloca em uma base de conhecimento transparente para reutilização. Na fase de solução de problemas, um usuário final insere problemas e, em seguida, o sistema fornece uma resposta ou sugestão, utilizando o conhecimento disponível na base de conhecimento.

A ideia desta técnica é separar e armazenar o conhecimento para futura manipulação, gerando uma reserva de conhecimento, chamada de base de conhecimento, que pode ser manipulada e examinada como qualquer outra estrutura de dados (BUCHANAN & DUDA, 1983). Neste ponto, o processo de aquisição de conhecimento desempenha um papel importante no desenvolvimento de SEs, onde diferentes estratégias podem ser utilizadas. De acordo com ABRAHAM (2005), o conhecimento pode ser obtido ao entrevistar especialistas de domínio e/ou aprendido por experiência.

Para a construção de SEs, diferentes técnicas são aplicadas, dentre essas estão (LIAO, 2005): I. Sistema Baseado em Regras; II. Raciocínio Baseado em Casos; III. Redes Neurais; e IV. Sistema de Inferência Difusa.

Segundo YANG & YAU (2000), um dos méritos associados aos SEs diz respeito ao suporte para modelar processos de tomada de decisão de especialistas de domínio. De acordo com LIAO (2005), há evidências positivas resultantes da aplicação de SEs no suporte ao processo de apoio à decisão e resolução de problemas em diferentes setores.

Embora diferentes técnicas sejam aplicadas para construção de SEs (*e.g.*, Redes Neurais, Sistema de Inferência Difusa), conforme indicado por LIAO (2005). Nesta dissertação de mestrado, duas técnicas são exploradas para especificar e implementar um SE com o objetivo de apoiar na atividade de definição de processo de software específico de projeto com base em artefatos de LPrS, sendo essas: I. Sistema Baseado em Regras (SBR); e II. Raciocínio Baseado em Casos (RBC). No cenário de LPrS, conforme observado nos resultados da revisão da literatura apresentada no capítulo anterior, os conceitos da técnica de SBR são aplicados por diferentes abordagens para apoiar na tomada de decisão para resolução de variabilidades de LPrS na fase de EPEP. Por outro lado, os conceitos da técnica de RBC são indicados na literatura como uma alternativa aos SBRs. Além disso, há evidências positivas para o uso combinado destas técnicas (YANG & YAU, 2000). Estas técnicas são descritas nas próximas subseções.

Este capítulo está organizado de forma que, além desta introdução ao conceito de SEs, a Seção 3.1 apresenta os conceitos da técnica de SBR. Na Seção 3.2, os conceitos da técnica de RBC são descritos, bem como os resultados de uma *Quasi-Revisão Sistemática da Literatura*, focada em abordagens que aplicam os conceitos da técnica RBC para definição de processo de software. Por fim, na Seção 3.3, as considerações finais do capítulo são apresentadas.

3.1. Sistema Baseado em Regras (SBR)

Sistema Baseado em Regras (SBR) é uma das técnicas aplicadas para construção de SEs (CARRICO *et al.*, 1989), tendo sua origem na área de IA no início da década de 1970. Esta técnica armazena e manipula o conhecimento obtido de especialistas humanos na forma de regras SE-ENTÃO ou sua variante SITUAÇÃO-AÇÃO (ABRAHAM, 2005, LIAO, 2005).

Nesta técnica, as regras são separadas do mecanismo de processamento, sendo essas armazenadas em uma base de conhecimento, também referenciada como base de regras. Cada regra disponível na base de conhecimento representa uma instrução condicional. Essas regras são utilizadas para realizar operações de inferir, com o objetivo de chegar a uma conclusão, resposta ou sugestão apropriada (HAYES-ROTH, 1985, LIAO, 2005).

Os SBRs evoluíram a partir dos sistemas de produção (NEWELL, 1973), que ao invés de visualizar a computação como uma sequência de operações pré-definidas, visualizam a computação como o processo de aplicação de regras de transformação em uma sequência determinada pelos dados (BUCHANAN & DUDA, 1983). Por outro lado, SBRs são utilizados para atender a necessidade de capturar, representar, armazenar, distribuir, raciocinar e aplicar conhecimento de especialistas humanos eletronicamente (HAYES-ROTH, 1985). Neste cenário, há evidência de aplicações em um vasto número de áreas (LIAO, 2005), como: I. Ensino; II. Desenvolvimento de software; III. Interpretação de histograma de DNA; IV. Avaliação de fraude de gestão; e V. Diagnóstico de falhas probabilísticas.

De acordo com HAYES-ROTH (1985), especialistas tendem a expressar a maior parte de suas técnicas de resolução de problemas em termos de um conjunto de regras de SITUAÇÃO-AÇÃO. Este ponto sugere que os SBRs deveriam ser utilizados para construção de sistemas especializados intensivos de conhecimento. Neste cenário, um SBR pode ser compreendido como um modelo que procura representar o raciocínio e conhecimento aplicado por especialistas na resolução de problemas específicos. Para isso, utiliza regras para representar o conhecimento do domínio e confronta essas regras com os fatos conhecidos sobre o problema, resultando na inferência de novos fatos para propor soluções, respostas ou sugestões.

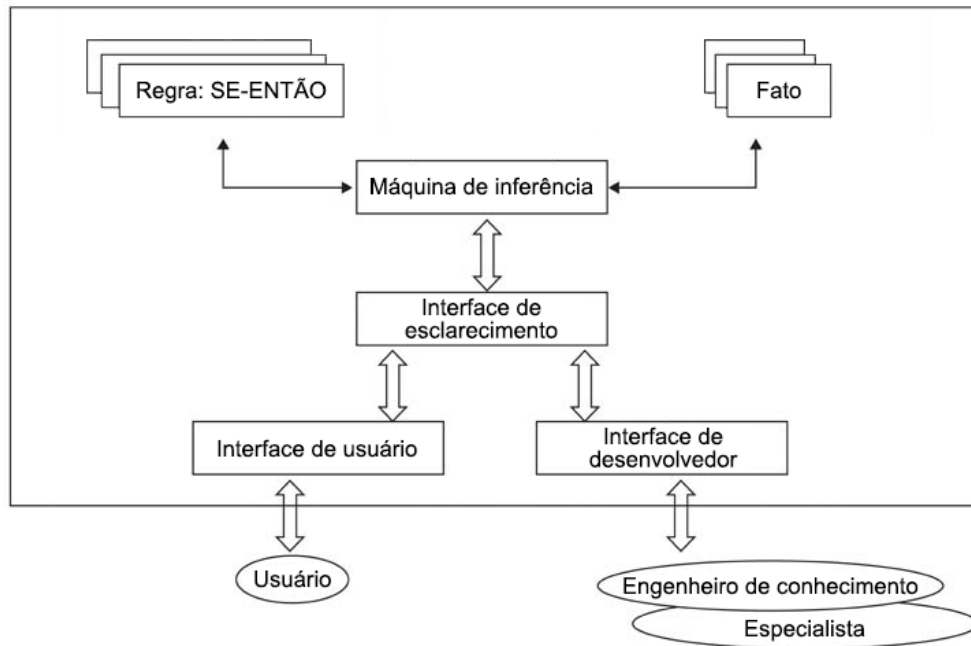


Figura 3.1 – Visão geral de SBRs, adaptado de WOO *et al.* (2014)

Conforme observado na Figura 3.1, um SBR é composto por diferentes componentes, sendo esses:

- **Base de conhecimento:** Conjunto de regras que representam o conhecimento de especialistas sobre um domínio específico;
- **Fatos:** Conjunto de fatos conhecidos, que podem estar armazenados no sistema ou serem obtidos em tempo de execução;
- **Máquina de inferência:** Componente responsável pelo processo de raciocínio (*i.e.*, inferência), interpretando regras e fatos para propor uma solução, resposta ou sugestão;
- **Interface de esclarecimento:** Componente responsável por oferecer informações ao usuário sobre as etapas de raciocínio;
- **Interface de usuário:** Componente responsável pela comunicação entre o usuário e o sistema para resolução de problemas;
- **Interface de desenvolvedor:** Componente responsável pela comunicação entre engenheiro de conhecimento/especialista e o sistema para criação e evolução da base de conhecimento.

Neste ponto, destaca-se que as regras armazenadas na base de conhecimento utilizam uma fórmula específica: SE {antecedente} ENTÃO {consequente}. Nesta fórmula, duas expressões são utilizadas, antecedente e consequente. A expressão antecedente representa uma premissa ou situação, que caso seja satisfeita (*i.e.*, verdadeira), resulta na execução da expressão consequente, que representa uma conclusão, sugestão, ação, etc. (HAYES-ROTH, 1985, ABRAHAM, 2005).

O componente de máquina de inferência, por outro lado, desempenha um papel central na resolução de problemas, interpretando as regras da base de conhecimento para gerar resultados. Neste ponto, duas formas distintas de configuração podem ser aplicadas para realizar o processo de inferência, sendo essas: I. Encadeamento direto; e II. Encadeamento reverso (HAYES-ROTH, 1985, ABRAHAM, 2005).

De acordo com HAYES-ROTH (1985), no encadeamento direto, uma regra é aplicada quando os fatos produzem uma situação que corresponde à sua expressão antecedente, ou seja, os fatos são processados usando as regras para obter conclusões (ABRAHAM, 2005). Por outro lado, no encadeamento reverso, segundo ABRAHAM (2005), uma hipótese (*e.g.*, solução, objetivo) é processada em busca de regras que comprovem sua veracidade, examinando sucessivamente quaisquer regras relacionadas. Neste ponto, as regras candidatas são consideradas individualmente e as condições da expressão antecedente não satisfeitas são analisadas, trabalhando de forma recursiva. O procedimento termina quando o objetivo principal (*i.e.*, hipótese) é reduzido a um conjunto de objetivos específicos (HAYES-ROTH, 1985).

Dessa forma, pode-se definir que o encadeamento direto é orientado a dados, enquanto o encadeamento reverso é direcionado por objetivos (ABRAHAM, 2005). Neste cenário, vale ressaltar que não existe uma configuração recomendada, cada cenário de aplicação irá exigir diferentes configurações da máquina de inferência, onde pode-se optar por um tipo de encadeamento específico ou até mesmo combinar os diferentes tipos para obter o resultado esperado.

3.2. Raciocínio Baseado em Casos (RBC)

Raciocínio Baseado em Casos (RBC) é uma técnica utilizada para a construção de SEs, originada na área de IA. Segundo WATSON (1999), RBC representa uma metodologia, onde diferentes tecnologias podem ser aplicadas para atingir seus objetivos. Neste ponto, a ideia básica desta técnica é adaptar soluções que foram usadas

para resolver problemas anteriores e usá-las para resolver novos problemas (RIESBECK & SCHANK, 1989, LIAO, 2005).

A ciência cognitiva indica que os humanos utilizam experiências adquiridas ao solucionar problemas passados para resolver problemas atuais. Neste ponto, o uso da experiência não se limita aos cenários de sucesso, onde experiências positivas são usadas para resolver um problema, mas também para evitar falhas ao chegar à solução utilizando experiências negativas. Os conceitos da técnica de RBC se baseiam nesta indicação para armazenar, recuperar e adaptar casos resolvidos anteriormente, a fim de resolver um problema específico (FOUQUE & MATWIN, 1992).

De acordo com LEAKE (1996), RBC é uma técnica computacional que combina a filosofia de SBCs e a simulação do raciocínio humano ao utilizar experiências passadas para propor soluções. Nesta técnica, um problema é solucionado por meio da percepção de sua similaridade com um ou vários problemas anteriores. Dessa forma, as soluções conhecidas destes problemas anteriores são adaptadas ao invés de elaborar uma solução do zero (MANTARAS & PLAZA, 1997).

De acordo com YANG & YAU (2000), na técnica de RBC, as experiências passadas desempenham um papel importante na resolução de problemas, uma vez que exibem como as decisões são tomadas em determinadas situações. Segundo LIAO (2005), as experiências passadas de especialistas humanos são armazenadas em uma base de conhecimento para posterior recuperação, onde cada experiência é representada como um caso. Neste cenário, quando um problema surge, casos semelhantes são recuperados. Se um dos casos retornados for idêntico ao problema descrito, sua solução será proposta; caso contrário, um dos casos retornados pode ser modificado para se adaptar ao problema, automaticamente ou manualmente, gerando uma solução específica (CHEN, 2013).

Quando um problema específico é resolvido, um caso é gerado combinando sua descrição e solução elaborada. Este caso é usado para alimentar a base de conhecimento com o objetivo de apoiar a resolução de problemas semelhantes no futuro. Este procedimento torna o resultado da técnica mais eficaz, além de possibilitar uma melhora de desempenho ao longo do tempo, adquirindo mais experiência e evitando repetições de erros já conhecidos (YANG & YAU, 2000).

Neste cenário, quantidade e diversidade de casos na base de conhecimento são pontos relevantes. Ao construir SEs por meio da técnica de RBC, alguns casos podem ser introduzidos com o objetivo de oferecer suporte inicial aos usuários com base em cenários básicos, ou seja, problemas específicos, idealizados pelos engenheiros de conhecimento (WANGENHEIM & WANGENHEIM, 2008).

Segundo WATSON (1999), três princípios orientam a adoção dos conceitos da técnica de RBC para construção de sistemas:

- O desejo de resolver problemas específicos por meio de soluções passadas;
- Reutilizar uma solução sugerida por um dos casos recuperados, com ou sem adaptação; e
- Aumentar o conhecimento ao armazenar casos na base de conhecimento.

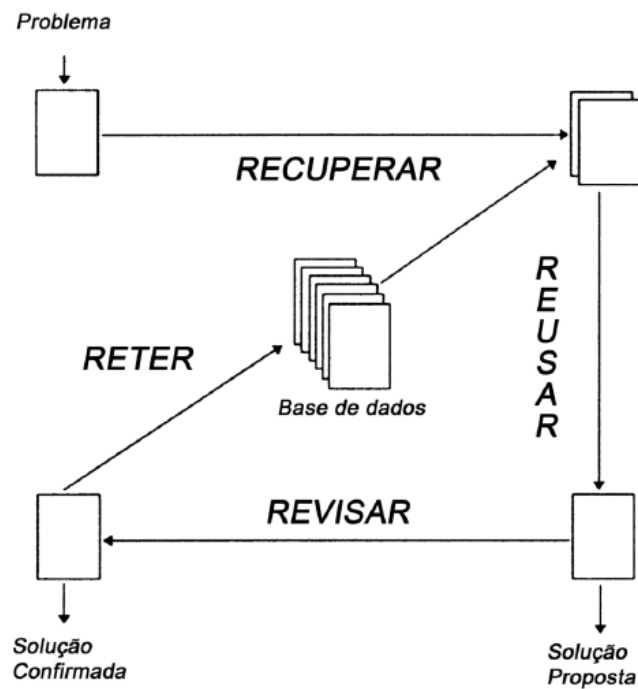


Figura 3.2 – Ciclo RBC, adaptado de AAMODT & PLAZA (1994)

Por outro lado, AAMODT & PLAZA (1994) indicam que RBC representa um ciclo, conforme observado na Figura 3.2, sendo este composto por quatro atividades:

- Recuperar: Nesta atividade, os casos armazenados são resgatados de acordo com sua similaridade ao problema descrito;
- Reusar: Neste ponto, um dos casos é selecionado, automaticamente ou manualmente, de forma que sua solução seja adotada;
- Revisar: Nesta atividade, a solução adotada é revisada e adaptada, se necessário, para se adequar ao problema descrito;
- Reter: Por fim, após elaborar e confirmar uma solução para o problema descrito, estes artefatos são armazenados, representando um caso.

De acordo com KOLODNER (1992), ao adotar os conceitos da técnica de RBC, duas estratégias costumam ser utilizadas sendo essas: I. Interpretativa; e II. Resolução de problemas.

Na estratégia interpretativa, o objetivo principal é discutir se uma situação específica deve ou não ser tratada como as anteriores, com base nas semelhanças e diferenças entre os casos. Na estratégia de resolução de problemas, o objetivo é construir uma solução para um caso específico, adaptando soluções de casos passados. Embora essa divisão seja útil para apresentar os conceitos da técnica, nem sempre esta divisão fica clara na prática, pois determinadas aplicações utilizam ambas as estratégias de forma combinada (MANTARAS & PLAZA, 1997).

Embora o uso da técnica de RBC seja indicado na literatura como uma alternativa para SBR, há evidências do uso combinado destas técnicas. Porém, quando comparadas, RBC apresenta algumas vantagens por reduzir o gargalo da elicitação de conhecimento, bem como pela introdução do conceito de aprendizado incremental (LIAO *et al.*, 1998, YANG & YAU, 2000). De acordo com LIAO (2005), há evidências na literatura de utilização da técnica de RBC em diferentes áreas, dentre elas: I. Gestão do conhecimento; II. Inspeção ultrassônica; e III. Diagnóstico de falha. Neste ponto, CHEN (2013) ressalta que diferentes áreas de pesquisa têm utilizado RBC e obtido resultados positivos ao explorar o conceito de aprendizado incremental.

3.2.1. Métrica de Similaridade

Segundo BURKHARD (1998), a atividade de recuperação, presente no ciclo RBC, tem como objetivo: I. Recuperar o menor número possível de casos; e II. Recuperar casos relacionados e semelhantes ao problema. Neste ponto, duas características se destacam, rapidez (*i.e.*, tempo de resposta) e eficiência (*i.e.*, relação entre os resultados obtidos e os recursos empregados), pois impactam diretamente na resolução de problemas (CHEN, 2013).

Embora diferentes tecnologias sejam aplicadas na atividade de recuperação, segundo WATSON (1999), o algoritmo *vizinhos mais próximos*, em inglês, *Nearest Neighbors* (NN), destaca-se pelo número de aplicações em diferentes áreas de pesquisa. Essa técnica considera um espaço multidimensional, onde cada caso é representado por um ponto. Nesta representação, a busca por casos semelhantes consiste em encontrar os casos que estão geometricamente mais próximos (WANGENHEIM & WANGENHEIM, 2008). Neste cenário, uma variação da técnica de vizinhos mais próximos, denominada K-NN, se diferencia do original ao limitar o número de casos recuperados (K).

Para encontrar os casos geometricamente mais próximos, um cálculo de semelhança é realizado entre o problema descrito e os casos da base de conhecimento, repetidamente, par a par, para todos os casos. Este cálculo é representado por uma métrica de similaridade, também referenciada na literatura como medida de similaridade ou função de distância. Nesta dissertação, conforme recomendado por FINNIE & SUN (2002), o termo métrica de similaridade é adotado.

Após realizar o cálculo de semelhança, os casos da base de conhecimento são ordenados com base no valor obtido e os casos com maior nível de similaridade são sugeridos como solução potencial para o problema (WANGENHEIM & WANGENHEIM, 2008).

De acordo com o princípio local-global (BERGMANN, 2002), para definir uma métrica de similaridade, deve-se considerar dois níveis de similaridade: I. Similaridade local, que determina como será realizado o cálculo da similaridade entre cada atributo que descreve o caso; e II. Similaridade global, que determina como os valores obtidos através da similaridade local serão computados para comparar o caso atual com os demais casos da base de conhecimento. No nível de similaridade local, diferentes

métricas podem ser propostas, visando os diferentes tipos de atributos utilizados na descrição dos casos e problema. Além disso, é possível definir pesos para destacar os atributos mais relevantes.

Diferentes métricas de similaridade podem ser encontradas nas revisões da literatura de WILSON & MARTINEZ (1997) e LIAO *et al.* (1998). Neste cenário, a métrica *vizinhos mais próximos ponderada*, em inglês, *Weighted Nearest Neighbors*, destaca-se como uma das métricas mais utilizadas (FINNIE & SUN, 2002, WANGENHEIM & WANGENHEIM, 2008), sendo esta apresentada na Figura 3.3.

$$\text{sim}(Q, C) = \frac{\sum_{i=1}^n f(Q_i, C_i) \times w_i}{\sum_{i=1}^n w_i}$$

Figura 3.3 – *Weighted Nearest Neighbors* (WANGENHEIM & WANGENHEIM, 2008)

Nesta métrica, w_i representa a importância ponderada de um atributo i (*i.e.*, valores numéricos entre 0 e 1). Q_i e C_i representam os valores para o atributo i , no problema atual e caso específico, respectivamente. Por fim, f representa a métrica de similaridade local, definida de acordo com o tipo do atributo. Dessa forma, os vizinhos mais próximos possuem valores mais próximos de 1, enquanto os vizinhos mais distantes têm valores próximos de 0.

3.2.2. Revisão da Literatura - RBC

Com base na questão de pesquisa que direciona esta dissertação de mestrado, e ciente que humanos utilizam experiências adquiridas ao solucionar problemas passados para resolver problemas atuais (FOUQUE & MATWIN, 1992), os conceitos da técnica de RBC são explorados nesta dissertação para apoiar a reutilização de processos de software dentro do tópico de LPrS.

Dessa forma, com o objetivo de compreender a relação entre os conceitos de RBC e a atividade de definição de processo de software, bem como analisar o suporte ofertado para tomada de decisão na definição de processo de software, uma

Quasi-Revisão Sistemática da Literatura foi conduzida. Esta seção apresenta os principais resultados obtidos.

Esta revisão está organizada de forma que, além desta introdução, na Seção 3.2.2.1, as etapas de planejamento e execução são detalhadas; na Seção 3.2.2.2, os resultados são apresentados; na Seção 3.2.2.3, as ameaças à validade são analisadas; e por fim, na Seção 3.2.2.4, uma discussão é introduzida.

3.2.2.1. Planejamento e Execução

Para atingir o objetivo desta revisão, um protocolo de pesquisa foi elaborado (Apêndice B). Com base neste protocolo, a *string* de busca definida foi utilizada nas bases eletrônicas selecionadas (*i.e.*, Scopus, Compendex e IEEE Xplore), com foco no conteúdo completo das publicações. A execução desta revisão da literatura foi realizada em dezembro de 2017. O número de documentos retornados e considerados relevantes é apresentado na Tabela 3.1. Neste ponto, para identificar os documentos relevantes, o título e o resumo de cada publicação retornada foram analisados, considerando os critérios de inclusão e exclusão especificados no protocolo de pesquisa desta revisão da literatura.

Tabela 3.1 – Números da etapa de execução nas bases eletrônicas (RBC)

Base	Resultado	Relevante	Disponível	Duplicado	Selecionado
Scopus	386	16	15	N/A	15
Compendex	24	12	11	11	0
IEEE Xplore	239	6	6	5	1
Total de documentos:					16

Conforme observado na Tabela 3.1, 16 documentos foram considerados relevantes pelo filtro inicial. Em seguida, estes documentos foram lidos e analisados, considerando todo seu conteúdo, e metade dos documentos foram retirados, resultando na seleção de 8 publicações.

Para apoiar na identificação de documentos, a técnica de *snowballing* (WOHLIN, 2016) foi aplicada sobre os documentos selecionados, utilizando a estratégia de *backward* (*i.e.*, análise de documentos referenciados). Como resultado, dois documentos relevantes foram identificados. Dessa forma, no total, 10 documentos foram selecionados para análise e extração de informações (Tabela 3.2).

Tabela 3.2 – Documentos selecionados para extração de informações (RBC)

Nº	Publicação	Snowballing	Ano
1	GRUPE, F. H., URWILER, R., RAMARAPU, N. K., <i>et al.</i> , 1998, "The application of case-based reasoning to the software development process", <i>Information and Software Technology</i> , v. 40, n. 9 (Sept), pp. 493-499.	X	1998
2	HENNINGER, S. "An environment for reusing software processes". In: <i>Proceedings of the Fifth International Conference on Software Reuse</i> , pp. 103-112, Victoria, BC, Canada, Aug. 1998.		1998
3	FUNK, P. J., CRNKOVIC, I. "Reuse, validation and verification of system development processes". In: <i>Proceedings of the Tenth International Workshop on Database and Expert Systems Applications</i> , pp. 300-304, Florence, Italy, Aug. 1999.	X	1999
4	HENNINGER, S., BAUMGARTEN, K. "A Case-Based Approach to Tailoring Software Processes". In: <i>Case-Based Reasoning Research and Development</i> , v. 2080, Lecture Notes in Computer Science, Springer, pp. 249-262, 2001.		2001
5	REIS, R. Q., REIS, C. A. L., NUNES, D. J. "Automated support for software process reuse: Requirements and early experiences with the APSEE model". In: <i>Proceedings of the Seventh International Workshop on Groupware</i> , pp. 50-55, Darmstadt, Germany, Sept. 2001.		2001
6	REIS, R. Q., REIS, C. A. L., SCHLEBBE, H., <i>et al.</i> "Early experiences on promoting explicit separation of details to improve software process reusability". In: <i>Proceedings of the 26th Annual International Computer Software and Applications Conference</i> , pp. 373-378, Oxford, UK, Aug. 2002.		2002
7	SANTOS, V., CORTÉS, M., BRASIL, M. "Dynamic Management of the Organizational Knowledge Using Case-Based Reasoning". In: <i>Evaluation of Novel Approaches to Software Engineering</i> , v. 69, Communications in Computer and Information Science, Springer, pp. 220-233, 2008.		2008
8	SANTOS, V., CORTÉS, M., BRASIL, M. "Reuse and Adaptation of Software Process using Similarity Measurement". In: <i>Proceedings of the 4th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)</i> , pp. 215-226, Milan, Italy, May. 2009.		2009
9	SANTOS, V. A., CORTÉS, M. I. "Organizational Knowledge Management through Software Process Reuse and Case-based Reasoning". In: <i>Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS)</i> , pp. 223-228, Madeira, Portugal, June. 2010.		2010
10	PARK, S. H., BAE, D. H., 2013, "Tailoring a large-sized software process using process slicing and case-based reasoning technique", <i>IET Software</i> , v. 7, n. 1 (Feb), pp. 47-55.		2013

3.2.2.2. Resultados da Revisão

3.2.2.2.1. Distribuição de Documentos por Ano

A Figura 3.4 apresenta a distribuição dos documentos selecionadas na etapa de execução por ano. Neste ponto, as publicações identificadas estão distribuídas no período entre 1998 e 2013.

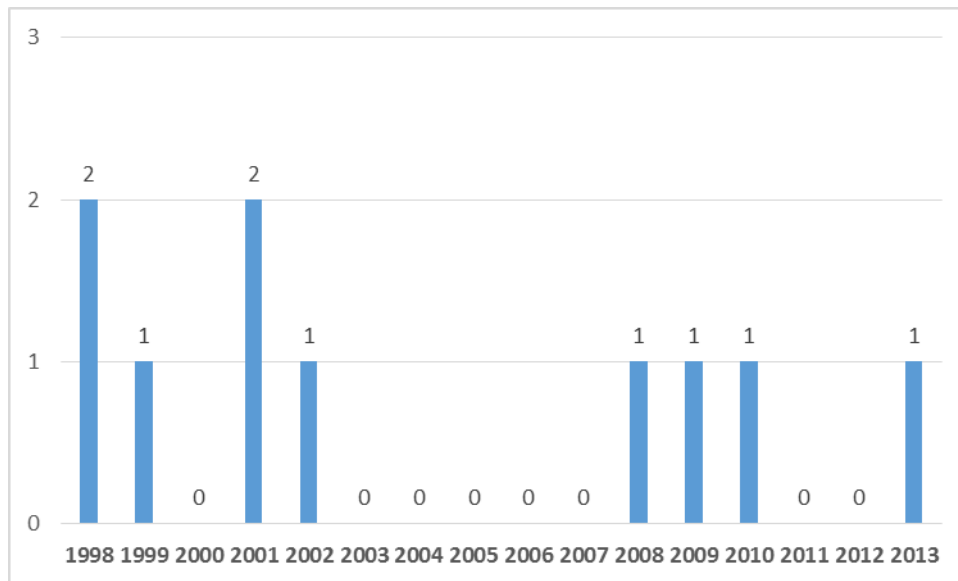


Figura 3.4 – Distribuição de documentos por ano (RBC)

3.2.2.2.2. Abordagens Identificadas

Com base nos documentos selecionados, 6 abordagens foram identificadas, conforme observado na Tabela 3.3.

Tabela 3.3 – Abordagens identificadas (RBC)

Abordagem	Documento (Tabela 3.2)
FUNK & CRNKOVIC	[3]
GRUPE <i>et al.</i>	[1]
HENNINGER & BAUMGARTEN	[2][4]
PARK & BAE	[10]
REIS <i>et al.</i>	[5][6]
SANTOS <i>et al.</i>	[7][8][9]

FUNK & CRNKOVIC apresentam uma abordagem focada na atividade de recuperação de processos de software. Trata-se de uma das primeiras aplicações RBC no cenário de processo de software. Neste ponto, não há relatos sobre a representação de modelos ou suporte ofertado pelo ferramental (CABS *system*) para apoiar na tomada de decisão.

GRUPE *et al.* apresentam uma aplicação RBC para reutilização de processos de software. Nesta abordagem, uma ferramenta comercial (CBR2) é adotada com foco na atividade de recuperação de processos de software utilizando fatores de estimativa da técnica de pontos de função para realizar o cálculo de similaridade entre os casos e o problema descrito. Não há relatos sobre como o processo de software é representado,

porém, informações sobre o processo de software (*e.g.*, nome, descrição e atributos) são disponibilizadas para apoiar na tomada de decisão ao retornar os casos similares.

HENNINGER & BAUMGARTEN propõem uma abordagem focada na técnica de SBR para apoiar a definição de processo de software específico de projeto. Além disso, os conceitos da técnica de RBC são aplicados de forma auxiliar para apoiar a reutilização de experiências de projeto (*e.g.*, observações de execução, documentações, artefatos). Esta abordagem utiliza *template* de processo de software, referenciado pelos autores como *schema* de processo de software. Porém, não há relatos sobre a representação de modelos de processo de software. Cada caso do repositório representa uma atividade selecionada para compor um processo de software específico de projeto. Neste cenário, ao definir um processo de software específico de projeto, para cada atividade selecionada a partir do *template*, um caso é gerado para armazenar as experiências relacionadas à sua execução. Neste ponto, a rastreabilidade entre as atividades do *template* e os casos é mantida, dessa forma, as experiências de projeto relacionadas à uma atividade específica podem ser observadas desde o momento da definição do processo de software específico de projeto até o término da execução deste no projeto de software.

PARK & BAE propõem uma abordagem que utiliza modelos de dependência de processos de software (sDPM) para apoiar na adaptação de processo de software. Não há relatos sobre a representação de modelos de processo de software. O repositório da abordagem armazena experiências de adaptação de processo de software, onde cada caso é composto por elementos de processo e ações de adaptação. Esta abordagem tem o objetivo de minimizar a intervenção humana na definição de processo de software. Dessa forma, os casos são recuperados do repositório e aplicados de acordo com seu nível de similaridade, sem ofertar o processo de tomada de decisão ao usuário, utilizando o caso de maior similaridade internamente.

REIS *et al.* propõem uma abordagem para definição de processo de software, onde *templates* e processos de software específicos de projeto são utilizados como base para apoiar a reutilização de processos de software. Para representar os *templates* e processos de software é utilizada a notação APSEE Model. Neste ponto, políticas de adaptação podem ser definidas nos *templates*. Na atividade de recuperação desta abordagem, o usuário deve optar pela busca de *templates* ou processos de software

específicos de projeto. Não há relatos sobre o suporte ofertado pelo ferramental (APSEE-Reuse *system*) para apoiar na tomada de decisão.

SANTOS *et al.* apresentam uma abordagem baseada em *templates* de processo de software, representados por meio da notação WebAPSEE-PML. Nesta abordagem, os *templates* de processo de software representam os casos, sendo estes descritos por meio de 13 atributos. Na atividade de recuperação, os casos são analisados de acordo com o nível de similaridade dos atributos. Os *templates* de processo de software retornados contam com informações para apoiar na tomada de decisão (*e.g.*, nome, nível de similaridade e nível de sucesso). Neste ponto, vale ressaltar que a abordagem incentiva o procedimento de *feedback*, onde ao realizar a retenção de um caso no repositório, o usuário pode indicar o nível de sucesso do mesmo, ou seja, o nível de aderência do processo de software.

3.2.2.2.3. Representação dos Casos

A Tabela 3.4 apresenta como cada abordagem representa os casos armazenados no repositório. Neste ponto, metade das abordagens identificadas utiliza o processo de software específico de projeto para representação dos casos.

Tabela 3.4 – Representação dos casos nas abordagens (RBC)

Abordagem (Tabela 3.3)	Representação dos Casos
GRUPE <i>et al.</i>	Processo de Software
HENNINGER & BAUMGARTEN	Atividade
FUNK & CRNKOVIC	Processo de Software
REIS <i>et al.</i>	<i>Template</i> de Processo de Software e Processo de Software
SANTOS <i>et al.</i>	<i>Template</i> de Processo de Software
PARK & BAE	Ações de Adaptação de Processo de Software

3.2.2.2.4. Descrição dos Casos

Todas as abordagens analisadas nesta revisão da literatura exploraram o uso de atributos para realizar o cálculo de similaridade, porém, poucas abordagens apresentaram os atributos utilizados.

A Tabela 3.5 apresenta um resumo da relação entre abordagem e atributos utilizados para descrever os casos.

Tabela 3.5 – Descrição dos casos nas abordagens (RBC)

Abordagem (Tabela 3.3)	Atributos	Agrupamento	Raciocínio	Peso por Relevância
GRUPE <i>et al.</i>	14 fatores de complexidade	N/A	Fatores da técnica pontos de função	Não
HENNINGER & BAUMGARTEN	N/A	N/A	N/A	Não
FUNK & CRNKOVIC	N/A	Processo e Projeto	N/A	Não
REIS <i>et al.</i>	N/A	Processo, Ambiente, Organização, Software e Projeto	N/A	Não
SANTOS <i>et al.</i>	13 informações de contexto	Processo e Projeto	Trabalhos relacionados	Sim
PARK & BAE	9 informações de contexto	N/A	N/A	Não

GRUPE *et al.* apresentam o conjunto de atributos utilizado para descrever os casos, 14 no total, sendo estes: Comunicação de Dados, Funções Distribuídas, Desempenho, Configuração Altamente Utilizada, Taxa de Transação, Entrada de Dados Online, Eficiência do usuário final, Atualização Online, Processamento Complexo, Reutilização, Facilidade de Instalação, Facilidade Operacional, Uso de Sites Múltiplos e Facilitação de Mudança. Neste ponto, os atributos representam fatores da técnica de análise de pontos de função.

HENNINGER & BAUMGARTEN e REIS *et al.* não apresentaram atributos específicos para descrever os casos no repositório. Nesta abordagem, os casos são recuperados a partir da rastreabilidade mantida a partir do *template* de processo de software, onde cada atividade de processos de software específicos de projeto representa um caso.

FUNK & CRNKOVIC não relatam nenhum atributo, porém, destacam dois agrupamentos, processo e projeto, como fonte de atributos, que podem ser interpretadas nesta dissertação de mestrado como entidades de contexto.

REIS *et al.* embora não apresentem atributos, destacam cinco agrupamentos, sendo estes: I. Processo; II. Ambiente; III. Organização; IV. Software; e V. Projeto.

SANTOS *et al.* relataram o uso de 13 atributos. Sete atributos do agrupamento projeto: Modelo de Ciclo de Vida, Complexidade, Tamanho, Tamanho da Equipe, Tempo, Conhecimento de Engenharia de Software, Paradigma de Desenvolvimento. Além disso, seis atributos do agrupamento processo são utilizados, sendo esses: Modelo

de Desenvolvimento, Modelo de Maturidade, Nível de Maturidade, Complexidade, Processo, Experiência no Uso do Processo. De acordo com os autores, trabalhos relacionados foram analisados para identificar os atributos adotados. Por fim, esta abordagem é a única identificada que trabalha com diferentes pesos por atributo para apoiar no cálculo de similaridade dos casos.

PARK & BAE indicam o uso de 9 atributos, sendo estes: Ciclo de Vida, Tipo de Desenvolvimento, Tipo de Software, Tamanho do Projeto, Complexidade, Estabilidade dos Requisitos, Criticidade, Risco de Cronograma, Risco de Custo.

3.2.2.2.5. Suporte para Tomada de Decisão

GRUPE *et al.* apresentam uma abordagem focada na reutilização de processos de software que oferece recursos para apoiar na tomada de decisão. Dentre os recursos ofertados, além do nível de similaridade, informações sobre os casos são fornecidas, como nome, descrição e os atributos que descrevem o caso no repositório.

SANTOS *et al.* focam no suporte ferramental para modelagem e reutilização de processos de software. Na atividade de reutilização de processos de software, o suporte para tomada de decisão ocorre ao ofertar informações como: nome, nível de similaridade e nível de sucesso (*i.e.*, aderência). Neste ponto, o nível de sucesso de cada caso é indicado pelo usuário. Logo, trata-se de uma informação subjetiva inserida com o objetivo de apoiar na tomada de decisão para reutilização futura.

Neste cenário, as demais abordagens não citaram informações ou recursos específicos, além do nível de similaridade, para apoiar na tomada de decisão.

3.2.2.2.6. Avaliação

Apenas duas abordagens apresentaram estudos de avaliação. SANTOS *et al.* apresentou uma prova de conceito, porém, este tipo de relato pouco diz respeito sobre os benefícios reais da abordagem. PARK & BAE apresentaram um experimento sobre a eficácia da abordagem, porém, este experimento limitou-se na análise da técnica de RBC, que é aplicada de forma auxiliar.

Dessa forma, nenhum dos estudos apresentou um estudo visando a correlação entre o nível de similaridade dos atributos utilizados para descrever os casos e a oportunidade de reutilização de processos de software. Além disso, nenhuma análise foi apresentada sobre o ferramental de suporte ofertado, com o objetivo de caracterizar sua

aceitação ou analisar se os recursos ofertados de fato apoiam na definição de processo de software.

3.2.2.3. Ameaças à Validade

Uma ameaça à validade desta revisão da literatura diz respeito a possível seleção incompleta ou inadequada de estudos primários. Para mitigar esta ameaça, adotou-se uma abordagem sistemática, incluindo um protocolo de pesquisa e o uso da técnica de *snowballing*. Neste cenário, o uso isolado da estratégia de *backward* da técnica de *snowballing* representa uma limitação deste estudo.

3.2.2.4. Discussões

Ciente da importância dos atributos que descrevem os casos armazenados na base de conhecimento para analisar o nível de similaridade de cada caso quando surge um problema, três abordagens apresentaram os atributos utilizados (GRUPE *et al.*, PARK & BAE, SANTOS *et al.*), porém, apenas duas abordagens justificaram os atributos adotados (GRUPE *et al.*, PARK & BAE). Neste ponto, em nenhum dos trabalhos analisados fica claro se houve uma investigação sistemática envolvida para definir os atributos ou avaliação para verificar se estes representam uma boa descrição para apoiar no cenário de definição de processo de software específico de projeto.

Os documentos analisados nesta revisão da literatura foram publicados no período entre 1998 e 2013, porém, nenhum dos estudos trabalhou sobre determinados desafios específicos da reutilização de processos de software (*e.g.*, gerência de variabilidades), bem como não foi encontrado nenhuma aplicação da técnica de RBC no tópico de LPrS. Neste cenário, considerando o ponto de vista de tomada de decisão, o suporte ofertado pelas abordagens analisadas apresenta oportunidades de melhoria, principalmente, para apoiar na resolução de variabilidades de domínios de processos de software.

Nenhuma das abordagens analisadas preocupou-se em apresentar uma comparação entre os atributos de cada caso e a descrição do problema utilizada para realizar a atividade de recuperação de casos similares, bem como não possibilitaram analisar o modelo do processo de software ao retornar os casos. Neste ponto, um estudo pode ser realizado para propor uma releitura do suporte ofertado por aplicações RBC, no que diz respeito à análise de casos recuperados visando apoiar na resolução de

variabilidades de domínios de processos de software ao observar as decisões tomadas em casos similares.

Por fim, todas as abordagens analisadas nesta revisão da literatura apresentaram mecanismos únicos para apoiar na definição de processos de software específicos de projeto. Considerando a existência de diferentes perfis de profissionais e organizações, observou-se pouca flexibilidade para atender profissionais com diferentes níveis de experiência e organizações com diferentes níveis de maturidade. Neste ponto, embora o estudo de HENNINGER & BAUMGARTEN apresente uma combinação entre as técnicas Sistema Baseado em Regras e Raciocínio Baseado em Casos, apenas um mecanismo é ofertado (*i.e.*, mecanismo híbrido).

Esta revisão da literatura teve como objetivo caracterizar abordagens de definição de processo de software que aplicam os conceitos da técnica de RBC para apoiar a reutilização de processos de software. Espera-se, por meio dos resultados apresentados, ressaltar a importância de interpretar os desafios de cada área de aplicação bem como adaptar o suporte ofertado os conceitos da técnica de RBC para apoiar na resolução de problemas específicos.

3.3. Considerações Finais

Neste capítulo, SEs são descritos como programas de computador construídos para resolver problemas específicos, que exigem alto nível de especialização, para desempenhar tarefas específicas e apresentar resultados similares aos produzidos por especialistas humanos. Neste cenário, embora uma variedade de técnicas seja aplicada para construção destes sistemas, duas técnicas são destacadas nesta dissertação de mestrado: I. Sistema Baseado em Regras (SBR); e II. Raciocínio Baseado em Casos (RBC).

SBRs são introduzidos como uma técnica que armazena e manipula o conhecimento de especialistas no formato de regras SE-ENTÃO ou SITUAÇÃO-AÇÃO. Para resolver problemas específicos, o conhecimento armazenado em forma de regras é interpretado por uma máquina de inferência, gerando uma solução, sugestão, etc.

Por outro lado, RBC é apresentado como uma técnica que representa um raciocínio que resolve problemas usando ou adaptando soluções de problemas antigos, usando como motivação o modo de funcionamento da competência humana

especializada, que recupera e modifica problemas previamente resolvidos e armazenados na memória.

Além disso, trabalhos relacionados são analisados por meio de uma *Quasi-Revisão Sistemática da Literatura*, conduzida no escopo desta pesquisa, sendo essa focada na aplicação dos conceitos de RBC para apoiar na definição de processo de software. Neste ponto, observou-se que o suporte ofertado pelas abordagens disponíveis na literatura não está adequado ao conceito de gerência de variabilidades de processo de software, o que ressalta oportunidades de melhoria para adequar o suporte ofertado pela técnica de RBC para apoiar na tomada de decisão visando a resolução de variabilidades.

As oportunidades de melhoria identificadas nesta revisão foram exploradas na especificação da abordagem *Odyssey-ProcessCase*, estabelecida nesta dissertação de mestrado, que trabalha com os conceitos de reutilização parcial e total de processo de software específico de projeto, descritos no Capítulo 4. Neste cenário, o conceito de reutilização parcial utiliza os conceitos da técnica de RBC para apoiar na resolução de variabilidades de domínios de processos de software, o que exigiu o desenvolvimento de funcionalidades específicas no ferramental de suporte.

CAPÍTULO 4 - ABORDAGEM *ODYSSEY-PROCESSCASE*

4.1. Introdução

Conforme destacado pelos resultados da revisão da literatura (LPrS), observou-se uma concentração de abordagens de LPrS que aplicam técnicas baseadas em mapeamento e regras para apoiar na tomada de decisão visando definir um processo de software específico de projeto. Porém, o processo de aquisição de conhecimento exigido por este tipo de técnica não é trivial, principalmente, por conta da indisponibilidade de especialistas e sobrecarga na fase de EDPS. Por outro lado, o uso da técnica de RBC é indicado na literatura como uma alternativa aos SBRs (LIAO *et al.*, 1998), tendo como motivação o modo de funcionamento da competência humana especializada, que recupera e modifica as soluções de problemas previamente resolvidos e armazenados na memória (SMYTH & CUNNINGHAM, 1992).

Esta dissertação de mestrado visa estabelecer uma abordagem, denominada *Odyssey-ProcessCase*, que aplica os conceitos das técnicas de RBC e SBR para ofertar mecanismos complementares de definição de processo de software. O objetivo desta abordagem é apoiar à tomada de decisão para resolução de variabilidades de LPrS visando a definição de um processo de software específico de projeto. Neste ponto, busca-se mitigar a complexidade desta atividade, bem como a dependência de especialistas para tomada de decisão.

Presume-se que a adoção da abordagem *Odyssey-ProcessCase* é ideal para organizações compostas por múltiplos projetos, com diferentes clientes ou requisitos contratuais variáveis, como fábricas de software, porém, não se restringem a este cenário de aplicação. Neste ponto, estudos experimentais e avaliações envolvendo a indústria são necessários para confirmar essa expectativa. Em organizações focadas em serviços e produtos de software individuais, este esforço pode não fazer sentido. Porém, caso exista uma LPrS disponível (*i.e.*, modelos de LPrS especificados visando um domínio), supõe-se que os benefícios da abordagem poderão ser obtidos até mesmo nestes cenários.

Para apresentar a abordagem *Odyssey-ProcessCase*, além desta seção introdutória, este capítulo está organizado da seguinte forma: a Seção 4.2 apresenta o contexto de pesquisa desta dissertação de mestrado; a Seção 4.3 descreve os requisitos

iniciais para definição da abordagem; a Seção 4.4 detalha uma visão geral da abordagem *Odyssey-ProcessCase*, suas etapas e atividades; e por fim, na Seção 4.5, as considerações finais do capítulo são apresentadas.

4.2. Contexto de Pesquisa

Esta dissertação de mestrado foi realizada no contexto do grupo de Reutilização de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ. Neste cenário, estudos conduzidos pelo grupo, bem como pesquisas externas, foram utilizados como base para definir a sistemática da abordagem *Odyssey-ProcessCase* e implementar seu ferramental de suporte.

Dentre os estudos utilizados, estão: I. metamodelo e notação *OdysseyProcess-FEX*, que oferece meios para representar explicitamente os conceitos de características, componentes, variabilidade e opcionalidade em LPrS; II. Abordagem *UbiFEX*, que fornece mecanismos para apoiar a modelagem de características sensíveis ao contexto, originada no tópico de LPS; III. Metodologia *OdysseyProcessReuse*, que descreve uma sistemática para reutilização de processos de software no tópico de LPrS; e IV. Modelo de Contexto para Adaptação de Processo de Software, que sugere 31 informações de contexto distribuídas em seis entidades de contexto para apoiar na adaptação de processos de software.

Estes trabalhos são apresentados brevemente nesta seção, que está organizada de forma que além desta introdução, na Seção 4.2.1, o metamodelo e notação *OdysseyProcess-FEX* é introduzido. Na Seção 4.2.2, a abordagem *UbiFEX* é apresentada. Na Seção 4.2.3, a metodologia *OdysseyProcessReuse* é descrita. Na Seção 4.2.4, o modelo de contexto para adaptação de processo de software definido por LEITE (2011) é apresentado.

4.2.1. Metamodelo e Notação *OdysseyProcess-FEX*

O metamodelo *OdysseyProcess-FEX* (TEIXEIRA *et al.*, 2016) foi definido com o propósito de representar explicitamente os conceitos de características, componentes, variabilidade e opcionalidade associados à representação dos elementos que constituem um processo de software em LPrS, como unidades de trabalho (atividades e tarefas), papéis, produtos de trabalho e ferramentas. Por outro lado, para representar simbolicamente os conceitos formalizados pelo metamodelo, uma notação foi especificada.

Diferentes fontes foram analisadas para realizar a definição do metamodelo *OdysseyProcess-FEX* (e.g., modelos de processos, metamodelos, ontologias). Neste ponto, para trabalhar os conceitos de modelagem de características da *OdysseyProcess-FEX*, o metamodelo e notação *Odyssey-FEX* (OLIVEIRA, 2006) foi adotado, levando em consideração sua abrangência e recursos para representar variabilidades referentes às características de domínio de software. Trata-se de uma abordagem anotativa para gerência de variabilidades, que utiliza estereótipos e anotações visuais (DIAS, 2015).

A Figura 4.1 apresenta um modelo de características de exemplo definido para ilustrar os recursos da notação *OdysseyProcess-FEX*.

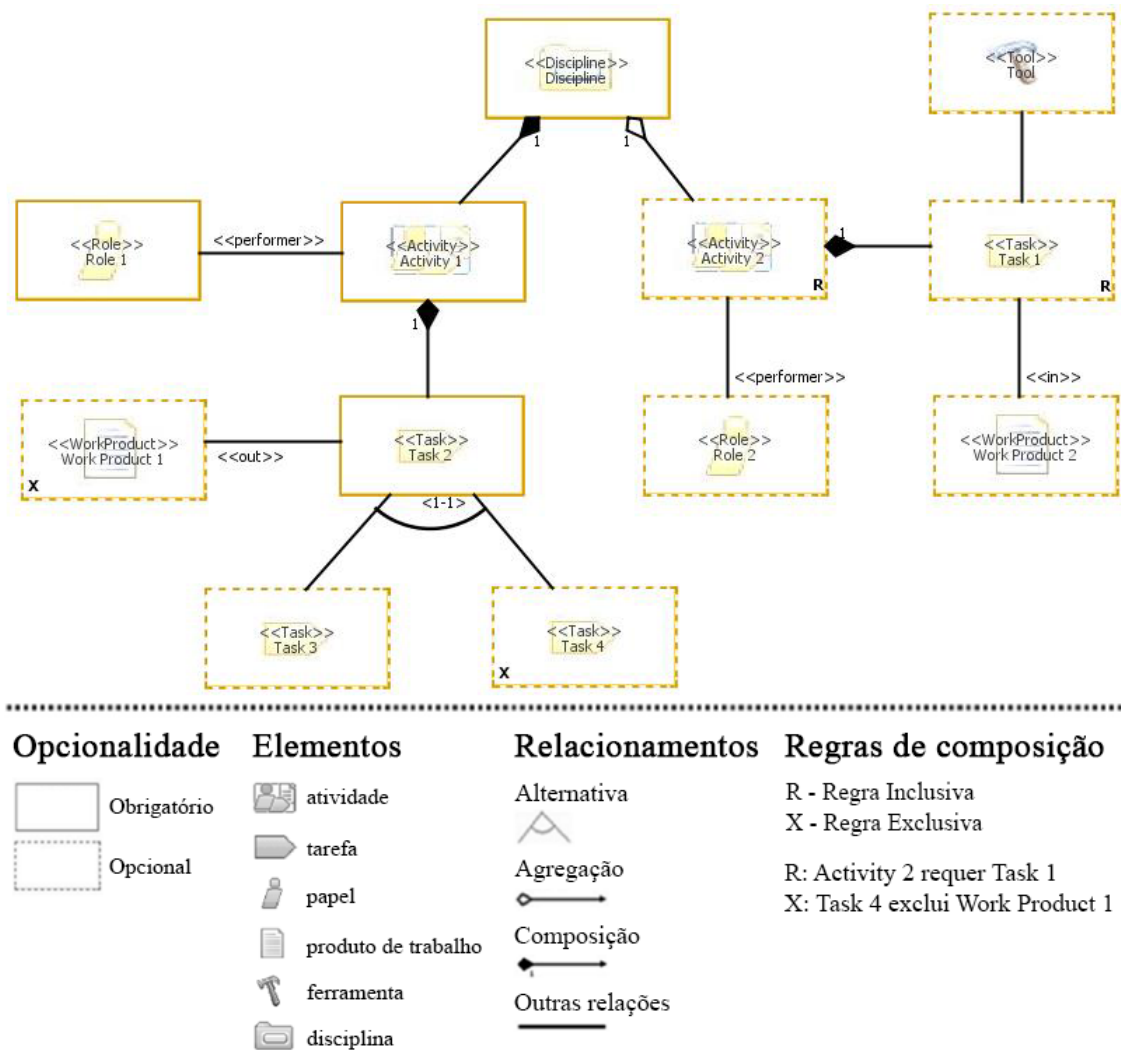


Figura 4.1 – Modelo de características de exemplo - *OdysseyProcess-FEX*

Neste cenário, as atividades consistem em grupos lógicos de unidades de trabalho menores, que podem corresponder a outras atividades ou tarefas. Uma tarefa pode ser descrita por meio de passos que a compõem. Ferramentas podem ser

associadas a unidades de trabalho (*i.e.*, atividade ou tarefa), representando o suporte ferramental para sua execução. Papéis estabelecem um conjunto de habilidades e conhecimentos requeridos por um indivíduo, ou por um grupo, para executar uma atividade ou tarefa. Produtos de trabalho descrevem artefatos a serem consumidos, modificados ou produzidos pelas atividades ou tarefas. Por fim, disciplinas representam uma forma para organização dos elementos de processo.

Para representar as variabilidades dos elementos de processo, três opções são disponibilizadas:

- **Pontos de variação:** Elementos que refletem a parametrização no domínio de maneira abstrata. Estes elementos são configuráveis por meio das variantes;
- **Variantes:** Elementos necessariamente ligados a um ponto de variação, que atuam como alternativas para configurar determinado ponto de variação;
- **Invariantes:** Elementos "fixos", não configuráveis no domínio.

As classificações de elementos com relação à variabilidade são mutuamente excludentes entre si. Desta forma, um elemento não pode receber simultaneamente dois tipos diferentes de classificação quanto à variabilidade.

Com o objetivo de representar a opcionalidade dos elementos de processo, duas opções são ofertadas:

- **Mandatário:** Elemento de processo que deverá estar presente em todos os processos derivados a partir do modelo de características;
- **Opcional:** Elemento de processo que pode ou não estar presente em processos derivados a partir do modelo de características.

As classificações de elementos com relação à opcionalidade são mutuamente excludentes entre si. Desta forma, um elemento não pode receber simultaneamente dois tipos diferentes de classificação quanto à opcionalidade.

Para representar os relacionamentos entre os elementos de processo são disponibilizados diversos tipos de representações, dentre eles:

- **Alternativo:** Relacionamento existente entre um ponto de variação e suas variantes;

- **Agregação:** Representa uma associação do tipo todo/parte;
- **Composição:** Representa uma associação do tipo todo/parte, porém, mais forte que relacionamentos do tipo agregação. Neste relacionamento, as partes não existem independentes do todo;
- **Ligação:** Relacionamento para associar elementos específicos: I. Unidade de trabalho (atividade ou tarefa) e Papel; II. Unidade de trabalho (atividade ou tarefa) e Produto de trabalho; III. Papel e Produto de trabalho; e IV. Unidade de trabalho (atividade ou tarefa) e Ferramenta.

Por fim, regras de composição são disponibilizadas para caracterizar restrições de dependência e mútua exclusividade entre elementos (*i.e.*, regras SE-ENTÃO), podendo envolver múltiplos elementos de processo. Tais regras são criadas por meio de expressões e operadores lógicos no domínio do processo de software para influenciar diretamente na definição de processo de software específico de projeto. Dois tipos de regras de composição são disponibilizados:

- **Regra de Composição Inclusiva:** Indica dependência entre dois ou mais elementos de processo (*i.e.*, regra do tipo *requer*). Estas regras são representadas por uma marcação "R_n", no canto inferior direito de todos os elementos pertencentes à regra;
- **Regra de Composição Exclusiva:** Indica mútua exclusividade entre dois ou mais elementos de processo (*i.e.*, regra do tipo *exclui*). Estas regras são representadas por uma marcação "X_n", no canto inferior esquerdo de todos os elementos pertencentes à regra.

Em ambos os tipos de regras de composição, a marcação também apresenta uma numeração (n), que representa a ordem sequencial de criação das regras de cada tipo.

4.2.2. Abordagem *UbiFEX*

A abordagem *UbiFEX* (FERNANDES, 2009) foi definida com o objetivo de fornecer mecanismos para apoiar a modelagem de características de Linha de Produtos de Software Sensíveis ao Contexto (LPSSC). Esta abordagem é dividida em dois componentes: I. *UbiFEX-Notation*, uma notação para modelagem de entidades e informações de contexto relevantes para um determinado domínio, bem como para representar a influência dessas informações na resolução de suas variabilidades do

domínio; e II. *UbiFEX-Simulation*, um mecanismo para apoiar na resolução das variabilidades do domínio em relação às variações de contexto.

A Figura 4.2 apresenta um modelo de contexto de exemplo especificado na notação *UbiFEX*.

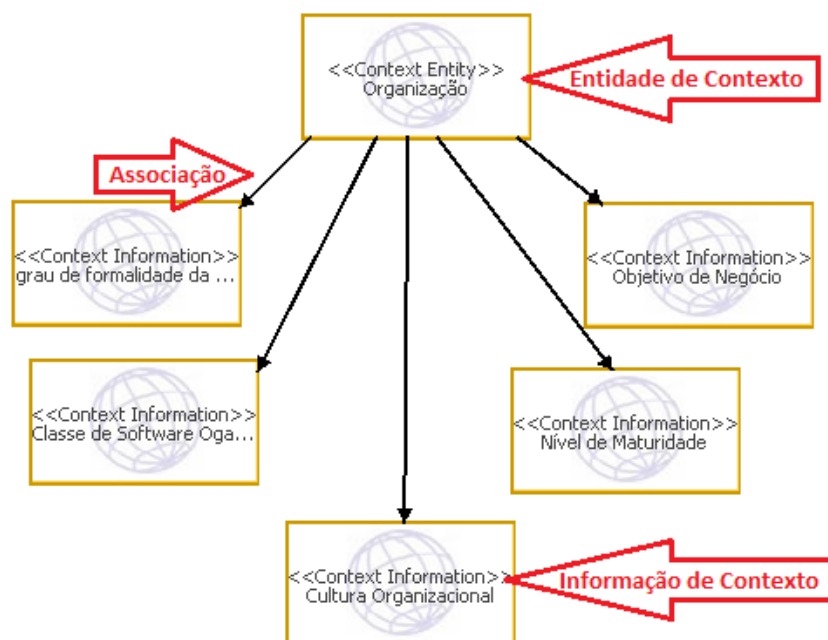


Figura 4.2 – Modelo de contexto de exemplo - *UbiFEX* (FERNANDES, 2009)

Neste cenário, para representar a influência das informações de contexto na resolução de variabilidades de domínio, a notação *UbiFEX* trabalha com os conceitos de entidade de contexto, informação de contexto, definição de contexto e regra de contexto.

Entidades de contexto representam as dimensões de software relevantes, que devem ser monitoradas e podem contribuir para adaptar o domínio. Cada entidade é caracterizada por informações de contexto, que representam os dados que devem ser coletados para descrever as entidades.

As definições de contexto descrevem situações relevantes para o domínio, com base nas entidades e informações de contexto. Cada definição de contexto possui um identificador e expressão (*i.e.*, condição para que o contexto ocorra), que pode envolver múltiplas informações de contexto, operadores relacionais (*e.g.*, maior que, menor que, maior ou igual) e valores específicos, respeitando os tipos *string*, inteiro, *float* ou

booleano. Neste ponto, uma definição de contexto pode combinar outras definições de contexto previamente especificadas, bem como utilizar o operador de negação.

As regras de contexto representam como determinadas situações impactam no domínio por meio de ações (*i.e.*, inclusão ou exclusão de características). Regras de contexto podem indicar, por exemplo, decisões para resolução de pontos de variação específicos (*e.g.*, sugestão de inclusão de característica variante) ou resolução de opcionalidade (*e.g.*, sugestão de inclusão de característica opcional). Cada regra de contexto possui um identificador e uma expressão. Neste ponto, a expressão da regra de contexto é formada por três componentes: antecedente, operador (*e.g.*, implica) e consequente. O antecedente pode ser formado por definições de contexto, operadores lógicos e/ou elementos de processo (*i.e.*, características). O consequente, por outro lado, é formado por um ou mais elementos de processo (*i.e.*, características).

As regras de contexto são responsáveis por fazer a ligação entre o modelo de contexto e o modelo de características do domínio, representando a influência de um modelo no outro. Neste ponto, vale ressaltar que as regras de contexto devem respeitar as regras de composição (*i.e.*, parte do modelo de características).

4.2.3. Metodologia *OdysseyProcessReuse*

A metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) descreve uma sistemática para reutilização de processos de software que adota os conceitos de LPrS e Desenvolvimento de Processos Baseados em Componentes (DPBC).

Segundo TEIXEIRA (2016), um componente de processo pode ser compreendido como uma abstração, um fragmento de processo baseado no princípio de "caixa-preta". Dessa forma, um componente de processo representa uma parte modular de um processo que encapsula seu conteúdo e se comunica com o ambiente via interfaces. Cada componente é composto por unidades de trabalho (*e.g.*, atividades, tarefas), unidades essas, que quando executadas apresentam o comportamento do componente e produzem os resultados esperados (*e.g.*, produtos de trabalho).

Cinco elementos são propostos pela metodologia *OdysseyProcessReuse*:

- Método para Engenharia de Linha de Processos de Software (ELPrS), focado no desenvolvimento *para* reutilização (EDPS);

- Proposta de representação de LPrS – modelagem de variabilidades (metamodelo e notação *OdysseyProcess-FEX*);
- Mecanismos de mapeamento entre artefatos de diferentes níveis de abstração;
- Procedimentos para suportar o agrupamento de componentes de processos; e
- Técnicas de verificação sintática e semântica dos artefatos.

A sistemática descrita pela metodologia *OdysseyProcessReuse* é apresentada por meio da visão de ELPrS, composta por duas fases: I. Engenharia de Domínio de Processo de Software (EDPS); e II. Engenharia de Processo Específico de Projeto (EPEP). Na fase de EDPS, dois níveis de abstração são apresentados. No primeiro nível de abstração, três modelos são especificados: modelo de características; modelo de contexto; e modelo de subescopos do domínio. No segundo nível de abstração, o modelo de componentes é proposto pela metodologia. Estes níveis de abstração são mapeados por meio de heurísticas.

Neste cenário, o modelo de características é definido por meio da notação *OdysseyProcess-FEX* (TEIXEIRA *et al.*, 2016) para refletir o conhecimento sobre o domínio, identificando similaridades e variabilidade. O modelo de contexto é definido por meio da notação *UbiFEX* (FERNANDES *et al.*, 2011), para caracterizar entidades, informações e situações de contexto esperadas (*i.e.*, definições de contexto), bem como definir regras de contexto para situações específicas, com o objetivo de apoiar na seleção de elementos de processo durante a definição de processo de software específico de projeto.

De forma complementar, o modelo de subescopos do domínio permite categorizar as características identificadas e organizar o domínio. Segundo TEIXEIRA (2016), trata-se de um recurso opcional, onde dependendo da abrangência do domínio, a definição de subáreas dentro do escopo total pode ser importante para estabelecer nichos de elementos e categorizar os elementos segundo alguns critérios definidos pela organização. O objetivo deste modelo é representar um escopo padrão e escopos específicos, sendo estes elaborados pela seleção de elementos do modelo de características.

Por fim, o modelo de componentes, definido por meio do metamodelo *OdysseyProcess-FEX* (TEIXEIRA *et al.*, 2016), representa o segundo nível de abstração

da metodologia *OdysseyProcessReuse*. Este modelo visa oferecer uma visão modular por meio de blocos de construção, que combinam elementos de processo relacionados. Neste ponto, o objetivo do modelo de componentes é reduzir o acoplamento e aumentar o nível de modularidade da LPrS ao organizá-la em componentes relacionados via interface de dados e controle. Interfaces de dados representam a troca de produtos de trabalho e interfaces de controle representam fluxos de execução entre componentes.

4.2.4. Modelo de Contexto para Adaptação de Processo de Software

O modelo de contexto para adaptação de processo de software proposto por LEITE (2011) é composto por seis entidades e 31 informações de contexto, identificadas por meio de uma revisão da literatura. Além disso, o modelo proposto especifica um conjunto de situações de contexto elaboradas com base nas entidades e informações de contexto identificadas. Os resultados do estudo conduzido por LEITE (2011) foram analisados por meio de duas etapas: I. Avaliação com especialistas, onde uma versão inicial foi avaliada por um conjunto de especialistas; e II. Estudo exploratório, onde o modelo adaptado com base na avaliação com especialistas foi aplicado em uma organização para verificar sua viabilidade.

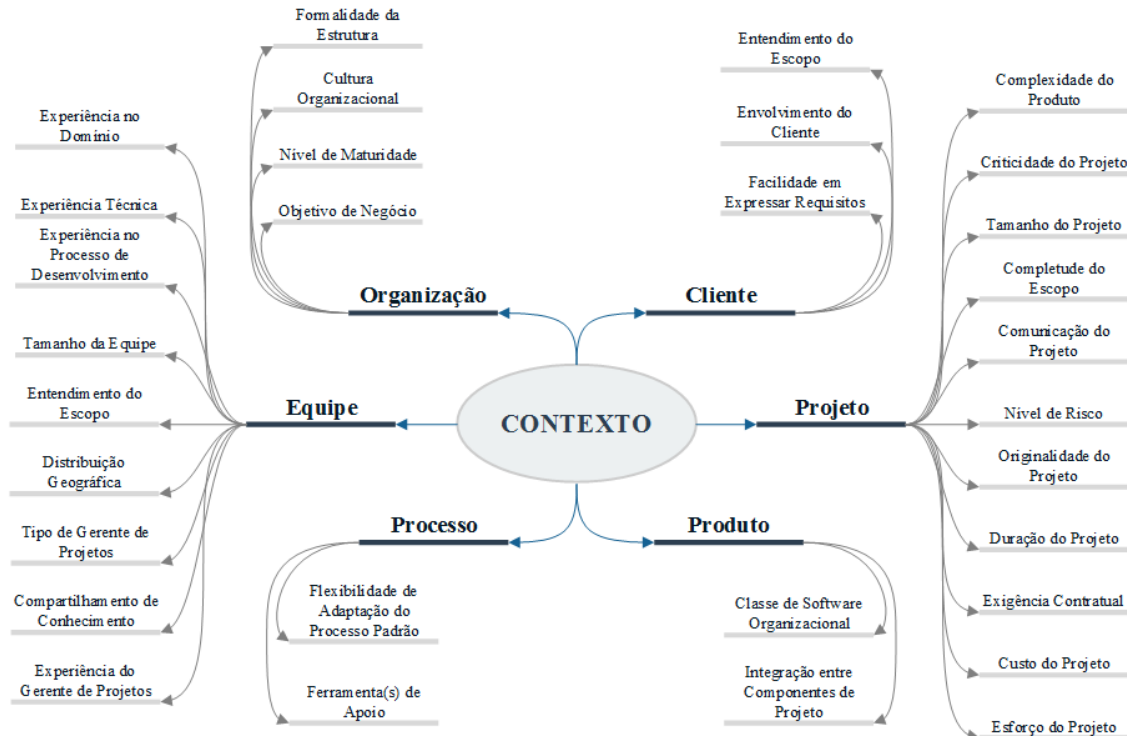


Figura 4.3 – Entidades e informações de contexto, adaptado de LEITE (2011)

As entidades e informações de contexto do modelo são apresentadas na Figura 4.3 e detalhadas no Apêndice C.

Este modelo apresenta as entidades e informações de contexto associadas à sua definição. Cada informação de contexto está relacionada com o seu tipo (*e.g.*, *string*), modo de aquisição (*e.g.*, dados históricos, entrevista, plano de projeto), alternativas pré-definidas, etc.

Considerando determinadas propostas de modelos de contexto disponíveis na literatura, como o de BOEHM & TURNER (2003) e o *Octopus Model* (KRUCHTEN, 2013), bem como as análises realizadas nas revisões da literatura conduzidas no escopo desta dissertação de mestrado, apresentadas no Capítulo 2 e 3, o modelo para adaptação de processos de software proposto por LEITE (2011) foi adotado parcialmente. Neste ponto, as entidades e informações de contexto do modelo foram utilizadas para representar o contexto de definição de processo de software, conforme descrito no próximo capítulo. Nenhuma melhoria foi realizada sobre o modelo nesta dissertação de mestrado.

Neste cenário, dois critérios foram utilizados para direcionar essa adoção: I. Conjunto de informações de contexto disponibilizado; e II. Nível de detalhamento das informações (*e.g.*, descrição, alternativas pré-definidas).

4.3. Requisitos da Abordagem *Odyssey-ProcessCase*

Com base nas observações realizadas ao conduzir as revisões da literatura, apresentadas no Capítulo 2 e 3, requisitos foram identificados para definição da abordagem, sendo estes:

- **Ofertar a possibilidade de análise de similaridade de informações de contexto:** Com base na primeira suposição de pesquisa apresentada no Capítulo 1, este requisito visa possibilitar a identificação de processos de software definidos em contextos similares, considerando suas caracterizações (*i.e.*, contexto de definição de processo de software). Este ponto representa uma das atividades essenciais da técnica de RBC, utilizada na abordagem *Odyssey-ProcessCase*;
- **Oferecer mecanismos complementares para a definição de processo de software:** Com base na segunda suposição de pesquisa apresentada no Capítulo

1, este requisito visa ofertar mecanismos complementares para apoiar na tomada de decisão durante a definição de processo de software específico de projeto usando LPrS, com o objetivo de aumentar a flexibilidade da abordagem para atender diferentes perfis de organizações e profissionais na indústria de software;

- **Centralizar artefatos de processo de software:** Os artefatos gerados pela abordagem devem ser centralizados por meio de um repositório de processos de software, com o objetivo de facilitar a identificação e recuperação destes artefatos para apoiar a reutilização de processos de software;
- **Ofertar recursos de avaliação para apoiar a reutilização de processos de software:** Permitir que os processos de software definidos por meio da abordagem sejam avaliados para apoiar sua reutilização, ofertando os recursos de submissão de feedback de aderência ao projeto e controle de disponibilidade para reutilização.

4.4. Visão Geral - *Odyssey-ProcessCase*

Durante a definição de processo de software a partir de artefatos de LPrS, presume-se que os gerentes de projetos necessitam de apoio para tomada de decisão, focado na resolução de variabilidades de LPrS para selecionar os elementos de processo que irão compor o processo de software específico de projeto.

Neste cenário, a abordagem *Odyssey-ProcessCase* oferece suporte por meio de diferentes mecanismos para definição de processo de software. Embora independentes, as funcionalidades dos mecanismos ofertados são complementares. Dessa forma, profissionais e organizações com diferentes perfis podem adaptar o uso da sistemática de acordo com a sua realidade.

A abordagem *Odyssey-ProcessCase* consiste na composição de recursos para representação de LPrS, gestão de contexto e definição de processo de software específico de projeto. Os recursos para representação de LPrS e gestão de contexto são ofertados ao adotar trabalhos anteriores, apresentados na Seção 4.2, de Contexto de Pesquisa (FERNANDES *et al.*, 2011, LEITE, 2011, TEIXEIRA *et al.*, 2016, TEIXEIRA, 2016), principalmente, para apoiar na fase de EDPS. Dessa forma, essa dissertação tem como foco de contribuição os mecanismos de definição de processo de

software da fase de EPEP, que aplicam os conceitos das técnicas de RBC e SBR como suporte para tomada de decisão durante a resolução de variabilidades de LPrS, conforme descrito ao longo deste capítulo.

A Figura 4.4 apresenta uma visão geral da abordagem *Odyssey-ProcessCase*, composta por uma etapa de pré-requisitos e três etapas principais: Engenharia de Domínio de Processo de Software (EDPS); Engenharia de Processo Específico de Projeto (EPEP); e Avaliação de Reusabilidade. A etapa de pré-requisitos centraliza informações organizacionais (*e.g.*, contexto da organização, colaboradores, projetos de software) para possibilitar o trabalho cooperativo por meio da sistemática da abordagem. Na etapa de EDPS, os modelos de LPrS são especificados. Na etapa de EPEP, o processo de software específico de projeto é definido. Por fim, na etapa de Avaliação de Reusabilidade, os processos de software específicos de projeto definidos são classificados, visando apoiar na sua futura reutilização. Detalhes sobre estas etapas são descritos nas subseções a seguir. Além disso, um cenário de exemplo que ilustra o uso da abordagem é apresentado no Capítulo 5.

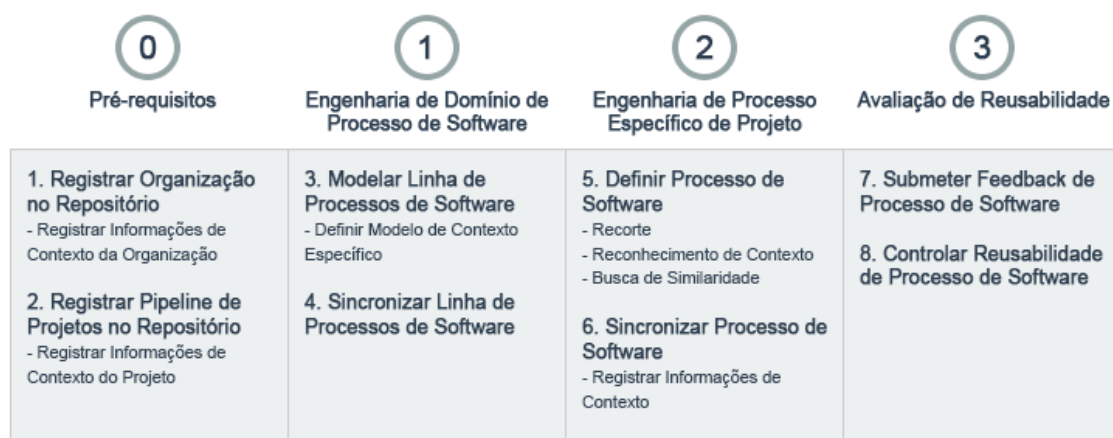


Figura 4.4 – Visão geral da abordagem *Odyssey-ProcessCase*

4.4.1. Pré-requisitos

Na etapa de pré-requisitos, duas atividades são requeridas para a utilização da sistemática da abordagem *Odyssey-ProcessCase*. Estas atividades devem ser realizadas no momento da adoção da abordagem, porém, as informações citadas nesta etapa podem ser gerenciadas ao longo do tempo, visando refletir novos projetos, mudanças organizacionais, etc.

Na atividade **(1) Registrar Organização no Repositório**, o repositório de processos de software utilizado pela organização (*e.g.*, Odyssey Repository) é configurado para criar o registro da organização, informando obrigatoriamente o nome da organização. Por outro lado, uma descrição de contexto da organização pode ser preenchida de forma opcional (*i.e.*, entidade de contexto de organização, Seção 4.2.4). Por fim, os colaboradores da organização devem ser vinculados para obter acesso aos artefatos de processo da organização (*e.g.*, Linhas de Processo de Software, Processos de Software).

Na atividade **(2) Registrar Pipeline de Projetos no Repositório**, projetos em andamento e previstos são registrados no repositório de processos de software utilizado pela organização. Neste cenário, para cada projeto, além de informar obrigatoriamente o nome do projeto, uma descrição de contexto pode ser preenchida de forma opcional (*i.e.*, entidade de contexto de projeto, Seção 4.2.4). Neste ponto, vale destacar que projetos passados da organização, se disponíveis, podem ser adicionados ao repositório, visando alimentar o repositório com experiências de projetos de software realizados no organização. Porém, os processos de software utilizados por estes projetos, só estarão disponíveis para reutilização tiverem sido derivados da LPrS utilizada pela organização.

Essas atividades estão relacionadas à obrigação de registrar a organização e seus projetos no repositório de processos de software utilizado pela organização, porém, suas descrições de contexto são opcionais. Caso estas descrições de contexto sejam fornecidas, o esforço para descrever o contexto de definição de processo de software nas atividades da etapa EPEP é mitigado.

4.4.2. Engenharia de Domínio de Processo de Software (EDPS)

Nesta primeira etapa da abordagem, destaca-se o uso da metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016) para especificação dos modelos de LPrS (*e.g.*, características, contexto, subscopos e componentes) por meio da abordagem *UbiFEX* (FERNANDES *et al.*, 2011) e notação *OdysseyProcess-FEX* (TEIXEIRA *et al.*, 2016). Nessa fase, duas atividades são definidas: I. Modelar Linha de Processos de Software; e II. Sincronizar Linha de Processos de Software.

Na atividade **(3) Modelar Linha de Processos de Software**, o engenheiro de processo de software realiza o processo de EDPS para especificar artefatos reutilizáveis, bem como identificar similaridades e variabilidades por meio dos modelos de LPrS

(e.g., características, contexto, subescopos e componentes), conforme sugerido pela metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016).

Neste ponto, apenas o modelo de características de LPrS é considerado obrigatório na abordagem *Odyssey-ProcessCase*. Porém, o modelo de contexto de LPrS está diretamente ligado à disponibilidade do mecanismo de reconhecimento de contexto, que representa uma das alternativas de suporte para definição de processo de software específico de projeto, descrito na próxima etapa (Seção 4.4.3).

Na construção do modelo de características, o engenheiro de processo de software especifica regras de composição entre os elementos de processo (*i.e.*, parte do modelo de características). Por outro lado, na definição do modelo de contexto, além de especificar as entidades e informações de contexto (Seção 4.2.4), definições de contexto e regras de contexto são elaboradas para representar diretrizes para a definição de processos de software específicos de projeto.

Neste cenário, as regras de composição, conforme descrito na Seção 4.2.1, oferecem suporte na definição de processo de software específico de projeto, restringindo a seleção dos elementos de processo do modelo de características de LPrS por meio de regras de dependência e mútua exclusividade. Essa restrição está relacionada aos mecanismos de recorte e reconhecimento de contexto, descritos na próxima etapa.

Por outro lado, as regras de contexto, conforme descrito na Seção 4.2.2, oferecem suporte na definição de processo de software específico de projeto, direcionando a inclusão ou exclusão dos elementos de processo para apoiar na resolução de variabilidades do modelo de características de LPrS. Este direcionamento está relacionado ao mecanismo de reconhecimento de contexto, descrito na próxima etapa (Seção 4.4.3).

Na atividade de **(4) Sincronizar Linha de Processos de Software**, o engenheiro de processo de software registra os modelos de LPrS no repositório de processos de software utilizado pela organização. Esta atividade tem como objetivo principal centralizar os artefatos de processo de software para futura reutilização.

4.4.3. Engenharia de Processo Específico de Projeto (EPEP)

Na segunda etapa da abordagem, destacam-se os mecanismos de definição de processo de software específico de projeto, utilizados para apoiar na tomada de decisão do gerente de projetos visando à resolução de variabilidades de LPrS. Nesta etapa, duas atividades são definidas: I. Definir Processo de Software; e II. Sincronizar Processo de Software.

Na atividade **(5) Definir Processo de Software**, o gerente de projetos é suportado por meio de três mecanismos de definição de processo de software específico de projeto a partir de artefatos reutilizáveis de LPrS, sendo estes:

1. **Recorte**, focado no suporte durante a seleção manual dos elementos de processo de software, sendo este um mecanismo limitado do ponto de vista do apoio à tomada de decisão para resolução de variabilidades;
2. **Reconhecimento de Contexto**, focado na identificação de situações de contexto esperadas para sugerir a inclusão ou exclusão de elementos de processo visando apoiar na resolução de variabilidades; e
3. **Busca de Similaridade**, focado na identificação de casos similares (*i.e.*, processos de software específicos de projeto), com o objetivo de adotar um processo de software definido em contexto similar como ponto de partida, ou apoiar na resolução de variabilidades ao observar as decisões tomadas em processos de software com caracterização de contexto similar.

Neste ponto, o gerente de projeto pode optar pela utilização independente ou complementar dos mecanismos ofertados.

O mecanismo de busca de similaridade pode ser utilizado de forma independente ou combinado com os outros dois mecanismos disponíveis. Se usado de forma independente, este mecanismo visa a reutilização total de um processo de software específico de projeto. O objetivo da reutilização total é adotar um processo de software definido em contexto similar como ponto de partida para definição de um processo de software específico de projeto.

Por outro lado, o uso do mecanismo de busca de similaridade de forma complementar, a partir de outros mecanismos (recorte ou reconhecimento de contexto), visa a reutilização parcial de processo de software específico de projeto. O objetivo da

reutilização parcial é analisar as decisões tomadas na definição de processos de software em contextos similares para suportar a resolução de variabilidades de LPrS.

Um processo de software específico de projeto é o resultado esperado desta atividade, composto pelos elementos de processo necessários para sua execução, com o objetivo de atender as necessidades do projeto de software para o qual foi definido.

Mecanismo de Recorte: Esse mecanismo corresponde à seleção manual de elementos de processo para atender as necessidades de um projeto de software específico. Tal seleção é realizada pelo gerente de projetos, que fica responsável pela resolução das variabilidades do modelo de características da LPrS. Neste cenário, o gerente de projetos recebe apoio para garantir a consistência do processo de software a ser definido por meio de verificações das resoluções realizadas, levando em consideração as dependências e restrições especificadas pelas regras de composição do modelo de características.

Para isso, inicialmente, os elementos de processo mandatórios do modelo de características da LPrS são pré-selecionados. Durante o recorte, verificações são realizadas para analisar as regras de composição com o objetivo de garantir a consistência na seleção dos elementos de processo, respeitando as restrições especificadas no modelo de características.

Este mecanismo utiliza os conceitos da técnica de SBR (Seção 3.1). Neste ponto, destaca-se o uso de um interpretador de regras baseado na estratégia de encadeamento direto (*i.e.*, orientado a dados) para analisar as regras de composição do modelo de características de LPrS.

Se utilizado de forma independente, este mecanismo oferece suporte limitado para tomada de decisão. Porém, o modelo de características é sua única dependência.

Neste cenário, as funcionalidades do mecanismo de busca de similaridade são disponibilizadas para suportar a resolução de variabilidades de LPrS. Neste ponto, o gerente de projetos busca por processos de software derivados da LPrS definidos em contexto similar, com base nas entidades e informações de contexto apresentadas na Seção 4.2.4, tendo como objetivo analisar as decisões tomadas para solucionar variabilidades específicas. Essa análise ocorre durante o uso do mecanismo de recorte.

Mecanismo de Reconhecimento de Contexto: Este mecanismo corresponde ao apoio à resolução de variabilidades do modelo de características de LPrS, que ocorre por meio de sugestões de inclusão ou exclusão de elementos de processo. Estas sugestões são realizadas ao interpretar as regras de contexto do modelo de contexto de LPrS, responsáveis por relacionar as situações de contexto esperadas (*i.e.*, definições de contexto) e os elementos de processo do modelo de características de LPrS.

No mecanismo de reconhecimento de contexto, o gerente de projetos descreve o contexto de definição de processo de software ao atribuir valores específicos para as informações de contexto do modelo de contexto de LPrS (*e.g.*, informações de contexto apresentadas na Seção 4.2.4). Com base nos valores atribuídos, bem como nas definições de contexto e regras de contexto especificadas na fase de EDPS, este mecanismo identifica ações (*i.e.*, inclusão ou exclusão de elementos de processo) que devem ser realizadas para apoiar a resolução de variabilidades do modelo de características de LPrS. Neste ponto, além dos elementos mandatórios, os elementos de processo sugeridos por meio da interpretação das regras de contexto são pré-selecionados.

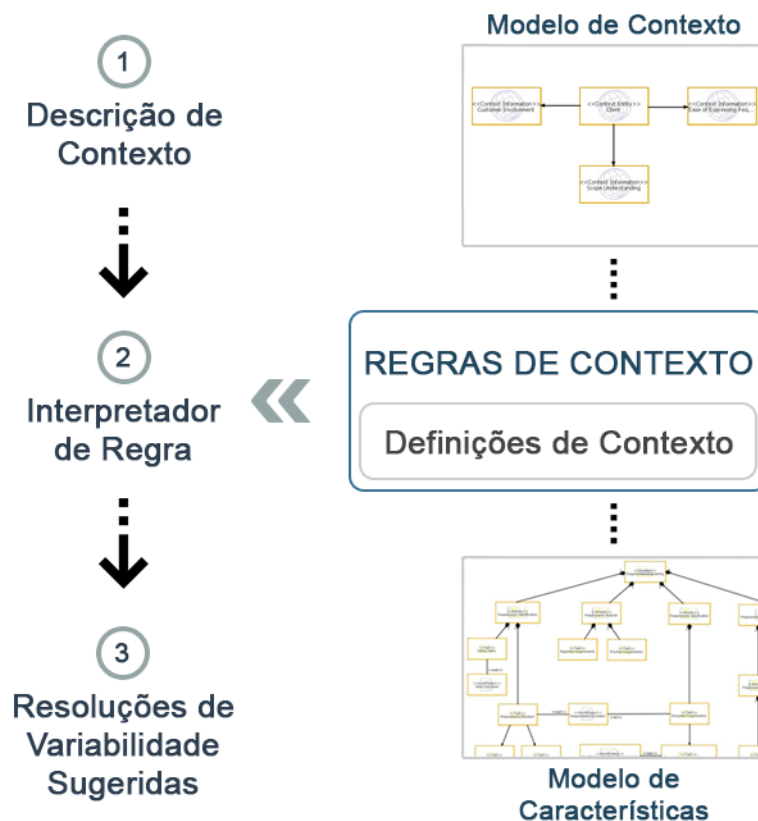


Figura 4.5 – Mecanismo de reconhecimento de contexto

Conforme ilustrado na Figura 4.5, após o gerente de projetos descrever o contexto, as definições de contexto são analisadas com base nos valores fornecidos, avaliando suas expressões para identificar quais definições de contexto estão ativas. Em seguida, as regras de contexto são analisadas, verificando quais regras devem ser executadas (*i.e.*, regras com expressão antecedente verdadeira). Essa etapa de análise é realizada pelo interpretador de regras. Dessa forma, ao interpretar quais regras devem ser executadas com base no contexto de definição de processo de software, o mecanismo de reconhecimento de contexto, por meio das ações relacionadas às regras, sugere um conjunto de elementos de processo para apoiar a resolução de variabilidades.

Este mecanismo utiliza os conceitos da técnica de SBR (Seção 3.1). Neste ponto, destaca-se o uso de um interpretador de regras baseado na estratégia de encadeamento direto (*i.e.*, orientado a dados) para analisar as regras de contexto.

Neste cenário, as funcionalidades do mecanismo de busca de similaridade (descrito a seguir) são disponibilizadas para suportar a resolução de variabilidades de LPrS. Neste ponto, o gerente de projetos busca por processos de software derivados da LPrS definidos em contexto similar, com base nas entidades e informações de contexto apresentadas na Seção 4.2.4, tendo como objetivo analisar as decisões tomadas para solucionar variabilidades específicas, seja para resolução de variabilidades que podem não ter sido contempladas pelo mecanismo de reconhecimento de contexto ou para reforçar as sugestões deste mecanismo. Essa análise ocorre durante o uso do mecanismo de reconhecimento de contexto.

Mecanismo de Busca de Similaridade: Neste mecanismo, o gerente de projetos descreve o contexto de definição do processo de software ao atribuir valores específicos para as informações de contexto apresentadas na Seção 4.2.4. Com base nesta caracterização, uma análise de similaridade é realizada nos processos de software específicos de projeto relacionados à LPrS.

Neste cenário, os processos de software definidos em contexto similar são priorizados de acordo com o nível de similaridade frente ao contexto descrito pelo gerente de projetos, usando a métrica de similaridade “vizinhos mais próximos ponderada”, em inglês, *Weighted Nearest Neighbors* (WANGENHEIM & WANGENHEIM, 2008), apresentada no Capítulo 3.

Segundo WATSON (1999), vizinhos mais próximos destaca-se neste cenário pelo número de aplicações em diferentes áreas. Além disso, há relatos de aplicação bem-sucedida desta métrica para reutilização de processos de software (SANTOS *et al.*, 2008).

O resultado desta análise de similaridade é uma lista de processos de software específicos de projetos definidos em contextos similares. Essa lista é ordenada de forma decrescente com base no nível de similaridade calculado.

Neste ponto, cada processo de software é acompanhado pelo nível de similaridade, suas informações básicas (*e.g.*, organização, projeto, LPrS, nível de aderência ao projeto e observações gerais), modelo de representação do processo de software (*i.e.*, modelo de características) e informações de contexto (*i.e.*, contexto de definição do processo de software). Estes recursos colaboram para compreensão dos processos de software retornados, e, conseqüentemente, para tomada de decisão envolvida na definição de processo de software específico de projeto. Dentre as possibilidades de análise proporcionadas por estes recursos, estão:

- I. Verificar se o processo de software atendeu às necessidades do projeto para o qual foi definido, por meio das informações de nível de aderência e observações gerais, preenchidas na atividade de avaliação de reusabilidade;
- II. Analisar os elementos de processo selecionados para compor o processo de software, bem como as decisões para resolução de variabilidades de LPrS (modelo de representação do processo de software); e
- III. Comparar o contexto de definição de processo de software para analisar as similaridades e divergências entre as informações de contexto descritas e as informações de contexto de cada processo de software retornado.

Este mecanismo utiliza os conceitos da técnica de RBC (Seção 3.2). Neste ponto, as duas estratégias (*i.e.*, resolução de problema e interpretativa) apresentadas no Capítulo 3 são utilizadas. A estratégia de resolução de problemas está relacionada ao conceito de reutilização total (*i.e.*, utilização do mecanismo de forma independente). Por outro lado, a estratégia interpretativa está relacionada ao conceito de reutilização parcial (*i.e.*, utilização do mecanismo de forma complementar).

As entidades e informações de contexto do modelo de contexto proposto por LEITE (2011), apresentadas na Seção 4.2.4, são adotadas para representar o contexto de definição do processo de software, ou seja, atributos que descrevem os casos armazenados no repositório de processos de software. Neste ponto, para cada informação de contexto, foi adicionado sua respectiva relevância para caracterização do projeto de software e definição do processo de software específico de projeto. A relevância de cada informação de contexto foi definida de forma *ad-hoc* (Apêndice C), por meio dos valores: I. Baixa (0.25); II. Moderada (0.75); e III. Alta (1.00). Dessa forma, a métrica de similaridade, *Weighted Nearest Neighbors*, analisa cada informação de contexto (*i.e.*, atributo) e sua respectiva relevância (*i.e.*, peso) durante a análise de similaridade.

Na atividade **(6) Sincronizar Processo de Software**, o gerente de projetos registra o processo de software específico de projeto no repositório de processos de software utilizado pela organização. Neste cenário, uma descrição do contexto de definição de processo de software (Seção 4.2.4) deve ser preenchida de forma obrigatória. Esta atividade tem como objetivo principal centralizar os artefatos de processo de software para futura reutilização.

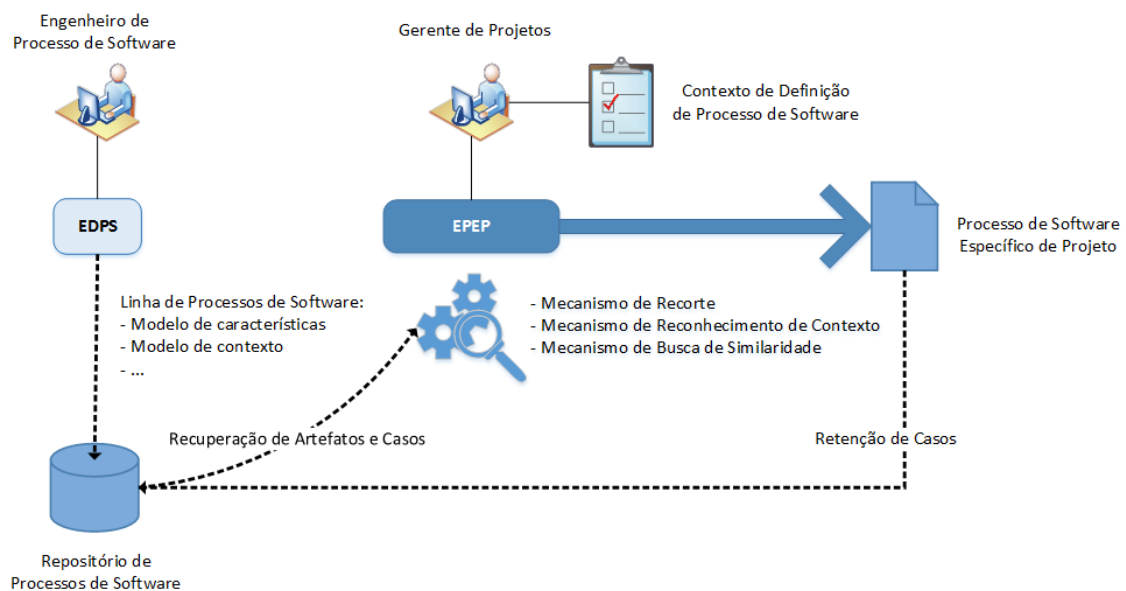


Figura 4.6 – Visão geral da etapa de EPEP da abordagem *Odyssey-ProcessCase*

A Figura 4.6 ilustra o funcionamento da etapa de EPEP na abordagem *Odyssey-ProcessCase*, onde o gerente de projetos realiza a definição do processo de software específico de projeto. Neste ponto, o gerente de projetos utiliza os artefatos definidos

pelo engenheiro de processo de software na fase de EDPS e casos similares (*i.e.*, processos de software específicos de projeto definidos em contexto similar) armazenados no repositório de processos de software utilizado pela organização. Três mecanismos de definição de processo são ofertados para apoiar a tomada de decisão durante a seleção dos elementos de processo necessários para atender aos requisitos do projeto de software (*i.e.*, contexto de definição de processo de software). O resultado é um processo de software específico de projeto, que após ser definido é retido no repositório de processos de software para incrementar o suporte para tomada de decisão ao surgir casos similares no futuro.

Neste cenário, conforme os processos de software específicos de projeto são definidos, estes artefatos são armazenados em um repositório de processos da organização. Consequentemente, o aumento do número de processos (*i.e.*, casos), visando diferentes contextos de definição de processo de software (*i.e.*, problemas), tende a evoluir o suporte à tomada de decisão para resolução de variabilidades de LPrS oferecido pela abordagem (*i.e.*, aprendizado incremental).

4.4.4. Avaliação de Reusabilidade

Na terceira e última etapa da abordagem, destacam-se recursos para apoiar a reutilização futura de processos de software específicos de projeto. Nesta etapa, duas atividades são definidas: I. Submeter *Feedback* de Processo de Software; e II. Controlar Reusabilidade de Processo de Software.

Na atividade **(7) Submeter *Feedback* de Processo de Software**, o gerente de projetos, após o término do projeto de software, classifica o nível de aderência do processo de software em relação as necessidades do projeto, utilizando alternativas pré-definidas (*e.g.*, baixa, moderada, alta), bem como preenche de forma opcional observações sobre o processo de software. Estas informações são influenciadas pelo conhecimento e percepção do gerente de projetos, e são disponibilizadas pelo mecanismo de busca de similaridade para apoiar na tomada de decisão.

Na atividade **(8) Controlar Reusabilidade de Processo de Software**, o engenheiro de processo de software analisa os processos de software específicos de projeto no repositório de processos de software utilizado pela organização, visando controlar a disponibilidade destes artefatos para futura reutilização. Essa atividade possibilita a remoção de ruídos (*e.g.*, processos de software definidos de forma incorreta

ou obsoletos frente à versão atual da LPrS), impactando diretamente a definição de processo de software específico de projeto utilizando o mecanismo de busca de similaridade. A frequência desta atividade deve ser definida pela organização.

4.5. Considerações Finais

Neste capítulo, o contexto de pesquisa desta dissertação de mestrado foi apresentado. Além disso, os requisitos iniciais para definição da abordagem foram descritos. Por fim, foi detalhada a abordagem *Odyssey-ProcessCase*, cujo objetivo principal é apoiar na tomada de decisão para a resolução de variabilidades de LPrS durante a definição de processo de software específico de projeto. A abordagem é composta por uma etapa de pré-requisitos e três etapas principais: I. EDPS, que especifica os modelos de LPrS; II. EPEP, que define o processo de software específico de projeto a partir dos artefatos reutilizáveis originados na EDPS; e III. Avaliação de Reusabilidade, que oferece meios para analisar os processos de software definidos para apoiar a reutilização.

Na sistemática da abordagem, um processo de software é definido para atender as particularidades de projeto de software a partir de artefatos reutilizáveis de LPrS, utilizando os modelos especificados na EDPS. Esta definição pode ser realizada por meio de três mecanismos complementares, que aplicam os conceitos das técnicas de RBC e SBR. Dessa forma, profissionais e organizações de diferentes perfis podem definir processos de software específicos de projeto de acordo com sua realidade.

Um ferramental de suporte integrado foi implementado no escopo desta dissertação para apoiar a sistemática da abordagem. Este ferramental é composto por uma versão estendida da ferramenta Odyssey (ODYSSEY, 2019) e um repositório de processos de software, denominado Odyssey Repository. A ferramenta Odyssey estendida oferece suporte para modelagem de LPrS e definição de processo de software específico de projeto, e a ferramenta Odyssey Repository reúne os artefatos para possibilitar a reutilização de processos de software.

Quando comparada aos estudos identificados pela revisão da literatura (LPrS), apresentados na Seção 2.1, a principal diferença da abordagem *Odyssey-ProcessCase* está relacionada ao suporte ofertado para apoiar a tomada de decisão para resolução de variabilidades de LPrS, que utiliza os conceitos das técnicas de RBC e SBR. O suporte ofertado pelas abordagens analisadas na revisão da literatura ocorre por meio de

mecanismo único de definição de processo de software. Além disso, os resultados da revisão ressaltaram a necessidade de ferramentas com suporte integrado para as atividades das fases de EDPS e EPEP. Na abordagem *Odyssey-ProcessCase*, os mecanismos complementares de definição de processo de software específico de projeto possibilitam explorar os conceitos de reutilização total e parcial de processo de software específico de projeto. Por fim, o ferramental de suporte implementado oferece suporte integrado para as atividades das fases de EDPS (TEIXEIRA, 2011, TEIXEIRA, 2016) e EPEP.

Quando comparada aos estudos identificados pela revisão da literatura (RBC), apresentados na Seção 3.2.2, a abordagem *Odyssey-ProcessCase* se diferencia pelo suporte ofertado para resolução de variabilidades de domínios de processos de software, por meio do conceito de reutilização parcial, e provimento de informações adicionais para o entendimento dos casos similares usados durante a tomada de decisão. Nenhum dos estudos analisados na revisão da literatura adotou o conceito de gerência de variabilidades de processo de software. Este ponto influenciou a implementação do ferramental de suporte da abordagem *Odyssey-ProcessCase*, bem como sua sistemática. Além disso, os estudos identificados na revisão da literatura (RBC) trabalharam apenas com o conceito de reutilização total de processo de software e apresentaram suporte ferramental limitado em relação à tomada de decisão, oferecendo recursos como nível de similaridade (porcentagem) e informações básicas do processo de software ao retornar casos similares, por meio da técnica de RBC. Neste ponto, além dos recursos mencionados, o ferramental de suporte da abordagem *Odyssey-ProcessCase* oferece a possibilidade de visualizar o modelo do processo de software, bem como visualizar/comparar as informações de contexto do processo de software retornado. Estas funcionalidades promovem o conceito de reutilização parcial de processo de software. Mais detalhes sobre o ferramental de suporte são encontrados no Capítulo 5.

CAPÍTULO 5 - IMPLEMENTAÇÃO

5.1. Introdução

No capítulo anterior, a sistemática da abordagem *Odyssey-ProcessCase* foi apresentada em detalhes. O objetivo desta abordagem de LPrS é apoiar a tomada de decisão, durante a definição de processo de software específico de projeto, para resolução de variabilidades de LPrS.

Neste cenário, para ofertar meios para utilização prática da abordagem, um ferramental de suporte foi implementado. Este capítulo descreve o processo de implementação do ferramental elaborado, focado na extensão do ambiente de reutilização de software Odyssey (ODYSSEY, 2019) e desenvolvimento do repositório de processos de software, denominado Odyssey Repository.

Além desta seção introdutória, este capítulo está organizado de forma que os requisitos iniciais para implementação do ferramental são apresentados na Seção 5.2. Na Seção 5.3, uma visão geral dos componentes do ferramental de suporte é detalhada. Na Seção 5.4, um cenário de exemplo é descrito para ilustrar o uso da abordagem por meio do ferramental de suporte implementado. Por fim, as considerações finais são apresentadas na Seção 5.5.

5.2. Requisitos do Ferramental de Suporte

Com base nas observações realizadas ao conduzir as revisões da literatura, apresentadas no Capítulo 2 e 3, requisitos foram identificados para implementação do ferramental de suporte da abordagem *Odyssey-ProcessCase*, sendo estes:

- **Ofertar suporte integrado para atividades de ambas as fases de LPrS:** Requisito identificado a partir da revisão da literatura focada na fase de EPEP, apresentada no Capítulo 2. Conforme observado nos resultados desta revisão, as abordagens identificadas apresentaram baixo nível de integração ferramental para suportar as atividades das fases de EDPS e EPEP;
- **Importação e exportação de artefatos:** O ambiente de reutilização, utilizado para modelagem de LPrS e processo de software, deve ofertar recursos para importação e exportação de modelos. Neste ponto, inicialmente, ficou definido o uso do formato XML como meio para que os artefatos gerados sejam

armazenados e recuperados. O formato XML, por meio do padrão XMI (XML *Metadata Interchange*) proposto pela OMG (*Object Management Group*), é utilizado para troca facilitada de metadados entre as ferramentas de modelagem UML (*Unified Modeling Language*). Embora não exista um padrão consolidado para modelagem de processo de software, o formato XML foi adotado.

5.3. Visão Geral do Ferramental de Suporte

Na Figura 5.1, pode-se observar a visão geral do ferramental de suporte idealizado para a abordagem *Odyssey-ProcessCase* para atender aos requisitos listados na seção anterior.

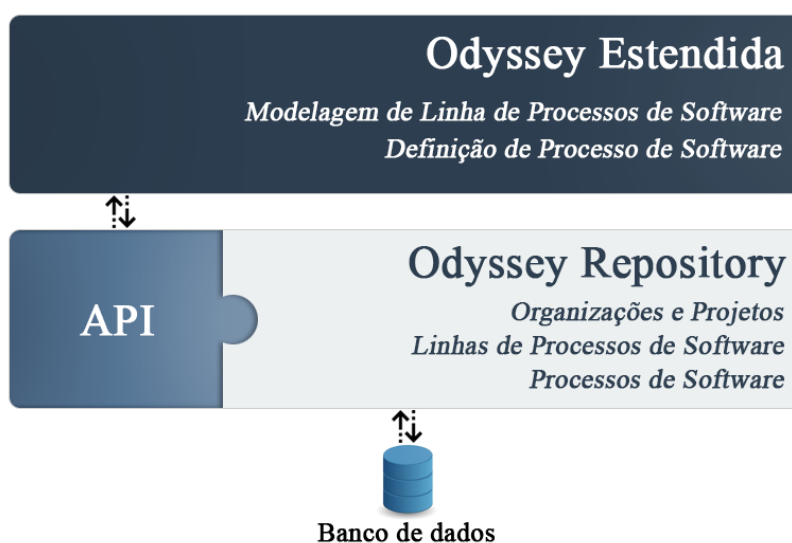


Figura 5.1 – Visão geral do ferramental de suporte

Conforme observado na Figura 5.1, o ferramental de suporte é composto por dois componentes:

- I. Ferramenta Odyssey Estendida (ODYSSEY, 2019): componente que integra as atividades de modelagem de LPrS e definição de processo de software específico de projeto (*i.e.*, EDPS e EPEP); e
- II. Ferramenta Odyssey Repository: componente responsável por centralizar os artefatos de processo de software e controlar o acesso aos mesmos, além de ofertar uma RESTful API para integração com ferramentas de modelagem de LPrS.

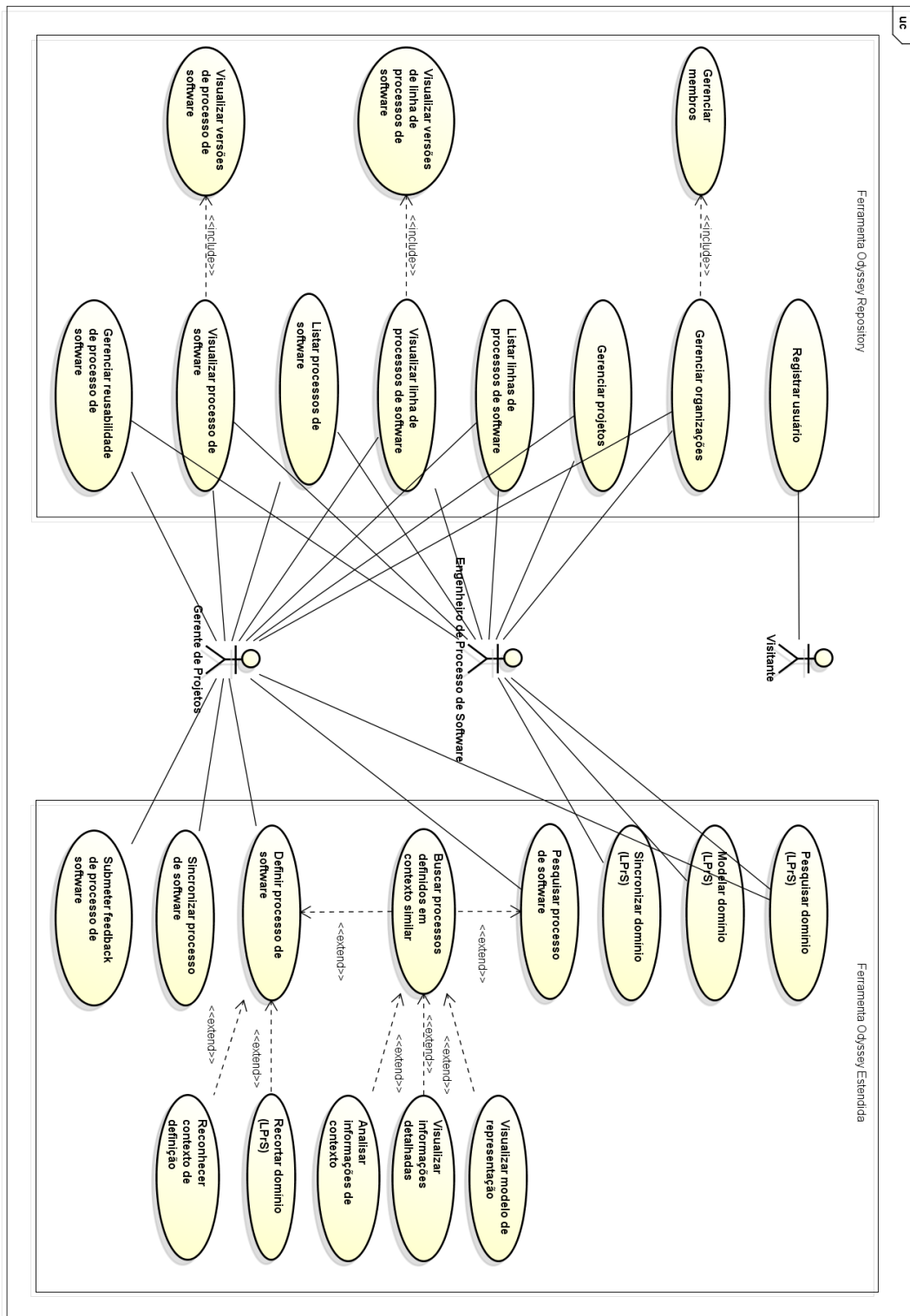


Figura 5.2 – Diagrama de caso de uso do ferramental de suporte

Na Figura 5.2, os casos de uso envolvidos no ferramental de suporte são apresentados em nível de contextualização, sendo estes distribuídos entre as ferramentas Odyssey Estendida e Odyssey Repository. Cada caso de uso apresentado neste diagrama é descrito nas seções específicas de cada ferramenta.

5.3.1. Ferramenta Odyssey Estendida

Embora outras ferramentas tenham sido observadas na revisão da literatura (LPrS), apresentada no Capítulo 2, a ferramenta Odyssey foi adotada nesta dissertação de mestrado como ambiente de reutilização. Neste ponto, foi levado em consideração a disponibilidade do código-fonte para extensão, bem como o suporte ofertado para as atividades da fase de EDPS.

O desenvolvimento da ferramenta Odyssey foi iniciado no ano de 1997, tendo como principal objetivo prover mecanismos, baseados em reutilização, para apoiar no desenvolvimento de software, servindo como um arcabouço onde modelos (*e.g.*, conceituais, arquiteturas de software) são especificados para domínios previamente selecionados (ODYSSEY, 2019). Essa ferramenta oferece uma infraestrutura baseada em modelos de domínio, desenvolvida em Java (ORACLE, 2019), que contempla as atividades de desenvolvimento *para* reutilização e atividades de desenvolvimento *com* reutilização.

Desde o seu surgimento, a ferramenta Odyssey tem sido evoluída. A partir de uma análise focada na identificação das funcionalidades básicas de um ambiente de reutilização, uma versão reestruturada da ferramenta foi disponibilizada como forma de tratar problemas de escalabilidade e desempenho, nomeada de *Odyssey Light* (MURTA *et al.*, 2004).



Figura 5.3 – Infraestrutura da ferramenta Odyssey, adaptado de TEIXEIRA (2011)

A versão *Odyssey Light* disponibiliza funcionalidades por meio do *kernel* e *plug-ins*, conforme observado na Figura 5.3. Entre os *plug-ins* ofertados estão: I. Documentação de componentes (MURTA, 1999); II. Especificação e instanciação de arquiteturas de domínios (XAVIER, 2001); e III. Suporte à engenharia reversa (VERONESE *et al.*, 2002).

5.3.1.1. Extensão da ferramenta Odyssey

Em sua versão atual, a ferramenta Odyssey conta com funcionalidades voltadas para técnica de LPrS. Porém, as funcionalidades de LPrS são focadas na fase de EDPS (TEIXEIRA, 2011, TEIXEIRA 2016). Dessa forma, para estender as funcionalidades da ferramenta Odyssey, essa dissertação explora as funcionalidades de LPrS implementadas pelos estudos anteriores, realizando melhorias e correções para ofertar funcionalidades focadas na fase de EPEP.

A Tabela 5.1 apresenta uma ficha técnica da ferramenta Odyssey Estendida, doravante chamada de ferramenta Odyssey simplesmente, bem como lista as principais bibliotecas adicionadas para extensão das funcionalidades neste trabalho.

Tabela 5.1 – Ficha técnica da ferramenta Odyssey Estendida

Informação	Descrição
Tecnologia	Java SE
Objetivo	Ofertar ambiente de reutilização baseado em modelos de domínio
Estrutura	<i>Kernel</i> e <i>Plug-ins</i>
Importação / Exportação	<i>Serialization</i> e XML
Bibliotecas adicionadas	XStream, Jersey Client, Jackson e Image Viewer

Na Tabela 5.2, os casos de uso com foco na ferramenta Odyssey, apresentados na Figura 5.2, são brevemente descritos. Além disso, destaca-se a ação realizada neste trabalho sobre cada caso de uso, dentro das categorias: I. Implementação, ou seja, implementado neste trabalho; II. Melhoria, funcionalidade existente que exigiu customização; e III. N/A, ou seja, funcionalidade existente implementada por trabalhos anteriores.

Tabela 5.2 – Casos de uso da ferramenta Odyssey Estendida

Ação	Caso de uso	Descrição
Implementação	Pesquisar domínio (LPrS)	Usuário (<i>i.e.</i> , Gerente de projetos ou engenheiro de processo de software) acessa a ferramenta Odyssey e busca por domínios de processo de software (<i>i.e.</i> , LPrS) utilizando suas credenciais do repositório. Uma lista de domínios é apresentada e após o usuário selecionar uma das opções, o artefato é importado para ferramenta Odyssey.
N/A (Existente)	Modelar domínio (LPrS)	Engenheiro de processo de software acessa a ferramenta Odyssey e realiza a modelagem do domínio de processo de software (<i>i.e.</i> , LPrS). Neste ponto, os diferentes modelos de LPrS (<i>e.g.</i> , características, componentes, contexto) podem ser especificados pelo engenheiro de processo de software de acordo com a metodologia <i>OdysseyProcessReuse</i> (TEIXEIRA, 2016). Este caso de uso não fez parte do escopo desta dissertação de mestrado, porém, trata-se de um caso de uso relevante para os demais casos de uso.
Implementação	Sincronizar domínio (LPrS)	Engenheiro de processo de software seleciona um domínio de processo de software (<i>i.e.</i> , LPrS) na ferramenta Odyssey e por meio de suas credenciais do repositório, sincroniza o domínio (<i>i.e.</i> , armazena) no repositório. Neste ponto, múltiplas sincronizações podem ser realizadas, para cada sincronização uma versão do domínio é gerada no repositório.
Implementação	Pesquisar processo de software	Gerente de projetos acessa a ferramenta Odyssey e pesquisa por processos de software específicos de projeto utilizando suas credenciais do repositório. Este caso de uso principal pode listar todos os processos ao qual o usuário tem acesso ou ser realizado pelo fluxo alternativo de busca de processos definidos em contexto similar. Uma lista de processos de software é apresentada e após o gerente de projetos selecionar uma das opções, o processo de software específico de projeto é importado para a ferramenta Odyssey. Neste ponto, caso o domínio de processo de software (<i>i.e.</i> , LPrS) não esteja presente, o domínio também é importado e o processo de software selecionado é atrelado ao domínio na ferramenta Odyssey.
Implementação	Definir processo de software	Gerente de projetos acessa a ferramenta Odyssey e após selecionar um domínio de processo de software, opta pela definição de um processo de software específico de projeto. Este caso de uso principal possui três fluxos alternativos, sendo estes apresentados a seguir.
Melhoria	Recortar domínio (LPrS)	Este caso de uso representa a definição de processo de software por meio do mecanismo de recorte (Capítulo 4). Neste ponto, o gerente de projetos seleciona de forma manual os elementos de processo que irão compor o processo de software específico de projeto. Enquanto seleciona os elementos, o gerente de projetos recebe apoio por meio de verificações de resolução, realizadas para garantir a consistência do processo de software.
Implementação	Reconhecer contexto de definição	Este caso de uso representa a definição de processo de software por meio do mecanismo de reconhecimento de contexto (Capítulo 4). Neste ponto, o gerente de projetos descreve o contexto de definição do processo de software. Em seguida, com base nas regras do modelo de contexto, elementos de processo são sugeridos ao gerentes de projetos para apoiar na tomada de decisão para resolução de variabilidades de LPrS. Este caso de uso não foi finalizado e representa uma limitação.

Ação	Caso de uso	Descrição
Implementação	Buscar processos definidos em contexto similar	Este caso de uso é utilizado nos casos de uso de pesquisar processo de software e definir processo de software, representando o mecanismo de busca de similaridade (Capítulo 4). Neste ponto, o gerente de projetos descreve o contexto de definição do processo de software. Em seguida, com base nesta caracterização, uma busca é realizada no repositório utilizando as credenciais do gerente de projetos para identificar os processos de software definidos em contexto similar, retornando apenas os que apresentarem maior nível de similaridade. Neste cenário, o gerente de projetos analisa as opções retornadas e após selecionar uma das opções, o artefato é importado para a ferramenta Odyssey (<i>i.e.</i> , reutilização total de processo de software).
Implementação	Visualizar modelo de representação	Este caso de uso representa um dos recursos disponíveis na interface de busca de similaridade/pesquisa de processo de software. Neste ponto, o gerente de projetos visualiza o modelo de representação do processo de software específico de projeto para analisar os elementos de processo selecionados e decisões tomadas para resolução de variabilidades específicas de LPrS.
Implementação	Visualizar informações detalhadas	Este caso de uso representa um dos recursos disponíveis na interface de busca de similaridade/pesquisa de processo de software. Neste ponto, o gerente de projetos visualiza as informações detalhadas do processo de software específico de projeto (<i>e.g.</i> , organização, LPrS, informações de <i>feedback</i> , etc).
Implementação	Analisar informações de contexto	Este caso de uso representa um dos recursos disponíveis na interface de busca de similaridade/pesquisa de processo de software. Neste ponto, o gerente de projetos analisa as informações de contexto de cada processo de software específico de projeto, comparando com as informações de contexto descritas (<i>i.e.</i> , caracterização do problema).
Implementação	Sincronizar processo de software	Gerente de projetos acessa a ferramenta Odyssey e após selecionar um processo de software específico de projeto, sincroniza (<i>i.e.</i> , armazena) o artefato no repositório usando suas credenciais. Para isso, o gerente de projetos descreve o contexto de definição do processo de software para caracterizá-lo no repositório visando uma futura busca de similaridade/reutilização.
Implementação	Submeter <i>feedback</i> de processo de software	Gerente de projetos acessa a ferramenta Odyssey e após selecionar um processo de software específico de projeto, submete informações de <i>feedback</i> sobre o processo de software utilizando suas credenciais do repositório de processos. Neste ponto, além de indicar um nível de aderência do processo de software ao projeto após sua execução, o gerente de projetos pode fornecer observações gerais sobre o processo de software no formato de texto.

5.3.2. Ferramenta Odyssey Repository

A ferramenta Odyssey Repository foi implementada para representar um repositório de processos de software, com o objetivo de possibilitar a centralização dos artefatos previstos na abordagem *Odyssey-ProcessCase* (Capítulo 4), bem como controlar o acesso aos artefatos armazenados.

O grupo de Reutilização de Software da COPPE/UFRJ conta com a biblioteca de componentes Brechó (BRECHÓ, 2019). Porém, a ferramenta Brechó, embora tenha sido idealizada para ser genérica, hoje apresenta recursos focados em componentes de software (*i.e.*, código-fonte) e questões relacionadas (*e.g.*, controle de versões, gerenciamento de fornecedores e consumidores de componentes). Nesta dissertação de mestrado, optou-se por implementar um repositório de processos de software específico para abordagem *Odyssey-ProcessCase*, mitigando o esforço de compreensão, customização e implementação de funcionalidades sem impactar os recursos existentes na ferramenta Brechó, desafios estes já explorados na extensão da ferramenta Odyssey. Uma análise pode ser realizada como trabalho futuro para verificar se faz sentido evoluir o repositório de forma independente ou integrar seus recursos na ferramenta Brechó.

Neste cenário, outras ferramentas foram observadas na literatura, porém, estas não foram encontradas em domínio público ou apresentaram uma forte relação com as abordagens para as quais foram implementadas. Além disso, repositórios genéricos foram identificados na literatura da técnica de RBC, porém, nesta dissertação, optou-se por englobar o *framework* jCOLIBRI e desenvolver uma solução customizada, adequada ao cenário de LPrS e conceitos da técnica de RBC.

Dessa forma, a ferramenta Odyssey Repository, focada no cenário de processo de software e LPrS, é composta por dois componentes: I. aplicação web (Figura 5.4), que oferece meios para visualização e gerenciamento dos registros e artefatos; e II. RESTful API, que disponibiliza serviços para integrar o repositório com ferramentas de modelagem de LPrS.

De acordo com BERNSTEIN & DAYAL (1994), um repositório é uma base de dados compartilhada de informações e artefatos usados por uma empresa, sendo considerado uma peça central para integrar ferramentas que alavancam as informações armazenadas. Por outro lado, SHAHZAD *et al.* (2010) destacam que um repositório de modelos de processos oferece um espaço para armazenar, manter e alterar o conhecimento do processo para reutilização futura. Além disso, um repositório permite que as partes interessadas recuperem modelos de processos para diversos fins, como compreender, atualizar, simular e analisar modelos de processos. Neste cenário, a ferramenta Odyssey Repository, em sua versão atual, ainda não atende aos pontos de

simulação e análise de modelos de processos, porém, espera-se que futuras versões contem com estes recursos.

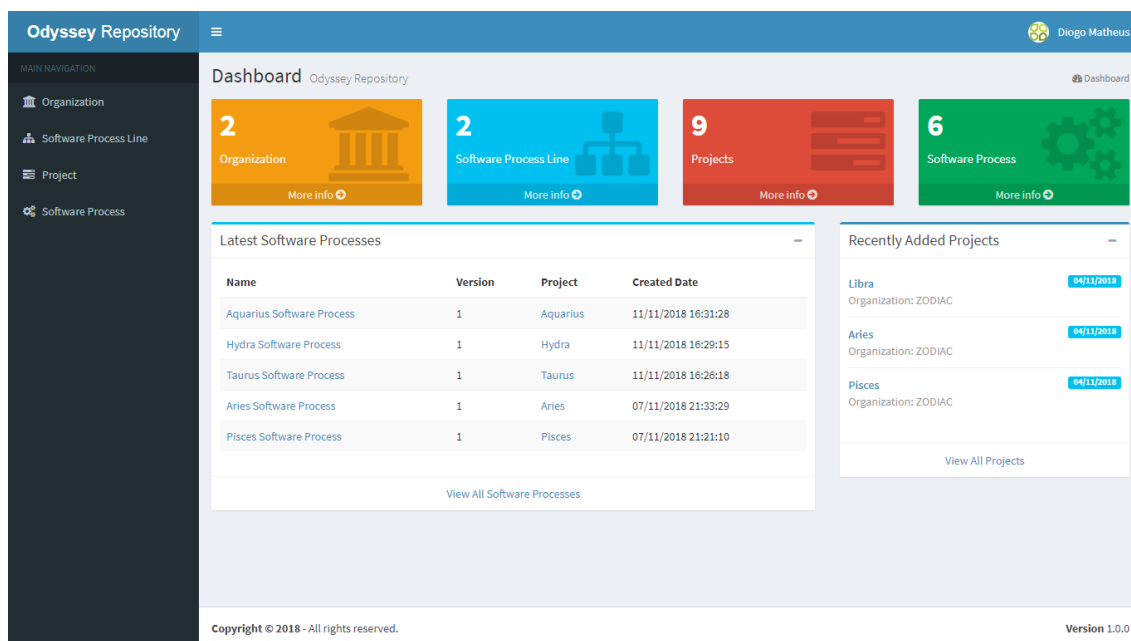


Figura 5.4 – Dashboard da ferramenta Odyssey Repository

Neste cenário, ao adotar a ferramenta Odyssey para modelagem de LPrS, que executa suas funcionalidades na máquina do usuário, o repositório foi idealizado de acordo com a estrutura cliente-servidor. Dessa forma, a ferramenta Odyssey Repository representa o lado do servidor, centralizando e distribuindo informações e artefatos aos clientes (*e.g.*, Odyssey), explorando o trabalho cooperativo entre os diferentes colaboradores de um time ou organização.

Em sua versão inicial, o repositório implementado oferece recursos focados nos conceitos da técnica de LPrS, porém, foram adicionados meios para representação de organizações e projetos, com o objetivo de oferecer futuramente uma plataforma, semelhante às plataformas de repositório de código-fonte existentes: I. GitHub; II. GitLab; e III. Bitbucket. Estas plataformas, além de trabalhar com os recursos de repositório de código-fonte, oferecem meios para criação de projetos públicos e privados, bem como permitem a criação de organizações (*i.e.*, times) para trabalho cooperativo entre um conjunto específico de membros. Por fim, dentre os recursos que colaboram para a evolução do código-fonte nas plataformas citadas, estão: I. submeter propostas de melhorias (*i.e.*, *Pull requests*); e II. reportar problema (*i.e.*, *Issues*).

Embora os recursos indicados façam sentido ao cenário de processos de software e LPrS, não foi possível implementá-los no escopo desta dissertação.

Tabela 5.3 – Ficha técnica da ferramenta Odyssey Repository

Informação	Descrição
Tecnologia	Java EE
Objetivo	Centralizar artefatos de processo de software
Estrutura	Aplicação Web e RESTful API
Bibliotecas / <i>Frameworks</i>	Spring Projects (<i>e.g.</i> , boot, web, data, security), Thymeleaf, jCOLIBRI.
Banco de dados	MySQL

A Tabela 5.3 apresenta uma ficha técnica da ferramenta Odyssey Repository. Na Tabela 5.4, os casos de uso com foco na ferramenta Odyssey Repository, apresentados na Figura 5.2, são brevemente descritos.

Tabela 5.4 – Casos de uso da ferramenta Odyssey Repository

Caso de uso	Descrição
Registrar usuário	Visitante acessa o repositório para realizar cadastro. Neste ponto, o visitante informa seu nome, email e senha. Caso o email informado não esteja sendo usado, o visitante é registrado e automaticamente autenticado.
Gerenciar organizações	Usuário (<i>i.e.</i> , Gerente de projetos ou engenheiro de processo de software) acessa o repositório para gerenciar organizações. Neste ponto, o usuário pode cadastrar uma organização nova ou editar uma organização da qual faz parte.
Gerenciar membros	Usuário acessa o repositório para gerenciar os membros de uma determinada organização, podendo adicionar ou remover colaboradores. Neste ponto, o usuário só tem permissão de gerenciar os membros de organização da qual faz parte.
Gerenciar projetos	Usuário acessa o repositório para gerenciar os projetos aos quais tem acesso por meio de suas organizações. Neste ponto, o usuário pode cadastrar projetos novos ou editar as informações dos projetos que tem acesso.
Listar linhas de processo de software	Usuário acessa o repositório para visualizar a listagem de linhas de processos de software sincronizadas (<i>i.e.</i> , armazenadas). Neste ponto, o usuário só visualiza os artefatos aos quais tem acesso por meio de suas organizações.
Visualizar linha de processos de software	Usuário acessa o repositório para visualizar informações de uma linha de processos de software. Neste ponto, o acesso do usuário ao artefato é verificado, se válido, o usuário visualiza as informações do artefato.
Visualizar versões de linha de processos de software	Usuário acessa o repositório para visualizar as diferentes versões de uma linha de processos de software. Neste ponto, o acesso do usuário ao artefato é verificado, se válido, o usuário visualiza as diferentes versões, bem como o autor da versão e data de criação.
Listar processos de software	Usuário acessa o repositório para visualizar a listagem de processos de software específicos de projeto sincronizados (<i>i.e.</i> , armazenados). Neste ponto, o usuário só visualiza os artefatos aos quais tem acesso por meio de suas organizações.
Visualizar processo de software	Usuário acessa o repositório para visualizar as informações de um processo de software específico de projeto. Neste ponto, o acesso do usuário ao artefato é verificado, se válido, o usuário visualiza as informações do artefato.
Visualizar versões de processo de software	Usuário acessa o repositório para visualizar as diferentes versões de um processo de software específico de projeto. Neste ponto, o acesso do usuário ao artefato é verificado, se válido, o usuário visualiza as diferentes versões, bem como o autor da versão e data de criação.
Gerenciar reusabilidade de processo de software	Usuário acessa o repositório para gerenciar a reusabilidade de processos de software específicos de projeto. Neste ponto, o acesso do usuário ao artefato é verificado, se válido, o usuário indica se o artefato é reutilizável.

5.3.2.1. RESTful API

Com base nos princípios da arquitetura *Representational State Transfer* (REST), ou seja, Transferência de Estado Representacional, uma RESTful API foi implementada como parte da ferramenta Odyssey Repository. Esta API foi desenvolvida em Java (ORACLE, 2019), utilizando o *framework* Spring, Swagger e JSON Web Token (JWT).

Neste cenário, o principal objetivo da API é fornecer meios para possibilitar a integração da ferramenta Odyssey Repository com diferentes ferramentas de modelagem de LPrS (*e.g.*, Odyssey). Para atingir este objetivo, pensando também nos casos de uso da ferramenta Odyssey que exigem interações específicas focadas em registros e artefatos, um conjunto de serviços foi implementado, 38 no total, sendo estes agrupados de acordo com os itens apresentados na Figura 5.5. O formato de transferência pode ser customizado, optando entre XML ou JSON.

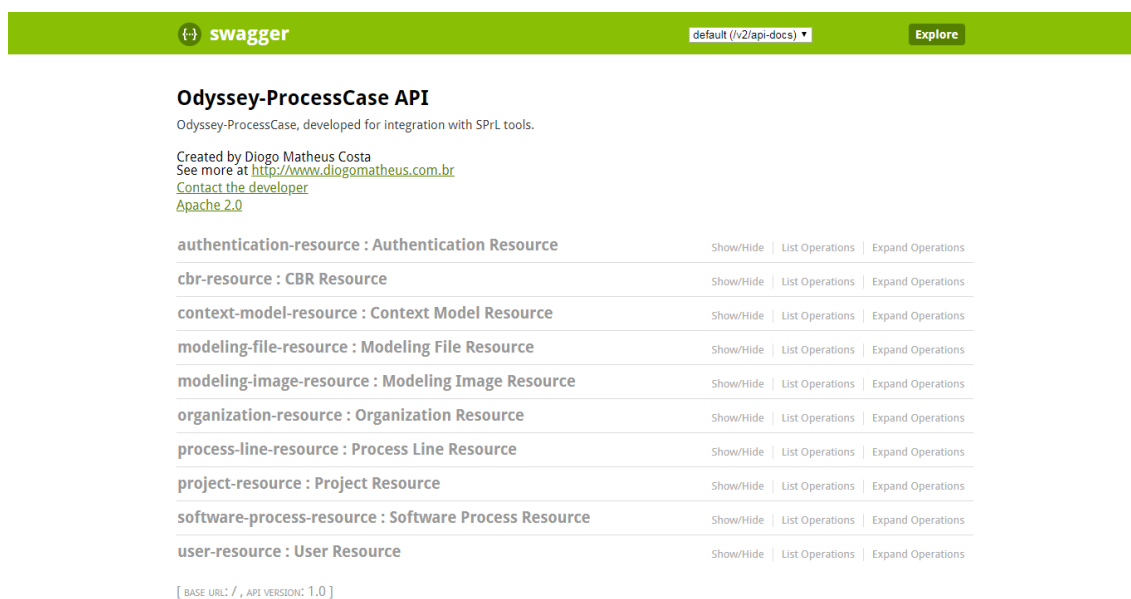


Figura 5.5 – Interface pelo Swagger UI da RESTful API (Odyssey Repository)

A Figura 5.5 apresenta uma interface gerada dinamicamente (Swagger UI) com base na especificação da API, definida por meio de *annotations*, ou seja, anotações em nível de código. Esta interface, além de possibilitar a compreensão dos serviços ofertados, permite que testes funcionais sejam realizados.

5.3.2.2. *Framework* jCOLIBRI

Com o objetivo de reduzir o esforço de implementação da ferramenta Odyssey Repository, uma pesquisa *ad-hoc* foi conduzida para identificar componentes reutilizáveis focados nos conceitos da técnica de RBC. Este componente tem papel importante no ferramental, pois representa o meio para realização da busca de similaridade, utilizada para a localização de processos de software definidos em contexto similar conforme previsto na abordagem *Odyssey-ProcessCase*.

Neste ponto, diferentes *frameworks* foram observados na literatura (ATANASSOV & ANTONOV, 2012). Porém, dois destes destacaram-se, sendo estes: I. myCBR; e II. jCOLIBRI, ambos desenvolvidos em Java (ORACLE, 2019). Para seleção do *framework* RBC, ficou definido como critério os recursos ofertados para implementar aplicações customizadas. Este ponto é relevante uma vez que o objetivo da ferramenta Odyssey Repository é se adequar ao cenário de LPrS.

No estudo de comparação realizado por ATANASSOV & ANTONOV (2012), focado nas ferramentas jCOLIBRI e myCBR, os autores destacam que o *framework* myCBR é recomendado para o desenvolvimento de aplicações de baixa complexidade e pequeno número de casos armazenados. Por outro lado, o *framework* jCOLIBRI é indicado para o desenvolvimento de aplicações complexas, contando com suporte para todo o ciclo da técnica de RBC. Neste ponto, além de ressaltar as características de caixa-branca do código do *framework* jCOLIBRI, os autores destacam dois níveis de uso dos módulos, um focado na implementação abstrata em alto nível, sem exigir conhecimentos de programação, e outro focado em módulos de baixo nível, focado em aplicações customizadas que exigem experiência no desenvolvimento de software.

Com base nessa análise, o *framework* jCOLIBRI foi adotado, explorando sua abertura para utilizar apenas os recursos necessários para a etapa de recuperação de casos da técnica de RBC. Neste ponto, RECIO-GARCÍA *et al.* (2014) em seu estudo apresentam uma visão interessante do *framework* jCOLIBRI, detalhando as diferentes formas que o *framework* pode ser utilizado para criar e customizar aplicações.

5.3.2.3. Modelo de Entidade Relacionamento (MER)

Para armazenar as informações e artefatos, um Modelo de Entidade Relacionamento (MER) foi idealizado para o banco de dados da ferramenta Odyssey Repository, sendo este apresentado na Figura 5.6.

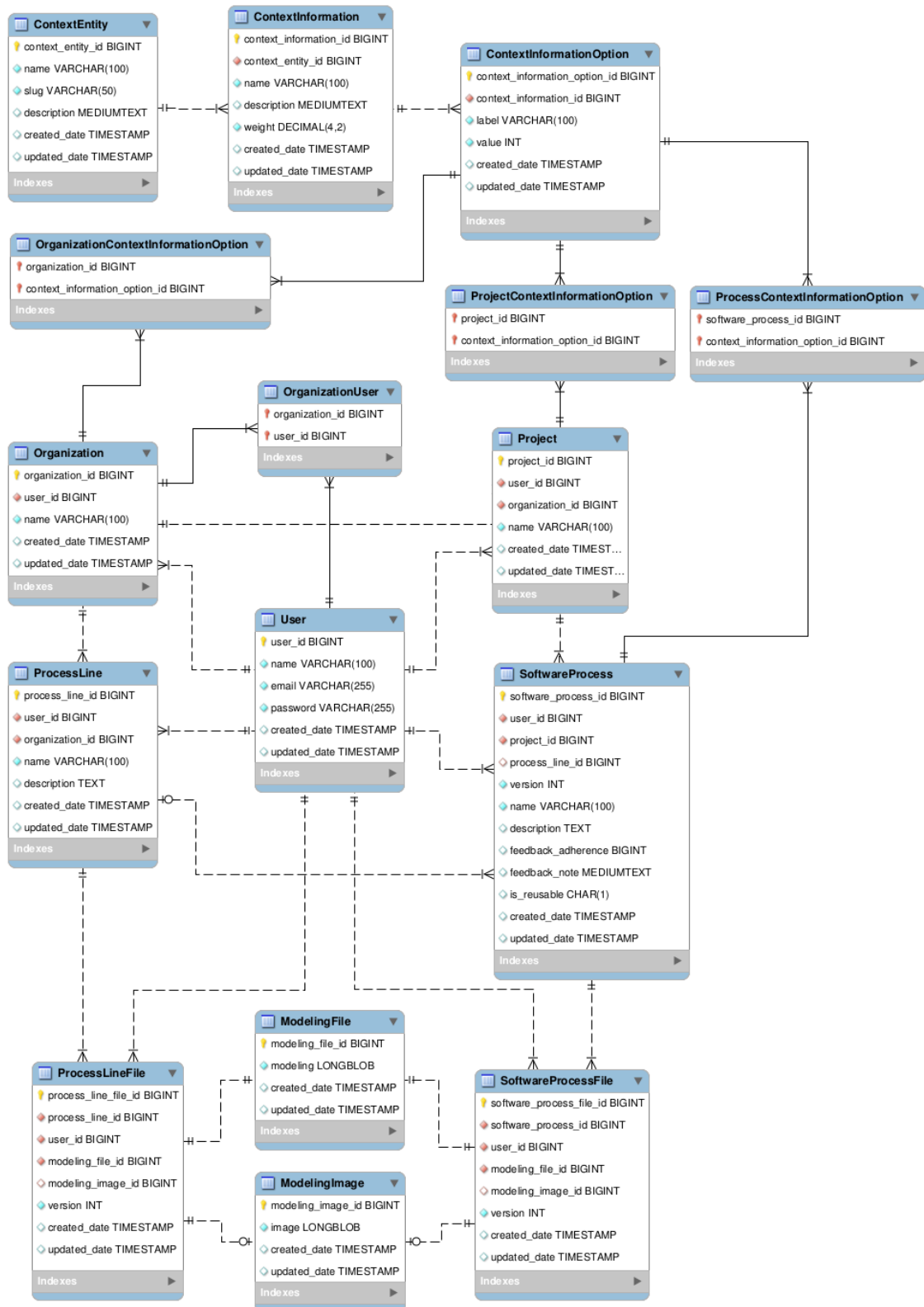


Figura 5.6 – Modelo de Entidade Relacionamento (Odyssey Repository)

Para especificar este modelo, alguns pontos foram mapeados para simplificar o desenvolvimento da ferramenta, bem como possibilitar melhorias e customizações futuras. Dentre estes pontos, destacam-se as entidades envolvidas para armazenar as

informações de contexto (*i.e.*, informações utilizadas para descrever os casos da técnica de RBC), *ContextEntity*, *ContextInformation* e *ContextInformationOption*, que facilitam a customização das entidades e informações de contexto por meio do banco de dados. Por outro lado, para evitar preocupações iniciais relacionadas ao armazenamento de arquivos no servidor, duas tabelas foram idealizadas para armazenar os modelos de LPrS (*i.e.*, domínio serializado exportado pela ferramenta de modelagem) e ilustrações que representam estes modelos (*i.e.*, figuras), *ModelingFile* e *ModelingImage*. Este ponto representa uma limitação da ferramenta Odyssey Repository, sendo ideal implementar uma melhoria neste cenário para retirar a responsabilidade de armazenar artefatos do banco de dados.

A ferramenta Odyssey Repository oferece uma infraestrutura para possibilitar a reutilização de artefatos de processo de software, adequada ao cenário de LPrS. Embora o foco deste trabalho tenha sido a integração deste repositório junto a ferramenta Odyssey (ODYSSEY, 2019), os recursos ofertados foram planejados de forma genérica, visando integrar o repositório com diferentes ferramentas de modelagem de LPrS.

5.4. Cenário de Exemplo

Para facilitar a compreensão dos detalhes da abordagem *Odyssey-ProcessCase* e seu ferramental de suporte, esta seção apresenta um cenário de exemplo que acompanha uma organização produtora de software fictícia, chamada XPTO, adotando a sistemática da abordagem por meio de artefatos e caracterizações elaboradas pelo pesquisador, para melhorar a qualidade de seus processos e, por consequência, produtos de software.

Neste cenário, para adotar as práticas da abordagem *Odyssey-ProcessCase*, o primeiro passo diz respeito às atividades da etapa de pré-requisitos (Seção 4.4.1). Neste ponto, a organização XPTO é registrada no repositório de processos Odyssey Repository (Figura 5.7). Além disso, os colaboradores da XPTO são inseridos, assim como o *pipeline* de projetos da organização (Figura 5.8).

Odyssey Repository

Diogo Matheus

MAIN NAVIGATION

- Organization
- Software Process Line
- Project
- Software Process

Create organization

Dashboard > Organization > Create

Organization information

Organization name

Back Send

Organization context

Business Objective:

Select an option

Maturity Level:

Select an option

Organizational Culture:

Select an option

Structure Formality:

Select an option

Copyright © 2018 - All rights reserved. Version 1.0.0

Figura 5.7 – Interface de cadastro de organização (Odyssey Repository)

No cadastro da organização (Figura 5.7) e de seus projetos de software (Figura 5.8), o usuário pode preencher, de forma opcional, as informações de contexto relacionadas com suas respectivas entidades de contexto (Seção 4.2.4). Essas informações de contexto são utilizadas no futuro apoio para a definição de processo de software específico de projeto, bem como na sincronização do processo de software com o repositório por meio da ferramenta Odyssey.

Odyssey Repository

Diogo Matheus

MAIN NAVIGATION

- Organization
- Software Process Line
- Project
- Software Process

Create project

Dashboard > Project > Create

Project information

Organization:

XPTO

Orion

Back Send

Project context

Product Complexity:

High

Project Criticality:

Very Low

Project Size:

Small

Scope Completeness:

High

Project Communication:

Low

Copyright © 2018 - All rights reserved. Version 1.0.0

Figura 5.8 – Interface de cadastro de projeto (Odyssey Repository)

Neste ponto, vale destacar que o *pipeline* de projetos pode englobar não só os projetos em andamento e previstos na organização XPTO, mas também projetos passados, executados pela organização. Caso projetos passados sejam cadastrados no repositório de processo, seus respectivos processos de software devem ser sincronizados por meio da ferramenta Odyssey.

Em seguida, na segunda etapa da abordagem, EDPS (Seção 4.4.2), o engenheiro de processo de software da organização XPTO realiza a modelagem de LPrS por meio da ferramenta Odyssey. Neste ponto, o ambiente de modelagem de domínio da ferramenta, apresentado na Figura 5.9, é utilizado com base nas notações *OdysseyProcess-FEX* e *UbiFEX*, para especificar os diferentes modelos de LPrS (e.g., características, componentes, contexto), conforme previsto na metodologia *OdysseyProcessReuse* (TEIXEIRA, 2016), onde mais detalhes sobre esta etapa podem ser encontrados.

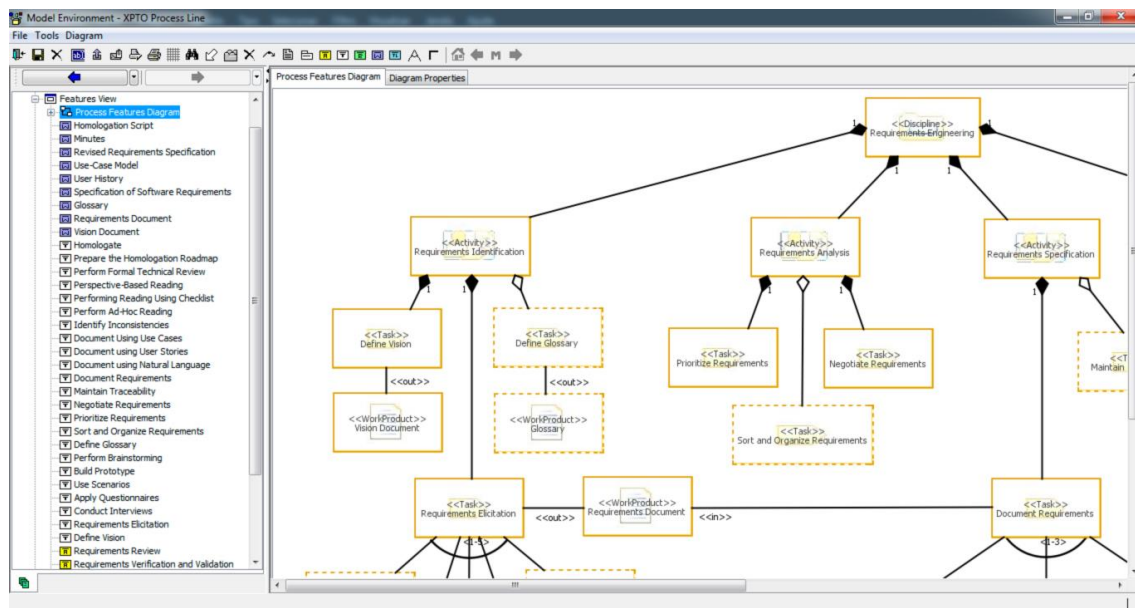


Figura 5.9 – Interface de modelagem de domínio de processo (Odyssey)

No término da modelagem, o engenheiro de processo de software sincroniza o domínio de processo definido com o repositório (Odyssey Repository) por meio da ferramenta Odyssey. Neste ponto, uma organização é selecionada, neste caso XPTO, conforme observado na Figura 5.10. Múltiplas sincronizações podem ser realizadas. Para cada sincronização, uma versão é gerada no repositório de processos.

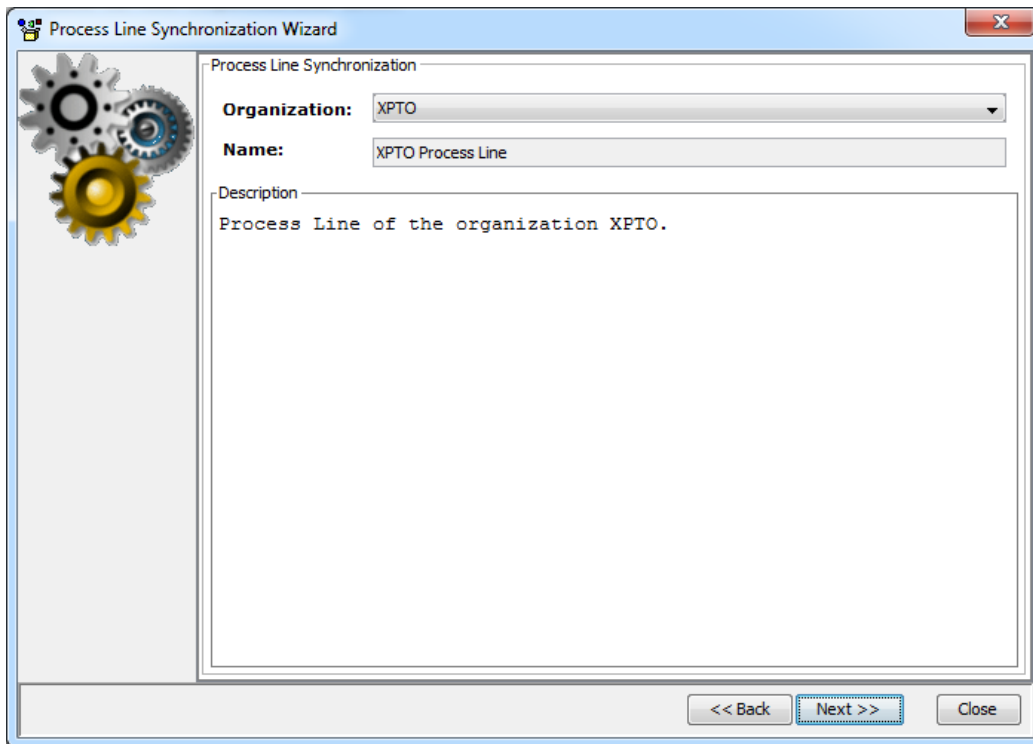


Figura 5.10 – Interface de sincronização de LPrS (Odyssey)

O resultado desta sincronização reflete no repositório de processos, conforme observado na Figura 5.11, onde se pode observar as informações da LPrS (e.g., usuário criador, histórico de versões, etc).

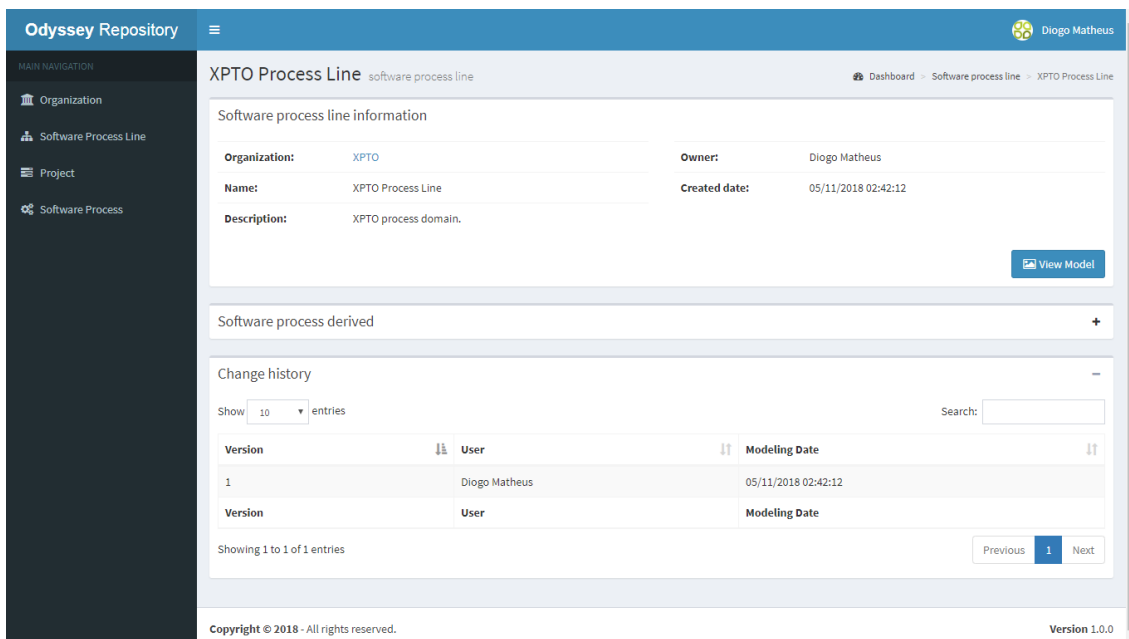


Figura 5.11 – Interface de visualização de LPrS (Odyssey Repository)

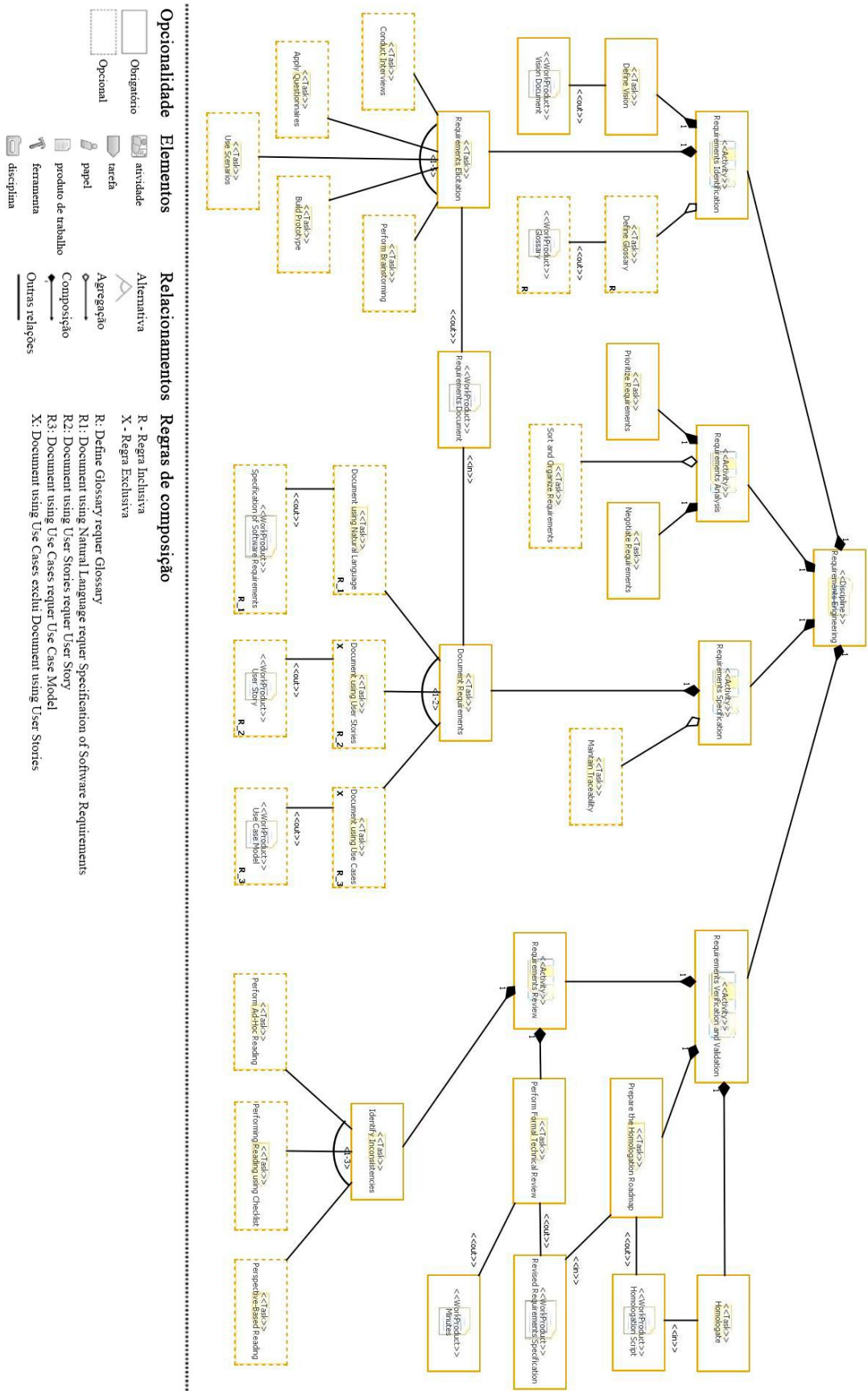


Figura 5.12 – Modelo de características da organização XPTO

A Figura 5.12 apresenta o modelo de características definido para organização XPTO, nomeada de XPTO Process Line, focado na disciplina de Engenharia de Requisitos para exemplificar este cenário de utilização da abordagem.

Com base na LPrS definida e sincronizada, ao surgir um projeto de software na organização XPTO, por exemplo, projeto Orion, que é caracterizado na Figura 5.13, o gerente de projetos conta com o suporte dos três mecanismos para realizar a definição do processo de software específico de projeto (Seção 4.4.3): I. Recorte; II. Reconhecimento de Contexto; e III. Busca de Similaridade. Neste ponto, este cenário de exemplo irá se concentrar na descrição dos mecanismos de recorte e busca de similaridade, pois o mecanismo de reconhecimento de contexto ainda não foi finalizado.

Cliente		Organização	
Envolvimento do Cliente	Alto	Objetivo de Negócio	Conservador
Entendimento do Escopo	Total	Nível de Maturidade	Em processo de amadurecimento
Facilidade em Expressar Requisitos	Alta	Cultura Organizacional	Tradicional
		Formalidade da Estrutura	Formal
Produto		Equipe	
Classe de Software Organizacional	Comum	Compartilhamento de Conhecimento	Baixo
Integração entre Componentes de Projeto	Baixa	Tipo de Gerente de Projetos	Líder
		Distribuição Geográfica	Não Distribuído
Projeto		Entendimento do Escopo	Total
Complexidade do Produto	Alta	Tamanho da Equipe	Muito Pequena
Criticidade do Projeto	Muito Baixa	Experiência no Processo de Desenvolvimento	Moderada
Tamanho do Projeto	Pequeno	Experiência Técnica	Alta
Completeness do Escopo	Alta	Experiência no Domínio	Alta
Comunicação do Projeto	Baixa	Experiência do Gerente de Projetos	Iniciante
Nível de Risco	Positivamente Alto		
Originalidade do Projeto	Comum	Processo	
Duração do Projeto	Curto Prazo	Flexibilidade de Adaptação do Processo Padrão	Moderada
Exigência Contratual	Simples	Ferramenta(s) de Apoio	Semi-Automatizado
Custo do Projeto	Baixo		
Esforço do Projeto	Baixo		

Figura 5.13 – Caracterização do projeto de software Orion

No mecanismo de recorte, o gerente de projetos utiliza sua experiência para definir um processo de software específico de projeto. Neste mecanismo, o gerente de projetos é apoiado durante o recorte manual do modelo de características por meio de alertas de resolução e restrição (Figura 5.14). Neste ponto, o gerente de projetos seleciona os elementos de processo com base na lista apresentada na interface de recorte, onde os elementos de processo mandatórios são pré-selecionados.

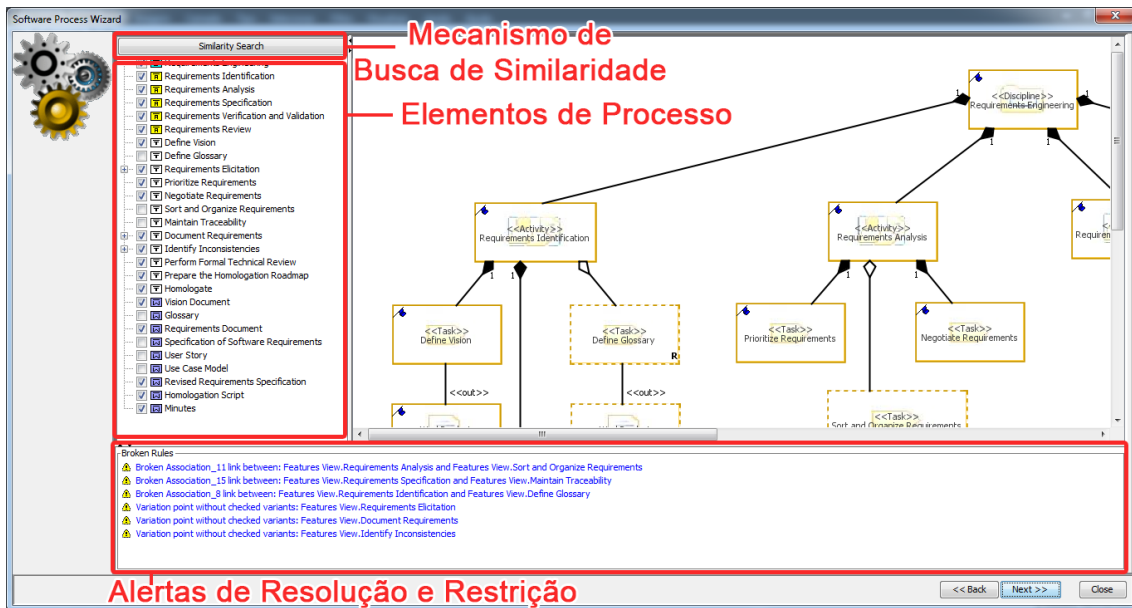


Figura 5.14 – Interface do mecanismo de recorte (Odyssey)

No mecanismo de busca de similaridade, o gerente de projetos descreve o contexto de definição do processo de software (Figura 5.15), atribuindo valores específicos para as informações de contexto adotadas pela abordagem *Odyssey-ProcessCase* (Seção 4.2.4).

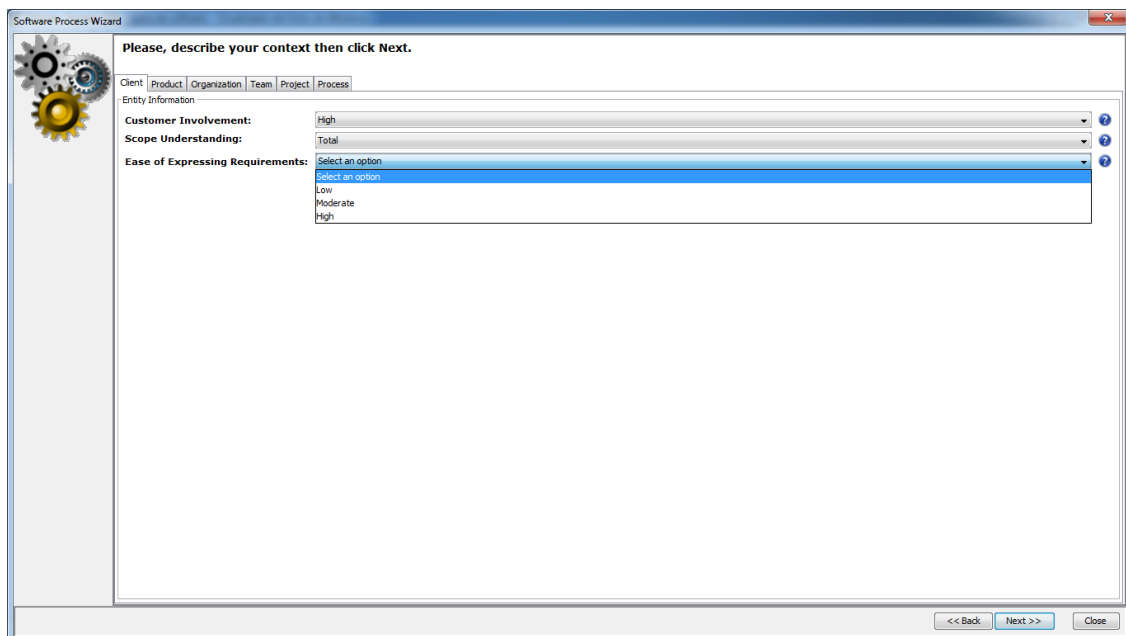


Figura 5.15 – Interface do preenchimento de contexto (Odyssey)

Com base nesta caracterização, os processos de software relacionados à LPrS, armazenados no repositório, são priorizados e retornados de acordo com seu nível de similaridade (Figura 5.16). Neste cenário, o gerente de projetos possui informações

sobre cada processo de software retornado, além disso, o mesmo pode optar pela visualização do modelo de representação do processo de software, bem como pode optar por comparar as informações de contexto de cada processo de software com as informações descritas na ferramenta Odyssey (*i.e.*, caracterização do problema), onde conta com um sumário de comparação, bem como comparações agrupadas por entidade de contexto.



Figura 5.16 – Interfaces do mecanismo de busca de similaridade (Odyssey)

Neste ponto, é importante ressaltar que o mecanismo de busca de similaridade tem como premissa a existência de processos de software no repositório. Além disso, o mecanismo de busca de similaridade está disponível nos demais mecanismos (Figura 5.17), visando complementar o suporte para tomada de decisão e resolução das variabilidades de LPrS, ao observar as decisões tomadas em projetos definidos em contextos similares.

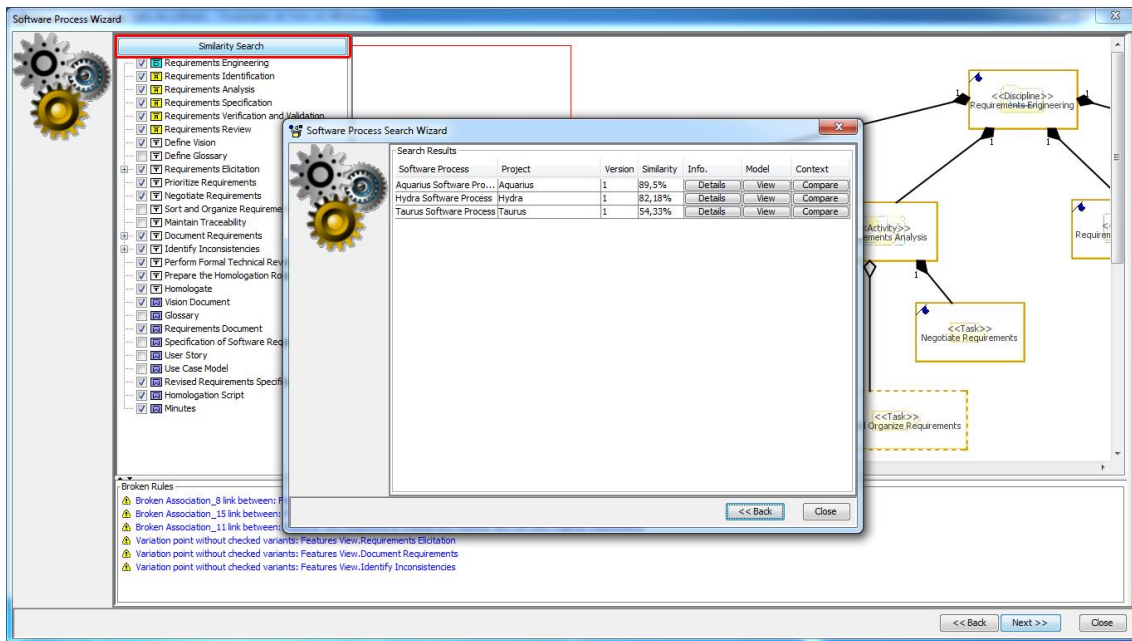


Figura 5.17 – Interface do mecanismo de busca de similaridade durante o recorte (Odyssey)

Independente do mecanismo utilizado pelo gerente de projetos para definição do processo de software do projeto Orion, o ambiente de modelagem da ferramenta Odyssey pode ser utilizado para adaptar o processo de software específico de projeto após sua definição inicial, resultado do uso direto dos mecanismos. Neste ponto, o gerente de projetos pode incluir ou excluir elementos de processo para contemplar necessidades específicas que podem não ter sido atendidas pelo domínio de processo de software (*i.e.*, LPrS).

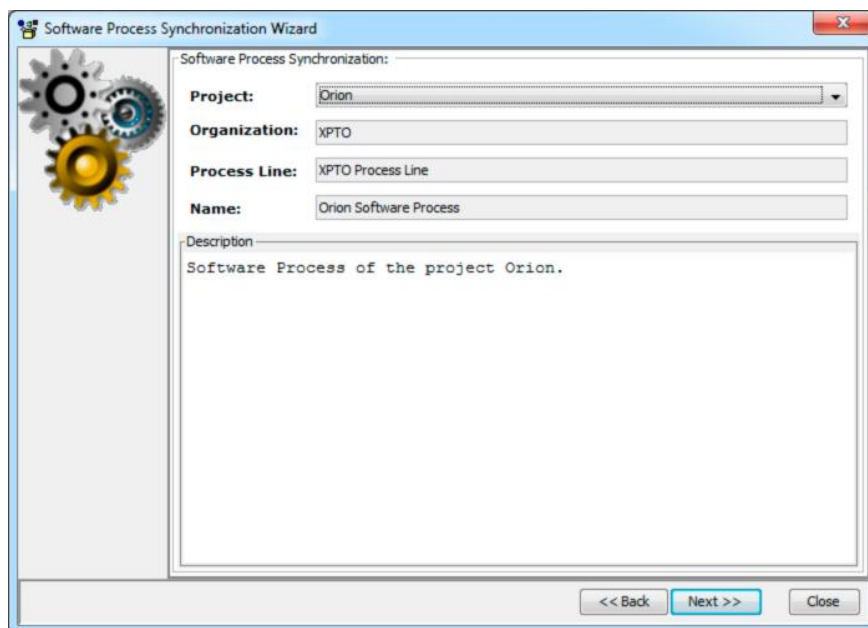
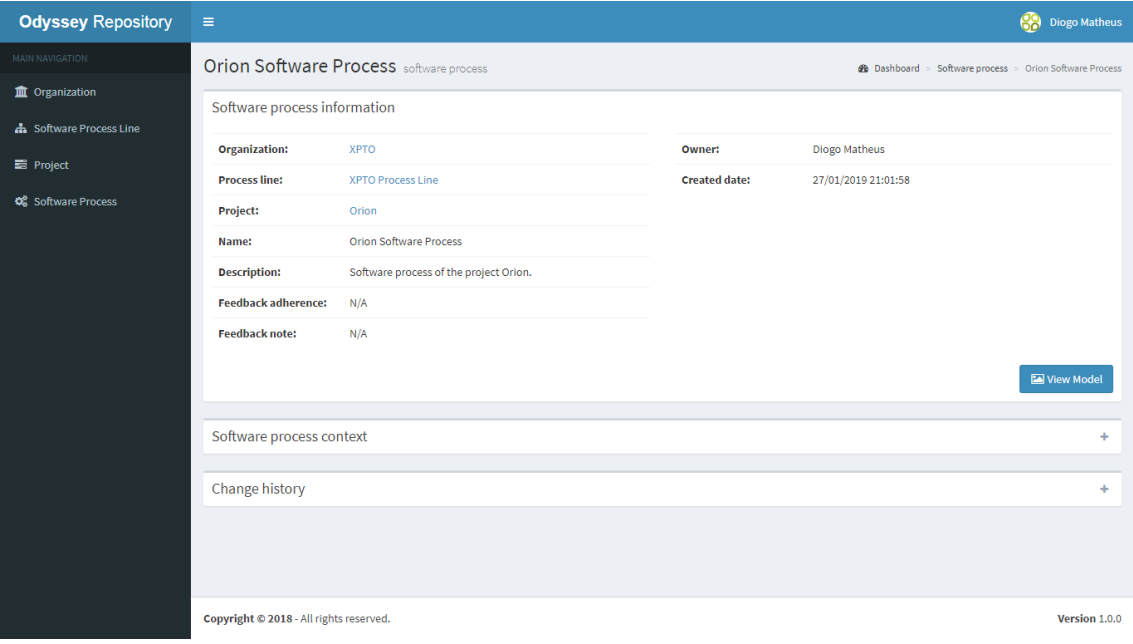


Figura 5.18 – Interface de sincronização de processo de software (Odyssey)

Após definir o processo de software específico de projeto, o gerente de projetos sincroniza o artefato com o repositório de processos Odyssey Repository, conforme observado na Figura 5.18. Neste ponto, o gerente seleciona o projeto, Orion neste caso, bem como preenche as informações de contexto de definição do processo de software (Seção 4.2.4), recebendo apoio (*i.e.*, preenchimento automático) por meio da ferramenta, caso tenha preenchido as informações utilizando um dos mecanismos de definição de processo de software. Múltiplas sincronizações podem ser realizadas. Para cada sincronização, uma versão é gerada no repositório de processos.

O resultado desta sincronização reflete no repositório de processos, conforme observado na Figura 5.19, onde se pode observar as informações do processo de software específico de projeto (*e.g.*, usuário criador, informações de contexto, histórico de versões, etc).



The screenshot displays the Odyssey Repository interface for the Orion Software Process. The main navigation menu on the left includes Organization, Software Process Line, Project, and Software Process. The main content area shows the Orion Software Process details, including the following information:

Field	Value	Field	Value
Organization:	XPTO	Owner:	Diogo Matheus
Process line:	XPTO Process Line	Created date:	27/01/2019 21:01:58
Project:	Orion		
Name:	Orion Software Process		
Description:	Software process of the project Orion.		
Feedback adherence:	N/A		
Feedback note:	N/A		

Below the information table, there are sections for Software process context and Change history, both with expandable arrows. A 'View Model' button is located at the bottom right of the information table. The footer of the interface includes 'Copyright © 2018 - All rights reserved.' and 'Version 1.0.0'.

Figura 5.19 – Interface de visualização de processo de software (Odyssey Repository)

No término do projeto Orion, o gerente de projetos pode fornecer informações sobre o processo de software por meio da interface de submissão de *feedback* de processo de software (Figura 5.20). Essas informações (*e.g.*, nível de aderência, observações) são disponibilizadas pela ferramenta Odyssey para apoiar a reutilização do artefato ao recuperar processos de software (*i.e.*, informações do processo de software do mecanismo de busca de similaridade).

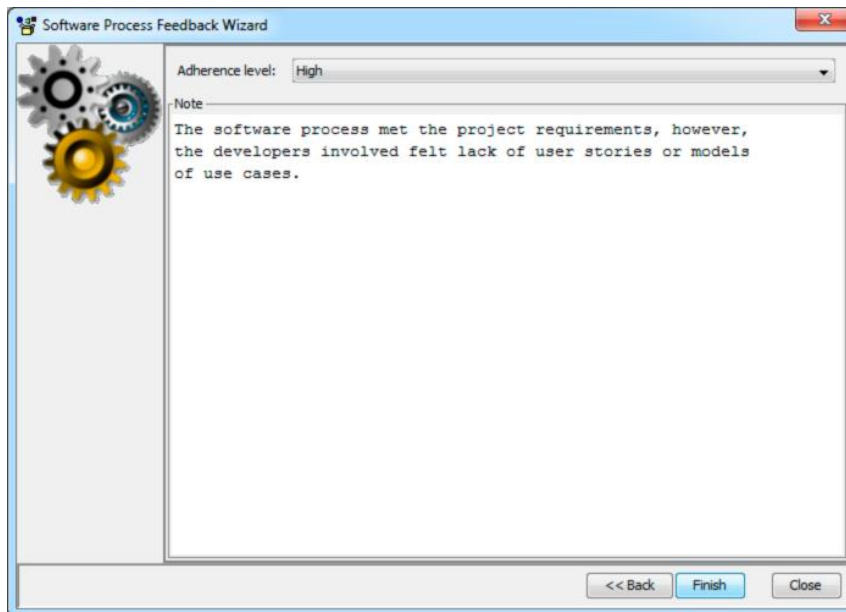


Figura 5.20 – Interface de submissão de *feedback* (Odyssey)

Por fim, o engenheiro de processo de software da organização XPTO pode gerenciar de forma contínua a reusabilidade dos processos de software definidos na organização por meio da ferramenta Odyssey Repository, conforme observado na Figura 5.21. Esse recurso visa ignorar (*i.e.*, não sugerir para reutilização) determinados processos de software definidos de forma incorreta ou obsoletos, seguindo critérios definidos pela própria organização. Esta dissertação não tem como objetivo definir critérios para análise de definição correta ou incorreta de processo de software.

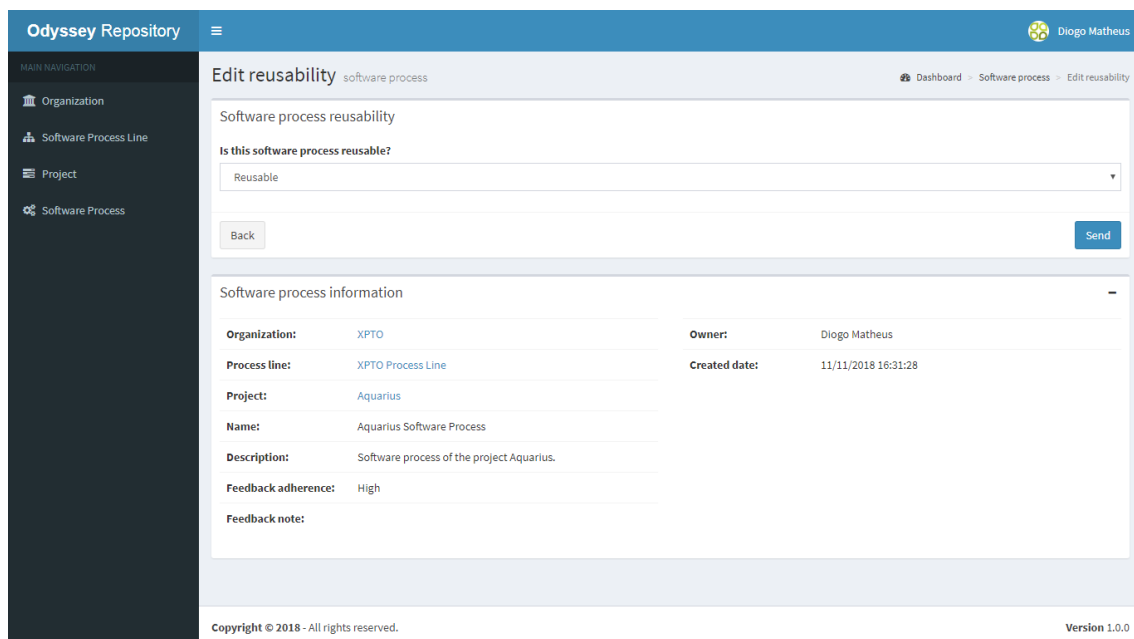


Figura 5.21 – Interface de controle de reusabilidade (Odyssey Repository)

5.5. Considerações Finais

Este capítulo apresentou o ferramental de suporte implementado para apoiar a sistemática da abordagem *Odyssey-ProcessCase* (Capítulo 4), sendo este composto pelas ferramentas Odyssey (ODYSSEY, 2019), que oferece uma infraestrutura para reutilização de software focada em modelos de domínio, e Odyssey Repository, um repositório de processos de software idealizado e implementado para centralizar as informações e artefatos previstos na sistemática da abordagem.

Neste cenário, ao ofertar meios para utilização prática da abordagem *Odyssey-ProcessCase*, o ferramental de suporte implementado pode ser classificado como um Sistema Especialista (SE) híbrido, pois explora diferentes técnicas para apoiar na resolução de problemas que exigem alto nível de especialização (*i.e.*, tomada de decisão para resolução de variabilidades de LPrS). Além disso, ao utilizar informações de contexto (*i.e.*, contexto de definição de processo de software), conforme previsto na sistemática da abordagem, este ferramental pode ser classificado como uma aplicação sensível ao contexto. Segundo FERNANDES (2009) um sistema é sensível ao contexto se utiliza informações de contexto para fornecer informações relevantes e/ou serviços para o usuário.

Algumas limitações podem ser observadas no ferramental de suporte da abordagem *Odyssey-ProcessCase*. Na ferramenta Odyssey, o uso do mecanismo de busca de similaridade de forma complementar exige que o usuário preencha as informações de contexto múltiplas vezes (*i.e.*, retrabalho), caso esta funcionalidade seja utilizada mais de uma vez durante a definição de processo de software específico de projeto. Neste ponto, melhorias podem ser realizadas para evitar o retrabalho do usuário ao utilizar os diferentes mecanismos de definição de processo de software, mantendo as informações de contexto em memória para realizar o preenchimento de forma automática. Além disso, o mecanismo de reconhecimento de contexto não foi finalizado no ferramental de suporte. Neste ponto, o mecanismo ofertado pela abordagem *UbiFEX*, denominado *UbiFEX-Simulation* (*i.e.*, *plug-in* da ferramenta Odyssey), oferece meios para obter diferentes processos de software específicos de projeto com base nas informações de contexto e regras do modelo de contexto de LPrS. Porém, este mecanismo precisa ser estendido para sistemática prevista na abordagem *Odyssey-ProcessCase* (Capítulo 4).

Na ferramenta Odyssey Repository, o fato de não ofertar recursos para controle de usuário por perfil (*e.g.*, Engenheiro de Processo de Software, Gerente de Projetos, etc) para restringir as ações de cada usuário, dificulta o gerenciamento dos artefatos relacionados à organização. Neste ponto, uma melhoria pode ser realizada para inserir o controle de usuário por perfil dentro de uma organização. Além disso, para um cenário de real utilização, supõe-se que uma melhoria deverá ser realizada visando armazenar os artefatos na nuvem, considerando questões relacionadas à segurança e sigilo de informações, para retirar essa responsabilidade do banco de dados. Por fim, melhorias podem ser realizadas para inserir recursos com o objetivo de apoiar a simulação e análise de processos de software, com o objetivo de atender a visão de repositório de SHAHZAD *et al.* (2010).

CAPÍTULO 6 - ESTUDO DE OBSERVAÇÃO

6.1. Introdução

Conforme observado no capítulo anterior, um ferramental de suporte foi implementado para ofertar meios para utilização prática da abordagem *Odyssey-ProcessCase* (Capítulo 4) com base na extensão da ferramenta Odyssey (ODYSSEY, 2019) e desenvolvimento da ferramenta Odyssey Repository. Porém, de acordo com a literatura de Engenharia de Software Experimental, a validade de qualquer corpo de conhecimento deve ser avaliada para que esse conhecimento possa ser considerado científico (JURISTO & MORENO, 2001).

Segundo BASILI *et al.* (1996), toda tecnologia proposta (*e.g.*, ferramenta, método, técnica) deve ser avaliada antes de ser disponibilizada para uso. Neste sentido, antes de avaliar a viabilidade e os benefícios da abordagem *Odyssey-ProcessCase*, este estudo tem o objetivo de caracterizar a aceitação do ferramental de suporte implementado com base no *Technology Acceptance Model* (TAM), ou seja, modelo de aceitação tecnológica (DAVIS, 1985). Este estudo pode ser classificado como um estudo de observação, onde o participante realiza alguma tarefa enquanto é observado por um experimentador (SHULL *et al.*, 2001).

O modelo de aceitação tecnológica foi proposto por Fred Davis em sua tese de doutorado (DAVIS, 1985), com base na *Theory of Reasoned Action* (TRA), ou seja, teoria da ação racional, uma teoria psicológica que procura explicar o comportamento (FISHBEIN & AJZEN, 1975). Em seu trabalho, DAVIS (1985) conclui que as pessoas tendem a usar ou não um sistema por meio dos conceitos de utilidade percebida e facilidade de uso percebida:

- **Utilidade percebida:** Grau em que um indivíduo acredita que o uso de um determinado sistema aumentaria seu desempenho no trabalho;
- **Facilidade de uso percebida:** Grau em que um indivíduo acredita que o uso de um sistema particular estaria livre de esforço físico e mental.

De acordo com LEE *et al.* (2003), o modelo de aceitação tecnológica é considerado a teoria mais influente e comumente empregada para descrever a aceitação individual de sistemas de informações. Neste ponto, há evidências de aplicações em

diferentes tecnologias (*e.g.*, sistemas hospitalares, processadores de texto) e diferentes perfis de participantes (*e.g.*, estudantes, especialistas).

Para definir o plano de pesquisa deste estudo de observação, além do modelo de aceitação tecnológica, foi utilizado o paradigma GQM (*Goal, Question, Metric*) (SOLIGEN & BERGHOUT, 1998). Neste cenário, os itens focados na utilidade e facilidade de uso percebida, propostos por DAVIS (1989) para caracterizar a aceitação e prever o uso de sistemas, são aplicados como questões. Por fim, para avaliar cada questão, foi adotada a escala psicométrica Likert (LIKERT, 1932, MCIVER & CARMINES, 1981), utilizada para determinar o nível de concordância ou discordância dos participantes. O trabalho de HERNANDES *et al.* (2012) foi utilizado como referência para elaboração do arranjo que combina estes diferentes conceitos.

Além desta seção introdutória, este capítulo está organizado de forma que o planejamento do estudo é apresentado na Seção 6.2. Na Seção 6.3, a etapa de execução do estudo é detalhada. Na Seção 6.4, os dados do estudo são analisados e interpretados. Na Seção 6.5, as ameaças à validade do estudo são enumeradas. Por fim, as considerações finais são apresentadas na Seção 6.6.

6.2. Planejamento

6.2.1. Objeto de Estudo

Ferramental de suporte da abordagem *Odyssey-ProcessCase*.

6.2.2. Objetivo Global

O objetivo global desse estudo é caracterizar a aceitação do ferramental de suporte da abordagem *Odyssey-ProcessCase*. Sendo este detalhado da seguinte forma:

- **Analisar** o ferramental de suporte da abordagem *Odyssey-ProcessCase*;
- **Com o propósito** de caracterizar sua aceitação;
- **Em relação** à utilidade e facilidade de uso percebida;
- **Do ponto de vista** de especialistas da área de processo de software;
- **No contexto** de definição de processo de software específico de projeto usando caracterizações e artefatos reutilizáveis de Linha de Processos de Software elaborados pelo pesquisador.

Este planejamento não busca nenhum tipo de comparação entre diferentes ferramentas de suporte. Além disso, esse estudo de observação não tem pretensão de responder qualquer questão relacionada ao desempenho para avaliar qual é o melhor mecanismo de definição de processo de software específico de projeto usando LPrS.

6.2.3. Objetivos, Questões e Métricas

Neste ponto, um modelo foi construído com base no paradigma GQM, sendo este composto por cinco objetivos específicos, quinze questões e cinco métricas. Os objetivos específicos deste estudo são apresentados na Tabela 6.1.

Tabela 6.1 – Objetivos específicos do estudo de observação

Objetivo	Descrição
G1	Avaliar a adequação do material de treinamento.
G2	Avaliar o suporte do mecanismo de busca de similaridade para o conceito de reutilização total de processo de software específico de projeto.
G3	Avaliar o suporte do mecanismo de busca de similaridade para o conceito de reutilização parcial de processo de software específico de projeto.
G4	Avaliar a facilidade de uso do ferramental de suporte.
G5	Avaliar a utilidade do ferramental de suporte.

Para analisar os objetivos específicos (Tabela 6.1), quinze questões foram elaboradas, sendo estas apresentadas na Tabela 6.2. Neste ponto, uma questão é focada na avaliação do material de treinamento, para identificar um possível impacto do treinamento sobre os resultados obtidos (Q1). Além disso, duas questões são focadas na avaliação do suporte oferecido pelo mecanismo de busca de similaridade no apoio aos conceitos de reutilização parcial e total de processo de software específico de projeto (Q2 e Q3). Por fim, doze questões propostas por DAVIS (1989) focadas na avaliação da utilidade e facilidade de uso percebida são adotadas (Q4-Q15).

Tabela 6.2 – Questões utilizadas para analisar os objetivos específicos

Objetivo	Questão	Descrição
G1	Q1	O material de treinamento aplicado foi suficiente para realização das tarefas.
G2	Q2	O mecanismo de busca de similaridade oferece suporte satisfatório para apoiar a reutilização de processos de software definidos em contexto similar como ponto de partida para adaptação visando definir um processo de software específico de projeto.
G3	Q3	O mecanismo de busca de similaridade oferece suporte satisfatório para visualizar as decisões tomadas em processos de software definidos em contexto similar visando apoiar a resolução de variabilidades de LPrS durante a definição de processo de software específico de projeto utilizando o mecanismo de recorte.

Objetivo	Questão	Descrição
G4	Q4	Aprender a utilizar a ferramenta Odyssey seria fácil para mim.
	Q5	Eu acharia fácil usar a ferramenta Odyssey para definir processos de software.
	Q6	Minha interação com a ferramenta Odyssey seria clara e compreensível.
	Q7	Eu acharia a ferramenta Odyssey flexível para interagir.
	Q8	Seria fácil para mim me tornar habilidoso na utilização da ferramenta Odyssey.
	Q9	Eu acharia a ferramenta Odyssey fácil de usar.
G5	Q10	Usar a ferramenta Odyssey no meu trabalho me permitiria realizar tarefas mais rapidamente.
	Q11	Usar a ferramenta Odyssey melhoraria meu desempenho no trabalho.
	Q12	Usar a ferramenta Odyssey no meu trabalho aumentaria minha produtividade.
	Q13	Usar a ferramenta Odyssey aumentaria minha eficácia no trabalho.
	Q14	Usar a ferramenta Odyssey tornaria mais fácil fazer o meu trabalho.
	Q15	Eu acharia a ferramenta Odyssey útil no meu trabalho.

Por fim, cada questão é avaliada pelas métricas apresentadas na Tabela 6.3, que utilizam uma escala *Likert* baseada em cinco alternativas (*i.e.*, cinco pontos).

Tabela 6.3 – Métricas utilizadas para avaliar as questões

Métrica	Descrição
M1	Número de participantes que escolheu "Discordo plenamente".
M2	Número de participantes que escolheu "Discordo parcialmente".
M3	Número de participantes que escolheu "Indiferente".
M4	Número de participantes que escolheu "Concordo parcialmente".
M5	Número de participantes que escolheu "Concordo plenamente".

6.2.4. Modelo de Interpretação

Para analisar os dados coletados por meio deste estudo de observação, um modelo de interpretação foi definido com base nos objetivos específicos, suas questões e métricas. Dessa forma, o modelo apresentado na Tabela 6.4, deve ser interpretado verificando as expressões de cada item, analisando sua veracidade, seguindo o formato “SE expressão, ENTÃO interpretação”.

Tabela 6.4 – Modelo de interpretação dos dados

Nº	Questão	Expressão	Descrição
1	Q1	$M5 + M4 + M3 \geq M2 + M1$	O material de treinamento fornecido foi suficiente para a condução do estudo de observação.
2	Q1	$M5 + M4 + M3 < M2 + M1$	O material de treinamento fornecido foi insuficiente e pode ter impactado o resultado do estudo de observação.
3	Q2	$M5 + M4 \geq M3 + M2 + M1$	O suporte ofertado pelo mecanismo de busca de similaridade foi suficiente para atender ao conceito de reutilização total.
4	Q2	$M5 + M4 < M3 + M2 + M1$	O suporte ofertado pelo mecanismo de busca de similaridade foi insuficiente para atender ao conceito de reutilização total. Dessa forma, os comentários dos participantes devem ser analisados visando identificar melhorias.
5	Q3	$M5 + M4 \geq M3 + M2 + M1$	O suporte ofertado pelo mecanismo de busca de similaridade foi suficiente para atender ao conceito de reutilização parcial.
6	Q3	$M5 + M4 < M3 + M2 + M1$	O suporte ofertado pelo mecanismo de busca de similaridade foi insuficiente para atender ao conceito de reutilização parcial. Dessa forma, os comentários dos participantes devem ser analisados visando aprimorar esta funcionalidade.
7	Qi, i=4 até 9	$M5 + M4 \geq M3 + M2 + M1$	O ferramental da abordagem <i>Odyssey-ProcessCase</i> é fácil de utilizar e o próximo passo é executar um estudo experimental na indústria.
8	Qi, i=4 até 9	$M5 + M4 + M3 \geq M2 + M1$	O ferramental da abordagem <i>Odyssey-ProcessCase</i> é fácil de utilizar, porém, o próximo passo é analisar os comentários dos participantes para identificar e implementar melhorias para incrementar a facilidade de uso antes de executar um estudo experimental na indústria.
9	Qi, i=4 até 9	$M5 + M4 + M3 < M2 + M1$	O ferramental da abordagem <i>Odyssey-ProcessCase</i> não apresentou indícios de facilidade de uso. O próximo passo é pesquisar por padrões de usabilidade, bem como analisar os comentários dos participantes para aprimorar a facilidade de uso antes de executar um estudo experimental na indústria.
10	Qi, i=10 até 15	$M5 + M4 \geq M3 + M2 + M1$	O ferramental da abordagem <i>Odyssey-ProcessCase</i> é útil e o próximo passo é executar um estudo experimental na indústria.
11	Qi, i=10 até 15	$M5 + M4 + M3 \geq M2 + M1$	O ferramental da abordagem <i>Odyssey-ProcessCase</i> é útil, porém, o próximo passo é analisar os comentários dos participantes para identificar e implementar melhorias para incrementar a utilidade antes de executar um estudo experimental na indústria.
12	Qi, i=10 até 15	$M5 + M4 + M3 < M2 + M1$	O ferramental da abordagem <i>Odyssey-ProcessCase</i> não apresentou indícios de utilidade. O próximo passo é revisar o design do ferramental, bem como analisar os comentários dos participantes para aprimorar a utilidade antes de executar um estudo experimental na indústria.

6.2.5. Expectativa

Neste estudo de observação, espera-se que os dados coletados apresentem indícios de que o material de treinamento foi suficiente para sua condução, bem como indícios de que o suporte ofertado pelo mecanismo de busca de similaridade atende aos conceitos de reutilização parcial e total de processo de software específico de projeto. Por fim, espera-se que o ferramental de suporte implementado para a abordagem *Odyssey-ProcessCase* apresente indícios de utilidade e facilidade de uso.

6.2.6. Seleção dos Participantes

Para a seleção dos participantes, foi definido como critério a experiência do mesmo na área de processo de software. Neste ponto, os participantes foram mapeados por conveniência, identificando na literatura pesquisadores da área de processo de software e LPrS. O perfil de especialista na área de processo de software foi escolhido assumindo como premissa que este pode colaborar com maior precisão para caracterização da aceitação do ferramental de suporte, bem como na identificação de melhorias.

Nenhum critério para formação de grupos foi definido, bem como nenhuma técnica de amostragem foi aplicada para seleção de participantes. Por fim, como critério para identificação de *outliers* ficou definido, a condução de uma análise *ad-hoc* com base na caracterização do participante, anotações realizadas pelo pesquisador responsável durante a execução das tarefas e o resultado reportado no questionário de avaliação.

6.2.7. Seleção de Variáveis

- **Independentes:** Ferramental de suporte para LPrS (Nominal), experiência do especialista (Ordinal), Informações de contexto de projeto de software (Ordinal) e modelo de característica de LPrS (Nominal);
- **Dependentes:** Utilidade percebida (*Likert*) e facilidade de uso percebida (*Likert*).

6.2.8. Instrumentação

Na etapa de planejamento, além do plano de pesquisa elaborado para direcionar o estudo de observação, um material de treinamento foi desenvolvido para reforçar os principais conceitos envolvidos na abordagem *Odyssey-ProcessCase*. Neste material de

treinamento, além dos conceitos envolvidos, um modelo de características de LPrS especificado para ser utilizado como base para a realização das tarefas é apresentado.

Para contextualizar o problema e as tarefas do estudo de observação, cinco caracterizações de contexto de definição de processo de software foram elaboradas, com base nas entidades e informações do modelo de adaptação de processo de software adotadas pela abordagem (Seção 4.2.4). Neste ponto, três caracterizações foram utilizadas como base para a definição de três processos de software específicos de projeto. As outras duas caracterizações elaboradas fazem parte do problema apresentado ao participante por meio das tarefas pré-definidas.

Por fim, outros cinco artefatos foram elaborados para apoiar na condução deste estudo de observação:

- **Formulário de consentimento**, que sinaliza o objetivo, bem como os termos do estudo de observação (Apêndice D.1);
- **Questionário de caracterização do participante**, que visa coletar informações para compreender o perfil do participante (Apêndice D.2);
- **Videoaula de treinamento**, focada na apresentação da abordagem e ferramental de suporte (<https://youtu.be/2kvRxVAS4vY>);
- **Documento de tarefas**, que lista tarefas pré-definidas. Neste ponto, duas versões do documento de tarefas foram definidas, invertendo a ordem das tarefas focadas no mecanismo de recorte e mecanismo de busca de similaridade (Apêndice D.3); e
- **Questionário de avaliação**, que visa coletar dados sobre o ferramental de suporte na percepção do participante para analisar as questões do estudo (Apêndice D.4).

6.2.9. Treinamento

Neste estudo, tópicos foram mapeados para apoiar no preparo do participante para a realização das tarefas pré-definidas. No total, cinco tópicos foram identificados, sendo estes:

- Linha de Processos de Software;
- Metamodelo e Notação *OdysseyProcess-FEX* (TEIXEIRA, 2016);
- Entidades e Informações de Contexto (LEITE, 2011);
- Abordagem *Odyssey-ProcessCase* (Capítulo 4); e
- Ferramenta Odyssey (ODYSSEY, 2019).

Neste cenário, os três primeiros tópicos são apresentados no material de treinamento, previamente enviado aos participantes por email. Além disso, os últimos dois tópicos são apresentados por meio de videoaula, disponibilizada no dia da execução do estudo. O conteúdo da videoaula tem como foco a definição de processo de software específico de projeto por meio dos mecanismos de recorte e busca de similaridade, escopo das tarefas pré-definidas a serem realizadas durante o estudo.

6.2.10. Recursos

Para a realização deste estudo de observação, alguns recursos necessários foram identificados, dentre eles estão:

- **Software:** Windows 7, Java 7, Odyssey 1.7.0, Odyssey Repository e *TeamViewer*;
- **Hardware:** Ultrabook LG U460 (5456), processador I5 e 8GB de memória.
- **Questionário:** Os questionários necessários para a realização deste estudo são apresentados no Apêndice D desta dissertação.

6.2.11. Avaliação do Plano de Pesquisa

Neste ponto, dois pesquisadores foram envolvidos na etapa de avaliação do plano de pesquisa. Essa avaliação teve foco na análise de viabilidade do plano, bem como a verificação dos artefatos elaborados para a execução do estudo. O processo de avaliação ocorreu por meio de iterações, onde melhorias eram identificadas e realizadas para mitigar as ameaças à validade, bem como o tempo e esforço necessário para realização das tarefas pré-definidas.

6.3. Execução do Estudo

6.3.1. Piloto

Com o objetivo de identificar melhorias no planejamento do estudo de observação, bem como equilibrar o tempo/esforço necessário para a realização das atividades na prática, um estudo piloto foi realizado no início de novembro de 2018. O estudo piloto foi conduzido com um participante, especialista em processo de software, caracterizado na próxima subseção. Os resultados obtidos não foram contabilizados na análise do estudo de observação, porém, colaboraram para evolução do plano.

6.3.1.1. Caracterização de Perfil do Participante

Os dados coletados por meio do questionário de caracterização do participante (Apêndice D) são apresentados na Tabela 6.5. Estes dados respeitam as opções e escala presentes no questionário em questão. Além disso, destaca-se na Tabela 6.5 a relação do participante com o grupo de reutilização de software (COPPE/UFRJ) e sua atuação.

Tabela 6.5 – Caracterização do participante do piloto

Item	Especialista
Grupo de reutilização de software	Membro
Formação	Doutorado
Atuação	Professor (CEFET-MG)
Projetos de software	4 (2-4 anos)
Implantação de processo de software	4 (1-2 anos)
Gerenciamento de processos de software	1 (Até 1 ano)
Definição de processo de software	2 (Acima de 4 anos)
Reutilização de processos de software	4 (1-2 anos)
Variabilidade e opcionalidade de processo	2 (Acima de 4 anos)
Engenharia de Requisitos	2 (Até 1 ano)

6.3.1.2. Execução

O participante convidado para execução do piloto realizou os procedimentos para verificar a adequação do plano de pesquisa e seus instrumentos.

A partir da execução deste piloto, foi possível observar que algumas tarefas estavam em nível macro. Por outro lado, observou-se que uma das tarefas exigia bastante esforço e tempo quando comparada as demais tarefas pré-definidas. Por fim, observou-se que um dos recursos do ferramental de suporte apresentava um *bug* de visualização de informação, onde uma informação importante ficava escondida.

Como solução, para as tarefas observadas em nível macro, subtarefas foram especificadas. Por outro lado, a tarefa observada como complexa foi simplificada sem

impactar o objeto de análise planejado, retirando seu foco em múltiplas tomadas de decisões. Dessa forma, uma tomada de decisão específica ficou definida. Por fim, o *bug* observado no ferramental de suporte foi corrigido para possibilitar a visualização da informação antes oculta.

6.3.2. Estudo de Observação

Após realizar as melhorias identificadas no piloto, o estudo de observação foi conduzido no período de 20 de novembro 2018 até 10 de dezembro 2018. No total, sete especialistas participaram do estudo. Esta seção apresenta a caracterização dos participantes, bem como descreve os procedimentos de execução do estudo.

6.3.2.1. Caracterização de Perfil dos Participantes

Os dados coletados por meio do questionário de caracterização do participante (Apêndice D) são apresentados na Tabela 6.6. Estes dados respeitam as opções e escala presentes no questionário em questão.

Tabela 6.6 – Caracterização dos participantes do estudo de observação

Item	P1	P2	P3	P4	P5	P6	P7
Grupo de Reutilização de Software	Externo	Externo	Externo	Ex-membro	Externo	Ex-membro	Externo
Formação	Mestrado	Mestrado	Doutorado	Pós-Doutorado	Pós-Doutorado	Doutorado	Doutorado
Atuação	Doutorando (PUC-PR)	Professor (ULBRA)	Pós-doutorando (UFRJ)	Professor (UFF)	Professor (UNIRIO)	Analista (BNDES)	Professor (IFRN)
Projetos de software	2 (2-4 anos)	3 (1-2 anos)	2 (Até 1 ano)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	3 (Acima de 4 anos)
Implantação de processo de software	0	3 (1-2 anos)	4 (Até 1 ano)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	3 (Acima de 4 anos)
Gerenciamento de processos de software	0	2 (1-2 anos)	0	4 (Até 1 ano)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	2 (Acima de 4 anos)
Definição de processo de software	1 (Até 1 ano)	2 (1-2 anos)	2 (Até 1 ano)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	3 (Acima de 4 anos)
Reutilização de processos de software	1 (Até 1 ano)	2 (1-2 anos)	2 (Até 1 ano)	4 (Acima de 4 anos)	2 (2-4 anos)	3 (Acima de 4 anos)	2 (2-4 anos)
Variabilidade e opcionalidade de processo	3 (2-4 anos)	2 (1-2 anos)	3 (1-2 anos)	4 (2-4 anos)	2 (2-4 anos)	3 (Acima de 4 anos)	3 (2-4 anos)
Engenharia de Requisitos	1 (1-2 anos)	2 (2-4 anos)	2 (Até 1 ano)	2 (Acima de 4 anos)	4 (Acima de 4 anos)	4 (Acima de 4 anos)	3 (Acima de 4 anos)

Além dos dados coletados por meio do questionário de caracterização do participante, destaca-se na Tabela 6.6 a relação dos participantes com o grupo de reutilização de software (COPPE/UFRJ) e sua atuação.

Embora os participantes deste estudo sejam considerados especialistas, seja pela relevância no cenário de processo de software ou pela participação em pesquisas de LPrS, os dados apresentados na Tabela 6.6, ressaltam a existência de diferentes níveis de experiência entre os participantes, porém, acredita-se que este fator é interessante para a caracterização da aceitação do ferramental de suporte.

Cada participante foi analisado com base nos critérios definidos para identificação de *outliers*, porém, nenhum *outlier* foi observado. Logo, os resultados dos sete participantes foram considerados.

6.3.2.2. Execução

Para a execução do estudo de observação, um roteiro foi definido na fase de planejamento, com o objetivo de padronizar e direcionar o estudo, contemplando as seguintes etapas:

1. Enviar convite aos especialistas;
2. Realizar agendamento e enviar material de treinamento;
3. Enviar formulário de consentimento, questionário de caracterização do participante e videoaula de treinamento;
4. Enviar documento de tarefas e informações para acesso remoto;
5. Enviar questionário de avaliação.

Neste cenário, para cada etapa, um *template* de email foi especificado para realizar a comunicação com os participantes. Dessa forma, respeitando o roteiro definido no planejamento, inicialmente, os especialistas da área de processo de software foram mapeados por conveniência. Em seguida, os especialistas identificados foram convidados para participar do estudo de observação, quinze no total. Por fim, aqueles que retornaram sinalizando disponibilidade foram agendados, sete no total, e o material de treinamento inicial foi enviado previamente à realização do estudo.

Na data agendada para execução, cada participante recebeu 30 minutos antes do combinado o formulário de consentimento, questionário de caracterização do participante e videoaula de treinamento.

Após assistir ao treinamento em videoaula (*i.e.*, 11 minutos de duração) e retornar os documentos preenchidos, o participante recebeu o documento de tarefas e acesso ao *TeamViewer* para realizar a conexão remota ao computador do pesquisador responsável. Neste momento, as tarefas pré-definidas foram executadas na máquina do pesquisador responsável enquanto este observava e realizava anotações. Por fim, o questionário de avaliação foi disponibilizado ao participante, preenchido e retornado ao pesquisador.

6.4. Análise dos Resultados

Esta seção tem como objetivo investigar os dados coletados por meio do questionário de avaliação, bem como as anotações realizadas pelo pesquisador responsável. Os dados foram usados para interpretar os resultados do estudo na visão dos objetivos específicos apresentados anteriormente, bem como para mapear melhorias visando a evolução do ferramental de suporte e, conseqüentemente, da abordagem *Odyssey-ProcessCase*.

6.4.1. Interpretação dos Dados Obtidos

Nesta etapa, com base nos objetivos específicos e modelo de interpretação definido, os dados obtidos são interpretados para analisar os resultados do estudo de observação. Cada objetivo específico é analisado a seguir.

Quanto ao primeiro objetivo específico, **avaliar a adequação do material de treinamento (G1)**, os dados coletados para questão Q1 foram analisados, sendo esses apresentados na Figura 6.1.

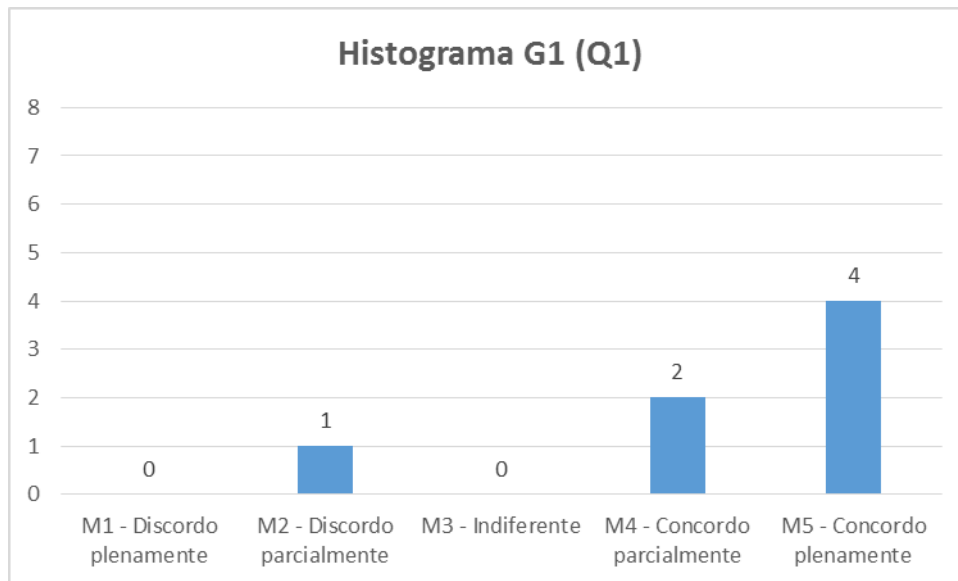


Figura 6.1 – Número de participantes X opção selecionada (G1)

Com base nos dados observados na Figura 6.1, o objetivo G1 pode ser interpretado de acordo com o item 1 da Tabela 6.4 ($M5 + M4 + M3 \geq M2 + M1$), onde $4 + 2 + 0 \geq 1 + 0$ é verdadeiro. Dessa forma, conforme descrito no item, os dados apresentam indícios de que: "O material de treinamento fornecido foi suficiente para condução do estudo de observação".

Neste ponto, o especialista P6 selecionou a opção "discordo parcialmente" e apresentou uma preocupação sobre o perfil dos participantes do estudo, indicando que o material poderia não ser suficiente para quem não tiver conhecimento sobre variabilidades de processo de software, LPrS, etc. Embora o perfil traçado para os participantes atenda o cenário ideal indicado pelo especialista, sua opção escolhida foi mantida e contabilizada (*i.e.*, discordo parcialmente).

Além disso, o especialista P4, que selecionou a opção "concordo plenamente", indicou que o material de treinamento poderia ser simplificado, reduzindo a duração da videoaula de treinamento.

Quanto ao segundo objetivo específico, **avaliar o suporte do mecanismo de busca de similaridade para o conceito de reutilização total de processo de software específico de projeto (G2)**, os dados coletados para questão Q2 foram analisados, sendo esses apresentados na Figura 6.2.

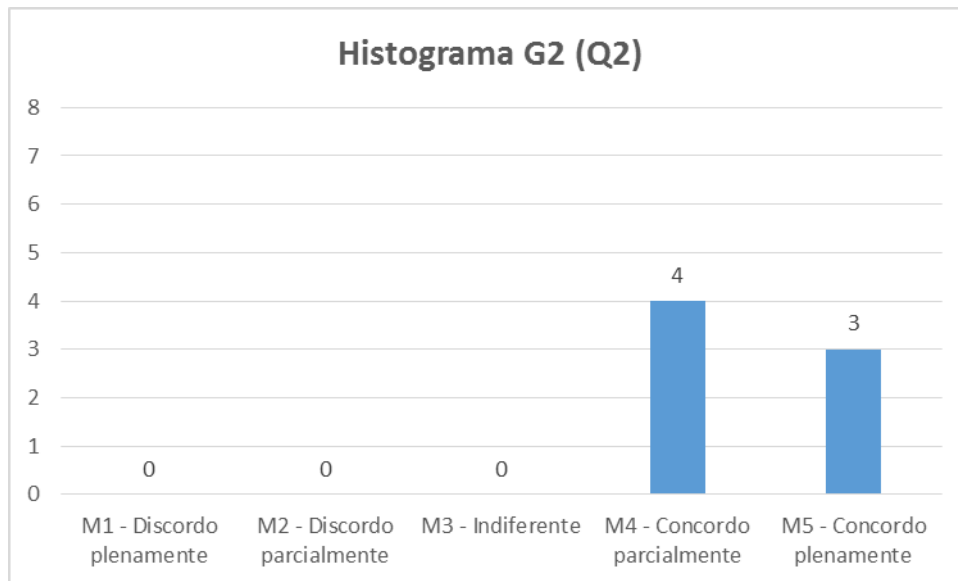


Figura 6.2 – Número de participantes X opção selecionada (G2)

Com base nos dados observados na Figura 6.2, o objetivo G2 pode ser interpretado de acordo com o item 3 da Tabela 6.4 ($M5 + M4 \geq M3 + M2 + M1$), onde $3 + 4 \geq 0 + 0 + 0$ é verdadeiro. Dessa forma, conforme descrito no item, os dados apresentam indícios de que: "O suporte ofertado pelo mecanismo de busca de similaridade foi suficiente para atender ao conceito de reutilização total".

Neste ponto, o especialista P1 embora tenha considerado o suporte ofertado pela busca de similaridade relevante, demonstrou preocupação e curiosidade sobre como o mecanismo iria se comportar em cenário real, contando com múltiplas disciplinas, elementos de processo, etc.

Além disso, o especialista P5 indicou que seria interessante possibilitar que o usuário filtre quais informações de contexto ele considera relevante, de forma que o cálculo de similaridade seja realizado sobre o subconjunto selecionado (funcionalidade não existente atualmente). O especialista P6, por outro lado, indicou que seria relevante para o usuário transparecer no ferramental como o cálculo de similaridade é realizado.

Por fim, os especialistas P4 e P7 indicaram que os recursos para análise e comparação de contexto são úteis e bastante claros.

Quanto ao terceiro objetivo específico, **avaliar o suporte do mecanismo de busca de similaridade para o conceito de reutilização parcial de processo de**

software específico de projeto (G3), os dados coletados para questão Q3 foram analisados, sendo esses apresentados na Figura 6.3.

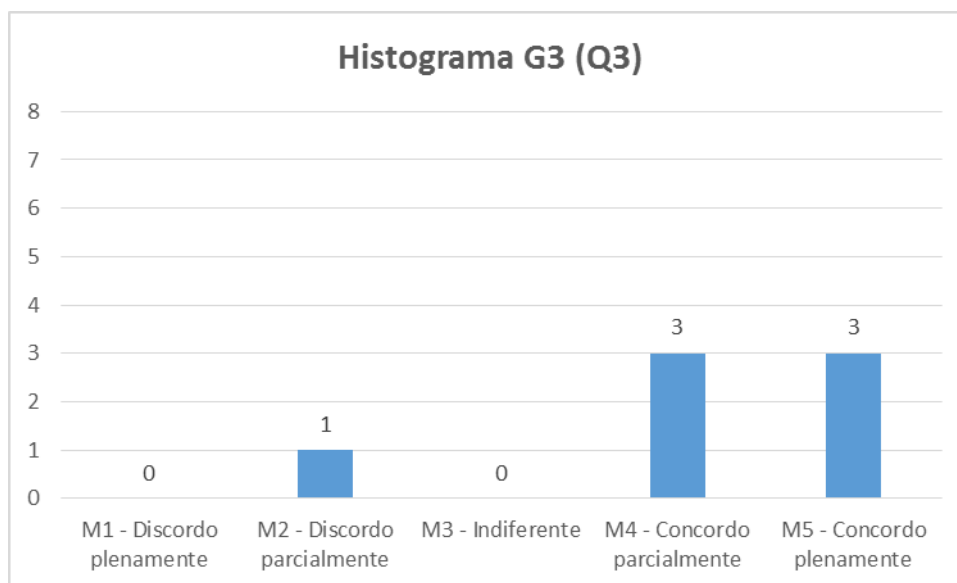


Figura 6.3 – Número de participantes X opção selecionada (G3)

Com base nos dados observados na Figura 6.3, o objetivo G3 pode ser interpretado de acordo com o item 5 da Tabela 6.4 ($M5 + M4 \geq M3 + M2 + M1$), onde $3 + 3 \geq 0 + 1 + 0$ é verdadeiro. Dessa forma, conforme descrito no item, os dados apresentam indícios de que: "O suporte ofertado pelo mecanismo de busca de similaridade foi suficiente para atender ao conceito de reutilização parcial".

Neste ponto, o especialista P3 selecionou a opção "discordo parcialmente". Porém, nenhum comentário foi fornecido no questionário de avaliação. Neste cenário, o especialista não demonstrou interesse em analisar as decisões tomadas em processos definidos em contexto similar, optando pela tomada de decisão própria.

Além disso, os especialistas P1 e P7 demonstraram preocupação e curiosidade sobre como o mecanismo iria se comportar em cenário real, contando com múltiplas disciplinas, elementos de processo, etc. O especialista P5 indicou que seria interessante comparar múltiplos casos retornados pela busca de similaridade em visão unificada (funcionalidade não existente atualmente).

Por fim, o especialista P4 sinalizou que é bastante positivo observar as escolhas (*i.e.*, decisões tomadas) em processos definidos em contexto similar, indicando que este tipo de recurso pode deixar o gerente de projetos mais seguro sobre as suas decisões.

Quanto ao quarto objetivo específico, **avaliar a facilidade de uso do ferramental de suporte (G4)**, os dados coletados para as questões Q4 até Q9 foram analisados, sendo esses apresentados na Figura 6.4.

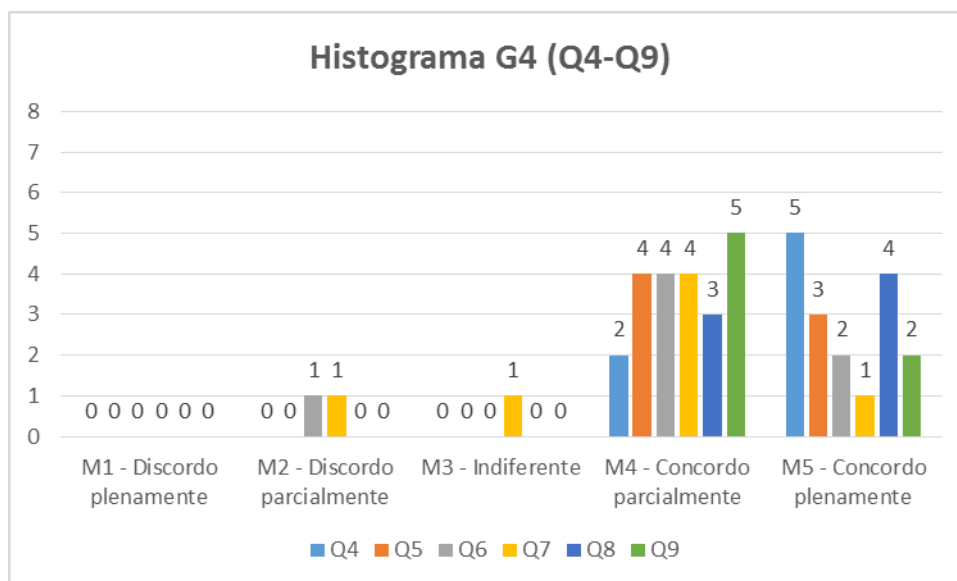


Figura 6.4 – Número de participantes X opção selecionada (G4)

Com base nos dados observados na Figura 6.4, o objetivo G4 pode ser interpretado de acordo com o item 7 da Tabela 6.4 ($M5 + M4 \geq M3 + M2 + M1$), onde $17 + 22 \geq 1 + 2 + 0$ é verdadeiro. Dessa forma, conforme descrito no item, os dados apresentam indícios de que: "O ferramental da abordagem *Odyssey-ProcessCase* é fácil de utilizar e o próximo passo é executar um estudo experimental na indústria".

Neste ponto, o especialista P1 sinalizou que o aspecto de flexibilidade do ferramental precisa ser melhorado, principalmente, no que diz respeito ao menu principal da ferramenta Odyssey. O especialista P5, por outro lado, sugeriu adicionar indicativos por meio de *wizards*, direcionando o usuário durante o uso do ferramental. Por fim, o especialista P7 destacou a simplicidade e objetividade da ferramenta Odyssey, sinalizando que o ferramental está em nível de utilização prática e que explorar experimentos em cenário real poderia levantar pontos de melhoria.

Quanto ao quinto objetivo específico, **avaliar a utilidade do ferramental de suporte (G5)**, os dados coletados para as questões Q10 até Q15 foram analisados, sendo esses apresentados na Figura 6.5.

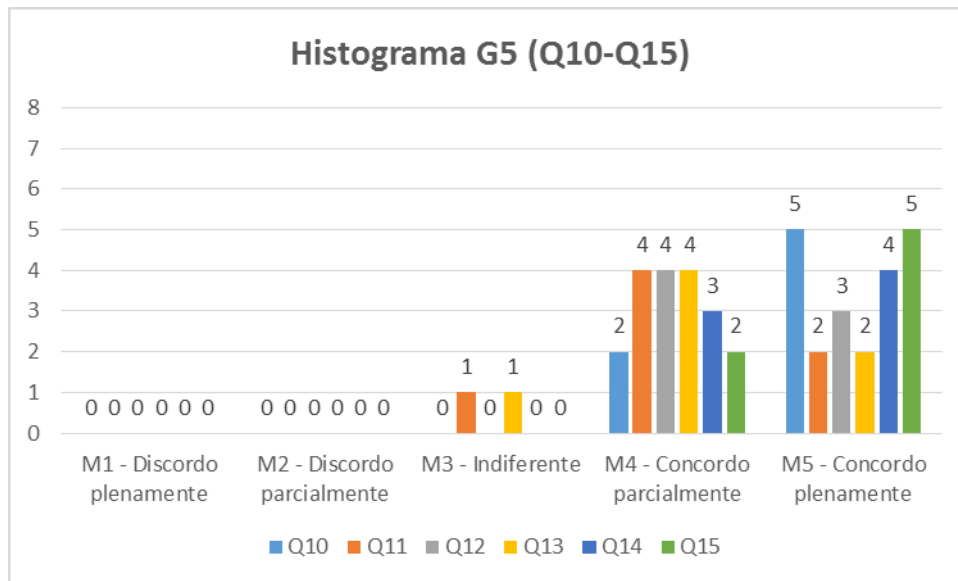


Figura 6.5 – Número de participantes X opção selecionada (G5)

Com base nos dados observados na Figura 6.5, o objetivo G5 pode ser interpretado de acordo com o item 10 da Tabela 6.4 ($M5 + M4 \geq M3 + M2 + M1$), onde $21 + 19 \geq 2 + 0 + 0$ é verdadeiro. Dessa forma, conforme descrito no item, os dados apresentam indícios de que: "O ferramental da abordagem *Odyssey-ProcessCase* é útil e o próximo passo é executar um estudo experimental na indústria".

Neste ponto, o especialista P4 destaca o recurso de comparação de informações de contexto do mecanismo de busca de similaridade, bem como sinaliza a utilidade da análise de decisões tomadas em processos definidos em contexto similar para apoiar o gerente de projetos. Por fim, o especialista P6 reforça que a consulta de casos similares pode ser de fato útil para apoiar na definição de processo de software específico de projeto.

6.4.2. Identificação de Melhorias

Nesta etapa, uma análise foi realizada para identificar oportunidades de melhoria no ferramental de suporte com base nos comentários fornecidos no questionário de avaliação e nas anotações realizadas por parte do pesquisador. No total, dez melhorias foram identificadas, sendo estas apresentadas na Tabela 6.7.

Tabela 6.7 – Melhorias identificadas no estudo de observação

Item	Descrição
1	Customizar menu da tela principal do Odyssey, oferecendo opções específicas de acordo com o item selecionado na árvore de domínios, que engloba processos de software (LPrS) e aplicações de produto (LPS).
2	Exibir informações específicas dos elementos de processo ao efetuar duplo clique no diagrama da interface do mecanismo de recorte.
3	Adicionar confirmação na interface de preenchimento das informações de contexto, apenas no cenário onde exista informação de contexto sem valor atribuído.
4	Manter em memória valores das informações de contexto do mecanismo de busca de similaridade, quando utilizado por meio de outros mecanismos (reutilização parcial).
5	Desacoplar mecanismo de busca de similaridade, quando utilizado por meio de outros mecanismos (reutilização parcial), com o objetivo de possibilitar a análise de casos similares e manipulação do mecanismo principal ao mesmo tempo.
6	Adicionar explicação sobre como o cálculo de similaridade é realizado na ferramenta.
7	Adicionar uma opção para customizar os pesos de cada informação de contexto do mecanismo de busca de similaridade, bem como filtrar informações, selecionando um subconjunto relevante para análise.
8	Adicionar recursos complementares para apoiar na tomada de decisões de casos similares, alternativas para visualização do modelo, como por exemplo, lista de elementos de processo, análise de elementos de processo específicos e suas ligações diretas e análise de múltiplos casos retornados simultaneamente.
9	Adicionar tutoriais internos na ferramenta Odyssey, com o objetivo de ensinar as etapas de determinada atividade.
10	Realizar um experimento para avaliar a necessidade das informações de contexto adotadas e compreender melhor seus diferentes pesos para apoiar a tomada de decisão na atividade de definição de processo de software específico de projeto usando LPrS.

Os itens da Tabela 6.7 são apresentados em ordem decrescente, de acordo com sua prioridade para implementação, que foi calculada de forma *ad-hoc*. Para calcular essa prioridade, o benefício de cada melhoria foi estimado (*i.e.*, contribuição para compreensão da atividade e interação entre usuário e ferramenta). Para verificar a contribuição real destas melhorias, o estudo deverá ser replicado após implementá-las.

6.5. Ameaças à Validade

Este estudo preliminar foi executado com o objetivo de caracterizar a aceitação do ferramental de suporte implementado para a abordagem *Odyssey-ProcessCase*. Porém, neste cenário, é comum que haja questões que possam impactar ou limitar a validade dos resultados observados. Estas questões são denominadas na literatura como ameaças à validade (WOHLIN *et al.*, 2000, TRAVASSOS *et al.*, 2002). Dessa forma, esta seção apresenta as principais ameaças identificadas, sendo essas distribuídas em quatro tipos de ameaças: I. Validade Interna; II. Validade Externa; III. Validade de Conclusão; e IV. Validade de Construção.

6.5.1. Validade Interna

- O plano de pesquisa propõe utilizar especialistas da área de processo de software, assumindo como premissa que estes representam um perfil de valor para o estudo de observação;
- A seleção por conveniência dos participantes representa uma ameaça à validade. Neste ponto, para mitigar essa ameaça, pesquisadores de diferentes estados brasileiros foram convidados para participar todo estudo.

6.5.2. Validade Externa

- O baixo número de participantes limita o poder de generalização dos resultados identificados para outros contextos;
- O estudo baseado em tarefas pré-definidas com foco na definição de processo de software específico de projeto limita o poder de generalização dos resultados observados, possibilitando sinalizar indícios de aceitação apenas do suporte para definição de processo de software específico de projeto usando LPrS por meio dos mecanismos explorados (recorte e busca de similaridade).

6.5.3. Validade de *Constructo*

- A ausência de métodos estatísticos para a verificação dos dados obtidos representa uma ameaça à validade. Neste ponto, um modelo de interpretação GQM foi idealizado na fase de planejamento para direcionar a análise dos dados obtidos, visando mitigar o impacto desta ameaça.

6.5.4. Validade de Construção

- O desenvolvimento de artefatos por parte dos pesquisadores (*i.e.*, modelo de características de LPrS, caracterização de contexto e processos de software específicos de projeto), ou seja, artefatos fictícios, não utilizados em projetos reais na indústria, representam uma ameaça à validade;
- O uso do software *TeamViewer* para possibilitar a participação de especialistas de diferentes estados brasileiros, por meio de acesso remoto, representa uma ameaça à validade dos resultados, por conta do risco de impacto (*e.g.*, lentidão na internet, resolução inadequada) na percepção do especialista sobre o ferramental de suporte.

6.6. Considerações Finais

Neste capítulo, foram apresentados os detalhes do planejamento e condução do estudo de observação, que teve como objetivo caracterizar a aceitação do ferramental de suporte implementado (Capítulo 5) para a abordagem *Odyssey-ProcessCase* (Capítulo 4). O foco de qualidade deste estudo foi delimitado pelos aspectos de facilidade de uso e utilidade.

O plano de pesquisa do estudo foi idealizado com base no modelo de aceitação tecnológica (TAM) e paradigma GQM, no contexto de especialistas da área de processo de software durante a definição de processo de software específico de projeto usando caracterizações e artefatos reutilizáveis de LPrS elaborados pelo pesquisador. Os dados obtidos no estudo foram interpretados com base no modelo de interpretação previamente definido na fase de planejamento. Além disso, um estudo piloto foi conduzido para melhorar o plano de pesquisa e artefatos do estudo.

Por meio da interpretação dos dados obtidos, comprovou-se o atendimento da expectativa do estudo de observação. Dessa forma, o estudo apresenta indícios de aceitação do ferramental de suporte da abordagem *Odyssey-ProcessCase*. Embora os resultados observados sejam positivos, as ameaças à validade representam limitações do estudo e de sua capacidade de generalização.

Por fim, mesmo ciente das limitações, melhorias foram identificadas para evoluir o ferramental de suporte antes de analisar os benefícios da abordagem na indústria.

CAPÍTULO 7 - CONCLUSÕES

7.1. Epílogo

Linha de Processos de Software (LPrS) surgiu como uma aplicação dos conceitos da técnica de Linha de Produtos de Software no domínio de processo de software visando obter uma sistemática para reutilização de processos (ROMBACH, 2005, JAUFMAN & MÜNCH, 2005, ARMBRUST *et al.*, 2009). Entre os benefícios dessa técnica, podemos citar: I. Aumento do potencial de reutilização; II. Redução dos riscos e esforço; e III. Aumento da qualidade e adequação dos processos de software gerados (ROMBACH, 2005, JAUFMAN & MÜNCH, 2005, BARRETO *et al.*, 2009).

Neste cenário, além das questões consideradas na fase de Engenharia de Domínio de Processo de Software (*e.g.*, representação de similaridades e variabilidade), na fase de Engenharia de Processo Específico de Projeto, as questões essenciais estão relacionadas à compreensão das particularidades de cada projeto de software, bem como da derivação de um processo de software específico de projeto que atenda estas particularidades a partir dos artefatos de LPrS. Para tanto, existe a necessidade de ofertar suporte para a tomada de decisão visando a resolução das variabilidades identificadas e representadas no domínio de processo de software.

A partir de uma revisão da literatura focada na definição de processo de software específico de projeto usando LPrS (Seção 2.1), observou-se uma concentração de abordagens que oferecem suporte para a tomada de decisão visando solucionar as variabilidades de LPrS, aplicando técnicas baseadas em mapeamento e regras de forma isolada. Porém, este tipo de técnica exige um processo de aquisição de conhecimento não trivial, que pode impactar na utilização da sistemática por conta da indisponibilidade de especialistas e sobrecarga na EDPS, principalmente, se não houver alternativas para a definição de processo de software específico de projeto.

Esta dissertação de mestrado apresentou uma abordagem sistemática para a definição de processo de software específico de projeto por meio de artefatos reutilizáveis de LPrS, denominada *Odyssey-ProcessCase*. A abordagem *Odyssey-ProcessCase* é uma abordagem para LPrS, focada no suporte para a tomada de decisão na resolução de variabilidades de LPrS que visa mitigar a dependência de especialistas. A sistemática da abordagem oferta mecanismos complementares para a definição de

processo de software específico de projeto, visando oferecer flexibilidade para diferentes perfis de profissionais e organizações.

Para o desenvolvimento deste trabalho, duas revisões da literatura foram realizadas (Capítulo 2 e 3), influenciando na sistemática da abordagem (Capítulo 4), bem como no ferramental de suporte implementado (Capítulo 5). Por fim, um estudo de observação foi conduzido (Capítulo 6) com a finalidade de caracterizar a aceitação do ferramental de suporte implementado com base no modelo de aceitação tecnológica (TAM). A partir dos resultados, verificou-se indícios de aceitação do ferramental de suporte no contexto de definição de processo de software específico de projeto usando LPrS, além disso, melhorias foram identificadas.

Neste capítulo, além desta seção, as contribuições desta dissertação são apresentadas na Seção 7.2. Na Seção 7.3, as limitações da pesquisa são discutidas. Na Seção 7.4, as publicações geradas no escopo desta dissertação de mestrado são apresentadas. Por fim, na Seção 7.5, algumas possibilidades de trabalhos futuros são descritas.

7.2. Contribuições

Esta dissertação de mestrado apresentou os resultados de uma pesquisa que estabeleceu uma abordagem para apoiar a tomada de decisão na definição de processo de software específico de projeto a partir de artefatos reutilizáveis de LPrS. Como contribuições alcançadas incluem-se:

- **Caracterização da fase de Engenharia de Processo Específico de Projeto (EPEP) em Linha de Processos de Software**, por meio da condução da *Quasi-Revisão Sistemática da Literatura* e resultados observados ao longo da Seção 2.1;
- **Caracterização de abordagens que aplicam Raciocínio Baseado em Casos para definição de processo de software**, a partir da *Quasi-Revisão Sistemática da Literatura* e resultados apresentados ao longo da Seção 3.2.2;
- **Definição de uma abordagem para Linha de Processos de Software**, com foco no processo de desenvolvimento *com* reutilização, visando o apoio à tomada de decisão para a definição de processo de software específico de projeto a partir de artefatos reutilizáveis. Neste ponto, destaca-se o uso das técnicas

RBC e SBR para ofertar mecanismos complementares de definição de processo de software;

- **Implementação de ferramental de suporte integrado**, oferecendo apoio para as atividades de ambas as fases de Linha de Processos de Software, ou seja, Engenharia de Domínio de Processo de Software, uma extensão do ferramental implementado em TEIXEIRA (2016), e Engenharia de Processo de Software Específico de Projeto, uma contribuição específica desta dissertação de mestrado;
- **Avaliação preliminar do ferramental de suporte**, por meio do estudo de observação baseado no modelo de aceitação tecnológica (TAM), no contexto de especialistas da área de processo de software durante a definição de processo de software específico de projeto usando caracterizações e artefatos reutilizáveis de LPrS elaborados pelo pesquisador.

A questão de pesquisa apresentada no Capítulo 1 foi utilizada para direcionar essa dissertação de mestrado. Neste ponto, os resultados da revisão da literatura (LPrS) apresentados no Capítulo 2, colaboram para compreender como as diferentes abordagens de LPrS solucionaram essa questão, identificando as técnicas de suporte aplicadas para apoiar na resolução de variabilidades durante a definição de processo de software específico de projeto usando LPrS.

Nesta dissertação, para responder esta questão, uma abordagem para LPrS foi estabelecida com base em mecanismos complementares para definição de processo de software. Dessa forma, a sistemática da abordagem *Odyssey-ProcessCase* (Capítulo 4), seu ferramental de suporte (Capítulo 5) e o estudo de observação focado na caracterização da aceitação do ferramental implementado (Capítulo 6), contribuem para solucionar esta questão ao aplicar diferentes técnicas de suporte para resolução de variabilidades de LPrS (*i.e.*, RBC e SBR) e obter indícios de aceitação do ferramental de suporte no contexto de especialistas da área de processo de software durante a definição de processo de software específico de projeto usando caracterizações e artefatos reutilizáveis de LPrS elaborados pelo pesquisador.

No entanto, LPrS é um tópico de pesquisa recente e estudos experimentais realizados por diferentes grupos de pesquisa são necessários para responder plenamente essa questão de pesquisa.

7.3. Limitações

Algumas limitações podem ser identificadas nesta pesquisa ao realizar uma análise crítica, tanto na abordagem apresentada quanto no seu ferramental de suporte. Dentre as principais limitações, estão:

- Para realizar o cálculo de similaridade na técnica de Raciocínio Baseado em Casos, uma métrica ponderada foi aplicada de forma que, para cada atributo (*i.e.*, informação de contexto) que descreve o problema e o caso, um peso é utilizado de acordo com sua influência para a resolução do problema (*i.e.*, definição de processo de software específico de projeto). Na versão atual do ferramental de suporte, os pesos foram definidos pelo pesquisador de forma *ad-hoc*. Este ponto pode ser melhor trabalhado ao realizar uma avaliação com especialistas para a definição dos pesos individuais de cada informação de contexto, considerando aspectos como relevância e disponibilidade ao planejar um projeto de software;
- O uso complementar dos diferentes mecanismos para definição de processo de software específico de projeto exige que o usuário preencha as informações de contexto múltiplas vezes no ferramental de suporte. Este ponto representa uma limitação e pode ser melhorado armazenando-se as informações de contexto em memória para realizar o preenchimento de forma automática;
- A ferramenta Odyssey em sua versão atual, não oferece meios para utilização do mecanismo de reconhecimento de contexto. Neste ponto, embora a abordagem *UbiFEX* ofereça o mecanismo *UbiFEX-Simulation* (*i.e.*, *plug-in* da ferramenta Odyssey), que oferta meios para obter diferentes processos de software específicos de projeto com base nas informações de contexto regras do modelo de contexto. Este mecanismo precisa ser adaptado para sistemática prevista na abordagem *Odyssey-ProcessCase* (Capítulo 4).
- A ferramenta Odyssey Repository em sua versão atual, não oferece recursos para realizar o controle de usuário por perfil (*e.g.*, Engenheiro de Processo de Software, Gerente de Projetos, etc) visando restringir as ações de cada usuário, o que dificulta o gerenciamento dos artefatos relacionados à organização. Neste ponto, uma melhoria pode ser realizada para inserir o controle de usuário por perfil dentro de uma organização específica;

- A avaliação executada nesta dissertação foi restrita, contando com sete especialistas da área de processo de software. Estes participantes utilizaram o ferramental de suporte por meio de tarefas pré-definidas para a definição de processo de software específico de projeto usando caracterizações e artefatos reutilizáveis de LPrS elaborados pelo pesquisador. Essa avaliação pode ser considerada um estudo inicial de caráter observatório. Para que exista evidência significativa da viabilidade e ganho da abordagem *Odyssey-ProcessCase*, estudos adicionais deverão ser realizados.
- Esta dissertação de mestrado não investigou os desafios relacionados ao cenário de evolução de LPrS (*e.g.*, análise de impacto, propagação de mudança). Neste ponto, uma revisão da literatura pode ser realizada para identificar os desafios relacionados à evolução de LPS e LPrS, bem como as práticas utilizadas para solucionar estes desafios nestas diferentes técnicas. Além disso, no contexto da abordagem *Odyssey-ProcessCase*, pode-se investigar a aplicação de diferentes técnicas para apoiar na evolução dos modelos de LPrS com base na análise dos casos gerados ao longo do ciclo de vida da LPrS (*e.g.*, Indução de Regras).

7.4. Publicações

No escopo desta dissertação de mestrado, dois artigos foram elaborados e submetidos para conferências nacionais, resultando nas seguintes publicações:

1. Os resultados da revisão da literatura (LPrS) foram apresentados na XLIV Conferência Latino-americana de Informática, no dia 2 de Outubro de 2018, no artigo *Software Process Definition using Process Lines: A Systematic Literature Review* (COSTA *et al.*, 2018a).
2. Uma visão geral desta dissertação foi apresentada no artigo *Odyssey-ProcessCase: A Case-Based Software Process Line Approach* (COSTA *et al.*, 2018b), no dia 19 de Outubro de 2018, na Trilha de Trabalhos Técnicos do XVII Simpósio Brasileiro de Qualidade de Software (SBQS).

Por fim, outros dois artigos estão planejados. O primeiro visa uma versão estendida dos resultados da revisão da literatura (RBC). O segundo tem como objetivo apresentar o ferramental de suporte implementado (Capítulo 5), bem como os resultados do estudo de observação conduzido (Capítulo 6).

7.5. Trabalhos Futuros

O tópicos de LPrS ainda apresenta diversos desafios em suas fases de EDPS e EPEP. A partir da abordagem apresentada e ferramental de suporte implementado ao longo desta pesquisa, é possível identificar oportunidades de melhoria. Entre as perspectivas de trabalhos futuros se destacam:

- Finalizar o mecanismo de reconhecimento de contexto no ferramental de suporte da abordagem *Odyssey-ProcessCase*, ofertando meios para utilização prática de todos os mecanismos previstos na abordagem;
- Realizar um estudo de avaliação focado na abordagem *Odyssey-ProcessCase*, visando verificar os benefícios obtidos pela utilização de sua sistemática, gerando evidência para apoiar um estudo na indústria;
- Realizar um estudo de avaliação na indústria, visando observar os benefícios da abordagem *Odyssey-ProcessCase* em cenário real de definição de projeto de software específico de projeto;
- Propor um conjunto de métricas para a análise de similaridade (*i.e.*, textual e estrutural) de modelos de processo de software especificados por meio do metamodelo *OdysseyProcess-FEX*;
- Especificar e implementar um mecanismo de composição automatizado para a definição de processo de software específico de projeto em LPrS a partir dos elementos de processo utilizados em casos similares;
- Inserir o debate sobre os riscos de impacto do processo de software definido sobre o projeto de software que será conduzido e produto de software que será desenvolvido. Este tipo de debate não foi observado na literatura;
- Conduzir um experimento para compreender a influência da complexidade de modelos de LPrS sobre a tomada de decisão para resolução de variabilidades de LPrS. Este ponto pode direcionar a escolha de diferentes estratégias de definição de processo de software com base no perfil do profissional;
- Identificar os desafios relacionados ao cenário de evolução de LPrS e propor um conjunto de práticas para apoiar a manutenção dos diferentes modelos de LPrS na abordagem *Odyssey-ProcessCase*.

REFERÊNCIAS BIBLIOGRÁFICAS

- AAMODT, A., PLAZA, E., 1994, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", *AI communications*, v. 7, n. 1, pp. 39-59.
- ABRAHAM, A., "Rule-Based expert systems". In: *Handbook of Measuring System Design*, v. 3, John Wiley & Sons Ltd, pp. 920-931, 2005.
- ALEGRIA, J. A. H., BASTARRICA, M. C., QUISPE, A., *et al.* "An MDE approach to software process tailoring". In: *Proceedings of the 2011 International Conference on Software and Systems Process (ICSSP)*, pp. 43-52, Waikiki, Honolulu, HI, USA, May. 2011.
- ALEIXO, F. A., FREIRE, M. A. SANTOS, W. C., *et al.*, "A Model-driven Approach to Managing and Customizing Software Process Variabilities". In: *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS)*, v. 3, pp. 92-100, Madeira, Portugal, June, 2010.
- ALEIXO, F. A., 2013, *Uma Abordagem Anotativa para a Gerência de Variabilidades em Linhas de Processos de Software: Concepção, Implementação e Avaliação*. Tese de D.Sc., Universidade Federal do Rio Grande do Norte (UFRN), Natal, RN, Brasil. Disponível em: <https://repositorio.ufrn.br/jspui/handle/123456789/17957>. Acesso em: 10 fev. 2019, 13:00:00.
- ARMBRUST, O., KATAHIRA, M., MIYAMOTO, Y., *et al.* "Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards". In: *Making Globally Distributed Software Development a Success Story*, v. 5007, *Lecture Notes in Computer Science*, Springer, pp. 160-172, 2008.
- ARMBRUST, O., KATAHIRA, M., MIYAMOTO, Y., *et al.*, 2009, "Scoping software process lines", *Software Process: Improvement and Practice*, v. 14, n. 3 (April), pp. 181-197.
- ATANASSOV, A., ANTONOV, L., 2012, "Comparative analysis of case based reasoning software frameworks jcolibri and myCBR", *Journal of the University of Chemical Technology and Metallurgy*, v. 47, n. 1 (January), pp. 83-90.

- BARRETO, A. S., MURTA, L. G. P., ROCHA, A. R. "Componentizando Processos Legados de Software Visando a Reutilização de Processos". In: *Anais do Simpósio Brasileiro de Qualidade de Software (SBQS)*, pp. 189-203, Ouro Preto, MG, Brasil, Junho. 2009.
- BARRETO, A., DUARTE, E., ROCHA, A. R., *et al.* "Supporting the definition of software processes at consulting organizations via software process lines". In: *Proceedings of the International Conference on the Quality of Information and Communications Technology (QUATIC)*, pp. 15-24, Porto, Portugal. 2010.
- BARRETO, A., 2011, *Uma Abordagem para Definição de Processos Baseada em Reutilização Visando à Alta Maturidade em Processos*. Tese de D.Sc, Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, RJ, Brasil. Disponível em: <https://www.cos.ufrj.br/uploadfile/publicacao/2247.pdf>. Acesso em: 10 fev. 2019, 23:00:00.
- BASILI, V., GREEN, S., LAITENBERGER, O., *et al.* "Packaging researcher experience to assist replication of experiments". In: *Proceedings of the ISERN Meeting*, pp. 3-6, Sydney, Australia, 1996.
- BERGMANN, R., *Experience Management - Foundations, Development Methodology, and Internet-Based Applications*. 1 ed. Berlin, Heidelberg, Springer-Verlag. 2002.
- BERNSTEIN, P. A., DAYAL, U. "An Overview of Repository Technology". In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pp. 705-713, Santiago, Chile. 1994.
- BOEHM, B., TURNER, R., 2003, "Using risk to balance agile and plan-driven methods", *Computer*, v. 36, n. 6 (June), pp. 57-66.
- BRECHÓ, 2019, "Projeto Brechó", In: http://reuse.cos.ufrj.br/site/pt/index.php?option=com_content&task=view&id=37&Itemid=46. Acesso em: 10 fev. 2019, 23:10:00.
- BUCHANAN, B. G., DUDA, R. O., "Principles of Rule-Based Expert Systems". In: YOVITS, M. C. (eds), *Advances in Computers*, 1 ed., chapter 4, Elsevier, 1983.

- BURKHARD, H. D. "Extending some concepts of CBR—Foundations of case retrieval nets". In: *Case-Based Reasoning Technology*, v. 1400, Lecture Notes in Computer Science, Springer, pp. 17-50. 1998.
- CARRICO, M. A., GIRARD, J. E., JONES, J. P. *Building knowledge systems: developing and managing rule-based applications*. 1 ed. New York, McGraw-Hill, Inc. 1989.
- CHEMUTURI, M. K., CAGLEY, T. M. *Mastering software project management: Best practices, tools and techniques*. 1 ed. Ross Publishing, Inc., 2010.
- CHEN, Y. "Research of the Case Retrieval Model Based on CBR". In: *Proceedings of the Third International Conference on Intelligent System Design and Engineering Applications (ISDEA)*, pp. 1132-1136, Hong Kong, China, Jan. 2013.
- COCKBURN, A. *Crystal Clear: A Human-Powered Methodology for Small Teams*. Pearson Education. 2004.
- COSTA, D. M., TEIXEIRA, E. N., WERNER, C. M. L., "Software Process Definition using Process Lines: A Systematic Literature Review". In: *Proceedings of the XLIV Latin American Computing Conference (CLEI)*, (to appear), São Paulo, Brazil, 01-05 October 2018a.
- COSTA, D. M., TEIXEIRA, E. N., WERNER, C. M. L., "Odyssey-ProcessCase: A Case-Based Software Process Line Approach". In: *Proceedings of the XVII Brazilian Software Quality Symposium (SBQS)*, pp. 170-179, Curitiba, Brazil, 17-19 October 2018b.
- DAI, F., LI, T. "Tailoring Software Evolution Process". In: *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, pp. 782–787, Qingdao, China, July. 2007.
- DAVIS, F., 1985, *A technology acceptance model for empirically testing new end-user information systems: theory and results*. Doctoral Thesis, MIT Sloan School of Management, Cambridge, MA. USA. Disponível em: <https://dspace.mit.edu/handle/1721.1/15192>. Acesso em: 11 fev. 2019, 18:00:00.

- DAVIS, F., 1989, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*, v. 13, n. 3 (Sep), pp. 319-340.
- DE CARVALHO, D. D., CHAGAS, L. F., LIMA, A. M., *et al.* "Software process lines: A systematic literature review", In: *Software Process Improvement and Capability Determination*, v. 447, Communications in Computer and Information Science, Springer, pp. 118-130, 2014.
- DEY, A. K., ABOWD, G. D., SALBER, D., 2001, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", *Human-computer interaction*, v. 16, n. 2-4 (Nov), pp. 97-166.
- DIAS, J. W., 2015, *Evidência Empírica das Abordagens Composicional e Anotativa para Gerência de Variabilidades em Linhas de Processo de Software*. Dissertação de M.Sc., Universidade Estadual de Maringá, Maringá, Brasil. Disponível em: <http://repositorio.uem.br:8080/jspui/handle/1/2519>. Acesso em: 06 abr. 2019, 12:00:00.
- FERNANDES, P., 2009, *UbiFEX: Uma Abordagem para Modelagem de Características de Linha de Produtos de Software Sensíveis ao Contexto*. Dissertação de M.Sc., Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, RJ, Brasil. Disponível em: <https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/2656>. Acesso em: 11 fev. 2019, 23:00:00.
- FERNANDES, P., WERNER, C., TEIXEIRA, E, 2011, "An Approach for Feature Modeling of Context-aware Software Product Line", *Journal of Universal Computer Science*, v. 17, n. 5 (Jan), pp. 807-829.
- FINNIE, G., SUN, Z., 2002, "Similarity and metrics in case-based reasoning", *International Journal of Intelligent Systems*, v. 17, n.3 (March), pp. 273-287.
- FISHBEIN, M., AJZEN, I., *Belief, attitude, intention and behavior: An introduction to theory and research*. 1 ed. California, Addison-Wesley. 1975.
- FORMAN, R. H., GASS, S. I., 2001, "The Analytic Hierarchy Process — An Exposition" *Operations Research*, v. 49, n. 4 (Aug), pp. 469-486.

- FOUQUE, G., MATWIN, S. "CAESAR: a system for case based software reuse" In: *Proceedings of the Seventh Knowledge-Based Software Engineering Conference*, pp. 90-99, Mclean, Sept. 1992.
- FRAKES, W. B., KANG, K., 2005, "Software reuse research: status and future", *IEEE transactions on Software Engineering*, v. 31, n. 7 (Aug), pp. 529-536.
- FUGGETTA, A. "Software process: a roadmap". In: *Proceedings of the Conference on The Future of Software Engineering*, pp. 25-34, Limerick, Ireland, June. 2000.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1 ed. Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc. 1989.
- HÁJEK, P., *Metamathematics of fuzzy logic*. 1 ed. Springer Science & Business Media. 1998.
- HAYES-ROTH, F., 1985, "Rule-based systems", *Communications of the ACM*, v. 28, n. 9 (Sept), pp. 921-932.
- HERNANDES, E., ZAMBONI, A., FABBRI, S., *et al.*, 2012, "Using GQM and TAM to evaluate StArt - a tool that supports Systematic Review". *CLEI Electronic Journal*, v. 15, n. 1, (April), pp. 13-25.
- JAUFMAN, I., MÜNCH, J. "Acquisition of a Project-Specific Process". In: *Proceedings of the International Conference on Product Focused Software Process Improvement (PROFES)*, pp 328-342 (June), Oulu, Finland. 2005.
- JURISTO, N., MORENO, A. M., 2001, *Basics of Software Engineering Experimentation*. 1 ed. Springer.
- KOLODNER, J. L., 1992, "An introduction to case-based reasoning", *Artificial Intelligence Review*, v. 6, n. 1 (March), pp. 3-34.
- KRUCHTEN, P., 2013, "Contextualizing agile software development" *Journal of Software: Evolution and Process*, v. 25, n. 4 (Sept), pp. 351-361.
- LEAKE, D. B., *Case-Based Reasoning: Experiences, Lessons and Future Directions*. 1 ed. Cambridge, MA, USA, MIT Press. 1996.

- LEE, Y., KOZAR, K. A., LARSEN, K. R., 2003, "The technology acceptance model: Past, present, and future", *Communications of the Association for information systems*, v. 12, n. 1 (Jan), pp. 752-780.
- LEITE, A. M. S., 2011, *Modelo de Contexto para Adaptação de Processos de Software*. Dissertação de M.Sc., Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, Brasil. Disponível em: <http://www2.uniriotec.br/ppgi/banco-de-dissertacoes-ppgi-unirio/ano-2011/modelo-de-contexto-para-adaptacao-de-processo-de-software/view>. Acesso em: 11 fev. 2019, 23:20:00.
- LIAO, S. H., 2005, "Expert system methodologies and applications - a decade review from 1995 to 2004", *Expert Systems with Applications*, v. 28, n. 1 (Jan), pp. 93-103.
- LIAO, T. W., ZHANG, Z., MOUNT, C. R., 1998, "Similarity measures for retrieval in case-based reasoning systems", *Applied Artificial Intelligence*, v. 12, n. 4 (Nov), pp. 267-288.
- LIKERT, R., 1932, "A technique for the measurement of attitudes", *Archives of psychology*.
- LU, R., SADIQ, S. "A reference architecture for managing business process variants". In: *Proceedings of 9th International Conference on Enterprise Information Systems (ICEIS)*. pp. 416-421, Madeira, Portugal, June. 2007.
- MAGDALENO, A. M., 2010, *Apoio à Decisão para o Balanceamento de Colaboração e Disciplina nos Processos de Desenvolvimento de Software*. Exame de Qualificação, Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil.
- MAGDALENO, A. M., DE ARAUJO, R. M., WERNER, C. M. L. "COMPOOTIM: An Approach to Software Processes Composition and Optimization". In: *Proceedings of the Iberoamerican Conference on Software Engineering (CIbSE)*, pp. 1-14, Buenos Aires, Argentina, April. 2012.
- MAGDALENO, A. M., BARROS, M. O., WERNER, C. M. L., *et al.*, 2015, "Collaboration optimization in software process composition", *Journal of Systems and Software*, v. 103, pp. 452-466.

- MALIK, K, CHOUDHARY, P., *Software Quality: A Practitioner's Approach*. McGraw-Hill Education. 2009.
- MANTARAS, R. L., PLAZA, E., 1997, "Case-Based Reasoning: an overview", *AI communications*, v. 10, n. 1, pp. 21-29.
- MCIVER, J., CARMINES, E. G., *Unidimensional scaling*. London, UK, SAGE. 1981.
- MURTA, L., 1999, *FRAMEDOC: Um Framework para a Documentação de Componentes Reutilizáveis*. Projeto Final de Curso, Universidade Federal do Rio de Janeiro (DCC/IM, UFRJ), Rio de Janeiro, Brasil. Disponível em: http://reuse.cos.ufrj.br/files/publicacoes/graduacao/PF_MURTA_Framedoc.pdf. Acesso em: 11 fev. 2019, 23:30:00.
- MURTA, L., BLOIS, A., LOPES, M., *et al.* "Run-Time Variability through Component Dynamic Loading" In: *Caderno de Ferramentas do XVIII Simpósio Brasileiro de Engenharia de Software*, pp. 67-72. 2004.
- NEWELL, A. "Production systems: Models of control structures". In: *Proceedings of the Eighth Annual Carnegie Symposium on Cognition*, pp. 463-526, Pittsburgh, Pennsylvania, May. 1973.
- NORTHROP, L. M., 2002, "SEI's software product line tenets", *IEEE software*, v. 19, n. 4 (July/August), pp. 32-40.
- NUNES, V. T., WERNER, C. M. L., SANTORO, F. M. "Context-Based Process Line". In: *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS)*, pp. 277-282, Funchal, Madeira, Portugal, June. 2010.
- ODYSSEY, 2019, "Projeto Odyssey", In: <http://reuse.cos.ufrj.br/odyssey>.
- OLIVEIRA, R. F., 2006, *Formalização e Verificação de Consistência na Representação de Variabilidades*. Dissertação de M.Sc., Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil. Disponível em: http://reuse.cos.ufrj.br/files/publicacoes/mestrado/Mes_OliveiraRegiane.pdf. Acesso em: 11 fev. 2019, 23:40:00.
- ORACLE, 2019, "Tecnologia Java", In: <https://www.java.com>.

- OSSHAN, H., HARRISON, W., TARR, P. "Software engineering tools and environments: a roadmap". In: *Proceedings of the Conference on the Future of Software Engineering*, pp. 261-277, Limerick, Ireland, June. 2000.
- OSTERWEIL, L. J. "Software processes are software too". In: *Proceedings of the 9th international conference on Software Engineering*, pp. 2-13, Monterey, California, USA. 1987.
- PAI, M., MCCULLOCH, M., GORMAN, J. D., *et al.*, 2004, "Systematic reviews and meta-analyses: an illustrated, step-by-step guide", *The National Medical Journal of India*, v. 17 n. 2 (Mar), pp. 86-95.
- PAULK, M. C., 2009, "A History of the Capability Maturity Model for Software", *ASQ Software Quality Professional*, v. 12, n. 1, pp. 5-19.
- PEDREIRA, O., PIATTINI, M., LUACES, M. R., *et al.*, 2007, "A systematic review of software process tailoring" *ACM SIGSOFT Software Engineering Notes*, v. 32, n. 3 (May), pp. 1-6.
- PEDRYCZ, W. *Fuzzy Control and Fuzzy Systems*. 2 ed. New York, NY, USA, John Wiley & Sons, Inc. 1993.
- PILLAT, R. M., OLIVEIRA, T. C., ALENCAR, P. S., *et al.*, 2015, "BPMNt: A BPMN extension for specifying software process tailoring", *Information and Software Technology*, v. 57, pp. 95-115.
- PRIETO-DÍAZ, R., 1991, "Making software reuse work: an implementation model", *ACM SIGSOFT Software Engineering Notes*, v. 16, n. 3 (July). pp. 61-68.
- RECIO-GARCÍA, J. A., GONZÁLEZ-CALERO, P. A., DÍAZ-AGUDO, B., 2014, "jcolibri2: A framework for building Case-based reasoning systems", *Science of Computer Programming*, v. 79 (Jan), pp. 126-145.
- RIESBECK, C. K., SCHANK, R. C. *Inside case-based reasoning*. 1 ed. Psychology Press. 1989.
- ROCHA, R. S., FANTINATO, M., 2013, "The use of software product lines for business process management: A systematic literature review", *Information and Software Technology*, v. 55, n. 8 (Aug), pp. 1355-1373.

- ROMBACH, D. "Integrated Software Process and Product Lines". In: *Unifying the Software Process Spectrum*, v. 3840, Lecture Notes in Computer Science, Springer, pp. 83-90, 2005.
- ROUILLÉ, E., COMBEMALE, B., BARAIS, O., *et al.* "Leveraging CVL to manage variability in software process lines". In: *Proceedings of the 19th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 148-157, Hong Kong, China, Dec. 2012.
- SANTOS, V., CORTÉS, M., BRASIL, M. "Dynamic Management of the Organizational Knowledge Using Case-Based Reasoning". In: *Evaluation of Novel Approaches to Software Engineering*, v. 69, Communications in Computer and Information Science, Springer, pp. 220-233, 2008.
- SHAHZAD, K., ELIAS, M., JOHANNESSON, P. "Requirements for a business process model repository: A stakeholders' perspective". In: *Business Information Systems*, v. 47, Lecture Notes in Business Information Processing, Springer, pp. 158-170, 2010.
- SHULL, F., CARVER, J., TRAVASSOS, G. H., 2001, "An Empirical Methodology for Introducing Software Processes", *ACM SIGSOFT Software Engineering Notes*, v. 26, n. 5 (Sept), pp. 288-296.
- SMYTH, B., CUNNINGHAM, P. "Déjà Vu: A Hierarchical Case-Based Reasoning System for Software Design" In: *Proceedings of the 10th European conference on Artificial Intelligence (ECAI)*, pp. 587-589, Vienna, Austria, Jan. 1992.
- SOLIGEN, R., BERGHOUT, E. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. 1 ed. London, UK, McGraw-Hill Education, 1998.
- SOMMERVILLE, I. *Engenharia de Software*. 9 ed. São Paulo, Pearson. 2011.
- TEIXEIRA, E. N., 2011, *OdysseyProcess-FEX: Uma Abordagem para Modelagem de Variabilidades de Linha de Processos de Software*. Dissertação de M.Sc., Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil. Disponível em: http://reuse.cos.ufrj.br/files/publicacoes/mestrado/Mes_Danny.pdf. Acesso em: 12 fev. 2019, 12:00:00.

- TEIXEIRA, E. N., VASCONCELOS, A., WERNER, C. M. L., 2016, *OdysseyProcess-FEX: Metamodelo e Notação para Representação de Variabilidades de Linha de Processos de Software*. Relatório Técnico, Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil. Disponível em: <https://www.cos.ufrj.br/index.php/en/publicacoes-pesquisa/details/20/2591>. Acesso em: 12 fev. 2019, 12:10:00.
- TEIXEIRA, E. N., 2016, *OdysseyProcessReuse: Uma Metodologia para Engenharia de Linha de Processos de Software Baseada em Componentes*. Tese de D.Sc., Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil. 2016. Disponível em: http://reuse.cos.ufrj.br/files/publicacoes/doutorado/Dou_Eldanae.pdf. Acesso em: 12 fev. 2019, 12:20:00.
- TEIXEIRA, E. N., VASCONCELOS, A., WERNER, C. M. L. "OdysseyProcessReuse - A Component-based Software Process Line Approach". In: *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS)*, pp. 231-238, Funchal, Portugal. 2018.
- TRAVASSOS, G. H., DOS SANTOS, P. S. M., NETO, P. G. M., et al. "An Environment to Support Large Scale Experimentation in Software Engineering". In: *Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, pp. 193–202, Belfast, United Kingdom, March. 2008.
- TRAVASSOS, G. H., GUROV, D., AMARAL, E. A. G. G., 2002, *Introdução à engenharia de software experimental*. Relatório Técnico. Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil. Disponível em: <http://www.cos.ufrj.br/uploadfile/es59002.pdf>. Acesso em: 12 fev. 2019, 12:30:00.
- VARGAS, R. V. "Using the analytic hierarchy process (AHP) to select and prioritize projects in a portfolio". In: *PMI global congress*, pp. 1-22, Washington, DC, Estados Unidos. 2010.
- VERONESE, G., CORREA, A., WERNER, C., et al. "ARES: Uma Ferramenta de Engenharia Reversa Java-UML." In: *Caderno de Ferramentas do XVI Simpósio*

- Brasileiro de Engenharia de Software*, pp. 347-352, Rio Grande do Sul, Jan. 2002.
- WANGENHEIM, C. G., WANGENHEIM, A. *Raciocínio Baseado em Casos*. 1 ed. Manole. 2008.
- WASHIZAKI, H. "Building software process line architectures from bottom up". In: *Product Focused Software Process Improvement*, v. 4034, Lecture Notes in Computer Science, Springer, pp. 415-421. 2006.
- WATSON, I. "Case-based reasoning is a methodology not a technology". In: *Research and Development in Expert Systems*, Springer, pp. 213-223. 1999.
- WILSON, D. R., MARTINEZ, T. R., 1997, "Improved heterogeneous distance functions", *Journal of Artificial Intelligence Research*, v. 6 (Jan). pp. 1-34.
- WOHLIN, C., RUNESON, P., HÖST, M., *et al.* *Experimentation in Software Engineering: An Introduction*. 1 ed. Springer. 2000.
- WOHLIN, C. "Second-generation systematic literature studies using snowballing". In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pp. 151-156, Limerick, Ireland, June. 2016.
- WOO, J. I., YANG, J. G., LEE, Y. H., *et al.*, 2014, "Healthcare decision support system for administration of chronic diseases", *Healthcare Informatics Research*, v. 20, n. 3 (July), pp. 173-182.
- XAVIER, J. R., 2001, *Criação e Instanciação de Arquiteturas de Software Específicas de Domínio no Contexto de uma Infraestrutura de Reutilização*. Dissertação de M.Sc., Universidade Federal do Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brasil. Disponível em: http://reuse.cos.ufrj.br/files/publicacoes/mestrado/Mes_XavierJoseRicardo.pdf. Acesso em: 12 fev. 2019, 12:40:00.
- XU, P., RAMESH, B., 2008, "Using Process Tailoring to Manage Software Development Challenges", *IT Professional*, v. 10, n. 4, pp. 39-45.
- YANG, J., YAU, N., 2000, "Integrating case-based reasoning and expert system techniques for solving experience-oriented problems", *Journal of the Chinese Institute of Engineers*, v. 23, n. 1 (May), pp. 83-95.

APÊNDICE A - PROTOCOLO DE REVISÃO DA LITERATURA (LPrS)

Objetivo Geral

O objetivo principal desta revisão da literatura é caracterizar o estado da arte da fase de Engenharia de Processo Específico de Projeto (EPEP), ou seja, definição de processo de software específico de projeto usando artefatos reutilizáveis de LPrS. Este objetivo envolve a identificação de publicações científicas sobre LPrS com o propósito de caracterizar seu suporte à fase de EPEP do ponto de vista do pesquisador.

Documentos de Controle

Para verificar a eficácia dos artigos selecionados, dois estudos foram definidos como documentos de controle. Estes estudos foram identificados por meio de pesquisa *ad-hoc* envolvendo estudos relacionados.

- MAGDALENO, A. M., DE ARAUJO, R. M., WERNER, C. M. L. "COMPOOTIM: An Approach to Software Processes Composition and Optimization". In: *Proceedings of the Iberoamerican Conference on Software Engineering (CibSE)*, pp. 1-14, Buenos Aires, Argentina, April. 2012.
- HURTADO, J. A., BASTARRICA, M. C., OCHOA, S. F., *et al.*, 2013, "MDE software process lines in small companies", *Journal of Systems and Software*, v. 86, n. 5 (May), pp. 1153-1171.

Escopo de Pesquisa

O escopo da revisão da literatura foi definido de acordo com a estrutura PICO (*Population, Intervention, Comparison and Outcomes* - População, Intervenção, Comparação e Resultado) (PAI *et al.*, 2004).

Estrutura	Descrição
População	Pesquisas de Linha de Processos de Software.
Intervenção	Abordagens que suportam a fase de Engenharia de Processo Específico de Projeto.
Comparação	Não é definido. O objetivo da revisão é caracterizar as abordagens.
Resultado	Abordagens de LPrS, técnicas, ferramentas e envolvimento da indústria.

Questões de Pesquisa

Com base no objetivo desta revisão da literatura e desafios da fase de EPEP, questões de pesquisa foram definidas para possibilitar a caracterização da fase de Engenharia de Processo Específico de Projeto em Linha de Processos de Software.

Tipo	Questão
Primária	Quais abordagens de Linha de Processos de Software com suporte à fase de Engenharia de Processo Específico de Projeto estão disponíveis na literatura?
Secundária (QS 1)	Existe evidência de uso de informações de contexto?
Secundária (QS 2)	Quais técnicas são aplicadas na fase de Engenharia de Processo Específico de Projeto?
Secundária (QS 3)	Quais ferramentas de suporte são relatadas?
Secundária (QS 4)	Quais avaliações são relatadas para validar a abordagem de LPrS?
Secundária (QS 5)	Existe evidência de envolvimento da indústria?

Máquinas de Busca

Três bases eletrônicas foram selecionadas para busca automática de publicações:

- I. Scopus: <https://www.scopus.com>; II. IEEEExplore: <https://ieeexplore.ieee.org>; e III. Compendex: <https://www.engineeringvillage.com>.

String de Busca

("process adaptation" OR "processes adaptation" OR "customization of processes" OR "software processes customization" OR "software process customization" OR "customizing software processes" OR "process definition" OR "processes definition" OR "process composition" OR "compose processes" OR "processes composition" OR "process tailoring" OR "processes tailoring" OR "tailoring of processes" OR "process development" OR "processes development" OR "process engineering" OR "processes engineering" OR "process design" OR "software process modeling" OR "software process modeling" OR "process implementation" OR "managing processes")

AND

("family of software process" OR "family of software processes" OR "families of software process" OR "families of software processes" OR "software process line" OR "software process lines" OR "software processes line" OR "software processes lines" OR "process-line" OR "process-lines" OR "processes-line" OR "processes-lines" OR "software process family" OR "software processes family" OR "software process families" OR "software processes families" OR "process-family" OR "processes-family" OR "process-families" OR "processes-families" OR "software process variability" OR "software process variabilities" OR "software processes variability" OR "software processes variabilities" OR "variabilities in software processes" OR "process domain engineering" OR "processes domain engineering" OR "process feature" OR "process features" OR "processes feature" OR "processes features" OR "process asset reuse")

Cr terios de Inclus o e Exclus o

Para direcionar a sele o de estudos prim rios relevantes, crit rios de inclus o e exclus o foram definidos.

Tipo	Cr�terio
Inclus�o	A abordagem deve usar o conceito de Linha de Processos de Software e apresentar algum suporte � fase de PSPE.
Exclus�o	Publica�es que n�o definem uma abordagem de Linha de Processos de Software, parcialmente ou totalmente, devem ser removidas.
Exclus�o	Publica�es com abordagens de Linha de Processos de Software que n�o suportam a fase de Engenharia de Processo Espec�fica do Projeto devem ser removidas.
Exclus�o	Publica�es do tipo de apresenta�o, tutorial, workshop ou tipo similar devem ser removidas.
Exclus�o	Publica�es sem acesso ao conte�do completo devem ser removidas.
Exclus�o	Publica�es n�o escritas em ingl�s devem ser desconsideradas. Para a realiza�o desta pesquisa foi selecionado o idioma ingl�s, devido a sua ado�o pela grande maioria das confer�ncias, peri�dicos internacionais e bases eletr�nicas relacionadas com o tema de pesquisa. A maioria das publica�es em portugu�s n�o se encontra indexada, embora se reconhe�a a sua import�ncia.

Formul rio de Extra o

Para apoiar na extra o de informa es, um formul rio foi definido visando realizar uma an lise sistem tica dos documentos selecionados na pesquisa.

Informa�o	Quest�o
T�tulo, resumo, autor(es) e ano de publica�o	N/A
Representa�o de contexto	QS 1
- <i>N�mero de informa�es de contexto sugeridas</i> - <i>Informa�es de contexto sugeridas</i>	QS 1
T�cnica de suporte aplicada na fase de EPEP	QS 2
Ferramental de suporte aplicado na abordagem	QS 3
- <i>Ferramental de suporte aplicado na fase de EPEP</i>	QS 3
Tipo de avalia�o da abordagem	QS 4
Envolvimento da ind�stria	QS 5

APÊNDICE B - PROTOCOLO DE REVISÃO DA LITERATURA (RBC)

Objetivo Geral

O objetivo principal desta revisão da literatura é caracterizar o estado da arte da aplicação da técnica de Raciocínio Baseado em Casos (RBC) na atividade de definição de processo de software. Este objetivo envolve a identificação de publicações científicas sobre definição de processo de software que utilizam os conceitos da técnica de RBC.

Documentos de Controle

Para verificar a eficácia dos artigos selecionados, dois estudos foram definidos como documentos de controle. Estes estudos foram identificados por meio de pesquisa *ad-hoc*.

- REIS, R. Q., REIS, C. A. L., NUNES, D. J. "Automated support for software process reuse: Requirements and early experiences with the APSEE model". In: *Proceedings of the Seventh International Workshop on Groupware*, pp. 50-55, Darmstadt, Germany, Sept. 2001.
- SANTOS, V. A., CORTÉS, M. I. "Organizational Knowledge Management through Software Process Reuse and Case-based Reasoning". In: *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS)*, pp. 223-228, Madeira, Portugal, June. 2010.

Escopo de Pesquisa

O escopo da revisão da literatura foi definido de acordo com a estrutura PICO (*Population, Intervention, Comparison and Outcomes* - População, Intervenção, Comparação e Resultado) (PAI *et al.*, 2004).

Estrutura	Descrição
População	Pesquisas de definição de processo de software.
Intervenção	Adoção dos conceitos da técnica de Raciocínio Baseado em Casos.
Comparação	Não é definido. O objetivo da revisão é caracterizar as abordagens.
Resultado	Abordagens, técnicas e ferramentas.

Questões de Pesquisa

Com base no objetivo desta revisão da literatura, questões de pesquisa foram definidas para possibilitar a caracterização das abordagens que adotam os conceitos da técnica de RBC para apoiar na definição de processo de software.

Tipo	Questão
Primária	Quais abordagens de definição de processo de software que adotam os conceitos da técnica de Raciocínio Baseado em Casos estão disponíveis na literatura?
Secundária (QS 1)	Qual é a representação dos casos na abordagem?
Secundária (QS 2)	Quais atributos são utilizados para descrever os casos na abordagem?
Secundária (QS 3)	Existe evidência de adoção do conceito de gerência de variabilidades de processo de software?
Secundária (QS 4)	Existe evidência de ferramental de suporte?
Secundária (QS 5)	Existe evidência de outra técnica aplicada para apoiar na definição de processo de software?
Secundária (QS 6)	Quais avaliações são relatadas para validar o estudo?

Máquinas de Busca

Três bases eletrônicas foram selecionadas para busca automática de publicações:

I. Scopus: <https://www.scopus.com>; II. IEEEXplore: <https://ieeexplore.ieee.org>; e III. Compendex: <https://www.engineeringvillage.com>.

String de Busca

("software process" OR "software processes")

AND

("case-based reasoning" OR "case based reasoning" OR "CBR")

Critérios de Inclusão e Exclusão

Para direcionar a seleção de estudos primários relevantes, critérios de inclusão e exclusão foram definidos.

Tipo	Critério
Inclusão	A abordagem deve usar o conceito da técnica de Raciocínio Baseado em Casos para apoiar na atividade de definição de processo de software.
Exclusão	Publicações do tipo de apresentação, tutorial, workshop ou tipo similar devem ser removidas.
Exclusão	Publicações sem acesso ao conteúdo completo devem ser removidas.
Exclusão	Publicações não escritas em inglês devem ser desconsideradas. Para a realização desta pesquisa foi selecionado o idioma inglês, devido a sua adoção pela grande maioria das conferências, periódicos internacionais e bases eletrônicas relacionadas com o tema de pesquisa. A maioria das publicações em português não se encontra indexada, embora se reconheça a sua importância.

Formulário de Extração

Para apoiar na extração de informações, um formulário foi definido visando realizar uma análise sistemática dos documentos selecionados na pesquisa.

Informação	Questão
Título, resumo, autor(es) e ano de publicação	N/A
Representação dos casos no repositório	QS 1
Atributos utilizados para descrever os casos no repositório	QS 2
- <i>Raciocínio para adoção dos atributos</i>	QS 2
- <i>Evidência de agrupamento dos atributos por categorias</i>	QS 2
- <i>Evidência de diferentes pesos para os atributos</i>	QS 2
Conceito de gerência de variabilidades de processo de software	QS 3
Técnica de suporte para definição de processo de software	QS 4
Ferramental de suporte da abordagem	QS 5
- <i>Recursos disponíveis para analisar os casos retornados</i>	QS 5
- <i>Nível de integração do ferramental visando modelagem e reutilização</i>	QS 5
Tipo de avaliação da abordagem	QS 6

APÊNDICE C - ENTIDADES E INFORMAÇÕES DE CONTEXTO PARA ADAPTAÇÃO DE PROCESSO DE SOFTWARE

Entidade de Contexto: Cliente

Conjunto de informações relacionadas ao cliente do projeto de software.

Número de informações de contexto: 3.

Informações de Contexto de Cliente

• **Envolvimento do Cliente**

Informação referente à presença e participação do cliente no projeto.

Aquisição	Valores	Relevância
Entrevista.	<ul style="list-style-type: none">Baixo (< 1 hora semanal);Moderado (1 a 5 horas semanais);Alto (> 5 horas semanais).	Alta

• **Entendimento do Escopo**

Nível de conhecimento do cliente sobre o escopo do projeto/produto.

Aquisição	Valores	Relevância
Entrevista.	<ul style="list-style-type: none">Nenhum (0% a 49%);Parcial (50% a 79%);Total (80% a 100%).	Alta

• **Facilidade em Expressar Requisitos**

Capacidade do cliente em expressar os requisitos do produto.

Aquisição	Valores	Relevância
Entrevista.	<ul style="list-style-type: none">Baixa;Moderada;Alta.	Alta

Entidade de Contexto: Produto

Conjunto de informações relacionadas ao produto que será desenvolvido, software gerado para atender as necessidades do cliente.

Número de informações de contexto: 2.

Informações de Contexto de Produto

- **Classe de Software Organizacional**

Relacionado aos tipos de software produzidos pela organização.

Aquisição	Valores	Relevância
Intranet organizacional.	<ul style="list-style-type: none">• Comum (mesmo tipo de software produzido na organização);• Diferenciado (tipo de software diferente do produzido pela organização).	Moderada

- **Integração entre Componentes de Projeto**

Grau de integração dos componentes de arquitetura do projeto com os componentes da organização.

Aquisição	Valores	Relevância
Plano de interface de projeto e área de arquitetura organizacional.	<ul style="list-style-type: none">• Baixa (nenhum ou poucos estão acoplados a outros componentes da organização);• Moderada (alguns componentes estão acoplados a outros componentes da organização);• Alta (todos os componentes estão acoplados a outros componentes da organização).	Baixa

Entidade de Contexto: Organização

Conjunto de informações relacionadas à organização produtora de software.

Número de informações de contexto: 4.

Informações de Contexto de Organização

• Objetivo de Negócio

Tipo de objetivo de negócio. Está relacionado ao planejamento estratégico da organização.

Aquisição	Valores	Relevância
Plano estratégico	<ul style="list-style-type: none">• Conservador (<i>e.g.</i>, diminuição de custos, aumento da qualidade);• Inovador (<i>e.g.</i>, uso de novas tecnologias).	Moderada

• Nível de Maturidade

Nível das melhores práticas adotadas pela organização de acordo com modelos de maturidade (*e.g.*, CMMI, MPS-BR).

Aquisição	Valores	Relevância
Intranet organizacional e área de qualidade.	<ul style="list-style-type: none">• Imaturo (não possui certificação);• Em processo de amadurecimento (em processo de certificação);• Maduro (possui certificação).	Moderada

• Cultura Organizacional

A cultura organizacional é formada por seus valores éticos e morais, princípios, crenças, políticas internas e externas, sistemas, e clima organizacional.

Aquisição	Valores	Relevância
Intranet organizacional.	<ul style="list-style-type: none">• Tradicional (controle sobre tarefas);• Democrática (tarefas ligadas à inovação).	Moderada

• Formalidade da Estrutura

Tipo de estrutura organizacional utilizada para estabelecer responsabilidades, distribuir autoridade e alocar recursos.

Aquisição	Valores	Relevância
Intranet organizacional.	<ul style="list-style-type: none">• Informal (estrutura voltada às pessoas);• Formal (estrutura rígida).	Moderada

Entidade de Contexto: Equipe

Conjunto de informações relacionadas aos participantes e papéis da equipe. Equipes formadas por diferentes participantes irão planejar e executar suas tarefas de formas diferentes.

Número de informações de contexto: 9.

Informações de Contexto de Equipe

• Compartilhamento de Conhecimento

Grau de compartilhamento de conhecimento entre os membros do projeto na organização (disponibilidade e troca).

Aquisição	Valores	Relevância
Sistema de gestão de conhecimento e dados históricos da organização (lições aprendidas).	<ul style="list-style-type: none">• Muito Baixo (informações obrigatórias);• Baixo (repositório de lições aprendidas);• Moderado (sistema de gestão do conhecimento);• Alto (sistema de gestão do conhecimento com práticas de compartilhamento e incentivo);• Muito Alto (rotineiro).	Moderada

• Tipo de Gerente de Projetos

Comportamento do gerente de projetos na execução do projeto.

Aquisição	Valores	Relevância
Dados históricos da organização.	<ul style="list-style-type: none">• Líder (orienta como se fosse um membro da equipe);• Gerente (centralizador de tarefas, controlando toda equipe).	Moderada

• Distribuição Geográfica

Grau de dispersão dos membros da equipe.

Aquisição	Valores	Relevância
Plano de projeto.	<ul style="list-style-type: none">• Não Distribuído (mesmo prédio);• Pouco Distribuído (bairros próximos);• Distribuído (estados diferentes);• Muito Distribuído (países diferentes).	Alta

• Entendimento do Escopo

Nível de conhecimento da equipe sobre o escopo do projeto/produto.

Aquisição	Valores	Relevância
Entrevista.	<ul style="list-style-type: none">• Nenhum (0% a 49%);• Parcial (50% a 79%);• Total (80% a 100%).	Alta

- **Tamanho da Equipe**

Quantidade de pessoas envolvidas no projeto.

Aquisição	Valores	Relevância
Plano de projeto.	<ul style="list-style-type: none"> • Muito Pequena (1-6); • Pequena (7-20); • Média (21-50); • Grande (51-100); • Muito Grande (+100). 	Alta

- **Experiência no Processo de Desenvolvimento**

Experiência no processo de desenvolvimento adotado pela organização.

Aquisição	Valores	Relevância
Entrevista e dados históricos da organização.	<ul style="list-style-type: none"> • Baixa (<= 1 ano); • Moderada (2 a 4 anos); • Alta (> 4 anos). 	Alta

- **Experiência Técnica**

Média do conhecimento prévio adquirido em projetos anteriores que utilizaram as mesmas tecnologias, ferramentas ou paradigmas.

Aquisição	Valores	Relevância
Entrevista, análise de currículo e participação em projetos da organização.	<ul style="list-style-type: none"> • Baixa (<= 1 ano); • Moderada (2 a 4 anos); • Alta (> 4 anos). 	Alta

- **Experiência no Domínio**

Conhecimento prévio do domínio da aplicação. Experiência adquirida por meio de participações em projetos anteriores do mesmo domínio.

Aquisição	Valores	Relevância
Entrevista, análise de currículo e participação em projetos da organização.	<ul style="list-style-type: none"> • Baixa (<= 1 ano); • Moderada (2 a 4 anos); • Alta (> 4 anos). 	Alta

- **Experiência do Gerente de Projetos**

Experiência do gerente de projetos em relação ao uso de habilidades gerenciais e liderança de equipes.

Aquisição	Valores	Relevância
Entrevista.	<ul style="list-style-type: none"> • Inexperiente (< 6 meses); • Iniciante (6 meses a 2 anos); • Intermediário (2 a 5 anos); • Experiente (5 a 10 anos); • Muito Experiente (> 10 anos). 	Moderada

Entidade de Contexto: Projeto

Conjunto de informações relacionadas ao projeto de software.

Número de informações de contexto: 11.

Informações de Contexto de Projeto

• Complexidade do Produto

Grau de dificuldade para desenvolver o produto.

Aquisição	Valores	Relevância
Conhecimento do gerente de projetos e cronograma de projeto.	<ul style="list-style-type: none">• Baixa;• Moderada;• Alta.	Alta

• Criticidade do Projeto

Classificação do projeto quanto à urgência e à importância de sua execução do ponto de vista da organização.

Aquisição	Valores	Relevância
Entrevista, plano estratégico e plano de projeto.	<ul style="list-style-type: none">• Muito Baixa (perda de conforto);• Baixa (prejuízos baixos, perdas facilmente recuperáveis);• Moderada (prejuízos moderados, perdas recuperáveis);• Alta (prejuízos altos, perdas irre recuperáveis);• Muito Alta (risco de descontinuidade do negócio).	Alta

• Tamanho do Projeto

O tamanho do projeto é definido pela sua capacidade de produção durante um período de trabalho considerado normal.

Aquisição	Valores	Relevância
Plano de projeto.	<ul style="list-style-type: none">• Pequeno (1 a 250 horas de trabalho);• Médio (251 a 2500 horas de trabalho);• Grande (mais de 2500 horas de trabalho).	Alta

• Completude do Escopo

Completude da definição do escopo do projeto.

Aquisição	Valores	Relevância
Documento de visão e documento de requisitos preliminar.	<ul style="list-style-type: none">• Baixa (0% a 49%);• Moderada (50% a 79%);• Alta (80% a 100%).	Alta

- **Comunicação do Projeto**

Comunicação com outros projetos e cliente de acordo com a quantidade de artefatos voltados à comunicação.

Aquisição	Valores	Relevância
Plano de comunicação.	<ul style="list-style-type: none"> • Muito Baixa; • Baixa; • Moderada; • Alta; • Muito Alta. 	Alta

- **Nível de Risco**

Média da probabilidade e impacto relacionados à ocorrência de eventos que afetem o projeto, seu processo ou produto.

Aquisição	Valores	Relevância
Plano de projeto e plano de riscos.	<ul style="list-style-type: none"> • Positivamente Alto (planejamento antecipado); • Positivamente Baixo (planejamento realista); • Moderado (planejamento apertado); • Negativamente Baixo (deslize no planejamento); • Negativamente Alto (planejamento irreal). 	Alta

- **Originalidade do Projeto**

Classificação de originalidade do projeto frente à experiência da organização no desenvolvimento de projetos desta natureza.

Aquisição	Valores	Relevância
Dados históricos da organização.	<ul style="list-style-type: none"> • Comum (semelhante aos demais projetos); • Original (diferente dos demais projetos). 	Alta

- **Duração do Projeto**

Classificação do tempo previsto para duração do projeto (*i.e.*, número de meses).

Aquisição	Valores	Relevância
Cronograma.	<ul style="list-style-type: none"> • Curto Prazo (abaixo de 6 meses); • Médio Prazo (entre 6 meses e 2 anos); • Longo Prazo (acima de 2 anos). 	Alta

- **Exigência Contratual**

Restrições contratuais ou legais as quais o projeto é submetido (e.g., exigência para a produção de certos artefatos).

Aquisição	Valores	Relevância
Contratos de projeto.	<ul style="list-style-type: none"> • Inexistente; • Simples; • Moderado; • Complexo. 	Alta

- **Custo do Projeto**

O custo do projeto é a aproximação dos custos de todos os recursos planejados.

Aquisição	Valores	Relevância
Termo de abertura de projeto.	<ul style="list-style-type: none"> • Muito Baixo (abaixo de 20 mil); • Baixo (entre 20 mil e 100 mil); • Moderado (entre 100 mil e 1 milhão); • Alto (entre 1 milhão e 10 milhões); • Muito Alto (acima de 10 milhões). 	Alta

- **Esforço do Projeto**

O esforço do projeto é a quantidade de unidades de mão de obra necessárias para terminar as atividades do cronograma (derivado do número de pessoas dedicadas ao projeto considerando o pico e a carga mensal média).

Aquisição	Valores	Relevância
Plano de projeto.	<ul style="list-style-type: none"> • Baixo (< 5H/mês); • Moderado (entre 5H/mês e 20H/mês); • Alto (> 20H/mês). 	Alta

Entidade de Contexto: Processo

Conjunto de informações relacionadas ao processo da organização produtora de software.

Número de informações de contexto: 2.

Informações de Contexto de Processo

- **Flexibilidade de Adaptação do Processo Padrão**

Grau de adaptação do processo de acordo com as diretrizes organizacionais.

Aquisição	Valores	Relevância
Área de processos organizacionais.	<ul style="list-style-type: none">• Baixa (poucos elementos do processo podem ser adaptados);• Moderada (elementos do processo podem ser adaptados);• Alta (todo o processo pode ser adaptado).	Baixa

- **Ferramenta(s) de Apoio**

Grau de automação e características das ferramentas de suporte utilizadas nas atividades do projeto.

Aquisição	Valores	Relevância
Área de qualidade e área de infraestrutura.	<ul style="list-style-type: none">• Manual (não utilização de ferramenta case);• Semi-Automatizado (utilização parcial de ferramenta case);• Automatizado (utilização total de ferramenta case).	Baixa

APÊNDICE D - INSTRUMENTOS UTILIZADOS NO ESTUDO DE OBSERVAÇÃO

D.1 - Formulário de Consentimento

OBJETIVO

Este estudo visa caracterizar o ferramental de suporte da abordagem *Odyssey-ProcessCase*, visando ao suporte para definição de processo de software específico de projeto a partir de artefatos reutilizáveis de Linha de Processos de Software.

IDADE

Eu declaro ter mais de 18 anos de idade e concordo em participar de um estudo conduzido por Diogo Matheus Costa na COPPE/UFRJ.

PROCEDIMENTO

Este estudo ocorrerá em sessão única, que incluirá um treinamento. Eu entendo que, uma vez que o estudo tenha terminado, os trabalhos que desenvolvi serão analisados visando à identificação de melhorias na abordagem *Odyssey-ProcessCase* e seu ferramental de suporte.

CONFIDENCIALIDADE

Toda informação coletada neste estudo é confidencial, e meu nome não será divulgado. Da mesma forma, me comprometo a não comunicar os meus resultados enquanto não terminar o estudo, bem como manter sigilo das técnicas e documentos apresentados e que fazem parte do experimento.

BENEFÍCIOS E LIBERDADE DE DESISTÊNCIA

Eu entendo que os benefícios que receberei deste estudo são limitados ao aprendizado do material que é distribuído e ensinado, bem como entendo que sou livre para realizar perguntas a qualquer momento. Além disso, entendo que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento de técnicas e processos da Engenharia de Software.

PESQUISADORES RESPONSÁVEIS

- Diogo Matheus Costa - PESC/COPPE/UFRJ;
- Eldânae Nogueira Teixeira - PESC/COPPE/UFRJ;
- Cláudia Maria Lima Werner - PESC/COPPE/UFRJ.

NOME: _____

ASSINATURA: _____ **DATA:** _____

D.2 - Caracterização do Participante

Código do participante: _____

Este formulário contém algumas perguntas sobre sua experiência.

Formação acadêmica:

Pós-doutorado Doutorado Mestrado Especialização Graduação
 Outro: _____

Tempo na academia (sistemas e computação): _____

Tempo na indústria (sistemas e computação): _____

Experiência em projetos de software:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Experiência implantando processo de desenvolvimento de software:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Experiência gerenciando processo de desenvolvimento de software:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Experiência definindo processo de desenvolvimento de software:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Experiência com reutilização de processos de software:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Experiência com os conceitos de variabilidade e opcionalidade em processo de software:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Experiência com domínio de Engenharia de Requisitos:

Nenhuma Até 1 ano Entre 1 e 2 anos Entre 2 e 4 anos Acima de 4 anos
Grau de experiência: _____ (escala apresentada abaixo)

Escala de grau de experiência:

0. Nenhuma experiência
1. Conhecimento teórico (sala de aula ou livros);
2. Conhecimento prático em ambiente de ensino (sala de aula);
3. Conhecimento prático em projetos pessoais;
4. Conhecimento prático na indústria.

Agradecemos seu apoio, disponibilidade e contribuição para esta pesquisa.

Diogo Matheus Costa
Eldânae Nogueira Teixeira
Cláudia Maria Lima Werner

D.3 - Documento de Tarefas

INSTRUÇÕES

Código do participante: _____

Este estudo será acompanhado pelo pesquisador. Sempre que possível, verbalize seus pensamentos. Pergunte e comente tudo que achar necessário.

CONTEXTUALIZAÇÃO

Para realização deste estudo, suponha que você é o gerente de projetos da organização XPTO, uma empresa de desenvolvimento de software. Neste cenário, você será exposto a duas caracterizações de projeto de software, visando os projetos Sirius (Apêndice D.3.A) e Orion (Apêndice D.3.B) da organização XPTO.

Com base nas caracterizações de projeto fornecidas, você deverá seguir as tarefas pré-definidas visando à definição de processos de software específico de projeto. Para apoiar nesta atividade, a organização XPTO trabalha com o conceito de Linha de Processos de Software (LPrS), disponibilizando uma LPrS nomeada de XPTO Process Line. Pensando na viabilidade deste estudo, o domínio fornecido foi simplificado e limita-se à disciplina de Engenharia de Requisitos (Apêndice D.3.C).

LISTA DE TAREFAS

Por favor, realize as tarefas descritas nesta seção de acordo com o roteiro apresentado por meio da ferramenta Odyssey.

[DEFINIÇÃO DE PROCESSO DE SOFTWARE – BUSCA DE SIMILARIDADE]

1 - Inicie a definição de processo de software utilizando a caracterização do projeto Sirius por meio do mecanismo de busca de similaridade (*based on similarity search*).

1.1 - Nomeie o processo de "Sirius Process V1", mantenha os campos estereótipo e descrição em branco.

2 - Execute os passos para realizar a busca, ao obter os processos similares, indique qual destes você acha mais apropriado para reutilizar como ponto de partida para definição do processo de software do projeto Sirius:

Resposta: _____

3 - Indique o processo de software com maior nível de similaridade retornado pelo ferramental de suporte:

Resposta: _____

4 - Analise as informações de contexto do processo de software com maior nível de similaridade retornado pelo ferramental de suporte (*context compare*).

4.1 - Indique o número de informações de contexto equivalentes:

Resposta: _____

5 - Selecione o processo de software que você acredita ser mais apropriado como ponto de partida para definição do processo de software do projeto Sirius e finalize o mecanismo para recuperar o modelo do processo de software escolhido.

[DEFINIÇÃO DE PROCESSO DE SOFTWARE – RECORTE]

6 - Inicie a definição de processo de software por meio do mecanismo de recorte (*based on domain model*).

6.1 - Nomeie o processo de "Orion Process V1", mantenha os campos estereótipo e descrição em branco.

6.2 - Selecione o escopo padrão da XPTO Process Line (*default*).

6.3 - Selecione o subtipo de definição tradicional (*traditional*).

7 - Indique um ponto de variação presente no modelo de características da Linha de Processos de Software da organização XPTO (*XPTO Process Line*):

Resposta: _____

8 - Utilize o mecanismo de busca de similaridade (*similarity search*) pela interface de recorte com base na caracterização do projeto Orion.

8.1 - Analise como o ponto de variação indicado anteriormente (Tarefa 7) foi solucionado em processos de software definidos em contexto similar.

9 - Selecione o(s) elemento(s) de processo adequado(s) para resolução do ponto de variação indicado anteriormente (Tarefa 7) de acordo com as necessidades do projeto Orion.

9.1 - Após solucionar o ponto de variação, avance para o próximo passo e finalize a definição de processo de software. (Não existe necessidade de solucionar as demais variabilidades)

[SALVAR]

10 - Salve o resultado deste estudo por meio da ferramenta Odyssey, no formato XML, na área de trabalho do sistema operacional.

10.1 - Nomeie o arquivo de "XPTO.xml".

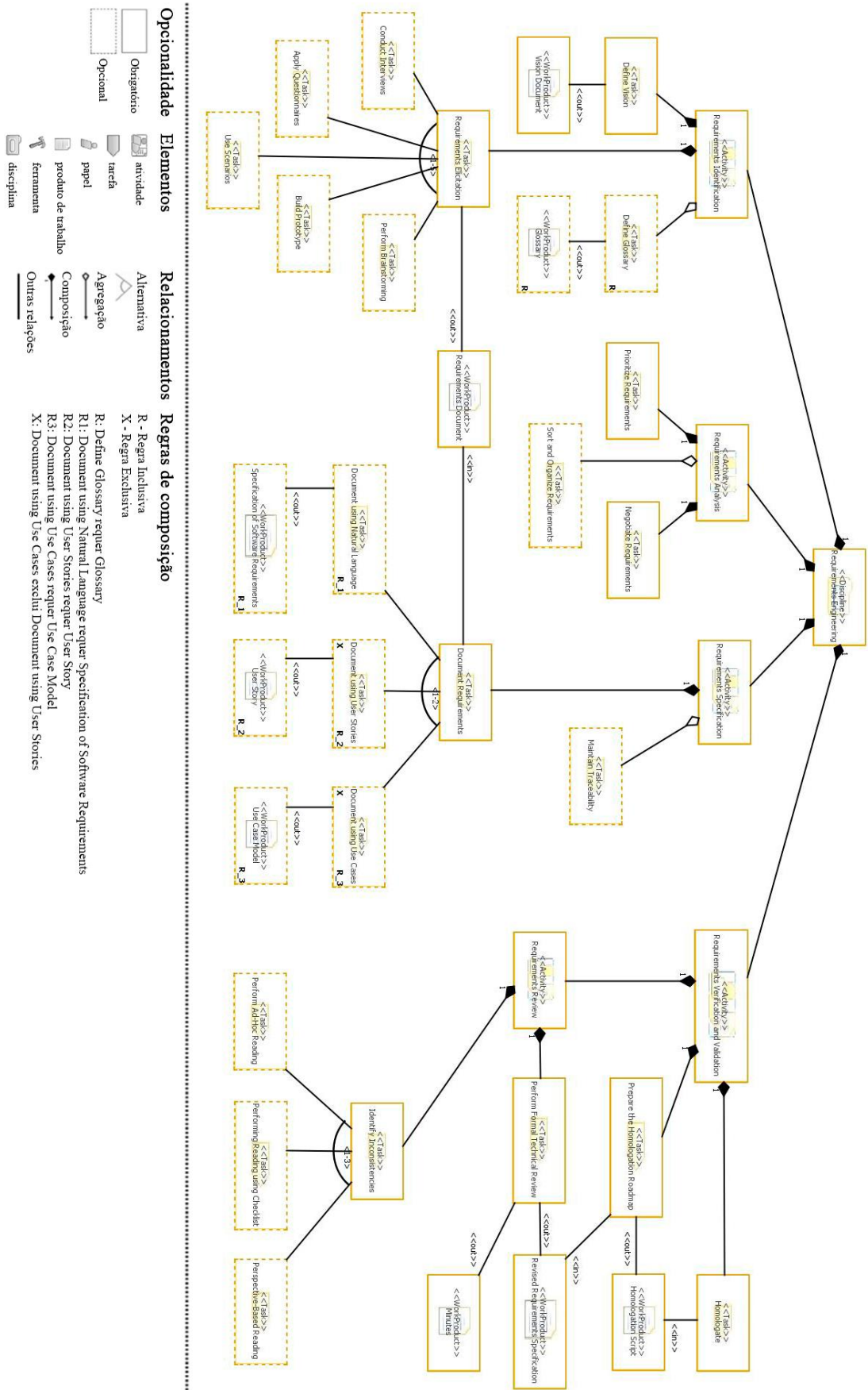
APÊNDICE D.3.A – Caracterização do Projeto SIRIUS

Cliente		Organização	
Envolvimento do Cliente	Moderado	Objetivo de Negócio	Conservador
Entendimento do Escopo	Parcial	Nível de Maturidade	Em processo de amadurecimento
Facilidade em Expressar Requisitos	Moderada	Cultura Organizacional	Tradicional
Produto		Formalidade da Estrutura	Formal
Classe de Software Organizacional	Diferenciado	Equipe	
Integração entre Componentes de Projeto	Moderada	Compartilhamento de Conhecimento	Moderado
Projeto		Tipo de Gerente de Projetos	Gerente
Complexidade do Produto	Alta	Distribuição Geográfica	Distribuída
Criticidade do Projeto	Muito Alta	Entendimento do Escopo	Parcial
Tamanho do Projeto	Grande	Tamanho da Equipe	Grande
Completeness do Escopo	Moderada	Experiência no Processo de Desenvolvimento	Baixa
Comunicação do Projeto	Alta	Experiência Técnica	Moderada
Nível de Risco	Negativamente Alto	Experiência no Domínio	Baixa
Originalidade do Projeto	Original	Experiência do Gerente de Projetos	Experiente
Duração do Projeto	Longo Prazo	Processo	
Exigência Contratual	Complexo	Flexibilidade de Adaptação do Processo Padrão	Moderada
Custo do Projeto	Muito Alto	Ferramenta(s) de Apoio	Semi-Automatizado
Esforço do Projeto	Alto		

APÊNDICE D.3.B – Caracterização do Projeto ORION

Cliente		Organização	
Envolvimento do Cliente	Alto	Objetivo de Negócio	Conservador
Entendimento do Escopo	Total	Nível de Maturidade	Em processo de amadurecimento
Facilidade em Expressar Requisitos	Alta	Cultura Organizacional	Tradicional
Produto		Formalidade da Estrutura	Formal
Classe de Software Organizacional	Comum	Equipe	
Integração entre Componentes de Projeto	Baixa	Compartilhamento de Conhecimento	Baixo
Projeto		Tipo de Gerente de Projetos	Líder
Complexidade do Produto	Alta	Distribuição Geográfica	Não Distribuído
Criticidade do Projeto	Muito Baixa	Entendimento do Escopo	Total
Tamanho do Projeto	Pequeno	Tamanho da Equipe	Muito Pequena
Completeness do Escopo	Alta	Experiência no Processo de Desenvolvimento	Moderada
Comunicação do Projeto	Baixa	Experiência Técnica	Alta
Nível de Risco	Positivamente Alto	Experiência no Domínio	Alta
Originalidade do Projeto	Comum	Experiência do Gerente de Projetos	Iniciante
Duração do Projeto	Curto Prazo	Processo	
Exigência Contratual	Simple	Flexibilidade de Adaptação do Processo Padrão	Moderada
Custo do Projeto	Baixo	Ferramenta(s) de Apoio	Semi-Automatizado
Esforço do Projeto	Baixo		

APÊNDICE D.3.C – Domínio de Engenharia de Requisitos



D.4 - Questionário de Avaliação

Código do participante: _____

Por favor, responda as questões deste formulário com base na experiência obtida ao realizar as tarefas propostas, considere a perspectiva de gerente de projetos apresentada no documento de tarefas.

[TREINAMENTO]

1 - O material de treinamento aplicado foi suficiente para realização das tarefas.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

[REUTILIZAÇÃO DE PROCESSOS DE SOFTWARE]

2 - O mecanismo de busca de similaridade oferece suporte satisfatório para apoiar a reutilização de processos de software definidos em contexto similar como ponto de partida para adaptação visando definir um processo de software específico de projeto.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

Justifique (opcional):

3 - O mecanismo de busca de similaridade oferece suporte satisfatório para visualizar as decisões tomadas em processos de software definidos em contexto similar visando apoiar na resolução de variabilidades durante a definição de processo de software específico de projeto através do mecanismo de recorte.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

Justifique (opcional):

[FERRAMENTAL DE SUPORTE]

4 - Aprender a utilizar a ferramenta Odyssey seria fácil para mim.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

5 - Eu acharia fácil usar a ferramenta Odyssey para definir processos de software.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

6 - Minha interação com a ferramenta Odyssey seria clara e compreensível.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

7 - Eu acharia a ferramenta Odyssey flexível para interagir.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

8 - Seria fácil para mim me tornar habilidoso na utilização da ferramenta Odyssey.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

9 - Eu acharia a ferramenta Odyssey fácil de usar.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

10 - Usar a ferramenta Odyssey no meu trabalho me permitiria realizar tarefas mais rapidamente.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

11 - Usar a ferramenta Odyssey melhoraria meu desempenho no trabalho.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

12 - Usar a ferramenta Odyssey no meu trabalho aumentaria minha produtividade.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

13 - Usar a ferramenta Odyssey aumentaria minha eficácia no trabalho.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

14 - Usar a ferramenta Odyssey tornaria mais fácil fazer o meu trabalho.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

15 - Eu acharia a ferramenta Odyssey útil no meu trabalho.

Concordo plenamente Concordo parcialmente Indiferente Discordo parcialmente Discordo plenamente

Fique à vontade para ressaltar pontos positivos e negativos, bem como propor melhorias para incrementar o suporte da abordagem:

Resposta (opcional):