

CLEITON DOS SANTOS GARCIA

PROPOSTA DE UMA LINHA DE PROCESSO DE
SOFTWARE PARA DESENVOLVIMENTO DE
APLICAÇÕES USANDO SOA E BPM

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná para obtenção do título de Mestre em Informática.

Curitiba
2012

CLEITON DOS SANTOS GARCIA

PROPOSTA DE UMA LINHA DE PROCESSO DE
SOFTWARE PARA DESENVOLVIMENTO DE
APLICAÇÕES USANDO SOA E BPM

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná para obtenção do título de Mestre em Informática.

Área de concentração: Ciência da Computação

Orientador: Prof. Dra. Sheila Reinehr

Curitiba
2012

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

G216p 2012	<p>Garcia, Cleiton dos Santos Proposta de uma linha de processo de software para desenvolvimento de aplicações usando SOA e BPM / Cleiton dos Santos Garcia ; orientador, Sheila Reinehr. – 2012. 164 f. : il. ; 30 cm</p> <p>Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2012 Bibliografia: f. 140-149</p> <p>1. Informática. 2. Arquitetura orientada a serviços (Computador). 3. Software. 4. Negócios. 5. Tecnologia da informação. I. Reinehr, Sheila. II. Pontifícia Universidade Católica do Paraná de Pós-Graduação em Informática. III. Título.</p> <p>CDD 20. ed. – 004.068</p>
---------------	--

DEDICATÓRIAS

Aos meus amigos pelo apoio e a minha melhor amiga e
esposa Katrin Grützmacher.

AGRADECIMENTOS

Primeiramente a Deus, o criador de todas as coisas, e ao seu filho Jesus Cristo pelas bênçãos e socorro nos momentos difíceis.

A minha esposa, Katrin Grützmacher, pelo apoio, compreensão e por ter suportado este período de ausência.

A minha querida mãe, Cleni Duarte dos Santos, e demais familiares pelo período de ausência com poucas visitas a Pelotas.

Aos meus sogros, Elmo e Iricina, pelo incentivo na realização deste trabalho.

Às professoras do grupo de pesquisa, Sheila e Andreia, pelo trabalho em conjunto, ensinamentos e experiência compartilhada.

Ao Departamento de Sistemas de Informação da WEG e a Católica de Santa Catarina que ofereceram condições para realização deste trabalho.

Aos membros do grupo de pesquisa pelas contribuições, críticas e apoio.

Ao departamento de Informática e Matemática Aplicada (DIMAp) da Universidade Federal do Rio Grande do Norte (UFRN), em especial pela contribuição do Uirá, Aleixo, Marília e Edmilson.

*“As palavras dos meus lábios e o meditar do
meu coração sejam agradáveis a tua presença,
SENHOR, rocha minha e redentor meu!”*

- Salmos 19:14

RESUMO

O gerenciamento de processos e sistemas é uma atividade complexa e morosa para as organizações e um desafio constante para a área de Tecnologia da Informação (TI). Entre as diferentes abordagens que buscam trazer agilidade aos processos de negócio e sistemas estão a Arquitetura Orientada a Serviços (SOA) e o Gerenciamento de Processos de Negócio (BPM). A abordagem SOA tem se popularizado entre as organizações, tratando de forma sistemática a disponibilização de componentes (serviços), oferecendo baixo acoplamento, interfaces bem definidas, e possibilitando a integração dos sistemas em plataformas heterogêneas e distribuídas. O BPM é uma abordagem essencial para que as organizações sejam capazes de sistematicamente realizar ciclos de identificação, monitoração, planejamento e execução das melhorias, controle e avaliação dos ganhos em seus processos. O objetivo desta pesquisa é propor uma linha de processo de software para tratar os métodos para SOA e BPM, de modo a simplificar o controle das variabilidades e possibilitar a derivação ágil de novos processos de desenvolvimento utilizando estas abordagens. Buscou-se ainda o desenvolvimento de um ambiente para suportar a linha de processo de software proposta, visando automatizar o processo. Quanto aos procedimentos técnicos, esta pesquisa se caracteriza como uma pesquisa de desenvolvimento, cuja avaliação dos resultados foi realizada utilizando opinião de especialistas. A pesquisa resultou na proposição de uma linha de processos de software voltada a derivar processos específicos para o desenvolvimento de aplicações usando SOA e BPM. Para apoiar a sua execução foi utilizado um ambiente computacional composto por ferramentas de mercado e por uma ferramenta desenvolvida para realizar a transformação do modelo UMA para a notação BPMN. A linha de processos foi avaliada por meio de doze especialistas provenientes do meio acadêmico e da indústria. A principal contribuição desta pesquisa foi a definição de uma linha de processo de software para a engenharia de produtos orientados a serviços e integrar os paradigmas e práticas envolvidas. A relevância para o mercado de TI é grande, pois a utilização de linhas de processos de software ainda é precária e carece de experimentos, práticas e ferramentas.

Palavras-chaves: Linha ou Família de processo de Software, Arquitetura Orientada a Serviços (SOA), Gerenciamento de Processos de Negócio (BPM).

ABSTRACT

The processes and systems management is a complex and time-consuming activity for organizations and also an ongoing Information Technology (IT) challenge. Among the different approaches for bringing flexibility to the business processes and systems there are the Service-Oriented Architecture (SOA) and Business Process Management (BPM). The SOA approach has become popular among organizations, addressing systematically the provision of components (services), offering loose coupling, well-defined interfaces, and enabling the integration of heterogeneous and distributed platforms. The BPM is an essential approach in order that the organizations are able to perform systematically cycles of identification, monitoring, planning and execution of improvements, control and evaluation of processes gains. This research aims at proposing a software process line to deal with SOA and BPM methods, in order to simplify variability control and enable the instantiation of new development process using such approaches. It also aims at developing an environment to support the proposed software process line, in order to automate the process. Regarding to the technical procedures, this research is characterized as a development research, which was evaluated using expert opinion. This research proposed a software line process focused on the generation of specific process to be used in the development of SOA and BPM applications. In order to support its execution it was proposed a computational environment composed by market tools and a tool specifically developed to perform the transformation of a UMA model into a BPM notation. The process line was assessed by twelve specialists coming either from academia and industry. Both approaches, SOA and BPM, are recent to the market and lack from well-defined development process and governance. The proposed line has explored the existent models for defining process lines that could improve the development of products of this nature and that could integrate paradigms, tools and practices. The main contribution of this work was the definition of the software process line to engineering service oriented products. The relevance for the IT market is huge, once the use of software process lines is still poor and lacks of experiments, practices and tools.

Keywords: Software Process Line or Family, Service-Oriented Architecture (SOA), Business Process Management (BPM).

SUMÁRIO

RESUMO.....	VII
ABSTRACT	VIII
LISTA DE FIGURAS	XII
LISTA DE TABELAS	XIV
LISTA DE ABREVIATURAS E SIGLAS	XV
CAPÍTULO 1 - INTRODUÇÃO	1
1.1 CONTEXTO.....	1
1.2 MOTIVAÇÃO	7
1.3 OBJETIVOS	8
1.4 PROCESSO DE TRABALHO	9
1.5 ESTRUTURA DO DOCUMENTO DA DISSERTAÇÃO	10
1.6 CONSIDERAÇÕES SOBRE O CAPÍTULO	10
CAPÍTULO 2 - REVISÃO DA LITERATURA	12
2.1 PROCESSOS DE ENGENHARIA DE SOFTWARE	12
2.1.1 Rational Unified Process	13
2.1.2 OpenUP.....	14
2.1.3 <i>Oracle Unified Method</i>	15
2.1.4 Accelerated SAP	15
2.1.5 Microsoft Solution Framework	16
2.2 LINHA DE PRODUTOS DE SOFTWARE.....	17
2.3 ENGENHARIA DE PROCESSO DE SOFTWARE.....	21
2.3.1 Notação para modelagem de processos de software	22
2.3.2 Ferramentas para modelagem de processo - <i>Eclipse Process Framework</i> 25	
2.3.3 Normas internacionais ISO/IEC 12207 e ISO/IEC 15504	27
2.3.4 CMMi.....	30
2.3.5 MPS.BR.....	31
2.3.6 Reutilização dos processos de software.....	32
2.4 LINHAS DE PROCESSO DE SOFTWARE	33
2.5 ARQUITETURA ORIENTADA A SERVIÇOS.....	35

2.5.1	Camadas de SOA.....	37
2.5.2	Projeto de serviços	39
2.5.3	Web Services	40
2.5.4	Segunda geração dos Web Services (WS-*)	44
2.5.5	REpresentational State Transfer - REST	44
2.6	BPM – BUSINESS PROCESS MANAGEMENT	45
2.7	MODELOS DE PROCESSOS PARA SOA E BPM	47
2.7.1	Modelos de maturidade para SOA.....	49
2.7.2	Guia para métodos para engenharia de software orientada a serviços ...	53
2.8	CONSIDERAÇÕES SOBRE O CAPÍTULO	56
CAPÍTULO 3 - ESTRUTURAÇÃO DA PESQUISA.....		57
3.1	METODOLOGIA E MÉTODOS DE PESQUISA.....	57
3.2	CARACTERIZAÇÃO DA PESQUISA.....	61
3.3	QUESTÃO DE PESQUISA.....	61
3.4	ESTRATÉGIA DE PESQUISA	62
3.4.1	Etapa 1 – Compreensão sobre as áreas de conhecimento.....	63
3.4.2	Etapa 2 – Identificação das variabilidades	63
3.4.3	Etapa 3 – Concepção da linha de processo.....	63
3.4.4	Etapa 4 – Definição de ferramentas para a linha de processo	63
3.4.5	Etapa 5 – Avaliação da linha de processos por especialistas.....	64
3.4.6	Etapa 6 - Análise dos resultados e revisão da linha	68
3.5	CONSIDERAÇÕES SOBRE O CAPÍTULO	69
CAPÍTULO 4 - DEFINIÇÃO DA LINHA DE PROCESSOS PARA SOA E BPM.....		70
4.1	INTRODUÇÃO	70
4.2	CONTEÚDO DO MÉTODO SOPLA.....	74
4.3	PROCESSO DA LINHA	78
4.3.1	Fase de Modelagem de Negócios e Transformação	80
4.3.2	Fase de Identificação e Definição da Arquitetura.....	85
4.3.3	Fase de Especificação e Projeto dos Serviços	88
4.3.4	Fase de Construção, Composição e Testes	95
4.3.5	Liberação, Execução e Monitoração	104
4.4	EXEMPLO DE SELEÇÃO DAS VARIABILIDADES.....	105
4.5	APOIO COMPUTACIONAL À LINHA DE PROCESSO PROPOSTA.....	108
4.6	CONSIDERAÇÕES SOBRE O CAPÍTULO	115

CAPÍTULO 5 - AVALIAÇÃO DA LINHA DE PROCESSOS PROPOSTA	116
5.1 CARACTERIZAÇÃO DOS ESPECIALISTAS PARA O ESTUDO.....	117
5.2 ANÁLISE DOS RESULTADOS	118
5.2.1 Análise da Visão Geral das Fases.....	118
5.2.2 Análise da Fase de Modelagem de Negócios e Transformação	120
5.2.3 Análise da Fase de Identificação e Definição da Arquitetura	123
5.2.4 Análise da Fase de Especificação e Projeto dos Serviços	125
5.2.5 Análise da Fase de Construção, Composição e Testes	127
5.3 SUMÁRIO DAS AVALIAÇÕES DOS ESPECIALISTAS	130
5.4 ANÁLISE DAS PROPOSIÇÕES INICIAIS	133
5.5 CONSIDERAÇÕES SOBRE O CAPÍTULO	135
CAPÍTULO 6 - CONSIDERAÇÕES FINAIS	136
6.1 RELEVÂNCIA DO ESTUDO.....	136
6.2 CONTRIBUIÇÕES DA PESQUISA.....	137
6.3 LIMITAÇÕES	138
6.4 TRABALHOS FUTUROS	138
6.5 MENSAGEM FINAL	139
REFERÊNCIAS BIBLIOGRÁFICAS	140
GLOSSÁRIO	150
APÊNDICE A – PROTOCOLO DE PESQUISA – CARTA DE APRESENTAÇÃO .	153
APÊNDICE B – PROTOCOLO DE PESQUISA – TERMO DE CONFIDENCIALIDADE	154
APÊNDICE C – FORMULÁRIO DE CARACTERIZAÇÃO DO ESPECIALISTA	155
APÊNDICE D – FORMULÁRIO DE AVALIAÇÃO DAS FASES DA LINHA	158

LISTA DE FIGURAS

Figura 1-1. Abordagem de reúso de software, adaptado de (SOMMERVILLE, 2007).	2
Figura 2-1. Matriz de priorização, adaptado de (TURNER, 2006).	17
Figura 2-2. Fluxo entre os procesos de engenharia de domínio e aplicação, adaptado de (GOMAA, 2004).	18
Figura 2-3. SPLE Framework, adaptado de (POHL; BÖCKLE; LINDEN, 2005).	19
Figura 2-4. Funcionalidades do Smartphone no FeatureIDE ³ (PERROUIN, 2011)	20
Figura 2-5. Funcionalidades do Smartphone no Pure::Variants (PERROUIN, 2011)	21
Figura 2-6. Áreas de conhecimento da engenharia de processo de software, adaptado de (IEEE, 2004).	22
Figura 2-7. Elementos básicos do SPEM, adaptado de (OMG, 2008a).	23
Figura 2-8. Elementos do vSPEM, adaptado de (MARTÍNEZ-RUIZ et al., 2011).	25
Figura 2-9. Etapas de edição de um processo no EPF (EPF, 2012).	26
Figura 2-10. Grupos de processos no ciclo de vida, de acordo com a ISO/IEC 12207 (ISO/IEC, 2008).	28
Figura 2-11. Contexto de utilização da ISO-15504, adaptado de (SQI, 2011).	29
Figura 2-12. Modelos do CMMi por área, adaptado de SEI.	30
Figura 2-13. Modelo MPS.BR, adaptado de (SOFTEX, 2011).	31
Figura 2-14. Evolução dos paradigmas e nível de reúso, adaptado de (LINDEN, 2002).	36
Figura 2-15. Linha do tempo SOA, adaptado de (ROSEN et al., 2008).	37
Figura 2-16. Camadas SOA, adaptado de (BIEBERSTEIN et al., 2008)(ARSANJANI et al., 2008).	38
Figura 2-17. Fluxo de mensagens para registrar e consumir serviços, adaptado de (ERL, 2009).	43
Figura 2-18. Fases do método WOA, adaptado de (THIES; VOSSEN, 2009).	48
Figura 2-19. SOPLE-DE Papéis e Atividades, adaptado de (MEDEIROS; ALMEIDA; MEIRA, 2010).	49
Figura 2-20. Níveis de maturidade SOA e benefícios (SOAMM, 2005).	50
Figura 2-21. Níveis de maturidade (RATHFELDER; GROENDA, 2008).	52
Figura 3-1. Classificação das pesquisas com base nos procedimentos técnicos, adaptado de (GIL, 2002).	59
Figura 3-2. Etapas da pesquisa. Fonte: o Autor, 2012.	62
Figura 3-3. Interpretação dos resultados. Fonte: o Autor, 2012.	69
Figura 4-1. Simbologia SPEM utilizada, adaptado de (OMG, 2008a).	72
Figura 4-2. Tipos de variabilidades. Fonte: o Autor, 2012.	73
Figura 4-3. Fases do SOPLA. Fonte: o Autor, 2012.	79

Figura 4-4. Fase de Modelagem de Negócios e Transformação. Fonte: o Autor, 2012.	81
Figura 4-5. Fase de Identificação e Definição da Arquitetura. Fonte: o Autor, 2012.	85
Figura 4-6. Fase de Especificação e Projeto dos Serviços. Fonte: o Autor, 2012.	89
Figura 4-7. Fase de Construção, Composição e Testes. Fonte: o Autor, 2012.	96
Figura 4-8. Atividades e ferramentas no apoio computacional à linha de processo, adaptado de (ALEIXO et al, 2011).	109
Figura 4-9. EPF com as atividades da fase de Construção, Composição e Testes. Fonte: o Autor, 2012.	111
Figura 4-10. Uma2Bpmn Eclipse <i>Plugin</i>	112
Figura 4-11. Arquivo BPMN resultante da transformação. Fonte: o Autor, 2012.	113
Figura 4-12. Diagrama gerado em BPMN. Fonte: o Autor, 2012.	113
Figura 4-13. Ajustes visuais no editor de BPMN. Fonte: o Autor, 2012.	114
Figura 4-14. Carregar o processo na ferramenta de BPM. Fonte: o Autor, 2012.	115
Figura 5-1. Avaliação da completude das fases do SOPLA. Fonte: o Autor, 2012.	119
Figura 5-2. Avaliação das sugestões das Fases do SOPLA. Fonte: o Autor, 2012.	120
Figura 5-3. Avaliação da completude da Fase de Modelagem de Negócios e Transformação. Fonte: o Autor, 2012.	121
Figura 5-4. Avaliação das sugestões para Fase de Modelagem de Negócios e Transformação. Fonte: o Autor, 2012.	123
Figura 5-5. Avaliação da completude da Fase de Identificação e Definição da Arquitetura. Fonte: o Autor, 2012.	124
Figura 5-6. Avaliação das sugestões para Fase de Identificação e Definição da Arquitetura. Fonte: o Autor, 2012.	125
Figura 5-7. Avaliação da completude da Fase de Especificação e Projeto dos Serviços. Fonte: o Autor, 2012.	126
Figura 5-8. Avaliação das sugestões para Fase de Especificação e Projeto dos Serviços. Fonte: o Autor, 2012.	127
Figura 5-9. Avaliação da completude da Fase de Construção, Composição e Testes. Fonte: o Autor, 2012.	128
Figura 5-10. Avaliação das sugestões para Fase de Construção, Composição e Testes. Fonte: o Autor, 2012.	129

LISTA DE TABELAS

Tabela 2-1. Partes da Norma ISO/IEC 15504.	28
Tabela 2-2. Open Group Service Integration Maturity Model, adaptado de (OSIMM, 2009).	50
Tabela 2-3. Comparação dos aspectos genéricos em métodos SOA – parte 1, adaptado de (GU; LAGO, 2011).	54
Tabela 2-4. Comparação dos aspectos genéricos em métodos SOA – parte 2, adaptado de (GU; LAGO, 2011).	54
Tabela 2-5. Atividades avaliadas nos métodos para SOA (GU; LAGO, 2011).	55
Tabela 3-1. Classificação da pesquisa.	61
Tabela 4-1. Classificação de tamanho de projeto.	73
Tabela 4-2. Atividades para fase de Modelagem de Negócios e Transformação	82
Tabela 4-3. Atividades para fase de Identificação e Definição da Arquitetura	86
Tabela 4-4. Atividades para fase de Especificação e Projeto dos Serviços	90
Tabela 4-5. Atividades para fase de Construção, Composição e Testes.	97
Tabela 5-1. Resumos do perfil dos especialistas.	117
Tabela 5-2. Sumário dos resultados das avaliações dos especialistas.	130

LISTA DE ABREVIATURAS E SIGLAS

BAM	<i>Business Activity Monitoring</i>
BPM	<i>Business Process Management</i>
BPMS	<i>Business Process Management Systems</i>
CBD	<i>Component-Based Development</i>
CMMI	Capability Maturity Model Integration
EPF	<i>Eclipse Process Framework</i>
ERP	<i>Enterprise Resource Planning</i>
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LPS	Linha de Produtos de Software
MPS.BR	Melhoria de Processo do Software Brasileiro
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
OMG	<i>Object Management Group</i>
OSIMM	<i>Open Group Service Integration Maturity Model</i>
PA	Process Area
PL	Product Line
QPI	<i>Quality Improvement Paradigm</i>
SEI	Software Engineering Institute
SOAMM	<i>Service-Oriented Architecture Maturity Model</i>
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SPEM	Software Process Engineering Metamodel
SPL	<i>Software Product Line</i>

SWEBOK	<i>Software Engineering Body of Knowledge</i>
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>
WOA	<i>Web-Oriented Architecture</i>
XML	eXtensible Markup Language
XP	eXtreme Programming

CAPÍTULO 1 - INTRODUÇÃO

Cada módulo é caracterizado por seu conhecimento (...) o qual este esconde de todos os outros. Sua interface é escolhida para revelar o mínimo possível sobre seu funcionamento interno.
– David Lorge Parnas, cientista da computação americano

A área da engenharia de software tem realizado diversas pesquisas que buscam trazer maior velocidade e menor custo para o desenvolvimento e manutenção de software. Entre as diversas abordagens têm se investido em processos de desenvolvimento de software que promovam o reúso dos artefatos de documentação e de software. O *Software Engineering Body of Knowledge* (SWEBOK) define o reúso de software como um fator chave para a manutenção e melhoria de produtividade e competitividade (IEEE, 2004). Já para (ENDRES; ROMBACH, 2003) “a promessa da reutilização é transmitida pela lei: Reutilização de software reduz o tempo de ciclo de desenvolvimento e aumenta a produtividade e a qualidade”.

1.1 Contexto

A Figura 1-1, resume a classificação das abordagens que buscam promover o reúso de software (SOMMERVILLE, 2007). Embora exista um número considerável de abordagens, o reúso de software é difícil de ser alcançado, pois a implantação desta prática requer uma visão estratégica e coordenação de esforços para gerar o poder único desta técnica (IEEE, 2004).

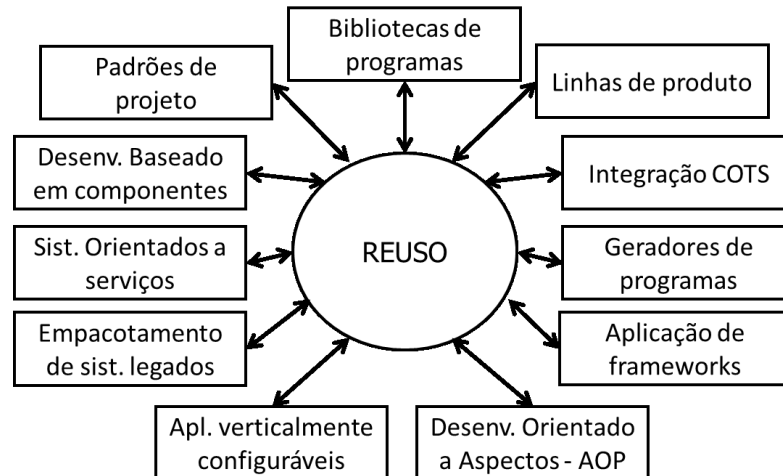


Figura 1-1. Abordagem de reuso de software, adaptado de (SOMMERVILLE, 2007).

Diversos métodos de desenvolvimentos de software surgiram para promover o reuso, entretanto, ainda hoje este é um objetivo de difícil sistematização. Em meados dos anos 80, a visualização *top down* da análise estruturada foi aprimorada, minimizando a redundância e incluindo orientação a eventos (YOURDON, 1992). A análise essencial enfatizou a identificação de pontos comuns de funcionalidade e estruturas de dados e a divisão em caixas conforme o comportamento dos módulos (PALMER; MCMENAMIN, 1991). No método estruturado, ainda como prática informal, já era possível o reuso de funções, procedimentos e módulos.

A partir da década de 80, com a ascensão do paradigma orientado a objetos, o reuso no desenvolvimento de software passou a ter maior ênfase. Foram propostas diversas abordagens e métodos voltados ao desenvolvimento orientado a objetos, mas, somente após a fusão das principais abordagens de modelagem na linguagem *Unified Modeling Language* (UML) é que houve uma maior aceitação e disseminação. Em 1997 a UML foi incorporada pela *Object Management Group* (OMG) como proposta de padrão de linguagem de modelagem. Desde então, este grupo tem dado continuidade à evolução da linguagem (FURLAN, 1998).

O software orientado a objetos geralmente é empacotado de forma estática para implantação gerando um módulo, framework, ou componente. Segundo (SZYPERSKI, 2002), objetos dificilmente são vendidos, comprados, ou implantados, enquanto os componentes têm facilitado com êxito a comercialização. Conforme Udell (1994) apud (SZYPERSKI, 2002): “A Orientação a Objetos falhou na promoção do reuso, mas a componentização está tendo sucesso”.

Visto que a orientação a objetos por si só não conseguiu estabelecer o reúso esperado, buscou-se gerenciar o reúso de software por meio de uma granularidade maior, o que deu espaço aos componentes. Entretanto, a ideia de componentes de software não é nova, pois surgiu há mais de 40 anos. Em 1968 na *NATO Software Engineering Conference*, McIlroy propôs que os componentes de software deveriam ser produzidos em massa. A abordagem de componentes foi vista como uma alternativa para a chamada “crise do software” (“*software gap/crisis*”), onde discutiu-se os fatores relacionados a: complexidade, conhecimento, controle de custos e prazo (NAUR; RANDELL, 1969). Em seu artigo (MCILROY, 1968) observou que: toda indústria madura é baseada em componentes, uma vez que estes permitem gerir processos (sistemas) de grande porte; os componentes devem ser produzidos em massa; e, por fim, montados para compor os produtos (sistemas). Conforme (SZYPERSKI, 2002) um componente de software caracteriza-se por: ser uma unidade de composição; ser uma unidade independente de implantação geralmente binária (sem código fonte); estar sujeita à composição por terceiros; possuir contratos de interfaces especificadas e somente dependências explícitas; e não possuir um estado observável externo.

Apesar do estabelecimento de componentes em todas as outras disciplinas de engenharia, até a década de 90 esta abordagem ainda não era bem sucedida na área de software (SZYPERSKI, 2002). O sucesso do desenvolvimento baseado em componentes (CBD – *Component-based development*) somente pôde ser alcançado com a adoção de métodos sistemáticos, como nas famílias de programas ou sistemas (PARNAS, 1976)(PARNAS, 1979), ou na engenharia de domínio introduzida por (NEIGHBORS, 1989) e mais tarde também desenvolvidas no *Software Engineering Institute* (SEI).

O poder do reúso tem sido aplicado há décadas na manufatura, inclusive na produção de hardware. O desenvolvimento de sistemas embarcados há muitos anos tem aplicado o reúso por meio de programas de reúso, em abordagens europeias e americanas. Conforme (PRIETO-DIAZ; 1991), um programa de reuso é uma estrutura organizacional e ferramentas que visem fomentar, gerenciar e manter a prática de um reúso. Nos EUA, destaca-se a adoção do programa do SEI, denominado *Software Product Line* (SPL) e traduzido aqui como Linha de Produto de Software. Ao final da década de 90, a SPL trouxe uma abordagem mais madura de reúso sistematizado. Uma SPL pode ser definida como um conjunto de sistemas

que compartilham um conjunto de características e funcionalidades comuns e gerenciadas, para satisfazer um determinado segmento de mercado com base em um conjunto comum de ativos de software (CLEMENTS; NORTHROP, 2002).

A partir de 1990, as fusões e aquisições de empresas começaram a ocorrer em dimensões cada vez maiores e em diversos setores. Este fator agravou problemas de integração entre soluções de TI incompatíveis, mas que precisavam ser integradas (BIEBERSTEIN et al., 2008). Pacotes padronizados, como *Enterprise Resource Planning* (ERP), foram propostos como uma solução a este problema. Do ponto de vista do fornecedor da solução, isto representava um ganho em escala, uma vez que o mesmo sistema é customizado para diversos clientes. Entretanto, sob a ótica das empresas adquirentes, os projetos de implantação de ERP representam um risco para operação da empresa, devido ao custo, prazo, modificação nos processos e grande volume de customizações inadequadas. O problema de integração também foi potencializado quando os fornecedores de software passaram a adquirir seus competidores.

A indústria de software passou a dedicar recursos para evolução das plataformas de sistemas distribuídos, integrações e padrões. Soluções como o *Enterprise Application Integration* (EAI) surgiram como uma proposta no suporte a integrações ponto a ponto de aplicações dedicadas (BIEBERSTEIN et al., 2008). A necessidade de padronização foi agravada com o advento da Internet e a utilização desta rede para comunicação dos processos de negócios de diversas organizações com seus parceiros de negócio.

Dentro deste contexto surgiu o termo SOA (*Service-Oriented Architecture*) em 1996, o qual tem sido fortemente discutido pela indústria e academia. As organizações vêm atuando de forma crescente na adoção de uma Arquitetura Orientada a Serviços na qual o reúso passa a atuar no nível de serviços que mapeiam os processos de negócio. O objetivo principal da arquitetura SOA é ligar o mundo dos negócios ao mundo da Tecnologia da Informação – TI, de forma a tornar ambos mais eficientes (BROWN et al., 2009). A ideia de SOA é buscar a padronização dos componentes conforme a necessidade de negócio, denominando-os de serviços, os quais possuem: infraestrutura autônoma; independência de linguagem e plataforma; são flexíveis e facilmente consumidos e combináveis (BIEBERSTEIN et al., 2008).

Arquiteturas orientadas a serviços encapsulam as funcionalidades, aprimorando a integração de sistemas e oferecendo visibilidade aos processos de negócios. Para coordenar a execução de sistemas baseados em SOA, o *Business Process Management* (BPM) possui um importante papel. BPM estabelece o suporte aos processos de negócio por meio da utilização de métodos de gestão, técnicas e software para desenhar, estabelecer, controlar e analisar a execução operacional do processo envolvendo pessoas, organizações, aplicações, documentos e outras fontes de informações (VAN DER AALST et al., 2003). Com a aplicação de SOA, as atividades dos processos de negócio são expostas como serviços e os sistemas de BPM orquestram ou coordenam o fluxo de execução das atividades humanas e automatizadas. Uma das tarefas principais do BPM é apoiar na análise e otimização dos processos, dando apoio às operações paralelas, na colaboração e distribuição dos processos de negócio em atividades dentro e fora da organização (ZHANG et al., 2010).

Apesar da expectativa gerada pelas abordagens e técnicas de BPM, SOA, CBD e OO, a qualidade do produto de software também depende das pessoas e das práticas envolvidas no processo de desenvolvimento. Visando responder a este problema, foram propostos padrões e modelos para definição, melhoria e avaliação de processos de software, tais como: ISO 9001, *Capability Maturity Model* (CMM), *Quality Improvement Paradigm* (QPI), ISO/IEC 12207, ISO/IEC 15288, ISO/IEC 15504, entre outras (SILVIA; JURISTO, 2005).

Visando aplicar os conceitos de reúso ao universo dos processos de software, a técnica de linha de produto de software foi adaptada e proposta para a engenharia de processos de software. Em 1996 surgiram as primeiras publicações sobre famílias ou linhas de processos, onde se observou que, assim como famílias de produtos, os processos de software também poderiam ser analisados e criados em famílias (SUTTON; OSTERWEIL, 1996), visando reduzir o esforço na sua definição e implantação. Após uma década, os trabalhos começaram a se intensificar em busca de trazer as técnicas, mecanismos e ferramentas para gerenciar a variabilidade no processo de software. Os principais benefícios trazidos por esta abordagem são a facilidade de adaptação do processo e a facilidade de aprendizado entre os processos dos diversos projetos de software (ROMBACH, 2005).

As linhas de processos de software atendem a necessidades de customização dos processos de desenvolvimento de software das organizações.

Seguidamente, as organizações possuem necessidades que requerem adaptações no processo de software, mas também existem razões convincentes para manter um processo padronizado. De acordo com (HUMPHREY; KELLNER, 1989), são elas:

- A padronização do processo é requerida para permitir treinamento, gerenciamento, revisões e o suporte de ferramentas;
- Utilizando-se de métodos padronizados, a experiência de cada projeto pode contribuir com melhorias no processo da organização;
- Prover uma estrutura básica para a medição;
- Facilitar o gerenciamento do processo;
- A definição de processos leva um grande tempo e esforço o que torna impraticável novas definições de processo para cada projeto;
- As tarefas básicas são comuns à maioria dos projetos, sendo que o modelo padronizado do processo precisa de pequenas customizações para atender às necessidades especiais de cada projeto.

Em SOA, a customização e o gerenciamento do processo de software têm sido um problema, especialmente devido à produção de diferentes artefatos entre os provedores e consumidores e suas camadas. Apesar de SOA ter se popularizado com o crescimento de serviço na WEB, nem sempre há um processo de software institucionalizado nas organizações. Para se obter sucesso no desenvolvimento de software baseado em SOA é necessário dar ênfase em algumas atividades do processo, tais como:

- A descoberta dos serviços é a atividade mais importante e, para esta ser efetiva, é preciso um mecanismo de busca que identifique os serviços a partir dos requisitos do usuário e do negócio (SPANOUidakis, 2005);
- Promover a análise dos processos e serviços existentes e quando estes não atenderem, estabelecer um processo sistemático para a criação de novas versões ou novos serviços compostos;
- Fortalecer a visão de processos de negócios com medição e orquestração; e,
- Um processo iterativo para refinar a especificação de requisitos (ZACHOS et al., 2006)(ZACHOS et al., 2007).

A proposta deste trabalho de pesquisa é a **definição de uma linha de processos de software voltado para SOA e BPM, a qual deve possuir**

variabilidades para apoiar o desenvolvimento de aplicações e serviços compostos a partir dos serviços existentes.

1.2 Motivação

O mercado brasileiro ocupa a 11^a posição no mercado mundial de software e serviços. Este segmento movimentou cerca de U\$ 19,4 bilhões em 2010, representando um avanço de 24% em relação ao ano anterior, de acordo com a ABES (ABES, 2011). Neste cenário, destaca-se o avanço de 15,7% das exportações, montante equivalente a U\$ 1,74 bilhões. Quanto à venda de licenças de software obteve-se um crescimento de 12%. A expectativa para 2012 é favorável, considerando: a expansão da computação em nuvem, o crescimento da procura por aplicativos de análise e de inteligência para os processos de negócio, o aumento nas vendas de dispositivos móveis e o advento da TV Digital (ABES, 2011).

Em contrapartida, os projetos de TI possuem um histórico de muitos fracassos com taxas de risco muito altas. O *Standish Group* coleta informações e métricas dos projetos de TI e publica periodicamente um relatório conhecido como *Chaos Report*. No ano de 2009 somente 32% dos projetos foram considerados de sucesso, enquanto 68% atrasaram e/ou ultrapassaram o orçamento ou foram cancelados (STANDISH GROUP, 2009).

A indústria de manufatura tem aplicado com sucesso técnicas de linhas de montagem, especialização de atividades e divisão do produto em partes menores (componentes). Estas práticas de sucesso têm sido base para pesquisa de melhorias onde o propósito é alcançar parte desta engenharia e reutilização planejada para o processo de software.

Contudo, o processo de software requer adaptações devido às peculiaridades de alguns projetos, pois eventualmente, as mudanças geram problemas para a organização. Algumas vezes as adaptações tomam proporções que não se tornar aceitável dizer que os projetos utilizam o mesmo processo de desenvolvimento. Alguns dos problemas gerados por estas adaptações são, de acordo com (ROMBACH, 2005):

- Carência de integração com outros processos da organização, tais como: aquisição, compras, controle de custos e prazos, engenharia e desenvolvimento de produto;

- Pouca orientação para guiar os desenvolvedores em suas atividades de trabalho ou guias muito genéricos;
- Problemas na medição devido à falta de adesão ao processo; e
- Dificuldade de propagar o aprendizado dos projetos até ao nível do processo corporativo, onde espera-se que outras equipes de projeto também aprendam.

Para organizações que desenvolvem software em unidades ou filiais geograficamente distribuídas, os problemas gerados pelas adaptações no processo de software é potencializado, assim como a dificuldade de controle do processo, métrica e integração de atividades. A terceirização de parte do processo de desenvolvimento também requer um esforço adicional para planejar, controlar, gerenciar a qualidade e contratos. Também em filiais centralizadas, quando se opta pela divisão do departamento em equipes de desenvolvimento, problemas como estes podem surgir em menor proporção. Mesmo divididas, estas equipes carecem de definições de melhoria contínua de seus processos para promover a produtividade e qualidade no desenvolvimento de software.

Assim como os grandes fornecedores de software estão construindo suas soluções baseadas em processos e atividades que são mapeados em serviços de negócios através de SOA e BPM, os desenvolvimentos em diversas organizações também poderiam usufruir destas abordagens. Os sistemas baseados nestas abordagens e tecnologias resultam em arquiteturas ágeis, alinhadas ao processo de negócio, flexíveis e adaptáveis às mudanças do negócio (ROSEN et al., 2008).

1.3 Objetivos

Visto o contexto de cada organização e as diferentes necessidades de cada projeto, observa-se a busca de adaptações e melhorias do processo de engenharia de software. Apesar de existirem diversos métodos de engenharia de software, os quais já foram comparados por (GU; LAGO, 2011), não são tratados aspectos de variabilidades no processo de software.

O objetivo geral desta pesquisa é **desenvolver uma linha de processos de software adaptada para o desenvolvimento de aplicações que utilizam BPM e SOA**. Para tanto, considera-se a aplicação da abordagem de linha de processos de software para gerenciar sistematicamente os pontos de variação e as variabilidades, objetivando acelerar a derivação de novos processos a partir desta linha. Através da

aplicação desta abordagem, será concebida uma linha de processos com ênfase no gerenciamento de processos de negócio (BPM) e desenvolvimento de sistemas orientados a serviços (SOA e WOA).

Para atingir o objetivo geral deste trabalho, os seguintes objetivos específicos serão atendidos:

- (I) Analisar os métodos para engenharia de software orientada a serviços e selecionar os elementos fundamentais, como: papéis, atividades, artefatos e ferramentas.
- (II) Construir a linha de processos de software, identificando as variabilidades de seus elementos.
- (III) Estabelecer um ferramental para controlar a execução do processo suportando as variabilidades e que incorpore mudanças no processo de desenvolvimento.
- (IV) Avaliar a linha proposta.

A contribuição deste trabalho é a definição de variações no processo de software, definidas em uma linha de processo de software para o desenvolvimento de soluções de negócio baseadas no paradigma de orientação a serviços. Esta abordagem difere-se dos modelos de maturidade SOA existentes (SOAMM, 2005) (OSIMM, 2009) (RATHFELDER; GROENDA, 2008) (SÖDESTRÖM; MEIER, 2007) e dos frameworks de processos de software SOA por definir o processo de software e as suas variabilidades em seus elementos, ou seja, analisando as características variáveis, semelhanças e diferenças dos elementos.

A questão principal desta pesquisa pode ser representada através da seguinte frase: **“É viável apoiar o desenvolvimento de aplicações SOA e BPM utilizando um processo de desenvolvimento de software instanciado a partir de um conjunto de variabilidades de uma linha de processos?”**.

1.4 Processo de trabalho

Nesta seção são descritas as atividades iniciais para a elaboração da pesquisa. Estas atividades foram organizadas em fases com o objetivo de estabelecer metas e controlar o resultado esperado.

- Fase 1 – Preparação da Pesquisa: fase que corresponde à delimitação da área de estudo, coleta e análise das referências bibliográficas, definição do tema e estabelecimento dos objetivos, questões e proposições.

- Fase 2 – Estruturação da Pesquisa: fase de elaboração de um quadro referencial teórico, seleção do método de pesquisa e definição das etapas de pesquisa.
- Fase 3 – Execução da Pesquisa: fase da investigação em si, iniciando pela coleta de dados até a análise dos principais modelos existentes na literatura e indústria para SOA e BPM. Nesta fase também será realizado o desenvolvimento inicial da linha de processos de software e incluído o gerenciamento das variabilidades conforme maturidade e tecnologias selecionadas.
- Fase 4 - Análise dos Resultados: fase da análise dos resultados extraindo as generalizações e conclusões.

1.5 Estrutura do documento da dissertação

Este Capítulo 1 apresentou uma visão geral da proposta de trabalho contextualizando o leitor sobre a relação com algumas áreas da engenharia de software. Apresentou a motivação, objetivo geral e específico, assim como a estruturação do processo de trabalho.

O Capítulo 2 apresenta o referencial teórico, focando os temas de processo de software, notação para modelagem de processos de software e SOA.

O Capítulo 3 estabelece o posicionamento metodológico, definindo detalhes da estrutura da pesquisa.

O Capítulo 4 descreve a concepção da linha de processo com alguns resultados preliminares obtidos na condução dos trabalhos de pesquisa.

O Capítulo 5 avalia a proposta do trabalho com especialistas da área.

Os Apêndices B até E apresentam as partes do protocolo de pesquisa e os formulários de avaliação da linha proposta.

1.6 Considerações sobre o capítulo

Este capítulo apresentou uma visão geral dos paradigmas de desenvolvimento de software e a importância da reutilização. Apresentou a orientação a serviço, onde o reúso ocorre em um nível granular compatível com as atividades dos processos de negócio. A arquitetura SOA e BPM tem aprimorado a medição, otimização, controle e suporte dos processos de negócios das organizações, mas existem diversos desafios. Esta arquitetura apresenta-se como

uma realidade sólida com diversos casos de sucesso na indústria. A definição de um processo de software bem definido e voltado ao reuso é fundamental para fomentar a utilização sistemática dos ativos existentes e consequente sucesso de SOA. A motivação, contribuição e os objetivos deste trabalho foram apresentados, bem como as limitações do escopo proposto e o processo de trabalho a ser utilizado para desenvolver esta dissertação.

CAPÍTULO 2 - REVISÃO DA LITERATURA

A experiência nunca é limitada e nunca é completa; ela é uma imensa sensibilidade, uma espécie de vasta teia de aranha, da mais fina seda, suspensa no quarto de nossa consciência, (...)
– Henry James, escritor norte-americano

Este capítulo descreve os conceitos relacionados ao processo de engenharia de software, os principais processos ou métodos existentes, definição e modelagem de processos, modelos de melhoria e avaliação. As Linhas de Produto de Software e Linhas de Processo de Software serão apresentadas. A Arquitetura Orientada a Serviços será abordada, assim como as propostas de métodos de desenvolvimento orientado a serviços disponíveis na academia e na indústria.

2.1 Processos de engenharia de software

O processo de engenharia de software possui um importante papel na construção de produtos de software e grande influência na sua qualidade. A ausência de um processo de software bem definido resulta em uma equipe trabalhando de modo empírico para gerar um produto, mas de maneira imprevisível e muitas vezes desordenada. Nesta situação o êxito do trabalho depende do esforço dos indivíduos, que muitas vezes tornam-se heróis pelo resultado e sucesso de seu trabalho. Entretanto, esta não é uma condição sustentável para a organização (KRUCHTEN, 2003).

Organizações maduras empregam processos bem definidos e que suprem as suas necessidades para o desenvolvimento de sistemas de modo consistente e independente do indivíduo que os produziu. A alta rotatividade das pessoas é uma realidade de mercado, devido à oferta aos profissionais de TI. A dependência de profissionais tem sido um problema para as organizações. Os produtos de software precisam continuar o seu ciclo de vida mesmo que os desenvolvedores sejam substituídos, esta preocupação deve ser agravada para os sistemas que suportam os processos críticos de negócio,.

A definição de um processo de software deve estabelecer e formalizar informações sobre: as atividades e os papéis responsáveis, os artefatos de entrada e saída que devem ser criados ou mantidos em cada atividade, os procedimentos e ferramentas utilizadas, e o modelo de ciclo de vida utilizado (FUGGETTA, 2000). Existem diversos processos de engenharia de software publicados, conforme será apresentado nas seções a seguir.

2.1.1 Rational Unified Process

Entre os diversos processos de desenvolvimento, o *Rational Unified Process* (RUP) é um dos processos mais relevantes no mercado. O RUP é um processo proprietário prescritivo e bem definido para o desenvolvimento de sistemas. A aplicação deste processo geralmente é realizada em conjunto com o desenvolvimento orientado a objetos e as tecnologias baseadas em componentes (AMBLER et al., 2005), juntamente com a utilização da notação UML. Uma das bases para a elaboração do RUP foi o processo Objectory, elaborado em 1988 por Jacobson e seus associados (SCOTT, 2003). O RUP também baseou-se em outros métodos como o Booch e *Object-modeling technique* (OMT). A empresa Rational, originalmente responsável pelo desenvolvimento do RUP, foi adquirida pela IBM em 2002 e o RUP tem sido mantido pela IBM a partir de então.

O RUP possui o seu conteúdo estruturado em duas dimensões (AMBLER et al., 2005):

- A dimensão das fases que representa as quatro fases principais, ou "estações", que um projeto percorre ao longo do tempo. Estas fases são denominadas de Iniciação, Elaboração, Construção e Transição.
- As disciplinas do RUP incluem as atividades lógicas que ocorrem durante todo o projeto de desenvolvimento. Este agrupamento lógico foi estabelecido para modelagem de negócios, requisitos, análise e projeto, implementação ou codificação, testes, implantação, gerenciamento de configuração e mudanças, gerenciamento de projetos e ambiente.

O RUP é um processo iterativo e incremental, ou seja, para cada iteração do processo é possível realizar atividades em todas as disciplinas, algumas com maior e outras com menor esforço. As iterações das fases iniciais geralmente possuem mais ênfase nas disciplinas de negócio e requisitos, enquanto que as iterações finais, em testes e implantação. No núcleo principal do RUP existem princípios que

suportam o processo de desenvolvimento e que representam o espírito do RUP. Estes princípios são baseados na experiência de um grande número de projetos e sintetizados em alguns guias para orientação dos trabalhos (KROLL; KRUCHTEN, 2003):

- Atacar os maiores riscos cedo e continuamente, para que estes não se tornem problemas atacando o projeto;
- Certificar-se de agregar valor ao cliente;
- Manter o foco em software executável: documentações e diagramas são muito importantes, mas o produto final é código correto e testado;
- Buscar acomodar as mudanças desde o início do projeto, projetando e construindo software adaptável;
- Manter a *baseline* da arquitetura desde o início;
- Construir o sistema baseado em componentes;
- Trabalhar junto como uma equipe; e
- Fazer da qualidade um modo de vida e não uma reflexão tardia.

2.1.2 OpenUP

O OpenUP (*Open Unified Process*) nasceu de uma iniciativa financiada pela IBM de propagar uma parte do RUP de forma livre e com códigos-fonte abertos. Este projeto foi anunciado em outubro de 2005 e chamava-se BUP – *Basic Unified Process*. Em março de 2006, o BUP foi rebatizado como OpenUP. Os princípios do manifesto ágil guiaram a elaboração deste processo, que busca, em primeiro plano, a colaboração entre os indivíduos da equipe. Assim como o RUP, o OpenUP segue o ciclo de vida iterativo e incremental, sendo suas fases também organizadas do mesmo modo em Iniciação, Elaboração, Construção e Transição. Entretanto, o OpenUP possui um conjunto de elementos simplificado e um número reduzido de atividades, tarefas, papéis, produtos de trabalho e guias (OpenUP, 2011).

Um conceito introduzido no OpenUP é o chamado micro incremento, onde a partir de um produto de trabalho que requer dias ou semanas de trabalho, divide-se em objetivos menores, de poucas horas ou dias, prática conhecida no *eXtreme Programming* (XP) como *baby-steps*. A utilização desta prática busca um acompanhamento mais próximo das atividades do projeto. Dentro da atividade de planejar e gerenciar a iteração, especificamente na tarefa de gerenciar a iteração, o OpenUP sugere a utilização de reuniões diárias no guia de orientação para

colaboração da equipe, prática conhecida no SCRUM como as reuniões diárias de acompanhamento (OpenUP, 2011).

2.1.3 Oracle Unified Method

A Oracle, preocupada em apoiar a implantação de suas soluções de negócio, construiu o seu método de desenvolvimento, o qual também foi concebido baseado no Processo Unificado e denominado *Oracle Unified Method* (OUM). Este método possui um grande número de métodos detalhados, estruturados e bem documentados (FEIN et al., 2011). Como diferencial deste processo estão as práticas e os guias que suportam a implantação de projetos de sistema orientados a serviço, arquiteturas de integração, gerenciamento de identidade, *Business Intelligence* (BI), entre outras ferramentas e produtos proprietários da Oracle.

2.1.4 Accelerated SAP

A SAP, líder mundial em software de gestão empresarial, introduziu o *Accelerated SAP* (ASAP) em 1996. Esta metodologia é direcionada à implantação do sistema SAP ERP, com o objetivo de diminuir o tempo dos projetos (ESTEVEZ; JORGE, 2001). O ASAP está estruturado para facilitar a adesão dos usuários do sistema SAP *Business Suite*, através de roteiros bem definidos, com atividades e modelos de documentação. Esta metodologia possibilita aos novos clientes da SAP utilizar os conhecimentos adquiridos pela SAP, que provêm de milhares de implantações realizadas a nível global que utilizaram o ASAP e dos profissionais (engenheiros de processo de software) que trabalham para manter este método aderente às necessidades dos clientes.

O ASAP unifica o processo de implantação, de modo a alcançar a missão crítica das funcionalidades do negócio da organização (KALE, 2010). O centro da metodologia ASAP é o roteiro que define cinco etapas que apoiam a empresa, desde a preparação inicial até a conclusão do projeto com a entrada em produção. As cinco fases que a compõem são (ASAP, 2011):

- Preparação do projeto;
- Análise dos processos de negócio, conhecida como *Business Blueprint*;
- Realização;
- Preparação final; e,
- Entrada em produção e suporte.

A metodologia ASAP foi estendida para uma nova versão ágil em 2011 (ASAP, 2011), com atividades mais leves e simplificadas, conceitos de iterativo incremental e gerenciamento baseado em práticas do SCRUM. Tanto a versão tradicional do ASAP, quanto a recente versão ágil, destacam-se por seus roteiros do método, por fases e atividades, modelos de plano do projeto, ferramentas e aceleradores, área de conhecimento, procedimentos dos processos de negócio e base de dados de perguntas e respostas.

2.1.5 Microsoft Solution Framework

O modelo *Microsoft Solution Framework* (MSF) teve sua versão 1.0 publicada em 1993. Atualmente está na versão 5.0. O MSF é uma abordagem estruturada e adaptável para gerenciar projetos de tecnologia, ou seja, tanto para desenvolvimento de software, como para projetos de infraestrutura. O modelo é baseado na definição de um conjunto de princípios, disciplinas, conceitos chaves, fases, guias e práticas validadas pela Microsoft. Os elementos do MSF são baseados nas melhores práticas da indústria em conjunto com a experiência de mais de 30 anos da Microsoft na indústria de alta tecnologia (TURNER, 2006).

O MSF busca combinar os princípios dos ciclos de desenvolvimento cascata e espiral, utilizando planejamento com marcos de controle estabelecidos, mas com agilidade para buscar o parecer do cliente e estimular a criatividade. A cada iteração são realizadas as fases de Visualização, Planejamento, Desenvolvimento, Estabilização, Liberação, que possuem como marcos, respectivamente: visão e escopo aprovado, plano do projeto aprovado, escopo concluído, versão aprovada para liberação e implantação concluída. Entre os guias do MSF destacam-se:

- Guia para montagem de equipes: define uma matriz entre os papéis que são possivelmente combinados, classificando-os entre possivelmente combináveis, preferivelmente não e não recomendada.
- Guia de montagem de equipes e responsabilidades.
- Matriz para priorização da gestão do projeto, onde as dimensões do triângulo: recursos, cronograma e funcionalidades são classificadas exclusivamente entre fixo, preferencial e ajustável, conforme exemplo da Figura 2-1.

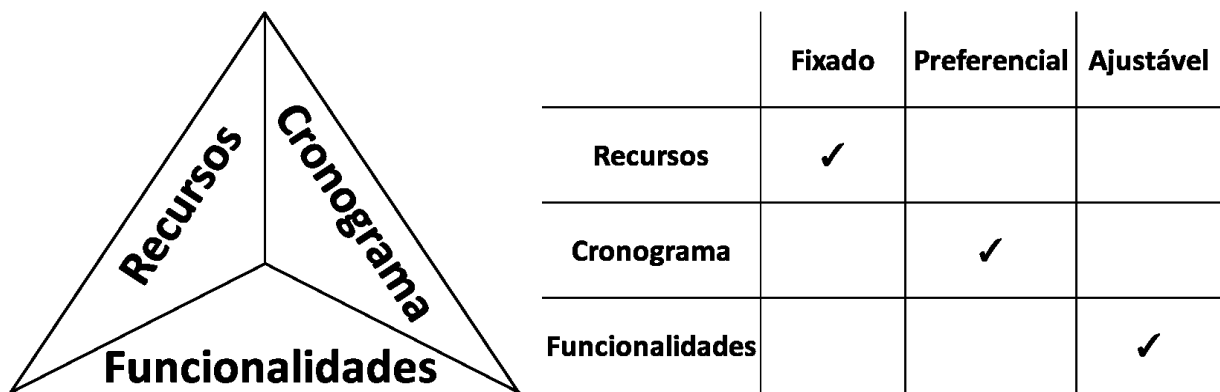


Figura 2-1. Matriz de priorização, adaptado de (TURNER, 2006).

O MSF possui uma especialização para facilitar a adoção de melhorias no processo e respectiva realização dos requisitos do *Capability Maturity Model Integration* (CMMi), chamado de MSF4CMMI. Assim como a especialização para o CMMi, também existe uma direcionada para práticas ágeis de desenvolvimento, chamada de MSF4ASD – *for Agile Software Development*.

A Microsoft fornece ferramentas para suporte e execução do MSF, chamada de Team Foundation. Este ferramental também suporta a execução de outros modelos além do MSF, como o suporte ao SCRUM. A Microsoft também publicou o seu modelo para serviços conhecido como *Microsoft Operations Framework – MOF*, o qual foi construído com base nas melhores práticas do *Information Technology Infrastructure Library* (ITIL) do escritório do gabinete do governo britânico. Enquanto o MSF possui ênfase na entrega de novas soluções, com ênfase em projetos, o MOF é dedicado a manter em execução o ambiente de produção. O MSF e o MOF possuem ligações para a entrega do domínio da solução desenvolvida para o domínio da operação (TURNER, 2006).

2.2 Linha de produtos de software

Os trabalhos iniciais em linhas de produtos de software iniciaram em 1976, por David Parnas (PARNAS, 1976). Entretanto, o assunto recebeu mais atenção da acadêmica e indústria somente após 20 anos. Uma família de sistemas é um conjunto de aplicações que compartilham funcionalidades comuns e mantêm funcionalidades específicas que variam de acordo com os sistemas abordados da família (PARNAS, 1976). A motivação de uma família de produto é explorar as similaridades, compartilhar código e reduzir os custos de manutenção, mas para isso

os produtos precisam ter aspectos comuns que compensem o esforço de analisá-los em conjunto, ao invés de analisá-los separadamente como sistemas individuais (PARNAS, 1979).

A engenharia de linha de produtos de software, ou em inglês *Software Product Line Engineering* (SPL ou SPLE) tem se mostrado uma área de muita pesquisa e interesse da indústria para a produção de software com baixo custo, redução do tempo de entrega e alta qualidade.

A abordagem de SPL difere-se dos outros métodos de desenvolvimento por possuir:

- Distinção entre dois processos de desenvolvimento: Processo de Engenharia de Domínio ou Engenharia Produto, em que as generalizações e as variabilidades de uma linha de produto são definidas e realizadas. Processo de Engenharia de Aplicação em que as aplicações da linha de produto são construídas através da reutilização de artefatos do domínio e exploração da variabilidade da linha de produto, conforme apresentado na Figura 2-2 (POHL et al.,2005).
- Gerenciamento explícito da variabilidade: durante a Engenharia de Domínio, os conceitos de variabilidade são introduzidos nos artefatos do domínio tais como: requisitos, modelos arquiteturais, componentes, casos de testes e qualquer outro que for necessário. Isto é explorado durante a Engenharia de Aplicação para derivar aplicações adaptadas para diferentes problemas ou clientes (KÄKÖLÄ; DUEÑAS, 2006).

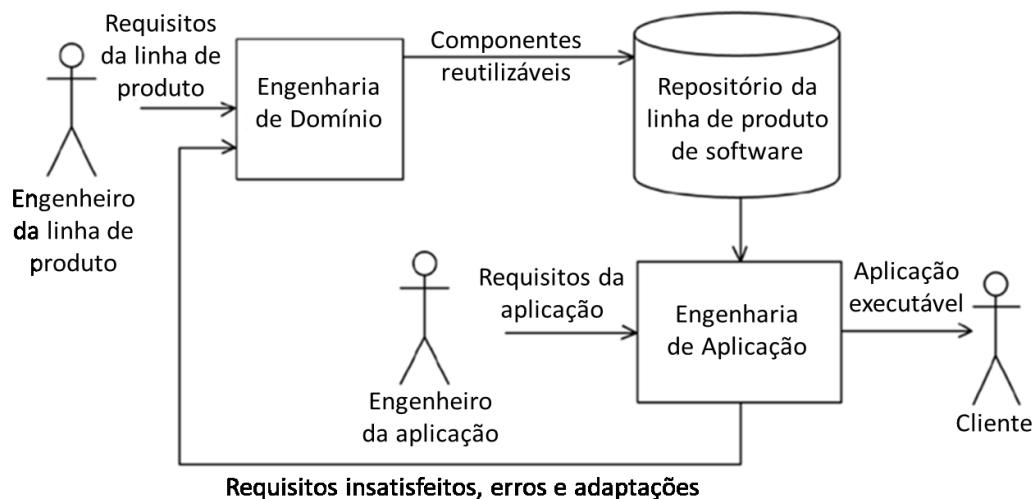


Figura 2-2. Fluxo entre os procesos de engenharia de domínio e aplicação, adaptado de (GOMAA, 2004).

Weiss e Lai propuseram um framework de SPLE, apresentado na Figura 2-3, que tem origem nos projetos da iniciativa ITEA: ESAPs, CAFÉ e FAMILIES (IEEE, 2004) (BOECKLE et al., 2004) (CAFÉ, 2011). Este framework é baseado na diferenciação entre engenharia de domínio e engenharia de aplicação (WEISS; LAI, 1999) (POHL; BÖCKLE; LINDEN, 2005). Neste framework pode-se analisar o ciclo entre as atividades de Engenharia de Domínio do produto e de semelhante forma o fluxo de Engenharia de Aplicação. Um importante ponto a ser ressaltado é a responsabilidade de gerenciar o produto pela Engenharia de Domínio.

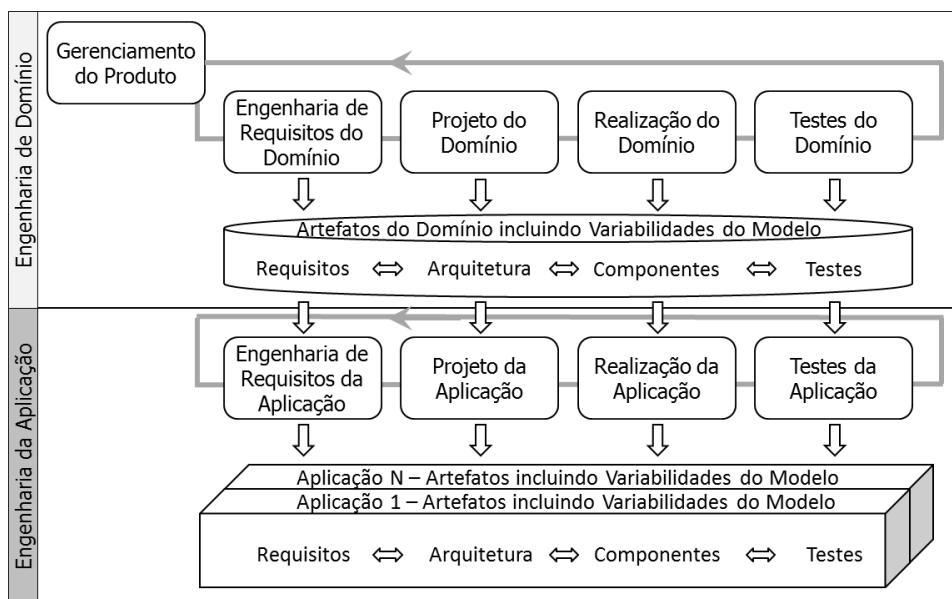


Figura 2-3. SPLE Framework, adaptado de (POHL; BÖCKLE; LINDEN, 2005).

Um dos pontos chaves de uma linha de produto de software é a modelagem das funcionalidades e características da linha. Esta modelagem apoia as decisões na análise de domínio, auxiliando assim o gerenciamento das semelhanças e diferenças dos artefatos do software, também chamado de gerenciamento de variabilidades. Para realização desta modelagem, diversas técnicas foram propostas, como: *Feature Model* (FM), *Basic Feature Model* (BFM), *Feature-Oriented Domain Analysis* (FODA) (KANG, 1990). Existem diversas ferramentas para FM, tais como FaMa-FW¹, Pure::Variants², FeatureIDE³, Software Product Lines online Tools (SPLOT)⁴, entre outras.

¹ Ferramenta disponível em <<http://www.isa.us.es/fama/>>.

² Ferramenta disponível em <<http://www.pure-systems.com/>>.

Para utilização do FM é necessário saber os tipos de relações. Para exemplificar será apresentado um modelo das funcionalidades de um celular com Android, conforme Figura 2-4, gerado a partir do Eclipse com o plugin FeatureIDE. Nesta linha todo o celular derivado possui funcionalidades básicas onde mandatoriamente estão presentes as funções básicas com mensagem e chamada de voz, porém a funcionalidade de mensagem MMS é opcional. Quanto às opções de comunicação, todos os tipos de conexões são opcionais, podendo não ter nenhum dos tipos de conexão. Já as funcionalidades adicionais (extras) serão ou MP3 ou câmera, podendo inclusive ser ambas. Contudo a câmera poderá ter a funcionalidade de uma e somente uma das alternativas 3 MP ou 8 MP, símbolo que caracteriza o ou exclusivo (PERROUIN, 2011).

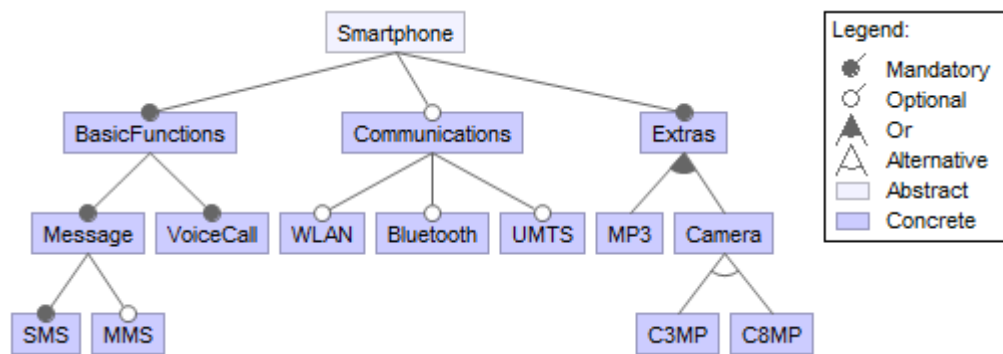


Figura 2-4. Funcionalidades do Smartphone no FeatureIDE³ (PERROUIN, 2011)

Algumas ferramentas também possibilitam a inclusão de relações, restrições e limitações entre as funcionalidades. A ferramenta Pure::Variants² também é um plugin do eclipse, semelhante ao FeatureIDE³, que permite representar estas restrições. Estas relações restritivas favorecem a identificação de dependências indiretas entre a hierarquia de funcionalidades, o que pode prever impactos na tomada de decisão, assim como, reduzir o esforço de teste através deste mapeamento detalhado. Na Figura 2-5 é possível observar a visualização gráfica da mesma linha de acordo com a representação da ferramenta Pure::Variants, onde os símbolos são semelhantes à figura anterior, sofrendo uma pequena variação para expressar o item mandatório (!), o item opcional (?) e o item alternativo (↔).

³ Ferramenta disponível em <<http://fosd.de/fide/>>.

⁴ O SPLOT pode ser acessado on-line em <<http://www.splot-research.org/>>.

Contudo, foi adicionada uma nova relação na linha, onde a funcionalidade MMS que requer a existência de uma câmera.

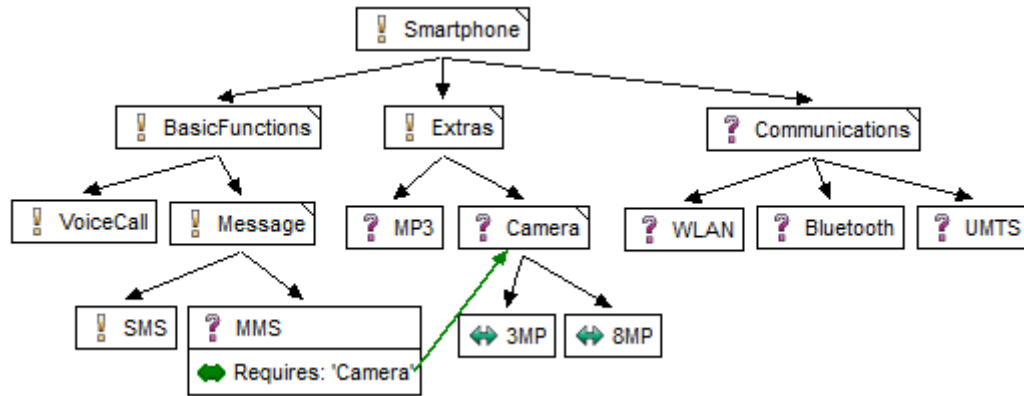


Figura 2-5. Funcionalidades do Smartphone no Pure::Variants (PERROUIN, 2011)

Um tipo de relação muito comum é a de exclusão, onde a seleção de uma característica da linha pode excluir a utilização de outras variações. Por exemplo, uma câmera de 8 MP poderia excluir desta linha de produto as telas de baixa resolução ou de tamanho pequeno.

2.3 Engenharia de Processo de Software

A engenharia de processo de software é responsável pela definição, modelagem, avaliação, medição, gestão, adaptação e melhoria do processo de ciclo de vida do software. Na seção anterior, foram abordados alguns de processos de engenharia de software, já a engenharia de processo de software é a criação de um novo processo do zero ou a partir de adaptações sobre um modelo existente. O termo Processo de Engenharia de Software também é utilizado, mas este confunde com as técnicas e atividades reais executadas dentro dos processos de ciclos de vida do software da organização, seja durante a aquisição de software, desenvolvimento, manutenção ou desativação (IEEE, 2004).

O SWEBOK – *A Guide to The Software Engineering Body of Knowledge* (IEEE, 2004) estruturou as áreas de conhecimento necessárias para a elaboração de um processo de engenharia de software, aqui referenciado como Engenharia de Processo de Software e apresentado na Figura 2-6.

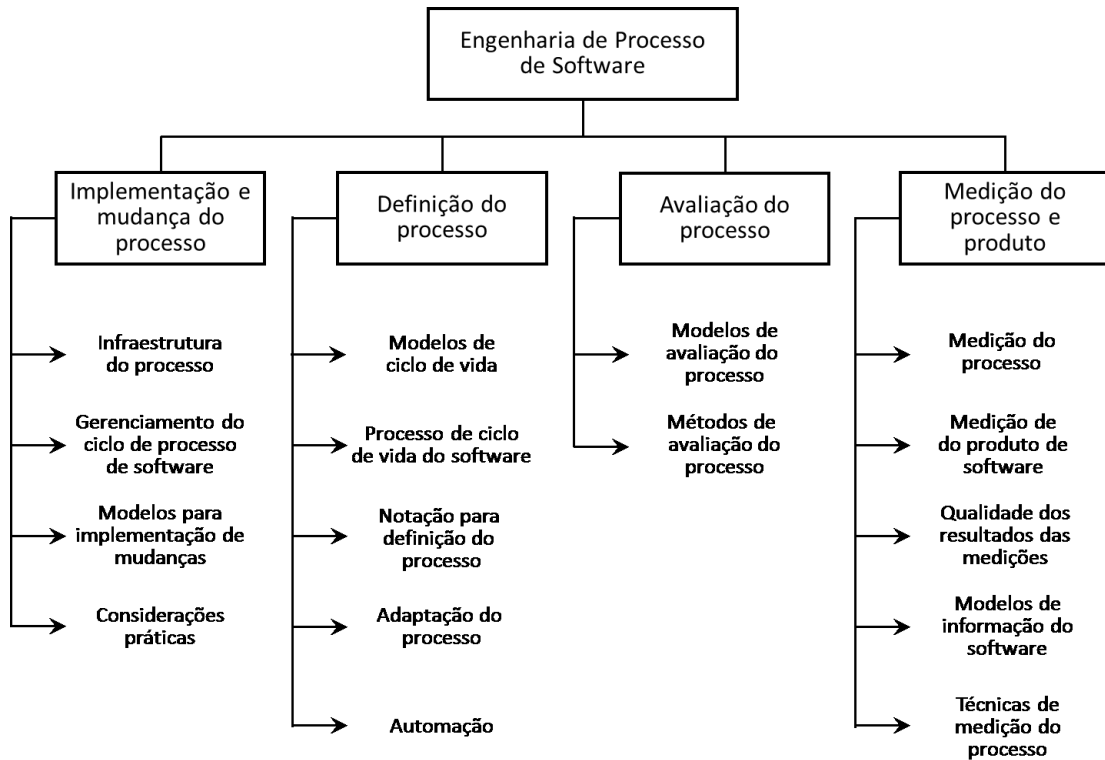


Figura 2-6. Áreas de conhecimento da engenharia de processo de software, adaptado de (IEEE, 2004).

A primeira subárea de conhecimento (Implementação e Mudança no Processo) é dedicada à avaliação e descrição das mudanças na organização e seu impacto na infraestrutura, atividades, modelos e outras considerações práticas (IEEE, 2004). Na segunda subárea (Definição do Processo) enumeram-se os conhecimentos necessários para definição do processo, dos modelos de ciclo de vida, diagramação (abordado no tópico seguinte 2.3.1), adaptação até ferramentas que suportem o controle da execução do processo. Já na terceira subárea (Avaliação do Processo) está a avaliação baseada em modelos de referência para a melhoria, como o CMMi e o MPS.BR; e os métodos de avaliação, com procedimentos estruturados que gerem uma avaliação quantitativa ou capacitiva, como o SCAMPI e MA-MPS, os quais serão abordados no tópico 2.3.4 e 2.3.2. Por fim, na última subárea são apresentados os conhecimentos relativos à medição (Medição do Processo e Produto).

2.3.1 Notação para modelagem de processos de software

A necessidade de customizar o processo foi motivada pela crescente demanda pela definição de melhorias contínuas para promover o desenvolvimento

de software de qualidade com baixo custo. A adaptação de um modelo de processo de software é uma tarefa complexa e dispendiosa. Antes de realizar adaptações nos processos faz-se necessária a análise e concepção das mudanças, atividade que requer muito tempo, de acordo com (ARMBRUST et al., 2008). Para elaborar, estabelecer e treinar as equipes no processo de software faz-se necessário uma notação que facilite a comunicação e a disseminação do processo.

O SPEM é o padrão da OMG destinado à modelagem de processos de software, cuja versão 2.0 foi disponibilizada em 2008. Sua estrutura favorece a reutilização separando elementos e aspectos entre o método de desenvolvimento e a instância de um processo em particular.

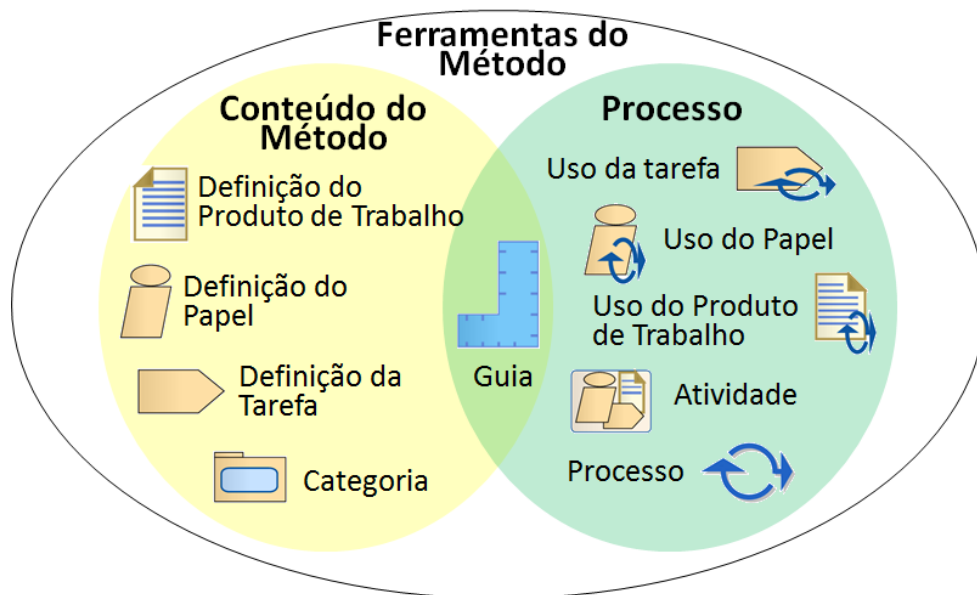


Figura 2-7. Elementos básicos do SPEM, adaptado de (OMG, 2008a).

Ao lado esquerdo da Figura 2-7 ficam as definições reutilizáveis chamadas de conteúdo do método, respondendo a questões sobre: “quem” deve fazer “o que” e “como”, sem envolver informações sobre o tempo. Entre os tipos de elementos do conteúdo do método estão: a definição de um produto de trabalho, definição de papéis, definição de tarefas, categorias e guias. Os papéis definem as funções das pessoas envolvidas, as quais podem realizar as tarefas como: Executor principal também chamado de responsável; Adicional também chamado de participante; e Assistente. Cada tarefa possui entradas e saídas, as quais são relacionadas com os produtos de trabalho, podendo ter guias para detalhar sua elaboração e utilização de ferramentas. Já ao lado direito da Figura 2-7 do diagrama estão os elementos para

representar o processo, “quando” estes são combinados para definir todas as sequencias de fases, iterações, o fluxo de atividade e marcos que definem o ciclo de desenvolvimento envolvendo aspectos temporais. Um processo é representado por uma coleção organizada de atividades. Já as atividades são um agrupamento que usam diferentes papéis, produtos de trabalho e tarefas. Os componentes de definição, no conteúdo do processo, podem ser combinados e reutilizados gerando diferentes processos (OMG, 2008a).

Existem outras propostas de linguagens para a modelagem de processos de software, porém diversas organizações utilizam simplesmente um diagrama de atividades ou fluxograma. Trabalhos anteriores realizaram a comparação entre os requisitos suportados entre as principais propostas de abordagens para a modelagem de processos, as quais estão avaliadas em diferentes perspectivas. Estas abordagens de representação são predominantemente baseadas em UML e possuem uma riqueza semântica semelhante (BENDRAOU et al., 2010):

- DiNitto's e a de Promenade são resultantes da especialização de um framework orientado a objetos, com um camada de modelagem UML (DI NITTO et al., 2002) (FRANCH et al, 1997);
- Abordagem de Chou's resultante de diagramas de alto nível, baseadas na UML, e uma linguagem adicional de baixo nível para o controle e execução do processo (CHOU, 2002);
- UML4SPM busca complementar o meta-nível, com suporte a tratamento a exceções e mecanismos para a execução (BENDRAOU et al., 2007);
- *Software Process Components* é uma abordagem simplificada que define basicamente três tipos de componentes: concretos, que não suportam variabilidades; abstratos, que são os pontos de variabilidades que serão substituídos pelos componentes variantes; e, opcionais. Estes componentes são ligados com associações do tipo: início-fim, início-início, fim-fim, fim-início (BARRETO et al., 2010).
- SPEM - *Software Process Engineering Metamodel* (versão 1.0 e 2.0) que possui origens tanto na forma do MOF meta-model quanto nos perfis da UML (OMG, 2008a). A versão 2.0 incluiu controles básicos de variabilidade como o uso da tarefa e o uso do produto de trabalho;
- vSPEM abordagem que complementa as variabilidades do SPEM versão 2.0. Permite a definição de pontos de variação e variantes, inclusive dentro

do conteúdo do processo, possibilitando a definição genérica de um processo com suas atividades (MARTÍNEZ-RUIZ et al., 2011). A Figura 2-8 apresenta na primeira linha os elementos que definem os pontos de variação e a segunda linha as variantes.

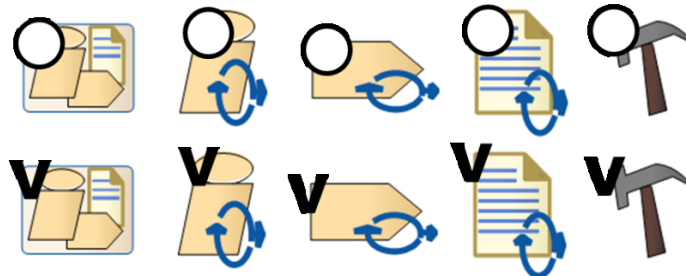


Figura 2-8. Elementos do vSPeM, adaptado de (MARTÍNEZ-RUIZ et al., 2011).

Destas abordagens de modelagem de processos de software somente o DiNitto's e o UML4SPM suportam a capacidade de execução do processo modelado, e com algumas restrições a abordagem de Chous's atendia parcialmente a este requisito (BENDRAOU et al., 2010). Em trabalhos mais recentes, novas abordagens tornaram possível a execução do processo para o SPeM (ALEIXO et al, 2011).

2.3.2 Ferramentas para modelagem de processo - *Eclipse Process Framework*

Buscando atender à natureza da cultura dos projetos e de diferentes organizações, faz-se necessário customizar determinadas partes do processo de software, visando trazer maior agilidade, qualidade e adequação à cultura organizacional. Devido à necessidade de documentação das customizações ou variabilidades dos processos de software surgiu o *Eclipse Process Framework* (EPF). O EPF é uma ferramenta destinada a engenheiros de software, líderes de projeto, gerentes de projeto responsáveis por adaptar e publicar o processo de desenvolvimento a ser seguido pela organização ou equipe (EPF, 2012). A IBM possui um produto comercial denominada IBM Rational Method Composer (RMC), muito semelhante ao EFP. O RMC diferencia-se do EPF devido: a integração com as demais ferramentas da Rational para gestão e execução do portfólio de projetos; e, os conteúdos prontos para acelerar a elaboração de diversos de métodos de desenvolvimento.

Esta ferramenta oferece apoio aos processos OpenUP, Extreme Programming e Scrum. Estes processos podem ser utilizados tal como estão ou com

customizações. Também está disponível uma ferramenta de Wiki para facilitar a colaboração dos membros da equipe na edição do processo de software. A Figura 2-9 descreve os componentes e as etapas para configuração e colaboração de processos de software através do EPF (EPF, 2012).

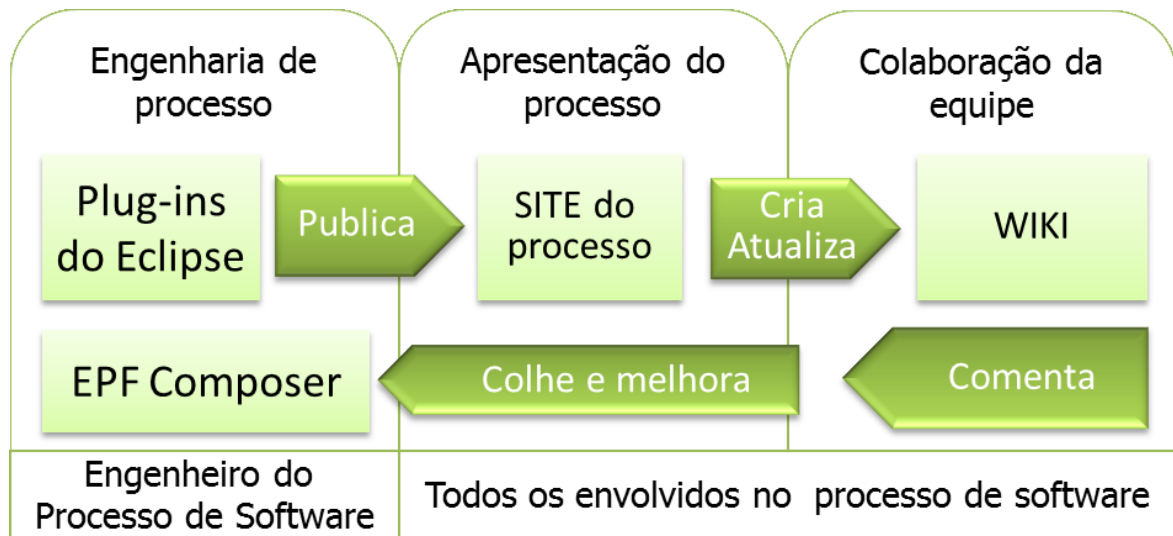


Figura 2-9. Etapas de edição de um processo no EPF (EPF, 2012).

O EPF é baseado no subconjunto da linguagem SPEM, o Unified Method Architecture (UMA). O UMA foi baseado na versão do SPEM 1.1 e contribuiu para a definição do SPEM 2.0. Ao utilizar o EPF para criar novos objetos pode-se selecionar alguns tipos de variabilidade as quais são associadas a elementos do mesmo tipo (por exemplo entre dois papéis, tarefas, ou produto de trabalho):

- Não aplicável: utilizado para elementos novos sem variabilidade.
- Contribui: adicionar elementos sem alterar diretamente as propriedades, objetivos e entrada e saída do elemento associado. Pode definir, por exemplo, uma atividade ou tarefa extra a ser realizada.
- Estende: cria um novo elemento modificando algumas características do elemento base, possivelmente substituindo atributos e associações.
- Substitui: sobrescreve o elemento com um novo elemento.
- Estende e substitui: Combina o efeito das duas variabilidades.

Estes tipos de variabilidades serão avaliados, posteriormente, para identificar se atenderão às necessidades da linha ou se será necessário buscar outras variabilidades a partir do SPEM ou extensões.

2.3.3 Normas internacionais ISO/IEC 12207 e ISO/IEC 15504

A Norma ISO/IEC 12207, que foi publicada nos Estados Unidos em 1995 e no Brasil em 1998 (ISO/IEC, 2008), auxilia na definição de processos de ciclo de vida de software. Atualmente, esta Norma encontra-se na versão 2008 (ISO/IEC, 2008). Esta Norma busca apoiar a organização na compreensão dos seus processos, atividades e papéis envolvidos em todo o ciclo de vida do software. Um ponto interessante é que esta Norma evita utilizar o termo processo de engenharia de software, visto que há diversos processos envolvidos como o processo de desenvolvimento, ou o processo de gerenciamento de configuração, entre outros. Esta Norma organiza os processos do ciclo de vida do software de forma estruturada e estabelecendo uma terminologia (nomenclatura) para facilitar a comunicação entre os envolvidos nos processos desde a concepção até a desativação do software (ISO/IEC, 2008).

Os processos desta Norma foram classificados em sete grupos de processos, conforme o objetivo e resultado esperado. A seguir é apresentada a descrição de cada grupo juntamente com a Figura 2-10 que lista os processos categorizados (ISO/IEC, 2008):

- Processos de acordo: definem as atividades para estabelecimento de contratos entre duas organizações;
- Processos organizacionais capacitadores de projetos: gerenciam a capacidade da organização de adquirir e fornecer produtos ou serviços;
- Processos de projeto: estabelecem as atividades para gerenciamento de projetos e podem ser aplicados a qualquer área da organização;
- Processos técnicos: representam as atividades que reduzem os riscos provenientes de decisões técnicas, desde a definição dos requisitos até a descontinuação do produto;
- Processos de implementação de software: contém as atividades para produção do item de software especificado;
- Processos de apoio ao software: atividades especializadas que contribuem para o sucesso e qualidade do projeto de software; e,
- Processos de Reúso de Software: destacam as atividades para reutilização de ativos, as quais devem ultrapassar as fronteiras do projeto.

Processos no Contexto de Sistema			Processos no Contexto de SW	
Processos de Acordo	Processos de Projeto	Processos Técnicos	Processos de Implementação de SW	Processos de Apoio ao SW
Aquisição	Planejamento de Projeto	Definição dos Requisitos dos envolvidos	Implementação de SW	Gerenciamento de Documentação de SW
Fornecimento	Controle e Avaliação de Projeto	Análises dos Requisitos de Sistema	Análise de Requisito de SW	Gerenciamento de Configuração de SW
Processos Organizacionais Capacitadores de Projeto	Tomada de Decisão	Projeto de Arquitetura de Sistema	Projeto de Arquitetura de SW	Garantia da Qualidade de SW
	Gerenciamento de Risco	Implementação	Projeto detalhado de SW	Verificação de SW
Processos Organizacionais Capacitadores de Projeto	Gerenciamento de Configuração	Integração de Sistema	Construção de SW	Validação de SW
	Gerenciamento da Informação	Teste de Qualificação de Sistema	Integração de SW	Revisão de SW
Gerenciamento de Portfólio de Projetos	Gerenciamento de Medição	Instalação de SW	Testes de Qualificação de SW	Auditoria de SW
Gerenciamento de Recursos Humanos		Suporte de Aceitação de SW	Processos de Reuso de Software	Resolução de Problema de SW
Gerenciamento da Qualidade	Operação de SW	Engenharia de Domínio		Gerenciamento do Programa de Reuso
	Manutenção de SW	Gerenciamento de ativos		
		Descontinuação de SW		

Figura 2-10. Grupos de processos no ciclo de vida, de acordo com a ISO/IEC 12207 (ISO/IEC, 2008).

Cada processo listado na Figura 2-10 é descrito em termos do seu propósito, resultados esperados, lista de atividades e tarefas que precisam ser executadas para atingir esses resultados.

Em 1993, o projeto denominado *Software Process Improvement and Capability dEtermination* – SPICE foi iniciado no âmbito da ISO. O objetivo do projeto era estabelecer um padrão internacional para avaliação processos de software. A Norma ISO/IEC 15504 representa o produto principal deste grupo, que tem atuado no suporte e organização de workshops e seminários. Esta Norma é complementar à Norma ISO/IEC 12207 sendo compatível com a ISO 9000 e está atualmente dividida partes, apresentadas na Tabela 2-1 (ISO/IEC, 2011).

Tabela 2-1. Partes da Norma ISO/IEC 15504.

Parte	Nome	Situação
15504-1	Conceitos e vocabulário	Publicada em 2004
15504-2	Realização de uma avaliação	Publicada em 2003
15504-3	Orientações para realização de uma avaliação	Publicada em 2004
15504-4	Orientação no uso para melhoria do processo e determinação da potencialidade do processo	Publicada em 2004
15504-5	Um exemplo de Modelo de Avaliação de Processo	Publicada em 2006
15504-6	Exemplo de modelo de avaliação de processo de ciclo de vida de sistema	Publicada em 2008
15504-7	Avaliação da maturidade de uma organização	Publicada em 2008
15504-8	Um exemplo de Modelo de avaliação de processo para Gerenciamento de Serviços de TI	Em elaboração
15504-9	Perfil do processo alvo	Publicada em 2011

A avaliação do processo pode ser utilizada com dois propósitos: melhoria dos processos ou determinação da capacidade dos processos, conforme pode ser observado na Figura 2-11 (SQI, 2011). A utilização contínua do processo de avaliação apoia a melhoria e busca pela eficiência e eficácia dos processos organizações. Também a manutenção dos processos é garantida pelo processo de avaliação, onde se verifica se as práticas previamente implantadas continuam sendo executadas adequadamente. O abandono de práticas pode ser gerado por inúmeros fatores, tais como: o crescimento da organização, a rotatividade da equipe, a gestão de projetos direcionada apenas pelo cumprimento dos prazos, entre outros.

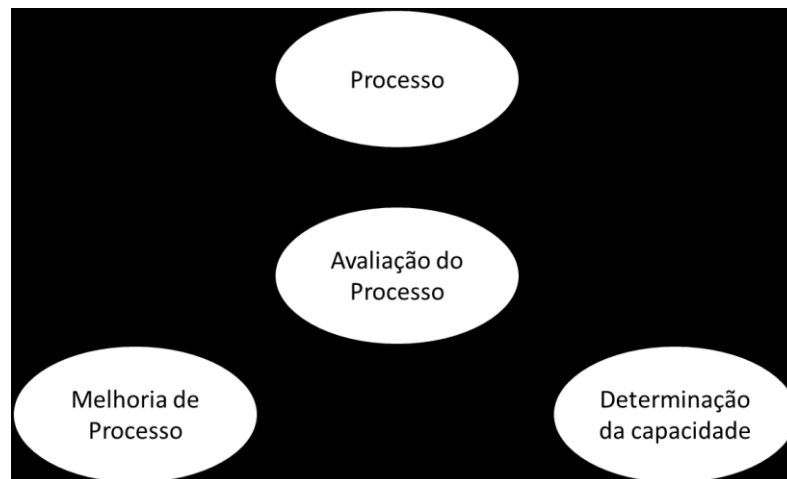


Figura 2-11. Contexto de utilização da ISO-15504, adaptado de (SQI, 2011).

A ISO-15504 define a classificação do processo de software das organizações em seis níveis de maturidade:

- 0 Incompleto: processo não implementado e em geral com baixo desempenho;
- 1 Executado: o processo é capaz de gerar o produto, mas é mantido graças ao esforço individual das equipes;
- 2 Gerenciado: os processos são planejados, acompanhados e controlados;
- 3 Estabelecido: busca práticas dos processos definidos sendo suportado por ferramentas, infraestrutura ou guias para a customização do processo;

- 4 Previsível: processos controlado quantitativamente para a alcançar os objetos de negócio;
- 5 Em otimização: processos com melhoria contínua e constantes inovações.

As Normas abordadas anteriormente são pouco utilizadas pela indústria de software, se comparadas à utilização dos modelos de melhoria e avaliação e tais como MPS.BR e CMMi. Contudo, o desenvolvimento destes modelos é fortemente embasado no conteúdo destas Normas, logo, o uso se dá de forma implícita.

2.3.4 CMMi

O CMMi - *Capability Maturity Model Integration* é um modelo de referência para melhoria dos processos de organizações ou projetos de software, o qual foi elaborado e patenteado pelo SEI – *Software Engineering Institute*. Em novembro de 2010 foi publicada a versão 1.3, sendo esta dividida em três modelos, denominados constelações: desenvolvimento de produtos de software (CMMI-DEV), aquisição e terceirização de soluções (CMMI-ACQ) e serviços (CMMI-SVC), conforme representado na Figura 2-12 (SEI, 2010a)(SEI, 2010b)(SEI, 2010c).



Figura 2-12. Modelos do CMMi por área, adaptado de SEI⁵.

O SEI é ligado à Universidade Carnegie-Mellon e financiado pelo Departamento de Defesa dos Estados Unidos (US-DoD). Além de sua atuação na

⁵ *CMMI for Services: Introducing the CMMI for Service Constellation*, disponível em <http://www.sei.cmu.edu/library/assets/hollenbach_07.pdf>, consultado em 06/11/2011.

melhoria de processos de software, o SEI também atua em outras áreas da engenharia de software com destaques em arquitetura, métricas e reúso, onde há ênfase no programa de sistemas de linha de produto (PLS – *Product Line System Program*).

2.3.5 MPS.BR

No Brasil existe um programa de melhoria e avaliação de processo de software, chamado de MPS.BR (Melhoria do Processo de Software Brasileiro). Este programa foi concebido pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), a qual organiza os comitês técnicos para revisão e expansão do programa.

O modelo de referência MR-MPS é baseado nas normas NBR ISO/IEC 12207 (ABNT, 2008) (ISO/IEC, 2008), na norma de avaliação de processo NBR ISO/IEC 15504 (ABNT, 2004) (ISO/IEC, 2011) e adicionalmente é compatível com o modelo CMMi-DEV (SEI, 2010a). Conforme pode ser observado na Figura 2-13, o MR-MPS, divide-se em Guia Geral, Guia de Aquisição e Guia de Implementação dividido conforme os níveis de maturidade. O modelo também possui o método de avaliação de maturidade MA-MPS e um modelo de negócio MN-MPS, conforme Figura 2-13.

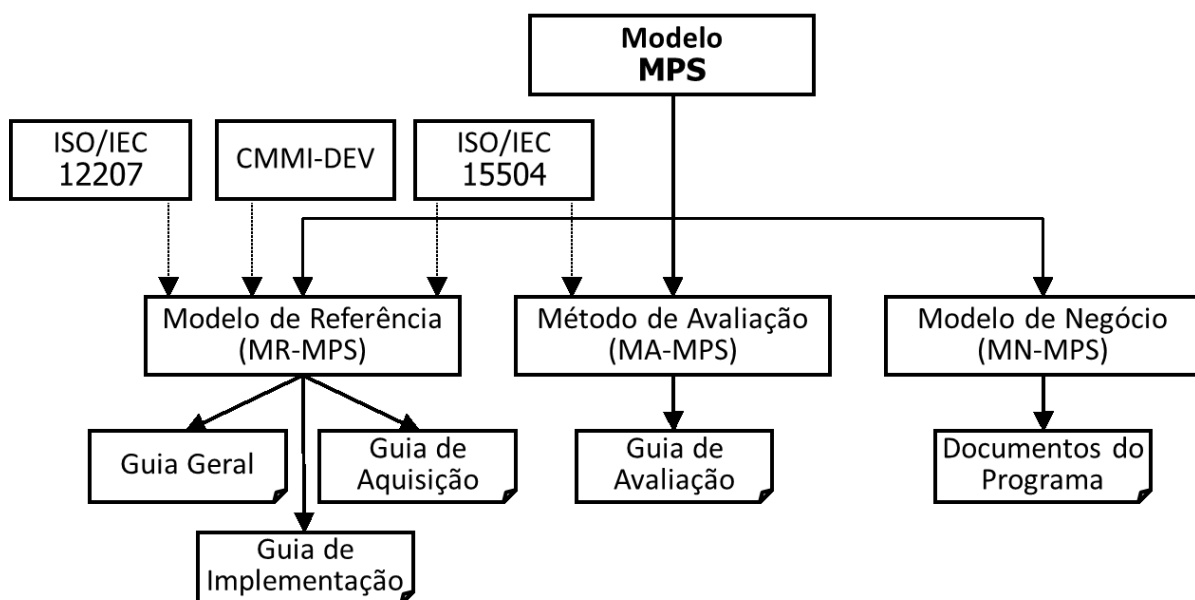


Figura 2-13. Modelo MPS.BR, adaptado de (SOFTEX, 2011).

2.3.6 Reutilização dos processos de software

Entre os métodos para desenvolvimento de software abordados anteriormente, o RUP possui uma disciplina chamada de Ambiente, a qual se propõe a tratar as customizações, adaptações do processo de software e ferramental a ser utilizado pela equipe. Para realização destas adaptações existe um guia, tanto para alterações para toda organização, quanto para alterações aplicáveis somente para projetos. Entretanto, o RUP não aborda técnicas de reutilização destas adaptações no processo.

Uma iniciativa precursora foi realizada no laboratório de engenharia de software do centro Goddard da NASA, onde foi criada uma nova versão do V-Modell denominada V-Modell-XT⁶ (ROMBACH, 2005). A primeira versão do V-Modell foi publicada em 1992, a qual foi atualizada em 1997 e rebatizada em 2005 para V-Modell XT. Este modelo é baseado em uma abordagem top-down de desenvolvimento e tem sido adotado na Alemanha para regulamentar as contratações, aquisições públicas de desenvolvimento e implantação de software (ROMBACH, 2005). Este processo é um dos precursores na aplicação de conceitos de customização e tratamento das variabilidades no processo de software, incluindo guias para estabelecer estas customizações.

Assim como nos processos tradicionais, os métodos de desenvolvimento ágeis também tratam o reuso de processo. O Método Crystal trata a organização de processos de software sistematicamente em nível organizacional. Este método busca elaborar uma família de processos conforme o nível de criticidade e tamanho do projeto, estes são categorizados por cores, por exemplo: de Claro (*Clear*), amarelo, laranja, laranja Web até o Vermelho (COCKBURN, 2006). As cores mais claras são para um processo mais leve para equipes pequenas, baixa criticidade e com pouco formalismo, até o vermelho ou rubi onde o processo precisa ter um alto nível de qualidade, para equipes grandes, com complexidade alta, em sistemas de alto risco ou de missão crítica.

⁶ O modelo pode ser acessado em <<http://www.v-modell-xt.de/>>, traduzido em inglês em <<http://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/1.3/V-Modell%20XT%20HTML%20English/>>

2.4 Linhas de processo de software

Uma das primeiras iniciativas sobre famílias de processos surgiu baseada nas famílias de produtos e inspirada no método Booch *Object-Oriented Design* (BOOD), onde observou-se que os processos de software também poderiam ser analisados e criados em famílias (SUTTON; OSTERWEIL, 1996).

De modo semelhante como as linhas de produto de software tratam características variáveis e gerenciam o reúso de seus componentes e arquitetura, também para processos de software a mesma técnica foi proposta e adaptada. As Linhas de Processo de Software, ou em inglês *Software Process Lines* (SPLs), ou ainda, *Software Process Families* são um conjunto de processos de software com um conjunto de características gerenciadas que satisfazem às necessidades específicas de uma organização específica e que são desenvolvidas a partir de um conjunto comum de processos centrais de um modo prescrito (ARMBRUST et al., 2009).

As principais características da engenharia de uma linha de processo são muito similares às de uma linha de produto (ROMBACH, 2005):

- Dois processos de desenvolvimento separados e distintos: o processo de engenharia de domínio e o processo de engenharia da aplicação. Com o domínio, processos para o reúso são criados, enquanto pela engenharia de aplicação os processos específicos são desenvolvidos.
- Um repositório de processos: os elementos do processo são reutilizáveis, em todos os níveis de abstrações, são armazenados e ficam disponíveis. Os tipos de elementos podem ser: atividades, tarefas, passos, papéis, guias, produtos de trabalho.
- Um processo de reúso sistematizado: para cada escolha de variabilidades, a escolha do componente do processo é pré-definida, com a documentação da decisão e justificativa.
- Um processo de gerenciamento dos processos: para cada exceção, como um comportamento não esperado do processo, opta-se pela inclusão ou não aos processos genéricos.

Os objetivos desta abordagem são descritos por (ROMBACH, 2005) como sendo:

“Os objetivos da utilização de linhas de processo de software são, da mesma forma como todas as abordagens de reutilização, aumento da previsibilidade, redução de custo e prazo e redução de riscos.” (p. 88, tradução nossa)

Outro benefício desta abordagem é a facilidade de customização do processo e do aprendizado entre processos nos diferentes projetos de software da organização (ROMBACH, 2005).

Um exemplo clássico de variabilidades em linhas de processo é a atividade de estimativa, em que uma linha de processos pode-se instanciar diferentes atividades para realiza-la. Este ponto de variabilidade poderia ser realizado através das técnicas de: análise de pontos de função, análise de pontos por casos de uso, COCOMO, histórico de projetos semelhantes e experiência da equipe.

Como abordado anteriormente, já em 1996 foram realizadas publicações sobre a utilização de uma linguagem para programação de processos ou famílias de processos, esta linguagem foi denominada de PDP - *Programmable Design Process*. Esta programação ocorria através da combinação entre: seleção, parametrização, especificação e customização de mecanismos (SUTTON; OSTERWEIL, 1996).

Em 2010, um trabalho propôs uma extensão para o SPEM tornando possível o controle do fluxo de execução do processo. Esta proposta reutilizou o EPF e dois processos de desenvolvimento Scrum e OpenUP adicionando as notações de extensão definidas no eSPEM (ELLNER et al., 2010).

Um trabalho de (ALEGRÍA, 2011) propôs uma estratégia de automação e de customização de processos, visando ganhos de tempo e custo, redução de erros e atendimento às necessidades dos projetos. O trabalho combinou a utilização de *Model-Driven Engineering* – MDE, um subconjunto de elementos do SPEM, uma abordagem de identificação dos processos da organização e a definição de regras de seleção e transformação.

A abordagem de (ALEIXO et al, 2011) propõe a definição de uma linha de processo de software para gerencia de variabilidades e realizar a derivação automática de processos executáveis. Para esta abordagem utilizaram-se os ferramentais: EPF Composer⁷ na definição do processo, o plugin do eclipse

⁷ Informações referentes à ferramenta EPF Composer, estão no site da ferramenta disponível em <<http://www.eclipse.org/epf/>>, consultado em 05/11/2011.

GenArch⁸ para gerenciamento de variabilidades (CIRILO; KULESZA; LUCENA, 2008) e a transformação dos artefatos do processo derivado da linha para um fluxo executável em jPDL – *JBoss Process Definition Language* (ALEIXO et al, 2011). O jPDL é uma especificação nativa do JBoss BPM⁹ utilizada em tempo de execução. A partir da versão 4.3 do jBPM, esforços tem sido dedicado para migração do jPDL para BPMN (*Business Process Modeling Notation*) 2.0 e também para a execução nativa do BPMN sobre o jBPM.

2.5 Arquitetura Orientada a Serviços

As abordagens baseadas na decomposição dos problemas do domínio têm evoluído trazendo novos paradigmas que facilitam a divisão das soluções e funcionalidades do produto de software. Esta evolução das abordagens de decomposição pode ser visualizada com o passar do tempo, conforme detalhado a seguir (ROSEN et al., 2008):

- Na década de 60 o software era dividido em múltiplas tarefas ou *Jobs*;
- Em meados de 70, passou-se a utilizar sub-rotinas e funções;
- A partir de 80, a decomposição começou a ser orientada a objetos;
- Já em 90 a abordagem de CBD passou a apresentar uma granularidade mais aderente as necessidades de TI;
- A partir de 2000, SOA contribui com a abordagem anterior trazendo facilidade de integração, independência de plataforma, infraestrutura para medição e gerenciamento do negócio.

A evolução dos paradigmas de desenvolvimento de software em relação ao nível do tipo de reuso pode ser visualizada, de forma simplificada, através da Figura 2-14 adaptada de (LINDEN, 2002). O fato da representação linear da evolução dos paradigmas, não impede que estes paradigmas sejam combinados de forma a propor uma solução adequada para as necessidades das organizações.

⁸ GenArch é uma ferramenta baseada em modelos (de característica, de arquitetura e de configuração) para derivação de produto, disponível em <<http://www.inf.puc-rio.br/~ecirilo/genarch/downloads.html>>, consultado em 05/11/2011.

⁹ Informações sobre a ferramenta de BPM da JBOSS estão em disponíveis no site do fornecedor, disponível em <<http://www.jboss.org/jbpm>>, consultado em 05/11/2011.



Figura 2-14. Evolução dos paradigmas e nível de reuso, adaptado de (LINDEN, 2002).

Prestar serviços de Tecnologia da Informação (TI) significa um esforço constante de: desenvolvimento, expansão e manutenção de um grande conjunto de sistemas, zelando constantemente pela sua integridade e interoperabilidade de forma a dar suporte aos negócios das empresas (BOTTO, 2004). Com as constantes mudanças e integrações presentes no mundo dos negócios, nota-se a crescente demanda tecnológica de serviços que atendam de melhor forma a estes requisitos. Este cenário trouxe a necessidade de uma arquitetura que pudesse facilitar a integração entre os processos, ponto favorável à Arquitetura Orientada a Serviços.

A Arquitetura Orientada a Serviços é a evolução da computação distribuída baseada na troca de mensagens de alto nível desenhadas conforme os processos de negócio. SOA pode ser definida como um conjunto de ferramentas para integração de processos de negócio e suporte a infraestrutura de TI com segurança, componentes padronizados como serviços, que podem ser reusados e combinados para atender as mudanças do negócio (BIEBERSTEIN et al., 2008).

Visto que o termo SOA foi introduzido em 1996, esta arquitetura não é mais um tema novo, conforme observado na Figura 2-15. (ROSEN et al., 2008). Contudo, muitas organizações ainda enfrentam dificuldades para implantar a orientação a objetos, o que torna a orientação a serviços ainda um assunto tratado como inovador.

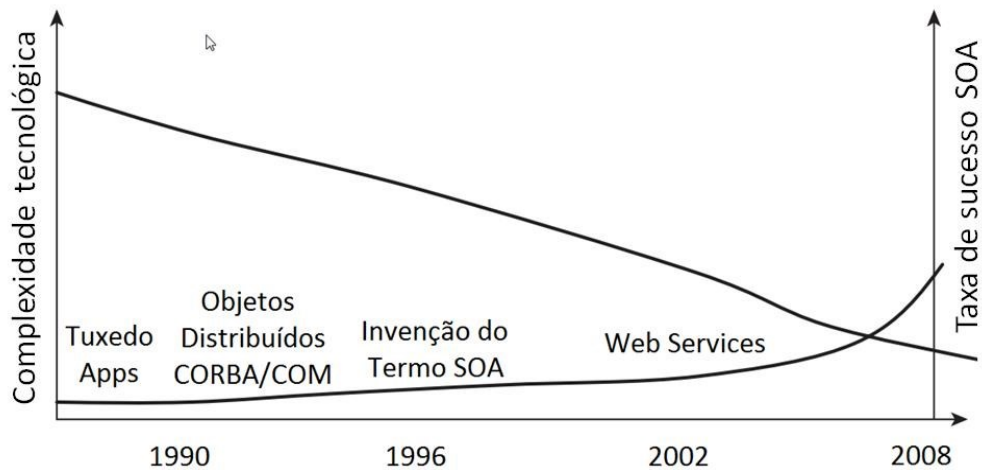


Figura 2-15. Linha do tempo SOA, adaptado de (ROSEN et al., 2008).

Os projetos de implantação de SOA anteriores aos Web Services apresentavam baixas taxas de sucesso, conforme apresentado na Figura 2-1. Os principais fatores de fracasso apontados eram a complexidade das tecnologias e a falta de conhecimento para definição de serviços na granularidade ideal para o negócio (ROSEN et al., 2008).

Um estudo de caso realizado nas instituições financeiras brasileiras observou a ênfase dada aos projetos para a implantação e promoção do reúso através de SOA, onde 60% dos participantes indicaram a intenção de buscar a orientação a serviços (REINEHR, 2008).

SOA possui um forte direcionamento ao reúso e tem facilitado e motivado a aquisição e venda de serviços. Conforme (GARTNER, 2011), em “SOA Is Evolving Beyond Its Traditional Roots”, graças ao sucesso de SOA que iniciativas como *Software as a Service* (SAAS), BPM e Cloud Computing se tornaram viáveis. No Brasil atualmente serviços como nota fiscal eletrônica, conhecimento do transporte eletrônico, validações de crédito de cartões, consulta a correios, entre outros diversos serviços de nosso dia a dia são suportados através de SOA. No próximo tópico são apresentadas as camadas desta arquitetura.

2.5.1 Camadas de SOA

Para o projeto de soluções baseadas em SOA sugere-se a divisão em camadas com responsabilidades bem definidas. Estas camadas buscam isolar os diferentes níveis de complexidade, assim como papéis e atividades na construção do software. O reúso também é promovido através das diferentes camadas,

identificando componentes conforme a granularidades de cada camada. A Figura 2-16 apresenta as principais camadas da Arquitetura Orientada a Serviços (ARSANJANI et al., 2008) (BIANCO et al, 2011) (BIEBERSTEIN et al., 2008).

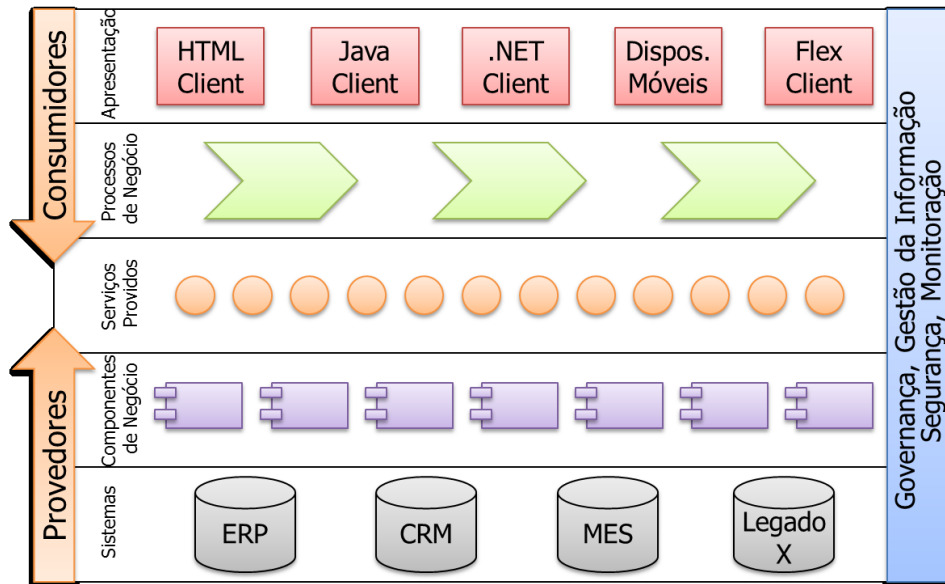


Figura 2-16. Camadas SOA, adaptado de (BIEBERSTEIN et al., 2008)(ARSANJANI et al., 2008).

SOA possui as seguintes as seguintes camadas:

- A camada de apresentação possui regras somente para apresentação da informação e consumo dos processos e serviços. Não envolve a complexidade do desenvolvimento das regras de negócio e pode ser reescrita facilmente para novas tecnologias. Algumas tecnologias atualmente aplicadas para desenvolvimento de interfaces para o usuário são: WEB (HTML+CSS+javascript), Java, .NET, Flex ou Flash, JME, Android, Objective C, entre outros.
- A camada de processos de negócio é responsável pelo fluxo de trabalho, realizando a composição ou orquestração dos serviços de negócio. Nesta camada estão os sistemas de BPM e Workflow.
- A camada de serviços é responsável por disponibilizar as regras de negócio dos componentes de forma simples, padronizada e desacoplada dos detalhes do desenvolvimento. Esta camada pode ser detalhada em outros níveis como:
 - Infraestrutura para barramentos de serviços (ESB), combinação de serviços, roteamento, controle de segurança, auditoria, etc.

- Semântica das entidades de negócio onde são gerenciadas e padronizadas as interfaces para troca de mensagens entre os provedores e consumidores de serviços.
- Registros de serviços que facilitam a pesquisa e descoberta dos serviços existentes, assim como a inclusão de novos serviços e assinatura de novos consumidores aos serviços existentes.
- A camada de componentes possui o código específico para a criação dos serviços, encapsulando a lógica de extração das funcionalidades dos sistemas internos ou externos à organização.
- A camada de sistemas representa os sistemas envolvidos na arquitetura de TI da organização, assim como toda a infraestrutura que os suportam. Esta camada representa desde o sistema operacional, banco de dados até as funcionalidades codificadas nos sistemas existentes.

A camada de serviços possui alguns princípios que devem ser observados para alcançar os ganhos efetivos de reusabilidade, manutenibilidade e consequente agilidade para TI e negócio. Estes princípios são abordados no próximo tópico.

2.5.2 Projeto de serviços

O projeto dos serviços deve ser guiado pelos princípios de manter um baixo acoplamento e alta coesão. Problemas de redundância ou duplicação muitas vezes são gerados devido ao acoplamento de serviços que dificultam o reúso devido ao incremento de responsabilidades em um único serviço (PAPAZOGLU; HEUVEL, 2006). O projeto de serviços mais simples e em maior quantidade, buscando a granularidade ideal, gera uma solução com maior potencial de reúso e manutenção.

A Orientação a Serviços possui oito princípios que são definidos a seguir, conforme (ERL, 2009):

- Padronização do contrato – Os serviços dentro do inventário ou repositório estão em conformidade com os padrões de projeto de contrato e semântica da informação de negócio.
- Fracamente acoplados – Os contratos dos serviços impõem requisitos de baixo acoplamento com o consumidor e garantem a dissociação do seu ambiente circundante (dependências).

- Abstração – Os contratos de serviços contêm somente as informações essenciais, limitando-se ao que está publicado no contrato e minimizando as meta informações.
- Reutilização – Os serviços contêm e expressam uma lógica agnóstica¹⁰ e podem ser posicionados como recursos corporativos reutilizáveis.
- Autonomia – Os serviços possuem um controle de alto nível sobre o ambiente de execução e tempo de suas operações e processos.
- Sem estado – Os serviços devem minimizar o consumo de recursos através do adiamento da gestão do estado da informação e independência da lógica do serviço.
- Fácil descoberta – Os serviços são complementados com metadados pelos quais podem-se fazer descobertos e interpretações automaticamente.
- Fácil composição – Os serviços são efetivos na participação de composições, independentemente do tamanho e complexidade da composição.

Considerando a importância dos *Web Services* para o sucesso dos projetos de SOA, assim como a efetividade desta proposta na integração entre diferentes plataformas e ambientes, na seção a seguir, serão abordadas as tecnologias que compõem os *Web Services*.

2.5.3 Web Services

Devido ao fato dos Web Services terem aprimorado a infraestrutura de TI e possibilitado a popularização de SOA, é comum confundir-se SOA com *Web Services*. SOA não é somente uma coleção de Web Services, mas um conjunto de conceitos que definem uma arquitetura. Web Services são uma coleção de tecnologias usadas para desenvolver diferentes aplicações, e, têm-se provado eficazes para a implementação da Arquitetura Orientada a Serviços (CHANNABASAVIAH; HOLLEY; TUGGLE, 2004).

As primeiras implementações concretas e práticas de SOA vieram com os Web Services. Com a aplicação dos Web Services, pôde-se colocar em prática os

¹⁰ Que expressa uma preocupação comum não específica para um problema, sendo independente da situação de negócio ou tecnologia utilizada.

conceitos de SOA, tornando esta arquitetura mais simples e aplicável. Oliveira (2005) descreve a relação entre Web Services e SOA "(...) Web Services são os músculos para a Arquitetura Orientada a Serviços".

O mercado tem atraído a atenção de muitas organizações e *Chief Information Officers* (CIOs) para SOA. Entretanto, ao tentar adotar esta arquitetura são encontrados diversos problemas com relação à maturidade da plataforma. Diversas empresas de software estão desenvolvendo ferramentas para proporcionar SOA de forma mais ágil, produtiva, segura, integrável e de baixo custo.

Para o desenvolvimento dos *Web Services* é necessário conhecer as tecnologias que o compõem. Nos tópicos a seguir, são apresentados os quatro pilares tecnológicos que sustentam os *Web Services*, são eles: XML, SOAP, WSDL e UDDI. A W3C é o grupo responsável pelo desenvolvimento de diversos padrões Web, como o XML, XML Schema, WSDL e inclusive o SOAP (ERL, 2004). O UDDI é a única tecnologia de Web Services (geração I) que não é controlada pela W3C.

2.5.3.1 XML

A linguagem de marcação estendida (eXtensible Markup Language) é um formato de texto simples e muito flexível, derivado do SGML (W3C; ISO 8879). Esta linguagem é utilizada em vários softwares e, principalmente, na Internet, devido ao seu compromisso de resolver a questão de conversação entre diferentes tecnologias e soluções. O XML é a linguagem para a comunicação e transporte da informação, no mundo tecnológico. O XML permite uma conversa baseada em texto plano, sendo de fácil leitura por diferentes tecnologias.

A linguagem XML tem redefinido a infraestrutura de aplicações, adicionando uma camada com inteligência informacional, oferecendo a possibilidade de usar esta camada com um laço que une diferentes sistemas empresariais entre si, e com a internet (ERL, 2004). Pode-se dizer que XML é uma tecnologia que permanecerá no mercado por um bom tempo, pois através dela foi realizado o desenvolvimento das relações entre diversas empresas em aplicações de comércio eletrônico B2B. Grande parte dos fornecedores de TI utiliza arquivos XMLs em suas soluções, mas o contato com esta tecnologia, geralmente, ocorre de modo indireto. Por exemplo, é possível utilizar XML ao realizar simples operações em um Web Site, realizando compras com a obrigatoriedade da nota fiscal eletrônica do governo, efetuando compras com cartão de crédito, consultando informações dos correios, entre outras.

2.5.3.2 SOAP

O protocolo de acesso a objetos simples (*Simple Object Access Protocol*) é responsável por manter e transportar a informação entre os aplicativos (W3C, 2011). O SOAP define um formato de como as mensagens devem trafegar na rede. Como o formato da mensagem é baseado no padrão XML, este protocolo pode ser usado em diferentes plataformas, linguagens e sistemas operacionais.

Dentro dos *Web Services*, o SOAP padroniza como a informação é remetida através do envelope identificando o destinatário e corpo do texto. Nota-se a semelhança com o mundo dos correios, onde o SOAP é comparado a um envelope com o seu destinatário, remetente, conteúdo e carimbo.

2.5.3.3 WSDL

A linguagem de descrição dos *Web Services* WSDL (*Web Services Description Language*) é usada para integração com os outros sistemas. Ela é um documento XML bem formado com as regras de integração. Estas regras de integração são baseadas nas definições dos documentos de entrada e/ou saída do serviço (HASAN, 2004). Os *Web Services* possuem um contrato que define o formato de sua mensagem de entrada e saída, conhecidas como requisição e resposta. Ambas as mensagens são formadas por um agrupamento de diferentes tipos de dados que são conhecidos como metadados ou contrato do serviço. Este contrato é definido em um XML Schema (XSD), o qual é inserido dentro do WSDL.

Uma definição WSDL pode ser usada para facilitar a criação de um consumidor ou provedor de Web Service. As plataformas Java (JDK6 ou superior) e Microsoft .NET possuem geradores de código a partir do WSDL (HASAN, 2004). Além dos contratos, são descritas as partes concretas para execução com as informações sobre *Service*, *Binding* e *Endpoint* – que definem o endereço físico para a troca de mensagens.

2.5.3.4 UDDI - Repositório de Serviços

Um dos pilares da tecnologia de Web Services é o catálogo universal de descrição e integração (*Universal Discovery, Description and Integration* - UDDI). O UDDI funciona como as páginas amarelas de um catálogo telefônico para os Web Services. De modo similar, são realizadas pesquisas por um especialista que preste determinado serviço que resolva o problema de negócio ou infraestrutura em

questão. O UDDI é um padrão controlado pela OASIS - *Organization for the Advancement of Structured Information Standards*.

Em SOA, estabelece-se um componente central para registro dos serviços, chamado de *Service Registry and Repository*, onde ficam armazenadas informações sobre cada serviço, formando um inventário ou catálogo. Este repositório é utilizado para descobrir, localizar e reusar os serviços, tanto em tempo de projeto da solução, como em tempo de execução, sendo mais comum tempo de projeto (ERL, 2009). Os Repositórios de Serviços e Registros são baseados ou integrados ao UDDI.

Conforme o passo 1 da Figura 2-17, o registro de um serviço se dá pelo cadastramento das entidades de negócio, os serviços de negócio, especificação do ponteiro (ou Binding Template), tipos do serviço, relacionamento com o negócio e subscrição (ERL, 2004). Os repositórios de serviços (Service Repository) são utilizados durante o projeto da solução, passo 2, onde se busca informações sobre a descrição e a definição, como e onde o serviço é executado, que informações contêm a resposta do serviço, quem e porque está usando o serviço (CARTER, 2007). Após entendimento do contrato no passo 3, o último passo é a execução do serviço, conforme Figura 2-17.

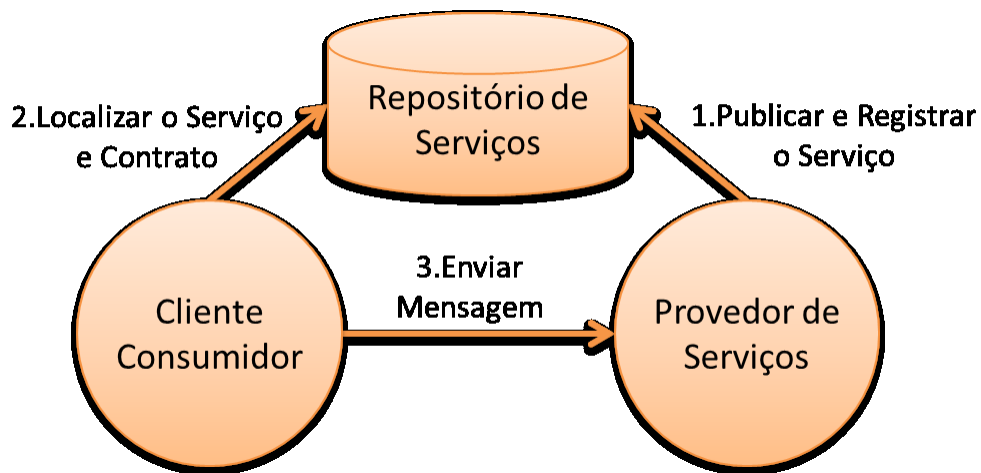


Figura 2-17. Fluxo de mensagens para registrar e consumir serviços, adaptado de (ERL, 2009).

O serviço de registro (*Service Registry*) está associado à execução em tempo real. Porém, algumas bibliografias e, até mesmo soluções de mercado referenciam como *Service Registry* para ambas as coisas (ERL, 2009). Como evidenciado anteriormente, o padrão UDDI é controlado pela OASIS, porém as soluções de repositório ainda possuem conceitos que necessitam padronização.

2.5.4 Segunda geração dos Web Services (WS-*)

Os *Web Services* possuem um conjunto de especificações que estendem os conceitos previamente descritos para dar suporte às especificações de segurança, confiabilidade, transações, arquivos de anexo, entre outros. Estas especificações são conhecidas como a segunda geração de *Web Services* e grande parte delas foram rotuladas com o prefixo WS-* (ERL, 2009). Existem diversas especificações que padronizam aspectos como: segurança (WS-Security, WS-Addressing, WS-Trust), envio de arquivos (WS-Attachments), controle transacional (WS-Transaction), gerenciamento de características não funcionais do serviço ou políticas (WS-Policy) entre outras.

As extensões de confiabilidade possuem dois padrões: o WS-Reliability e o WS-ReliableMessaging. O WS-Reliability foi desenvolvido pela Sun, Oracle e Sonic, este protocolo provê *Web Services* assíncronos, com garantia de entrega, sem duplicação e com as mensagens ordenadas. O WS-ReliableMessaging é um protocolo muito similar ao WS-Reliability, que permite a independência da tecnologia de transporte e enlace de rede. Esta especificação foi produzida pela BEA, IBM e Microsoft (HOHPE; WOOLF, 2004).

2.5.5 Representational State Transfer - REST

O REST ou Transferência de Estado Representacional é um estilo arquitetural definido por princípios, os quais foram elaborados na tese do pós-doutorado de Roy Fielding's (WEBBER et al., 2010). Neste princípio busca-se uma arquitetura mais leve, resgatando a utilização do protocolo HTTP, identificado unicamente cada registro de informação por um endereço único (URI- *Uniform Resource Identifier*) e mapeando as operações através dos verbos HTTP (POST, GET, PUT e DELETE).

A partir dos conceitos de REST, também emergiu um novo termo muito utilizado no mercado chamado de *Web-Oriented Architecture* – WOA. Este novo conceito de arquitetura pode ser definida como uma especialização da SOA com ênfase no simples uso das tecnologias e padrões da estabelecidos pela Web 2.0 (THIES; VOSSSEN, 2009). Tanto REST, assim como WOA, buscam reduzir a complexidade de SOA facilitando a adoção de uma arquitetura de serviço de mais baixo nível, ou seja, transformando as operações das aplicações Web em serviços. Esta redução de complexidade de SOA fica explícita pela simples utilização do HTTP, ao invés do protocolo SOAP utilizado pelos *Web Services*. Como abordado

anteriormente os *Web Services* possuem o WSDL para descrição, já para o REST esta descrição é realizada pelo WADL - *Web Application Description Language*.

No próximo tópico será abordado o BPM. Atualmente os fornecedores de ferramentas para BPM possuem suporte completo para *Web Services*, porém, o consumo e exposição de serviços mais leves, como a utilização do REST, não é uma funcionalidade suportada por alguns fornecedores.

2.6 BPM – Business Process Management

Dentro da área de TI muitas vezes se confunde o gerenciamento de processos de negócios (BPM) com ferramentas ou iniciativas de levantamentos de processos, conforme (JESTON; NELIS, 2008).

“Existe um perigo das organizações acreditarem que uma vez que tenham comprado uma ferramenta de modelagem de processo, esta vai resolver todos seus problemas e melhoria de processo irá seguir. Nada poderia estar mais distante da verdade. Uma ferramenta de modelagem de processos é apenas um pedaço de software, e sem uma metodologia ou framework, recursos qualificados para usá-la e um comprometimento autêntico da liderança organizacional, isto será inútil.” (p. 9, tradução nossa)

Conforme o Guia para **Gerenciamento de processos de Negócio** corpo comum de conhecimento (BPM CBoK) elaborado pela associação internacional de profissionais de gerenciamento de processos de negócio (ABPMP), BPM pode ser definido como (BPM-CBOK, 2009):

“uma abordagem disciplinada para identificar, desenhar, executar, documentar, medir, monitorar, controlar e melhorar processos de negócio automatizados ou não a fim de atingir resultados consistentes com metas estratégicas da organização”.

BPM é uma disciplina que deve ser realizada continuamente buscando alcançar os objetivos de negócio da organização. Logo, a definição da direção estratégica da organização é um pré-requisito. Para realização do gerenciamento de processo utiliza-se o ciclo PDCA (Plan-Do-Check-Act / Planejar-Fazer-Verificar-Agir), buscando identificar melhorias, implantar, medir e controlar continuamente o processo de negócio (JESTON; NELIS, 2008). Semelhante aos modelos de maturidade para processos de desenvolvimento software, CMMi, também foram propostas abordagens para melhoria de capacidade dos processos organizacionais, como o *Business Process Maturity Model* (BPMM) (OMG, 2008b).

Muitas melhorias podem ser endereçadas ao processo através de procedimentos e sem necessariamente o desenvolvimento de Software e

mecanismos de automação. Contudo, as ferramentas de BPM representam uma posição de destaque apoiando o ciclo PDCA e controle em tempo de execução, assim como bases analíticas da execução do processo. Antes mesmo de 1990, a academia e indústria já atuavam sobre gestão de processos de negócio e ferramentas de suporte. Nesta época todos os profissionais da indústria e pesquisadores usavam o termo *Workflow Management* – WfM para o que atualmente se chama de BPM. Atualmente observa-se constantes equívocos no uso destes termos considerando a definição e escopo associada aos trabalhos publicados (HAVEY, 2005).

Apesar da robustez das ferramentas presentes na indústria chamadas de WfMS – Workflow Management Systems, muitas vezes existe falta de mecanismos que gerem informação de diagnóstico, relatórios em tempo real para a análise, a fim de identificar gargalos e falhas de processos de negócios. As ferramentas que dão suporte ao gerenciamento e operação dos processos são chamadas de BPM Systems – BPMS, as quais atendem com maior qualidade estes quesitos de análise e diagnóstico dos processos de negócio (KO, 2009).

As ferramentas de BPMS apoiam a melhoria contínua dos processos de negócio, centralizando a modelagem, execução e monitoração dos processos no ambiente corporativo (ZHANG et al., 2010). Entretanto, conforme Jeston e Nelis (2008), uma organização não pode simplesmente comprar BPM, pois para se implantar BPM é necessário observar todos os detalhes da equação:

- “BPM = organização + pessoas + tecnologia”.

Conforme abordado anteriormente, sabe-se que muitos projetos de BPM e SOA deram maior ênfase à tecnologia e à venda de infraestrutura¹¹, do que aos fatores críticos para o sucesso do reúso (BIANCO et al, 2011)(JESTON; NELIS, 2008). Tanto BPM quanto SOA são princípios que precisam ser adotados e praticados, não podem ser simplesmente adquiridos. O próximo tópico trata de modelos e métodos para SOA e BPM, como evidenciado estes possuem um papel fundamental.

¹¹ Informações para evitar as armadilhas na adoção de SOA estão disponíveis em <<http://www.ibm.com/developerworks/library/ar-soapit/>>, consultado em 15/11/2011.

2.7 Modelos de Processos para SOA e BPM

Pesquisas da academia e indústria constataram a necessidade de organizar o processo de software de melhor forma para dar suporte às soluções para SOA e BPM. Estes processos caracterizam-se por enfatizar atividades de reutilização de processos e serviços de negócio, sendo geralmente suportado por um repositório de serviços e processos. Práticas para agrupar processos e serviços são estabelecidas para facilitar o gerenciamento e suporte a requisitos não funcionais. Um método para projetos SOA, WOA e BPM foi proposto por (THIES; VOSSEN, 2009), sendo este dividido em 3 fases, conforme representado pela Figura 2-18:

- Fase de modelagem de processo, subdividida nas atividades:
 - Identificar os sistemas envolvidos;
 - Analisar os casos de negócio, papéis e módulos dos sistemas;
 - Construir a especificação do processo usando BPMN; e,
 - Identificação de cada tarefa no processo que originaram serviços, após análise do arquiteto de software.
- Fase de Refinamento dos processos e serviços, subdividida nas atividades:
 - Identificar serviços existentes;
 - Definir tecnologia SOA; e
 - Descrever serviços, funcionalidades e informações.
- Realizar serviços, subdividida nas seguintes atividades:
 - Desenvolver serviços previamente projetados;
 - Configurar conexões para serviços existentes ou novos;
 - Verificar e validar serviços; e
 - Estabelecer SLA e disponibilizar em produção a solução.

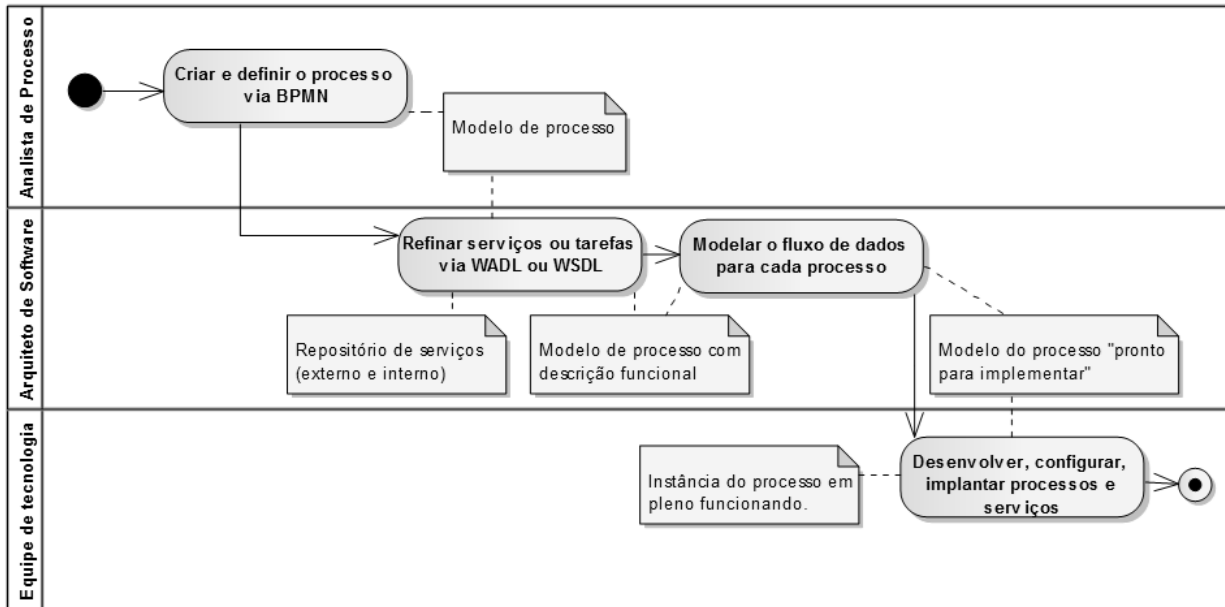


Figura 2-18. Fases do método WOA, adaptado de (THIES; VOSSSEN, 2009).

Como já abordado, o desenvolvimento baseado em Linha de Produtos de Software (SPL) é uma linha de pesquisa em evidência, que busca maximizar o reuso, aprimorar a qualidade, reduzir os custos e tempo de entrega da solução. A abordagem SOPLE-DE de (MEDEIROS; ALMEIDA; MEIRA, 2010) é uma proposta para desenhar sistemas orientados a serviços utilizando o método de desenvolvimento SPL.

A atividade inicial do SOPLE-DE é a identificação dos elementos arquiteturais, a qual se divide em: definir serviços para o processo de negócio, definir serviços e componentes candidatos a partir dos casos de uso, descrever funcionalidades e definir atributos de qualidades. Na atividade seguinte, a análise de variabilidade considera os ativos existentes, a coesão, os acoplamentos, a parametrização, as extensões de contratos dos serviços e define a técnica de variabilidade a ser realizada. Na terceira e quarta atividades, as visões da solução são projetadas descrevendo componentes, serviços, composições, extensões e variabilidades com suas condições. Uma atividade paralela é realizada com o objetivo de documentar as motivações e decisões realizadas pelos envolvidos no projeto. Este fluxo de atividades, juntamente com os papéis executores, é apresentado na Figura 2-19.

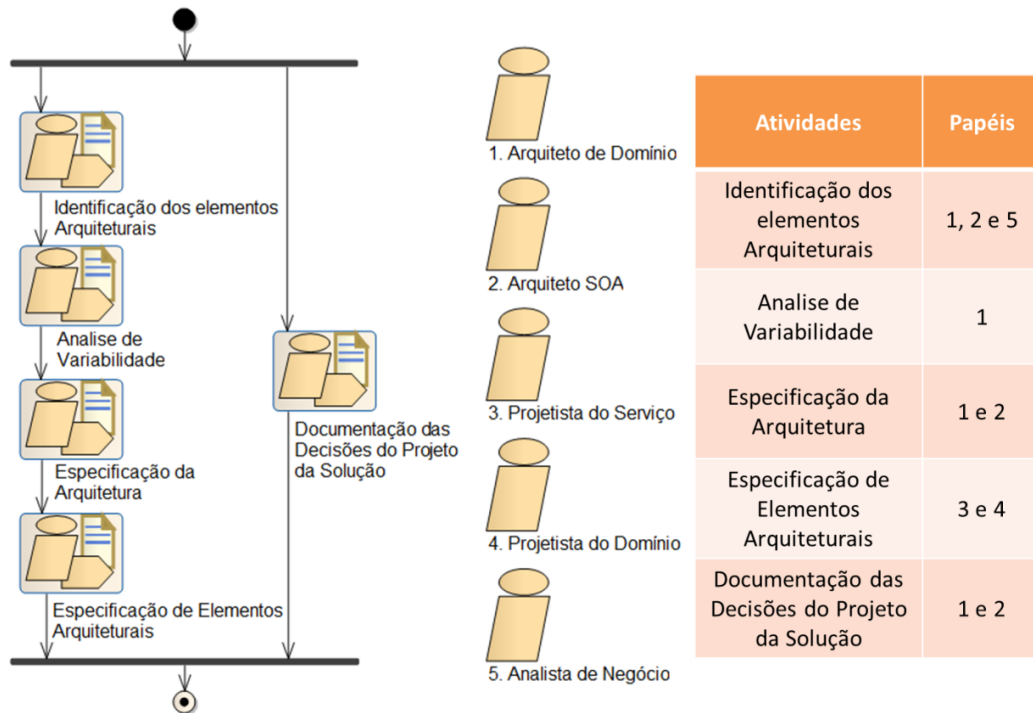


Figura 2-19. SOPLE-DE Papéis e Atividades, adaptado de (MEDEIROS; ALMEIDA; MEIRA, 2010).

O forte direcionamento arquitetural e a identificação de artefatos existentes das SPL permitiram uma boa integração com SOA, provando entregar serviços com bom acoplamento, operações coesas e baixa instabilidade, conforme os estudos experimentais realizados para validação do modelo (MEDEIROS; ALMEIDA; MEIRA, 2010).

2.7.1 Modelos de maturidade para SOA

Diversos fornecedores de software possuem métodos próprios de implantação de SOA, logo existem diversas classificações para medir a sua maturidade, o que torna o assunto polêmico. Em 2005, as empresas Sonic Software, Systinet, AmberPoint e BearingPoint fizeram uma aliança para a criação de um modelo único de maturidade, conhecido como SOAMM (SOA Maturity Model), o qual foi publicado no fórum da OMG SOA SIG. Este modelo teve os seus níveis de maturidade relacionados aos do modelo CMMI - Capability Maturity Model Integration (SOAMM, 2005).

Conforme apresentado na Figura 3-3, no modelo SOAMM, o nível 1 tem como escopo treinamento, projetos pilotos com Portal e a construção de um pequeno número de serviços. Já no nível 2, o escopo é estendido para múltiplas aplicações,

para institucionalizar o uso de SOA e solidificar a arquitetura SOA dentro da liderança e governança de TI. No nível 3, surge uma camada de software para combinar os serviços, conforme o processo de negócio e também a disponibilização para parceiros de negócio. O nível 4 é caracterizado pelo estabelecimento e controle de métricas sobre a eficiência do processo e o 5 pela otimização contínua e automatizada (SOAMM, 2005).

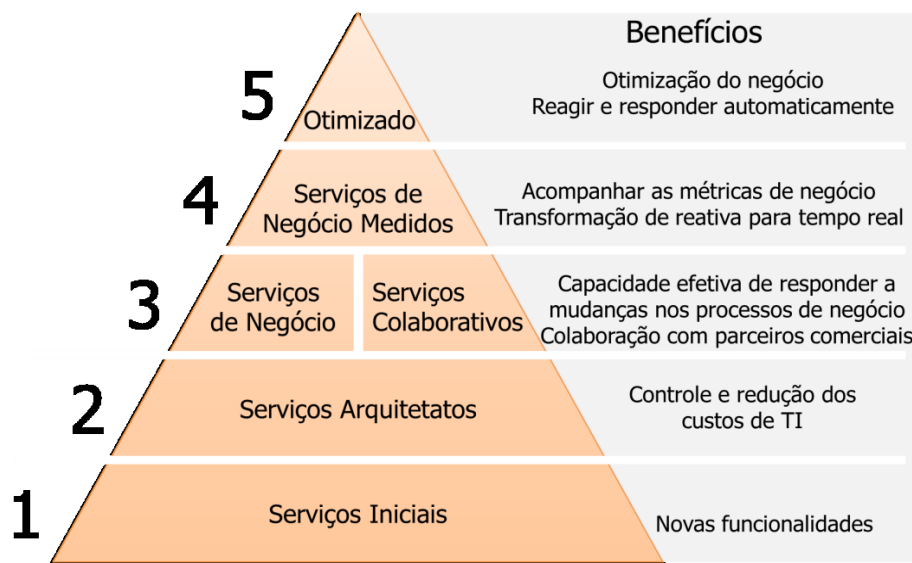


Figura 2-20. Níveis de maturidade SOA e benefícios (SOAMM, 2005).

Assim com o modelo SOAMM, a IBM também definiu o seu modelo de maturidade, chamado de *Service Integration Maturity Model* (SIMM) e foi incorporado pelo *Open Group Service Integration Maturity Model* (OSIMM, 2009). O OSIMM define um roteiro para adoção incremental de SOA, buscando maximizar os benefícios ao negócio e à TI em cada um dos sete estágios de sua evolução. O Open Group submeteu o OSIMM ao processo de publicação da ISO, com vistas a tornar-se um padrão internacional, sendo denominado de ISO/IEC 16680.

Comparado aos níveis do SOAMM, o OSIMM divide o nível inicial em três níveis menores (silos, integrado e componentizado), facilitando a evolução inicial em passos mais curtos de maturidade. Dentro deste modelo são analisadas as diferentes dimensões da TI em cada linha e as colunas correspondem aos níveis de maturidade, conforme representado na Tabela 2-2.

Tabela 2-2. Open Group Service Integration Maturity Model, adaptado de (OSIMM, 2009).

	Silo	Integrado	Componentizado	Serviços	Serviços Compostos	Serviços Virtualizados	Serviços Dinamicamente Reconfiguráveis
Negócios	Dirigido a linhas de Negócio Isoladas	Integração de Processos de Negócios	Negócio Componentizado	Negócio disponibilizado como Serviços	Processos realizados pela composição de serviços	Centros de serviços geograficamente independentes	Misturar e combinar serviços adaptando-se ao contexto
Organização	Estratégia e Governança indefinida	Transformação de TI	Governança comum de processos	Surgimento Governança SOA	SOA e Governança de TI alinhadas	Alinhamento c/ estratégia de Infraestrutura	Governança conforme políticas
Métodos	Análise e projeto estruturado	Modelagem Orientada a Objetos	Desenvolvimento baseado em componentes	Modelagem Orientada a Serviços	Modelagem Orientada a Serviços	Arquitetura orientada a serviços para Infra	Modelagem Orientada a Gramática de Negócio
Aplicações	Módulos	Objetos	Componentes	Serviços	Integração de processos via serviços	Integração de processos via serviços	Montagem dinâmica; Invocação cfm. contexto
Arquitetura	Arquitetura Monolítica	Arquitetura em Camadas	Arquitetura de Componentes	Estabelecimento do SOA	SOA	SOA suportada em Grid	Arquitetura dinamicamente reconfigurável
Informação	Específica por Aplicação	Específico por Linha de Negócio	Modelos canônicos	Informação como serviço	Dicionário de dados de negócio e repositório.	Serviços de dados virtualizados	Vocabulários de dados semânticos
Infra-estrutura	Plataforma Específica por Linha de Negócio	Padrões corporativos	Infraestrutura comum e reutilizável	Projetos baseados em SOA	Ambiente SOA comum	Ambiente Virtual SOA c/ respostas reativas	Percepção dinâmica c/ decisões e resposta
	Nível 1	Nível 2	Nível 3	Nível 4	Nível 5	Nível 6	Nível 7

A partir da combinação do SOAMM e do SIMM, foi proposto um novo modelo chamado de CSOAMM - *Combined SOA Maturity Model*. Este modelo apresenta uma divisão de níveis mais detalhada e em menores passos. A proposta apresenta dez níveis, porém três estágios são iniciais e ainda não relacionados a SOA, o que resulta em sete níveis de SOA, tal como o SIMM. Os níveis da proposta foram, de acordo com (SÖDESTRÖM; MEIER, 2007):

- Nível 7 – **Serviços dinamicamente reconfiguráveis**, controle de eventos automatizados e composição dinâmica de serviços.
- Nível 6 – **Serviços e processos medidos** e monitorados em tempo real.
- Nível 5 – Estabelecimento de **serviços internos e externos**, em toda a cadeia de suprimentos.
- Nível 4 – **Serviços arquitetados**, múltiplas aplicações integradas, controle de usuário e autenticação unificada com SSO – *Single Sign-On* e o ROI passa a ser provado.

- Nível 3 – **Institucionalização** através da inclusão no processo de software, centro de competências, estabelecimento de políticas de controle e redução de custos em integração.
- Nível 2 – **Publicação dos primeiros serviços**, implantação de infraestrutura de barramento de serviços, repositório de serviços, com a estimativa inicial do ROI.
- Nível 1 – **Testes de tecnologias** em projetos pilotos, capacitação e integração com legados.
- Nível 0 – **Arquitetura baseada em componentes** identificando funcionalidades e modularidades.

Outra importante contribuição realizada foi a definição de um modelo de avaliação da maturidade conforme as diferentes perspectivas dos departamentos de TI dentro da organização, conforme apresentado na Figura 2-21 (RATHFELDER; GROENDA, 2008).

Ponto de vista		Arquitetura de Serviços	Infraestrutura	Estrutura Corporativa	Serviço de Desenvolvimento	Governança
Nível de Maturidade						
5	SOA Sob Demanda	Serviços Dinâmicos	Serviços de Mercado	Serviços como Negócio	Serviços Sob Demanda	Automatizado
4	SOA Corporativa	Processos	Gerenciamento e orientação a eventos	Serviços Alinhados	Orientado a Modelo	Controle de Concorrência Justo
3	SOA Administrada	Serviços Orquestrados	Monitoração e Segurança	Centralmente Gerenciado	Documentado e com suporte de ferramentas	Regras
2	SOA Integração	Integração de Aplicações	Comunicação	Orientado a TI	Experiência na prática	Guias
1	SOA Avaliação	Ilhas	Heterogênea	Separada	Não Estruturada	Nada

Figura 2-21. Níveis de maturidade (RATHFELDER; GROENDA, 2008).

Devem-se gerenciar os investimentos na melhoria da maturidade de forma coesa, de modo a abranger as diversas áreas. A evolução de somente uma das áreas não é possível de ser realizada por completo, pois existem dependências entre serviços de outras áreas que precisam responder em um mesmo nível.

2.7.2 Guia para métodos para engenharia de software orientada a serviços

Devido à diversidade de métodos de engenharia de software aplicados à Orientação a Serviços, realizou-se um guia de seleção de métodos conforme as necessidades da organização (GU; LAGO, 2011). Neste trabalho foram avaliadas as características de cada um dos doze métodos: CBDI-SAE-TM (Component Based Development and Integration for Service Architecture & Engineering), SOA SDLC (Systems Development Life Cycle), SeCSE (Service Centric Systems Engineering), SOMA (Service Oriented Modeling and Architecture), OASIS SOA-RM (Reference Model), SOAD (Service-Oriented Analysis and Design), SOSE (Service-Oriented Software Engineering), TRUE, CHANG, SODIUM (Service-Oriented Development In a Unified fraMework), SOAF (Service-oriented architecture framework) e SOUP (Service-Oriented Unified Process).

Esta avaliação definiu critérios comuns para avaliação das diferentes propostas de métodos de engenharia de software aplicados a SOA. Os critérios utilizados foram baseados nos questionamentos a seguir:

- Objetivo: Qual o objetivo deste método?
- Artefatos: O método define produtos de trabalho para suas atividades?
- Guias: O método especifica como realizar os produtos de trabalho?
- Procedimentos: O método descreve os procedimentos (atividades, tarefas ou passos) para cada fase do ciclo de vida?
- Princípios: O método possui orientação ou guias sobre os princípios chaves do ambiente de desenvolvimento? Alguns princípios, geralmente apresentados, são o baixo acoplamento e alinhamento de negócio e TI.
- Ferramentas: Existem ferramentas para dar suporte aos procedimentos e técnicas do método?
- Maturidade: Este método tem sido aplicado na indústria?
- Gerenciamento: Este método suporta a disciplina de gerenciamento de projetos?
- Integração: Quais são as técnicas existentes compatíveis com o método?

As comparações dos aspectos genéricos representados pelas questões estão representadas na Tabela 2-3 e

Tabela 2-4. Os aspectos avaliados por escala foram representados com: “-“ para sem suporte, “+” para suporte implícito, “++” para suporte explícito, e “+++” para suporte robusto e detalhado.

Tabela 2-3. Comparação dos aspectos genéricos em métodos SOA – parte 1, adaptado de (GU; LAGO, 2011).

	CBDI-SAE	SDLC	SeCSE	SOMA	SOAF	SOUP
Objetivo	Engenharia de Processo	Engenharia de Processo	Engenharia de Processo	Engenharia de Processo	Migração SOA	Gerenciamento de projetos
Artefatos	+++	+	+++	++	+++	+++
Notação	-	++	++	+++	+	-
Procedimentos	++	++	+++	++	+++	++
Princípios	+++	+++	-	+++	+++	-
Ferramentas	-	++	+++	-	-	-
Maturidade	Indisponível	Não	Sim	Sim	Sim	Indisponível
Gerenciamento	++	-	++	++	++	+++
Integração	-	RUP, CBD, BPM	-	OOAD, RUP, BPM	-	RUP, XP

Tabela 2-4. Comparação dos aspectos genéricos em métodos SOA – parte 2, adaptado de (GU; LAGO, 2011).

	OASIS	SOAD	SOSE	TRUE	CHANG	SODIUM
Objetivo	Análise e Projeto	Análise e Projeto	Análise e Projeto	Análise e Projeto	Composição de Serviços	Composição de Serviços
Artefatos	+++	-	++	+++	+++	+++
Notação	++	-	++	+++	+++	++
Procedimentos	+++	-	++	+++	++	++
Princípios	++	-	++	++	++	++
Ferramentas	-	-	-	-	-	+++
Maturidade	Sim	Sim	Sim	Sim	Indisponível	Não
Gerenciamento	+++	-	-	-	-	-
Integração	-	OOAD, BPM	-	-	SOUP, SOMA	-

O trabalho analisou o critério de formalidade para a tomada de decisão, onde, buscou-se por matrizes decisórias baseadas em definições matemáticas ou lógicas, assim como, também analisou a presença de suporte a customização do método para atender as necessidades da organização ou do domínio de negócio. Entretanto, os doze métodos avaliados não atenderam a ambos os pontos (formalismos e o suporte à customização).

Para cada atividade do ciclo de vida dos métodos SOA avaliou-se a escala de atendimento para os doze métodos, conforme a Tabela 2-5. De modo geral,

percebeu-se que a maioria dos métodos dá ênfase às atividades de prover serviços. Em segundo plano estão as atividades para consumo de serviços. As atividades de registro no repositório possuem um fraco suporte em todos os métodos, ocorrendo apenas na CBDI-SAE, SOAD e SODIUM. Quanto às atividades para construção de aplicações baseadas em serviços, a única atividade coberta é a engenharia de requisitos e apenas nos métodos CBDI-SAE, SeCSE, CHANG e SODIUM.

Tabela 2-5. Atividades avaliadas nos métodos para SOA (GU; LAGO, 2011).

Papel	Atividade do ciclo de vida
Provedor de Serviços	Análise de mercado
	Engenharia de requisitos
	Modelagem de negócios
	Projeto de serviços
	Desenvolvimento de serviços
	Testes de serviços
	Publicação de serviços
	Provisionamento de serviços
	Monitoração de serviços
Gerenciamento de serviços	
Repositório de serviços	Seleção do registro
	Atualização do registro
	Manutenção do registro
Consumidor do serviço	Descoberta de serviços
	Composição de serviços
	Negociação de serviços
	Invocação de serviços
	Monitoração de serviços
Construção de aplicação	Engenharia de requisitos
	Projeto da aplicação
	Desenvolvimento da aplicação
	Testes do módulo
	Testes da aplicação
	Manutenção da aplicação

Um conjunto de aspectos específicos direcionados aos métodos SOA foram avaliados, tais como: o suporte às mudanças no ambiente externo, à definição de serviços, à criação de serviços, à descrição dos papéis, à associação entre papéis e atividades, nas mudanças arquiteturais, nas atividades em tempo de execução, nos requisitos não funcionais, à variabilidade entre os diferentes conjuntos de serviços, às perspectivas (fornecimento e consumo), ao desenvolvimento colaborativo por múltiplas organizações. Nestes aspectos o método SeCSE se destacou devido à sua completude e descrições detalhadas no suporte à engenharia de serviços.

2.8 Considerações sobre o capítulo

Este capítulo apresentou uma revisão sobre o cenário atual da literatura, pesquisas da academia e da indústria de software. Com esta fundamentação foi possível analisar assuntos relacionados à engenharia de processo de software, os modelos de maturidade e processos que estão em pleno uso pela indústria. Os conceitos de Linhas de Produto de Software e de Linhas de Processo de Software foram apresentados. Por fim, SOA e BPM foram fundamentados juntamente com os modelos de processos de engenharia aplicados.

CAPÍTULO 3 - ESTRUTURAÇÃO DA PESQUISA

O segredo de seguir em frente é começar. O segredo de começar é dividir suas tarefas complexas e difíceis em tarefas pequenas e gerenciáveis, e depois começar pela primeira.

- Mark Twain, escritor norte-americano

Este capítulo visa apresentar a abordagem metodológica aplicada para desenvolvimento deste trabalho de pesquisa. Para tanto, se faz necessário abordar algumas referências para estabelecer os conceitos necessários. Após, será caracterizada a forma de trabalho e definida a estratégia e as etapas utilizadas pelo pesquisador para estruturar seu trabalho para alcançar os objetivos previamente definidos.

3.1 Metodologia e Métodos de Pesquisa

Para Marconi e Lakatos (2010) a pesquisa é “um procedimento formal, com método de pensamento reflexivo, que requer um tratamento científico e se constitui no caminho para conhecer a realidade ou para descobrir verdades parciais”. De modo similar Gil (2002) define pesquisa como:

“(...) o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. A pesquisa é requerida quando não se dispõe de informação suficiente para responder ao problema, ou então quando a informação disponível se encontra em tal estado de desordem (...)”. (p. 17)

De acordo com Silva e Menezes (2005) existem quatro perspectivas clássicas para classificação de uma pesquisa, são estas: **natureza, forma de abordagem do problema, objetivos e procedimentos técnicos**.

Dentro da primeira perspectiva da pesquisa está o **ponto de vista de sua natureza**, onde a pesquisa é definida como sendo (SILVA; MENEZES, 2005):

- Pesquisa Básica: busca-se gerar novos conhecimentos para avanço da ciência envolvendo interesses universais e sem aplicações práticas.

- Pesquisa Aplicada: busca-se gerar conhecimentos para aplicações práticas e destinados a problemas e interesses específicos ou locais.

A classificação da pesquisa quanto ao **ponto de vista da forma de abordagem do problema**, classifica-se em dois grupos: pesquisas quantitativas e pesquisas qualitativas (SILVA; MENEZES, 2005). A pesquisa quantitativa aplica procedimentos para classificação, quantificação e análises utilizando-se de técnicas estatísticas. Já a pesquisa qualitativa explora opiniões, características e interpretações do universo em análise, sem requerer a aplicação de técnicas estatísticas. Para apoiar a escolha do método de pesquisa Martins (2008), orienta que: “(...) pode-se afirmar que avaliações quantitativas são mais adequadas ao processo de testar teorias, enquanto as avaliações qualitativas são mais aplicáveis em situações onde se deseja construir teorias (...)”.

Quanto ao **ponto de vista dos objetivos** as pesquisas são classificadas em três grandes grupos, sendo estes (GIL, 2002):

- Pesquisa Exploratória: busca uma visão geral sobre o assunto aprimorando as ideias do pesquisador, aprofundando os conhecimentos sobre determinado assunto, auxiliando na identificação de problemas e a definição de hipóteses e guiando os passos iniciais para descobertas. O planejamento desta pesquisa é bastante flexível, geralmente esta pesquisa envolve levantamento bibliográfico, ou estudos de casos, ou entrevistas com pessoas com experiência relacionada ao problema.
- Pesquisa Descritiva: busca realizar a descrição das características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis. Esta é uma classificação de estudo muito comum onde geralmente são utilizadas técnicas padronizadas na coleta de dados, como: questionário e observação sistemática.
- Pesquisa Explicativa: busca identificar os fatores determinantes para ocorrência dos fenômenos, ou os fatores contribuintes. Entre as três classificações quanto ao objetivo, esta é a que mais aprofunda o conhecimento da realidade, buscando o porquê e a razão das coisas. Este é o tipo de pesquisa mais complexa, onde os riscos de erros nos resultados são potencializados. Uma pesquisa pode iniciar como descritiva e torna-se explicativa, onde a descrição das características/variáveis leva

à busca da identificação dos fatores determinantes. Geralmente utilizam-se métodos experimentais ou quase experimentais.

Na classificação relacionada ao objetivo é estabelecido um marco teórico, mas para realização da pesquisa é necessário torná-la aplicável através de modelos e procedimentos padronizados, onde a pesquisa é definida quanto ao **ponto de vista dos procedimentos técnicos**. Dentro desta classificação, (GIL, 2002) divide em dois grupos, são eles: fontes em papel e dados fornecidos por pessoas, conforme ilustrado na **Erro! Fonte de referência não encontrada.**

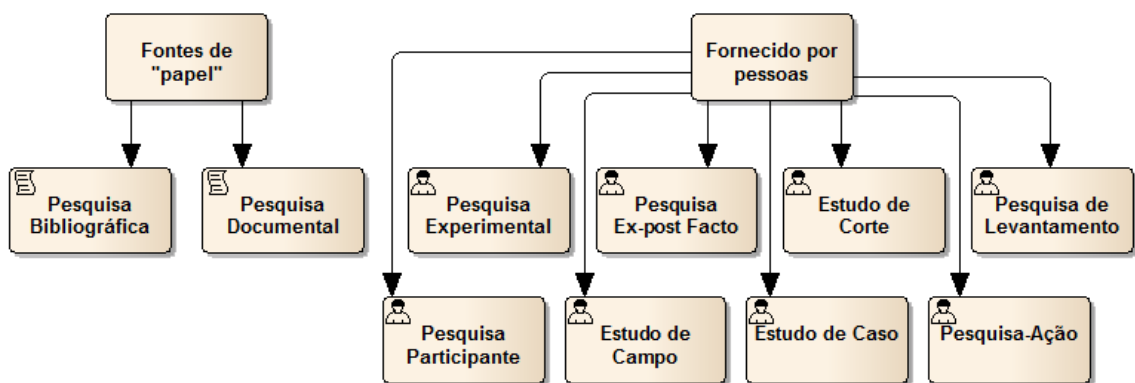


Figura 3-1. Classificação das pesquisas com base nos procedimentos técnicos, adaptado de (GIL, 2002).

Dentro do primeiro grupo, das fontes de papel, estão (GIL, 2002):

- Pesquisa Bibliográfica: desenvolvida com base em material já elaborado e publicado, sendo as fontes preferenciais livros, artigos científicos e normas.
- Pesquisa Documental: muito semelhante à pesquisa bibliográfica, esta é diferenciada devido à origem das fontes, onde, ao invés de materiais publicados, utilizam-se materiais que ainda não receberam tratamento analítico, ou que ainda podem ser reelaborados.

Dentro do segundo grupo, onde os dados são fornecidos por pessoas, estão as pesquisas (GIL, 2002):

- Pesquisa Experimental: representa o melhor exemplo de pesquisa científica, onde através da observação de um objeto, deve-se: selecionar as variáveis, propor um modelo comportamental, definir as formas de controle, mensurar, analisar e avaliar o resultado. O experimento pode ser repetido para coleta de novos dados, evolução do modelo proposto com

novas hipóteses. A principal limitação é a dificuldade ou impossibilidade de manipulação de algumas variáveis, seja devido a custos, esforço ou questões éticas.

- *Ex-post Facto*: busca a partir de fatos passados, analisar e relacionar as variáveis, identificando um modelo comportamental. Diferencia-se da pesquisa experimental devido à falta de formas de controle. Esta pesquisa não apresenta resultados totalmente seguros sobre causa e efeito, mas constata a relação dentro do histórico analisado, por isso, costuma-se denominá-la de pesquisa correlacional.
- Estudo de Corte: analisa-se uma amostra, grupos de pessoas com características comuns, investigando os acontecimentos relacionados em um período determinado de tempo.
- Pesquisa de Levantamento: é realizada através da interrogação direta às pessoas envolvidas no problema estudado, seguido da análise quantitativa dos dados coletados.
- Estudo de Campo: assemelha-se à pesquisa de levantamento, porém possui um nível maior de profundidade, ao invés de amplitude. O pesquisador realiza grande parte do trabalho pessoalmente, através de imersão na realidade para entendimento das regras e costumes do grupo em estudo (geralmente uma comunidade).
- Estudo de Caso: consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira a permitir um amplo e detalhado conhecimento.
- Pesquisa-Ação: estabelece ampla atuação dos pesquisadores e participantes da situação (problema) de modo cooperativo e participativo, caracteriza-se por uma ação planejada de caráter social, educacional, técnico, ou outro (THIOLLENT, 2005).
- Pesquisa Participante: caracteriza-se pela interação entre participantes e pesquisadores, muito semelhante à pesquisa-ação, porém sem ações planejadas pelo pesquisador. Os participantes, juntamente com o pesquisador, definem as bases de pesquisas, objetivos, hipóteses e o cronograma de atividades (MARCONI; LAKATOS, 2010).

3.2 Caracterização da pesquisa

Conforme as classificações de pesquisa apresentadas no **Erro! Fonte de referência não encontrada.**, a **natureza** desta pesquisa é classificada como **aplicada**, pois trabalha em um problema específico buscando uma abordagem prática para solução. A **abordagem do problema** é tanto **qualitativa** como **quantitativa**, utilizando métodos estatísticos nas etapas de avaliação. Quanto ao **objetivo** esta pesquisa é **exploratória**, pois busca uma visão geral das propostas atuais de processos e suas fraquezas, e propõe uma linha de processos de software para agilizar a definição de modelos para SOA e BPM. Quanto aos **procedimentos técnicos**, esta pesquisa se caracteriza como uma **pesquisa de desenvolvimento**, que será avaliada utilizando **opinião de especialistas (survey)**. A Tabela 2-1 resume a classificação da pesquisa para a concepção da linha de processo de software orientado a serviços.

Tabela 3-1. Classificação da pesquisa.

Do ponto de vista:	Classificação
da sua natureza	Aplicada
da forma de abordagem do problema	Qualitativa e Quantitativa
de seus objetivos	Exploratória
dos procedimentos técnicos	Desenvolvimento e Survey

Fonte: o Autor, 2012.

3.3 Questão de pesquisa

A questão principal desta pesquisa pode ser representada através da seguinte frase: **“É viável apoiar o desenvolvimento de aplicações SOA e BPM utilizando um processo de desenvolvimento de software instanciado a partir de um conjunto de variabilidades de uma linha de processos?”**

Baseados no levantamento bibliográfico, realizado e apresentado nos capítulos anteriores foram identificadas as seguintes proposições:

- **Proposição 1:** Os elementos fundamentais da linha proposta atendem às necessidades para a derivação de novos processos.
- **Proposição 2:** O modelo proposto é capaz de gerenciar as variabilidades, complementando os elementos fundamentais com os elementos adicionais relativos à realidade das organizações.
- **Proposição 3:** A transição entre as fases e as atividades da linha atendem às necessidades, sem precisar de customizações.

- **Proposição 4:** A fusão das práticas de SOA e BPM em uma única linha de processos é adequada e aplicável para as atividades dos engenheiros de software.
- **Proposição 5:** O tempo para derivação de um novo processo de uma linha de processo pode ser reduzido e ou automatizado através de ferramentas adequadas.

3.4 Estratégia de pesquisa

Nesta seção está descrita a estratégia da pesquisa, resumida de forma gráfica na Figura 3-2. Para realizar o desenvolvimento da pesquisa, cada uma das etapas definidas foi detalhada nas subseções seguintes.

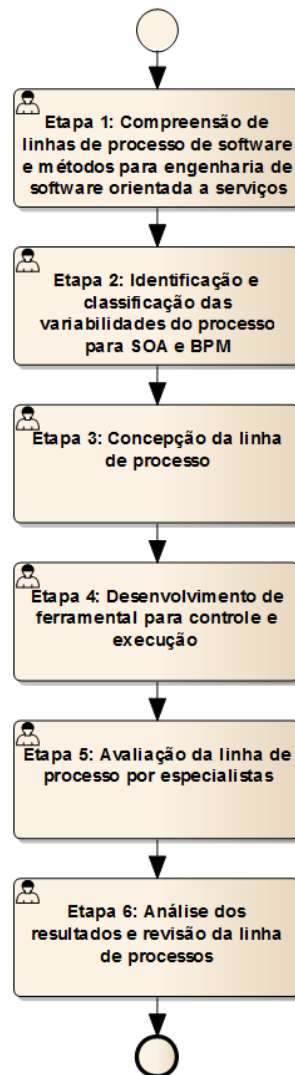


Figura 3-2. Etapas da pesquisa. Fonte: o Autor, 2012.

3.4.1 Etapa 1 – Compreensão sobre as áreas de conhecimento

A compreensão sobre as áreas de conhecimento busca capacitar o pesquisador para torná-lo apto à realização deste trabalho. O trabalho envolve a compreensão dos modelos de processo de software existentes, desde os métodos clássicos aplicados na academia e mercado, até os métodos propostos para a construção de aplicações baseadas em SOA. O pesquisador também deve dominar a atividade de um engenheiro de processo de software, que é responsável pela customização dos processos de engenharia de software. Por fim, o pesquisador deve realizar um estudo sobre as propostas existentes para linha de processo de software, baseado no levantamento bibliográfico da academia e da indústria.

3.4.2 Etapa 2 – Identificação das variabilidades

A Etapa 2 analisa os métodos para desenvolvimento de software orientados a serviços (SOA e WOA) e busca identificar as variabilidades existentes nos modelos para construção da linha. Da mesma forma que para a orientação a serviços, também para BPM deve-se identificar as variabilidades no processo de engenharia de software. Juntamente com a identificação das variabilidades, os elementos do processo, ou seja, atividades e tarefas, devem ter suas variabilidades classificadas, de modo a aprimorar a produtividade no processo de instanciação de novos processos da linha.

3.4.3 Etapa 3 – Concepção da linha de processo

A etapa 3 busca realizar a concepção e documentação da linha de processos de software proposta, incluindo a definição e descrição de todas as fases, atividades, transições, tarefas, entradas, saídas, produtos de trabalho, papéis, variabilidades e seus respectivos critérios de seleção.

Uma forma de representação da linha de processo será utilizada para concepção e documentação baseada na notação vSPEM (MARTÍNEZ-RUIZ et al., 2011)..

3.4.4 Etapa 4 – Definição de ferramentas para a linha de processo

Definição, seleção e contribuição ao ferramental para modelagem da linha de processo de software. Além da modelagem, busca-se utilizar o mesmo componente

de infraestrutura de BPM para realizar o controle, execução e monitoração dos processos derivados da linha.

3.4.5 Etapa 5 – Avaliação da linha de processos por especialistas

A avaliação por especialistas torna-se uma boa opção uma vez que a análise de um modelo de processo envolve um grande custo e tempo. O preparo de um experimento, identificando os cenários e variáveis presentes na indústria, de modo a permitir o controle e medições sobre as variáveis é uma tarefa árdua. Como alternativa, a pesquisa sobre opiniões de especialistas ajuda na obtenção de informações valiosas para avaliação de modo mais ágil. Segundo Cooke (1991), a opinião de especialistas está relacionada com: “as especulações, suposições e estimativas de pessoas que são considerados especialistas, na medida em que estes servem como **entrada cognitiva** em algum processo decisório”. Neste processo, o mais importante para Cooke (1991) é: “(...) o desenvolvimento de modelos práticos com **fundamentos matemática transparente** para o uso das opiniões dos especialistas na ciência”.

O processo de avaliação por opinião de especialistas seguiu os seguintes passos conforme (LI, SMIDTS, 2003)(MOSLEH; BIER; APOSTOLAKIS, 1987):

- Passo 1 - Declaração de problema: O problema deve ser claramente definido, detalhado e compreendido, assim como os demais problemas e variáveis indiretamente relacionadas ou presentes no mesmo ambiente.
- Passo 2 - Seleção dos especialistas: Um número razoável de especialistas deve ser identificado através de um conjunto de critérios, que deve incluir: a credibilidade, o conhecimento, a habilidade e a confiabilidade.
- Passo 3 – Preparação e treinamento dos especialistas: O treinamento possui um importante aspecto de familiarizar os especialistas com os objetivos da pesquisa e os detalhes das questões envolvidas. O objetivo do treinamento é assegurar que haja um entendimento comum do conteúdo abordado e que os especialistas sejam capazes de responder as mesmas questões.
- Passo 4 - Elicitação das opiniões: Este passo capta as questões certas de todos os especialistas e busca oferecer as condições favoráveis para este exercício de elicitación.

- Passo 5 - Agregação de opinião: A ideia básica é alcançar uma opinião agregada ou um consenso embasado, o qual deve permitir que decisões sejam tomadas. Esta opinião agregada é geralmente uma escala de valores que representam as alternativas e que é chamada de “utilidade”. A decisão por análise e melhorias geralmente é necessária nas questões com o menor ou o maior valor de utilidade.
- Passo 6 - Tomada de Decisão: Este passo realiza a decisão com base no consenso das opiniões.

Logo, conclui-se que a principal atividade do pesquisador na avaliação por especialistas é definir um modelo prático para transformar a opinião dos especialistas em um processo decisório baseado em fundamentos matemáticos (COOKE, 1991). Optou-se, neste trabalho, por seguir os passos propostos por (LI, SMIDTS, 2003) (MOSLEH; BIER; APOSTOLAKIS, 1987), conforme detalhado a seguir.

3.4.5.1 Declaração do problema

A avaliação buscou identificar a completude e aplicabilidade da proposta de linha de processo para processos de engenharia de software para produtos orientados a serviços. A pesquisa foi realizada por meio de especialistas que analisaram os diagramas e o conteúdo descritivo das fases, atividades, tarefas e respectivas variabilidades da linha de processos. Além disso, esta avaliação buscou captar sugestões dos especialistas com ênfase na: necessidade de adição de elementos no processo; transição entre as atividades; e, inclusão de variabilidades na linha de processos.

3.4.5.2 Seleção dos especialistas

Para escolha dos especialistas os critérios de seleção foram definidos pelo conhecimento do profissional, explorado pela participação em projetos de boa representação e experiência. Considerou-se projetos de boa representação os que tiveram um volume de trabalho superior a 5.000 horas. Já a experiência do profissional buscou-se pelo menos 2 anos com atividades ligadas a pelo menos uma das seguintes áreas: processo de engenharia de software e suas respectivas disciplinas; engenharia de produtos orientados a serviços; e, modelagem de processos de negócio.

Estes critérios selecionaram 30 especialistas com conhecimento nas áreas de processos de software, orientação a serviços ou modelagem de negócio. Entretanto, 8 destes profissionais estavam trabalhando em projetos no exterior e não puderam participar da avaliação. Após contato, 16 deles se dispuseram a participar da avaliação da proposta. Evitou-se a conveniência no processo de seleção, pois esta abordagem traria uma ótica de uma ou poucas organizações e regiões, inserindo certo viés. Desta forma, foi aceito no máximo um profissional para cada organização, o que eliminou 4 profissionais, resultando em 12 especialistas selecionados.

Inicialmente, cada especialista preencheu itens para caracterização do seu perfil. Este formulário está apresentado no A. O objetivo era analisar o conhecimento e o tempo de experiência em relação a cada uma das áreas necessárias ao estudo: processo de software, orientação a serviços e BPM. A seguinte escala foi definida com relação conhecimento sobre o assunto:

- 5 – Excelente;
- 4 – Bom;
- 3 – Razoável;
- 2 – Superficial;
- 1 – Desconheço o assunto.

3.4.5.3 Preparação e treinamento dos especialistas

A capacitação do especialista para entendimento da proposta foi realizada através do comunicado com um material preparado com diagramas e o conteúdo descritivo da proposta de linha de processo. Também foram realizadas conferências telefônicas para explicar e facilitar o entendimento dos participantes sobre o processo de avaliação. O contato direto facilitou o esclarecimento de dúvidas e captação preliminar de algumas opiniões.

3.4.5.4 Elicitação das opiniões

Para elicitar a opinião dos especialistas foi elaborado um instrumento de pesquisa na forma de questionário de avaliação da proposta. Este questionário foi introduzido aos especialistas por uma carta de apresentação onde as partes da avaliação foram apresentadas. O questionário está dividido em seis etapas, conforme APÊNDICE , onde:

- Parte 1 – Caracterização do especialista;

- Parte 2 – Avaliação da visão Geral das fases da linha;
- Parte 3 – Análise da fase de Modelagem de negócios;
- Parte 4 – Análise da fase de Identificação;
- Parte 5 – Análise da fase de Especificação;
- Parte 6 – Análise da fase de Construção;
- Parte 7 – Sugestões gerais de melhoria

A parte 1 identifica e caracteriza o conhecimento e o nível de experiência dos especialistas em processo de software, SOA e BPM. Para a parte 2 do questionário até a parte 6, foram definidas 4 questões objetivas. Nestas questões objetivas buscou-se avaliar: a completude e suficiência da linha de processo; a necessidade de adição de elementos na linha de processos; a coerência nas transições definidas; e, a existência de mais pontos de variabilidades e também variabilidades propriamente ditas, que não tivessem sido descritas. Por fim, a parte 7 abre para sugestões gerais do trabalho proposto. Ao final do questionário está o Termo de Confidencialidade, o qual pode ser visualizado no APÊNDICE .

Para coleta das opiniões foi optado por utilizar uma escala de Likert, composta por cinco valores:

- 5 – Concordo totalmente: o especialista concorda com o item apresentado e julga que não sejam necessárias mudanças;
- 4 – Concordo parcialmente: o especialista concorda com o item apresentado, mas julga que sejam necessárias pequenas mudanças;
- 3 – Indiferente: o especialista julga o item avaliado como algo muito próximo na média mínima para atendimento o objetivo;
- 2 – Discordo parcialmente: o especialista discorda em diversos aspectos do item proposto;
- 1 – Discordo totalmente: o especialista discorda totalmente do item proposto e sugere mudanças ou até mesmo a exclusão e substituição total do item.

3.4.5.5 Agregação de opinião e avaliação dos resultados

Os dados coletados com os especialistas foram agregados, visando produzir uma visão qualitativa sobre cada item da avaliação.

No formulário, juntamente as respostas optativas, havia campos de justificativa onde os especialistas puderam contribuir com opiniões pessoais no formato de questão aberta. Estas informações foram agregadas e analisadas, os pontos críticos foram destacados e itens de correções foram identificados para melhoria da proposta. Estes resultados serão apresentados no Capítulo 5.

3.4.6 Etapa 6 - Análise dos resultados e revisão da linha

Com base na agregação de resultados, é possível identificar, para cada item da linha de processo proposta, se é possível considerar o item como satisfatório ou se é necessário introduzir alguma mudança. Para a análise dos resultados, optou-se por uma escala, conforme a representada na Figura 3-3:

- Aceito: caso todos os especialistas assinalem os itens CP (concordo parcialmente) ou CT (concordo totalmente), o item será considerado como adequado.
- Tendência de Aceitação: caso haja algum especialista apontando para o item IND (indiferente), então o item será considerado como tendo uma tendência de aceitação, mas as sugestões serão analisadas. Será considerado nesta situação também o caso em que houver menos do que 3 votos para o lado oposto.
- Neutro: caso todos os especialistas apontem para o item IND (indiferente), então será considerado que são indiferentes. Neste caso o item também será considerado como aceito.
- Revisão: caso todos os especialistas assinalem os itens DT (discordo totalmente) ou DP (discordo parcialmente), o item será considerado como sujeito à revisão.
- Tendência de Rejeição: caso haja algum especialista apontando para o item IND (indiferente), então o item será considerado como tendo uma tendência de rejeição, mas as sugestões serão analisadas. Será considerado nesta situação também o caso em que houver menos do que 3 votos para o lado oposto.
- Demais casos: quando as opiniões estiverem distribuídas entre as várias faixas, o item será considerado como sem consenso. Neste caso serão avaliadas as sugestões e incorporadas, caso pertinente.

A Figura 3-3 resume a abordagem utilizada para análise dos resultados da revisão da linha.

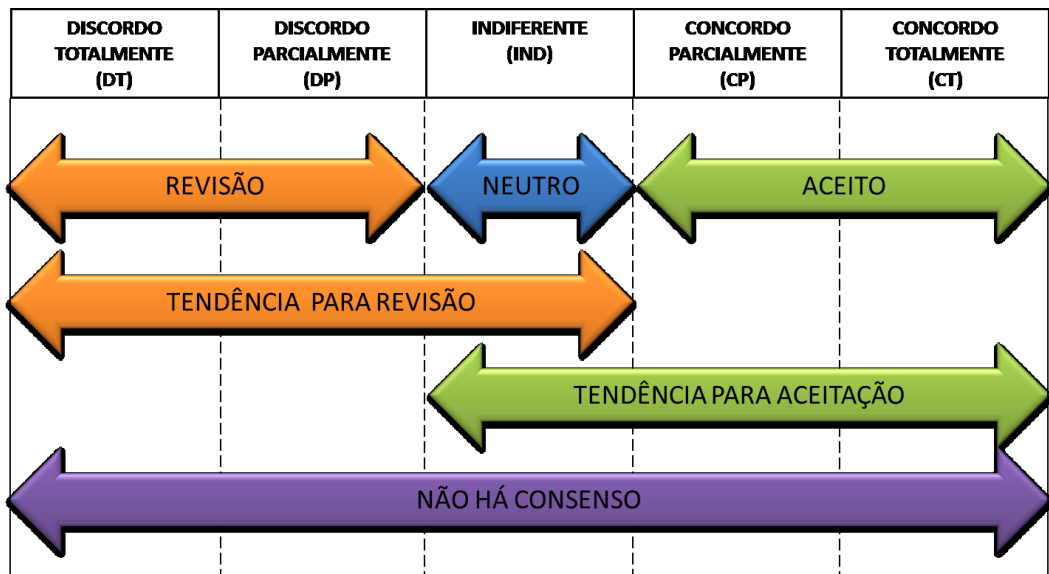


Figura 3-3. Interpretação dos resultados. Fonte: o Autor, 2012.

3.5 Considerações sobre o capítulo

Neste capítulo foram apresentados os conceitos de métodos de pesquisas científicas e suas classificações quanto a objetivo e procedimentos técnicos, para desta forma, caracterizar o trabalho de pesquisa aqui proposto. As etapas da pesquisa foram apresentadas e detalhadas buscando esclarecer e delimitar os procedimentos a serem executados ao longo da execução deste trabalho.

CAPÍTULO 4 - DEFINIÇÃO DA LINHA DE PROCESSOS PARA SOA E BPM

A primeira regra de qualquer tecnologia é que a automação aplicada sobre uma operação eficiente irá engrandecer a eficiência. O segundo é que a automação aplicada a uma operação ineficiente irá ampliar a ineficiência. - Bill Gates, fundador da Microsoft

Este capítulo define as características e elementos da linha que foi denominada como SOPLA – *Service-Oriented Process Line Approach*. Conforme evidenciado em estudos empíricos, os fatores críticos para o sucesso de programas de reuso mais apontados são respectivamente (REINEHR, 2008): Gerenciamento, Treinamento, Presença de um processo para reuso, Projeto para reuso, Bibliotecas e componentes. Visando atender a estes pontos, este trabalho busca propor uma linha de processos voltada ao reuso de produtos de software, direcionada para projetos SOA e BPM.

Para implantação sistemática de SOA e BPM são necessários procedimentos, treinamentos e governança organizacional, pois isto requer análise e decisões contínuas para o projeto e desenvolvimento desta arquitetura de processos de negócio, sistemas e infraestrutura. A linha SOPLA busca acelerar a definição de um processo para suportar a entrega de soluções baseadas nesta arquitetura.

4.1 Introdução

A proposta SOPLA é uma linha de processos de software para reduzir o trabalho do engenheiro de processos, na qual, com a utilização da seleção de variabilidades, este será capaz de gerar um novo processo. Os passos que o engenheiro de processos fará através da linha são:

- Passo 1: Ao iniciar um novo projeto, deve-se analisar suas características;
- Passo 2: Selecionar as variabilidades da linha de processo;
- Passo 3: Gerar uma instância do processo derivando da linha somente as variabilidade selecionadas. Este passo é realizado através do ferramental EPF (EPF, 2012) e GenArch-P (ALEIXO et al, 2012);

- Passo 4: Converter a instância do processo para BPMN (*Business Process Modeling Notation*) e configurar o fluxo para execução do processo. Esta conversão é realizada através de um ferramental proposto, que realiza a conversão do metamodelo do EPF (UMA - *Unified Method Architecture*) para BPMN 2.0. Para execução do BPMN 2.0 utilizou-se a plataforma de BPM Activiti¹², mas devido a padronização pode-se utilizar diversas plataforma compatíveis de BPM.

O passo inicial realizado para concepção da linha foi o levantamento das características e requisitos para definição do escopo da linha. Ao analisar os modelos de governança para BPM e SOA percebe-se a necessidade de um processo muito bem integrado para planejamento conforme os objetivos organizacionais, elaboração do plano estratégico, realização dos projetos para as mudanças e a monitoração ativa dos indicadores de negócio para alcance da melhoria contínua. Entretanto, estes processos possuem muitas características próprias de cada organização respeitando a cultura e a forma de gerenciamento organizacional.

A concepção da linha enfatizou um escopo direcionado à execução, ou seja, foram detalhadas as atividades de projeto, construção e testes do produto. A documentação da linha deteve-se aos elementos de maior granularidade do SPEM (fases e atividades), detalhamento os elementos de menor granularidade (tarefas e passos) nos casos mais relevante ao contexto SOA e BPM. As atividades relacionadas ao planejamento, gerenciamento de mudanças, medição, treinamento, operação e monitoração não fazem parte do escopo deste trabalho. Enfatizando as atividades de execução do processo, foram identificadas as seguintes características para concepção da linha e suas variabilidades:

- Suportar várias abordagens de mapeamento e modelagem de processos;
- Incentivar o reúso sistemático;
- Permitir a inclusão de atividades de gestão da informação e segurança;
- Utilizar diferentes técnicas de estimativas;
- Suportar a especificação de novos serviços, adaptação ou melhorias;

¹² Informações referentes à ferramenta Activiti, estão no site da ferramenta disponível em <<http://activiti.org/>>, consultado em 12/07/2012.

- Permitir instâncias de processos mais simples e leves para realizar a construção através de ferramentas de composição;
- Permitir a inclusão de diferentes combinações de tipos de testes manuais e automatizados, tanto para interfaces, quanto para os serviços de negócio;
- Mapear atividades de configuração geralmente realizadas nos componentes de infraestrutura;
- Mapear os diferentes tipos de testes automatizados utilizados em SOA.

A notação utilizada para documentar a linha proposta foi baseada no SPEM, onde se descreveu o processo com 3 elementos, conforme Figura 4-1. Os elementos utilizados são (OMG, 2008a):

- O uso da tarefa: elemento que a partir da definição de uma tarefa estabelece sua localização temporal no processo;
- A atividade: que define um trabalho em um nível mais alto podendo agrupar tarefas e papéis; e,
- A Fase: que representa um período significativo de tempo com um objetivo estabelecido, um marco de finalização juntamente com um conjunto de entregas.

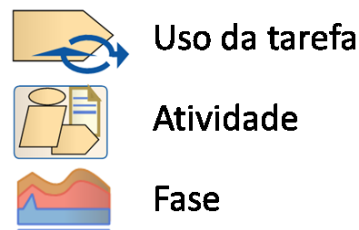


Figura 4-1. Simbologia SPEM utilizada, adaptado de (OMG, 2008a).

A abordagem utilizada para documentar a linha foi baseada no vSPEM (MARTÍNEZ-RUIZ et al., 2011) utilizando o conceito de **ponto de variação**, o qual permite instanciar um novo processo da linha que será substituído por uma de suas **variantes**. Por padrão, cada elemento representado na linha de processo é **mandatório**, sempre que omitida a definição, este é obrigatório. Cada ponto de variação, também denominado de componente abstrato (BARRETO et al., 2010), contém um tipo de classificação da sua variabilidade, a saber:

- Tipo Alternativo, também chamado de “XOR”, onde apenas uma das variantes pode ser instanciada;
- Tipo “OR” que permite a instância de mais de uma das variantes.

Para compor a linha, a variabilidade padrão do SPEM versão 2.0 chamada **contribui** também foi utilizada para a adição de elementos extras para colaboração com outro elemento existente na linha. A variabilidade padrão utilizada para representação das variantes foi a **estende e substitui**, onde nas variantes as informações omitidas são herdadas do ponto de variação. Para documentar os elementos que podem ou não existir em uma instância da linha de processo, especificou-se a variabilidade como **opcional**, representado por ‘?’. A variabilidade opcional pode ser aplicada tanto para pontos de variação, quanto para elementos sem variantes. A Figura 4-2 apresenta de forma gráfica o resumo dos tipos de variabilidade utilizados na linha, semelhante ao vSPeM Figura 2-8 previamente apresentada de (MARTÍNEZ-RUIZ et al., 2011), porém com símbolos específicos para representar as restrições nos pontos de variabilidades.









	Mandatário mínimo 1	Opcional mínimo 0	Instância das variabilidades
Elemento do processo somente com variabilidade opcional			-
Ponto de variação Alternativo máximo 1			
Ponto de variação OU máximo N			

Figura 4-2. Tipos de variabilidades. Fonte: o Autor, 2012.

Um dos critérios para seleção das variabilidades da linha está associado ao tamanho do projeto. Para isso definiu-se uma regra geral para orientação com a definição de três tamanhos de projeto, conforme descrito na Tabela 4-1. Estes critérios podem ser ajustados para melhor atender ao planejamento e gerenciamento, conforme as necessidades de cada organização.

Tabela 4-1. Classificação de tamanho de projeto.

Tamanho do projeto	Esforço total de horas
Tamanho P	Projeto pequeno, com esforço até 400h
Tamanho M	Projeto mediano, com esforço até 1200h
Tamanho G	Projeto grande, como esforço superior a 1200h

Fonte: o Autor, 2012.

4.2 Conteúdo do método SOPLA

O conteúdo do método é a parte conceitual utilizada para a definição da base de conhecimento das organizações, onde são mantidos os elementos independentes do processo e dos projetos de desenvolvimento (OMG, 2008a). Esta parte do processo é (OMG, 2008a) facilmente reutilizável entre qualquer processo, até mesmo fora da linha proposta, pois estes elementos não possuem vínculo temporal. Dentro da linha proposta foram utilizados alguns papéis tradicionais do UP, acrescidos de papéis adicionais mais comuns em grandes e médias organizações. A lista de papéis a seguir está vinculada às atividades como responsáveis e participantes.

- Patrocinador: papel fundamental, responsável por fornecer o cenário de negócios e finanças para o projeto. Exerce a governança do projeto, processo de negócio, indicadores de medição do processo e soluções que suportam o processo.
- Usuário chave: representa o patrocinador no levantamento de informações mais detalhadas sobre os processos de negócio.
- Analista de negócio: busca conhecer profundamente o processo de negócio mapeando e conhecendo as pessoas envolvidas e suas atividades e ferramentas. Modela e projeta o modelo futuro e indicadores (SHUJA; KREBS, 2008);
- Arquiteto da solução: responsável por definir os princípios e plataformas que devem orientar o projeto detalhado da solução. O arquiteto de solução precisa ser um profissional sênior com domínio e experiência em diversas plataformas de desenvolvimento e nos diferentes componentes de infraestrutura SOA.
- Analista de sistemas: deve compreender claramente o problema capturando, descrevendo e delimitando as funcionalidades a serem entregues pela solução. Possui extensões com perfil em:

- SOA: responsável pela modelagem dos serviços e suas respectivas mensagens. Este profissional deve dominar os componentes SOA para reutilizá-los e compor a solução.
- *User experience*: responsável pelo projeto da interface do usuário que deve conhecer o perfil deste para projetar uma experiência agradável de uso.
- Projetista de banco de dados: responsável pela modelagem, construção e otimização do banco de dados e deve possuir conhecimento das informações da organização para reutilizá-las.
- Arquiteto de segurança da informação: profissional que preza pela segurança das informações estruturadas e não estruturadas (documentos). Responsável pela classificação da informação, recomendações e definição de padrões de segurança. Apoia o *Chief Security Officer* (CSO), e deve primar pela conformidade com as normas, como: ISO 27001, Sarbanes-Oxley, entre outras.
- Arquiteto de infraestrutura: profissional responsável pela administração e instalação das plataformas utilizadas na organização que apoia processo de seleção das plataformas utilizadas na solução.
- Administrador de dados: profissional que apoia o desenvolvimento através da definição, revisão e padronização dos tipos de dados da organização.
- Programador: responsável pela construção do produto com domínio e experiência nas ferramentas e linguagens a serem utilizadas. Possui extensões com perfil em:
 - SOA: profissional que possui os conhecimentos necessários para desenvolver as regras de negócio e expô-las em serviços;
 - *User experience*: profissional especialista na codificação de interfaces para o usuário com conhecimentos avançados em tecnologias de apresentação.
 - Testador: papel responsável pela condução dos testes; definição, execução e análise dos problemas; registro de defeitos; coleta de evidências e comunicação com a equipe.

A definição da entrada e saída de cada atividade é realizada por meio dos produtos de trabalho. Para a linha SOPLA foram propostos os seguintes artefatos:

- Documento de visão: documento que define as necessidades, os objetivos, o(s) processo(s) de negócio envolvido(s) e os cenários envolvidos, além dos requisitos e diferenciais de alto nível (SHUJA; KREBS, 2008);
- Requisitos de negócio: representam um esboço dos principais requisitos para o negócio, geralmente apresentando um esboço dos requisitos da solução segundo a ótica do patrocinador do projeto.
- Modelo futuro do processo de negócio: define o modelo futuro do processo de negócio e é geralmente decomposto em sub-processos, atividades e tarefas. Devem ser capturadas informações sobre os pontos fortes e fracos e estabelecer indicadores de desempenho para o processo de negócio (KPI = *key performance indicators*). Para realizar a modelagem do processo de negócio futuro, dois outros artefatos podem ser necessários, são estes: Modelo atual do processo de negócio e Inventário de ferramentas e relação com as atividades do processo (JESTON; NELIS, 2008).
- Modelo de domínio: especifica as informações envolvidas na solução, descrevendo os tipos de dados utilizados em serviços SOA, pacotes de classes ou banco de dados. Os elementos de dados utilizados devem estar vinculados ao Glossário dos termos do negócio juntamente com a definição das informações.
- Lista de serviços e aderência: representa a principal saída referente à identificação de reuso, onde se deve avaliar o percentual de aderência dos serviços existentes e classificá-los. O documento deve conter a lista de serviços candidatos para a solução, bem como os serviços e componentes dependentes para compor a solução.
- Documento de Arquitetura de Software: documento que registra as decisões, raciocínio e suposições para orientar a equipe no projeto e construção da solução. Deve identificar os principais componentes arquiteturais, como: serviços, componentes, camadas da aplicação, mecanismos de abstração, banco de dados e componentes de infraestrutura (SHUJA; KREBS, 2008).
- Modelo de caso de uso: este artefato permite uma visão rápida e geral dos casos de uso e inter-relacionamentos (SHUJA; KREBS, 2008).

- Modelo de entidade e relacionamento: Utilizado para construção do banco de dados.
- Especificação descritiva dos requisitos da solução: especificação que busca descrever a solução de maneira simplificada.
- Especificação de caso de uso: documento descritivo que contém o conjunto de ações sequenciais entre os fatores (geralmente um tipo de usuário e o sistema) (SHUJA; KREBS, 2008).
- Especificação suplementar: descreve as demais restrições da solução complementando as definições comportamentais do caso de uso (SHUJA; KREBS, 2008).
- Mapa de navegação: define a sequência de navegação entre as telas do usuário.
- Protótipo das interfaces da solução: este artefato descreve o projeto das interfaces tendo como principal intuito o atendimento aos requisitos e a validação da solução a ser desenvolvida e entregue (OpenUP, 2011).
- Especificação do serviço: descreve o objetivo do serviço, o nível de serviço, as dependências, as regras de negócio, o contrato com a descrição das mensagens, os tipos de dados e o canal de comunicação (ARSANJANI et al., 2008).
- Registro de aceite e lista de não conformidades: artefato utilizado nas etapas de revisão com o objetivo de reportar problemas e não conformidades nos artefatos entregues. Este artefato é utilizado para reportar para avaliação da documentação, tanto em verificações internas, como em validações externas realizadas pelo usuário chave.
- Plano de Teste: documento que estabelece os objetivos, dependências e a sequência de execução dos casos de testes.
- Caso de Teste: descreve um conjunto de entradas, condições e resultados esperados para uma parte determinada da solução (SHUJA; KREBS, 2008).
- *Mock* dos serviços: artefato que criado para simular o comportamento do serviço que, conforme as condições de entrada, gera um conjunto de saídas estáticas.
- Registro de execução dos testes e aceite: registra as informações coletadas durante a execução dos casos de testes. Com a utilização deste

documento é gerada uma evidência da execução dos testes (SHUJA; KREBS, 2008).

4.3 Processo da linha

Inicialmente foram definidas as fases do ciclo de vida, para posteriormente identificar as suas atividades e variabilidades. Essas fases foram baseadas na análise nos 12 métodos de engenharia de software aplicados à Orientação a Serviços (GU; LAGO, 2011), sendo mais influenciada pelos modelos com forte integração com: BPM, RUP, OOAD e CBD. Esta integração foi documentada na última linha da Tabela 2-3 e Tabela 2-4

O ciclo de vida foi representado por meio de fases que também possuem variabilidades. Por exemplo, pode não ser necessária a execução de determinada fase, conforme a classificação de tamanho do projeto. As fases do ciclo de vida propostas são:

- Modelagem de Negócios e Transformação: esta fase é opcional para projeto sem impacto nos processos de negócio e de tamanho P, conforme Tabela 4-1;
- Identificação e Definição da Arquitetura;
- Especificação e Projeto dos Serviços: esta fase é opcional para projeto de tamanho P e que não requeira a criação de serviços;
- Construção, Composição e Testes;
- Liberação, Execução e Monitoração.

A fase inicial é a Modelagem de Negócio e Transformação, onde são analisadas mudanças dos processos de negócio da organização. A fase seguinte é a Identificação e Definição da Arquitetura, a qual pode iniciar a fase de Especificação e Projeto dos Serviços e também a fase de Construção, Composição e Testes. A partir da fase de Construção, Composição e Testes, são liberados novos pacotes para a execução, assim como, é realizada a monitoração dos KPIs definidos previamente. Cada uma das fases apresentadas na Figura 4-3 foram detalhadas por fluxos, apresentando as atividades e suas variabilidades nos sub-tópicos 4.3.1, 4.3.2, 4.3.3 e 4.3.4.

O modelo adotado para o ciclo de vida entre estas fases da linha processo é o iterativo e incremental (apresentado na Figura 4-3) que se caracteriza pela presença central da fase de identificação onde se deve buscar e analisar os serviços e o

domínio de dados existentes. A Identificação é um dos passos chaves no reúso, pois esta é a fase principal que centraliza as atividades de busca de artefatos para reutilização, entretanto o reúso também ocorre nas demais, porém com menor ênfase. O fluxo principal entre as fases está representado pela numeração e passa pela especificação que é opcional, alternativamente pode-se ir diretamente para a fase de Construção, Composição e Testes. Após cada fase de Construção, pode-se optar pela fase de Liberação, Execução e Monitoração ou por retornar para a fase de Identificação e Definição da Arquitetura. As transições entre as fases do modelo foram muito influenciadas pelo modelo SOMA (ARSANJANI et al., 2008) com adaptações do autor e inserções das variabilidades.

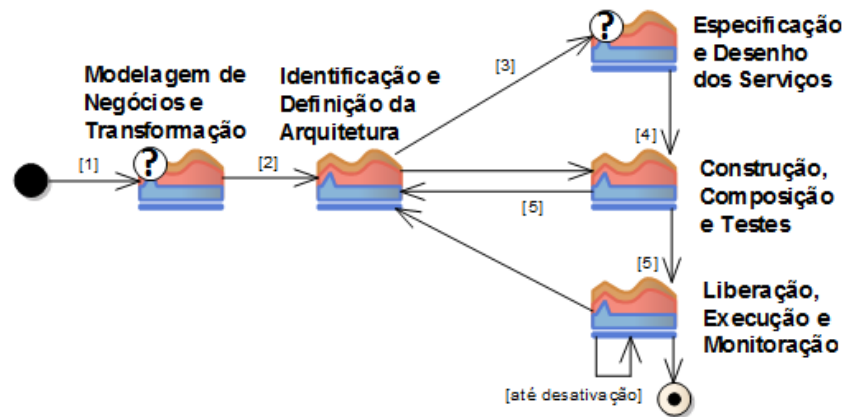


Figura 4-3. Fases do SOPLA. Fonte: o Autor, 2012.

Ao final de cada iteração ou das fases algumas das características do projeto podem ser alteradas, gerando a necessidade de adequação do processo. Quando mudanças nas características ocorrerem, os passos 1 a 4 apresentados no início do capítulo 4.1 devem ser novamente executados, para seleção das variabilidades adequadas, geração de uma nova instância do processo e atualizar a versão do processo em execução na plataforma de BPM.

Para cada fase serão apresentadas as atividades em tabelas descrevendo: o nome da atividade; a descrição; o tipo da atividade; opcionalmente são informadas as variabilidades e os critérios de seleção; o profissional responsável pela realização; os participantes envolvidos; os pré-requisitos (artefatos de entrada); o artefato de saída (produto resultante da execução) e as ferramentas utilizadas.

4.3.1 Fase de Modelagem de Negócios e Transformação

A fase de Modelagem de Negócios e Transformação busca estabelecer os objetivos de negócio para o projeto, motivação da mudança e principais requisitos. Estes objetivos de negócio devem ser claros, compreensíveis para o patrocinador, assim como para possíveis comitês de gestão do projeto que devem ter um consenso sobre a mudança proposta. Para a transformação do processo de negócio são fundamentais: o tratamento adequado das informações, a comunicação no momento correto e o convencimento da organização sobre a mudança do processo de negócio. Ao final desta fase, devem ter sido avaliados e simulados os cenários futuros e selecionado o cenário adequado juntamente com seus indicadores de desempenho. A Figura 4-4 apresenta a transição das atividades e a

Tabela 4-2 descreve as atividades para a fase de Modelagem de Negócios e Transformação.

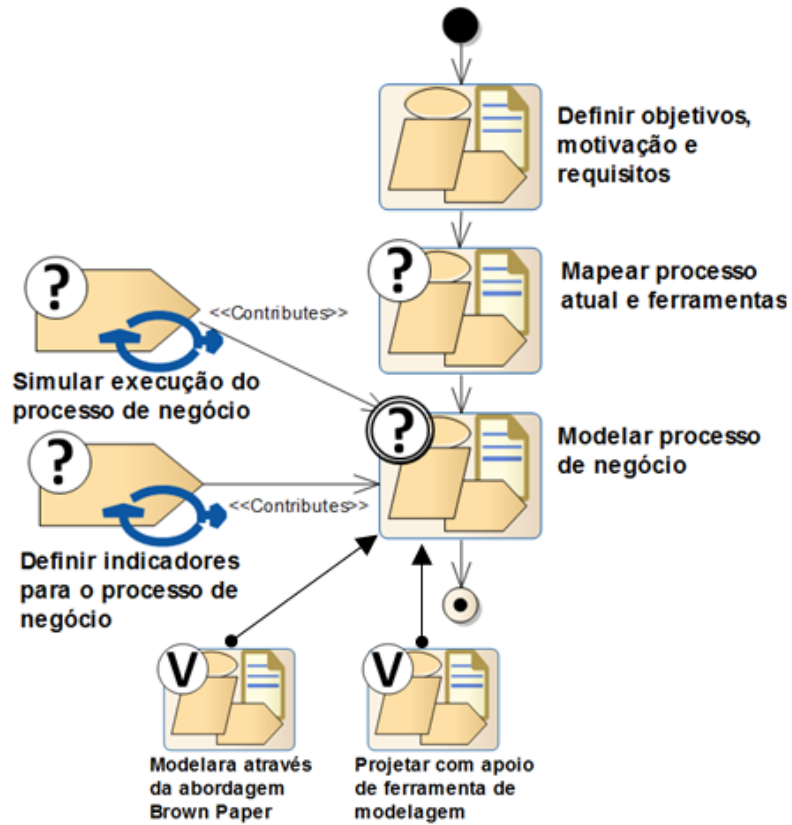


Figura 4-4. Fase de Modelagem de Negócios e Transformação. Fonte: o Autor, 2012.

Tabela 4-2. Atividades para fase de Modelagem de Negócios e Transformação

Identificador:	ATIV_OBJETIVO_REQ_NEG
Nome:	Definir objetivos, motivação e requisitos
Descrição:	Esta atividade busca identificar as necessidades do mercado, elicitar os objetivos estratégicos do projeto e a motivação do negócio na realização do investimento. Deve-se alinhar as metas indicadoras estratégicas do negócio com os ganhos operacionais esperados. A visão da mudança de processo e/ou sistema é captada, estabelecida e documentada de forma textual através de restrições/requisitos do projeto e do produto.
Tipo:	Atividade
Tarefas:	<ul style="list-style-type: none"> - Identificar os patrocinadores; - Compreender e descrever a necessidade ou mudança do mercado; - Elicitar e descrever os objetivos estratégicos do projeto e os requisitos de negócio; - Vincular os objetivos do projeto ao plano estratégico da organização; - Estabelecer o escopo da solução e as funcionalidades; - Validar a visão do produto.
Responsável:	Analista de negócio
Participante:	Patrocinador e usuário chave
Entrada:	Requisição de solução de negócio ou mudança
Saída:	Documento de visão Requisitos de negócio Glossário dos termos do negócio
Ferramenta:	Editor de texto
Origem:	Adaptado de (OpenUP, 2011) e (SHUJA; KREBS, 2008)
Identificador:	ATIV_PROC_ASIS
Nome:	Mapear processo atual e ferramentas
Descrição:	Esta atividade descreve como compreender e dar visibilidade ao processo atual através da representação deste modelo, identificando os atores, as atividades atuais e passos realizados.
Tipo:	Atividade
Variabilidade:	Opcional
Quando:	Processo novo, que ainda não é executado na organização.
Tarefas:	<ul style="list-style-type: none"> - Identificar o representante principal do processo (usuário-chave); - Descrever objetivo do processo, relação com a estratégia de negócio e indicadores utilizados; - Listar os papéis envolvidos no processo e localizá-los na estrutura (departamentos e organizações parceiras); - Levantar as ferramentas utilizadas no processo atual, como: planilhas de controle; sistemas existentes e as <i>user interfaces</i> utilizadas; pontos de decisão e demais fontes de informação. - Identifica e avaliar os problemas e oportunidades de ganhos.
Responsável:	Analista de negócio
Participante:	Usuários chave
Entrada:	Documento de visão
Saída:	Modelo atual do processo de negócio Inventário de ferramentas e relação com as atividades do processo
Adaptado de:	Adaptado de (SHUJA; KREBS, 2008) (JESTON; NELIS, 2008) (BPM-CBOK, 2009)
Identificador:	ATIV_PROC_TOBE
Nome:	Modelar processo de negócio
Descrição:	Nesta atividade o(s) processo(s) de negócio futuro (<i>TO-BE</i>) é detalhado e subdividido. A definição do fluxo de trabalho é realizada

	identificando as pessoas, ferramentas, sistemas e aplicativos envolvidos para cada tarefa do fluxo. O objetivo é aprimorar este fluxo de trabalho para melhorar o seu desempenho, aprimorar o nível de serviço, e/ou reduzir custo.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU opcional
Tarefas:	<ul style="list-style-type: none"> - Analisar o fluxo de trabalho, passos e atividades críticas, buscar por ineficiências e oportunidades de melhoria, supressão de atividades, alteração de executores, automação e padronização; - Consultar repositórios de melhores práticas e normas relacionadas ao processo; - Desenhar o modelo futuro; - Descrever os papéis envolvidos no processo; - Detalhar os requisitos de negócio; - Identificar as tarefas manuais e oportunidade de automatização; - Novos requisitos de negócio são capturados complementando os requisitos identificados previamente; - Iniciar a definição dos requisitos para realização da solução; e - Capturar os impactos do processo futuro para iniciar um plano das mudanças organizacionais.
Responsável:	Analista de negócio
Participante:	Usuários chave
Entrada:	Documento de visão Requisitos de negócio
Saída:	Modelo futuro do processo de negócio Requisitos de negócio
Origem:	Adaptado de (SHUJA; KREBS, 2008) (JESTON; NELIS, 2008) (BPM-CBOK, 2009)
Identificador:	ATIV_PROC_TOBE_BP
Nome:	Modelar processo de negócio – Brown Paper
Descrição:	Esta atividade favorece o trabalho coletivo, onde no processo são detalhados os passos, pontos de decisão, fontes de informação e identificação. A técnica estabelece o entendimento comum, compartilhando o conhecimento entre área de negócio, cliente e equipe de TI. Devem-se identificar os pontos fracos e as respectivas oportunidades e mudanças para alcance da melhoria.
Variabilidade:	Variante de ATIV_PROC_TOBE
Quando:	Reengenharia do processo de negócio que requer reinventar o processo de negócio e envolver as pessoas chaves que realizam o processo. Abordagem recomendada para grande impacto no negócio, ou quando equipe de TI possui baixo conhecimento no processo.
Ferramenta:	Mesa grande, local amplo com paredes largas e espaçosas, Brown Paper, Post-its e canetas.
Identificador:	ATIV_PROC_TOBE_FE
Nome:	Modelar processo de negócio – Ferramenta de Modelagem
Descrição:	Esta atividade favorece a realização de melhorias, sem reinventar o processo de negócio. Pode trazer uma mudança significativa, sem grandes impactos na organização. Abordagem favorece o trabalho individual ou de poucos agentes de mudança.
Variabilidade:	Variante de ATIV_PROC_TOBE
Quando:	Organização controla seus processos (certificação) com normatização dos processos de negócio. Pode ser realizada sem a abordagem anterior quando impacto baixo ou moderado e a organização possui pessoas capazes de realizar esta atividade de forma autônoma.
Ferramenta:	Editor de diagramas para processo de negócio,

	preferencialmente para BPMN – Business Process Modeling Notation. É possível documentar através de outros tipos de diagrama como: o Diagrama Event-driven Process Chain (EPC), o Diagrama de Atividades da UML ou o Diagrama Integration DEFinition for Function Modeling (IDEF0).
Identificador:	TAR_PROC_SIMULA
Nome:	Simular execução do processo de negócio
Descrição:	Esta tarefa busca simular a execução do processo e fluxos de trabalhos para validar os ganhos do novo processo ou melhoria. A simulação contribui com o modelo futuro de negócio explorando as possibilidades de melhoria e automação e validando os ganhos efetivos no processo. O resultado da simulação sobre o processo negócio futuro apoia a tomada de decisão, assim como a escolha entre diferentes modelos projetados.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Projeto gera mudança no processo negócio, modifica o fluxo das atividades, papéis dos envolvidos, e resulta em impacto financeiro significativo.
Passos:	<ul style="list-style-type: none"> - Capturar informações para simulações como volume de trabalho, tempos do ciclo, quantidade das operações e qualidade; - Realizar simulações para os cenários; - Analisar resultados, impactos, riscos e resumir informações para decisão; e - Elaborar um comparativo das propostas e alternativas avaliadas.
Responsável:	Analista de negócio
Entrada:	Modelo atual do processo de negócio Modelo futuro do processo de negócio
Saída:	Modelo futuro do processo de negócio
Ferramenta:	Ferramenta para simulação, tais como: Aris ou Enterprise Architect ou ferramentas de propósito comum para simulação de logística.
Origem:	Adaptado de (OMG, 2008b) (BPM-CBOK, 2009) (JESTON; NELIS, 2008)
Identificador:	TAR_PROC_KPI
Nome:	Definir indicadores para o processo de negócio
Descrição:	Esta tarefa descreve uma abordagem para definição dos indicadores chave de desempenho relacionando com: os objetivos de negócio (<i>top-down</i>) e também com as tarefas no fluxo de trabalho (<i>bottom-up</i>).
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	O impacto da solução no processo negócio precisa ser medido, devido à criticidade, tamanho da mudança, ou para justificar o valor da solução.
Passos:	<ul style="list-style-type: none"> - Levantar e detalhar metas para o processo de negócio e suas atividades; - Definir ou selecionar indicadores para controle dos processos que estejam em conformidade com os objetivos previamente capturados. - Descrever objetivos de negócio e indicadores; - Avaliar indicadores e esforço de implantação; e - Validar com o gestor do processo de negócio da organização.
Responsável:	Analista de negócio
Entrada:	Documento de visão Requisitos de negócio Modelo futuro do processo de negócio
Saída:	KPI - Indicadores chave de desempenho
Ferramenta:	Repositório de processos e indicadores Guia sobre indicadores referenciais de mercado
Origem:	Adaptado de SOMA (ARSANJANI et al., 2008) (JESTON; NELIS,

4.3.2 Fase de Identificação e Definição da Arquitetura

A fase de Identificação e Definição da Arquitetura tem como metas fortalecer o reúso e selecionar o padrão de arquitetura adequado, assim como estimar custo e prazo para a solução proposta. Os custos estimados devem ser avaliados e aprovados pelo patrocinador e/ou comitê gestor do projeto (atividade de planejamento fora do escopo da linha).

A entrada inicial desta fase é após a modelagem de negócio, pois como o ciclo do processo é iterativo, ao final de uma iteração (passando pelas fases de Especificação, Construção e Liberação) pode-se novamente executar a fase de Identificação. Neste caso, é possível realizar uma nova iteração de Especificação e Construção sem a Liberação do produto e ao final desta fase, os serviços de negócio candidatos são listados e classificados conforme nível de aderência aos requisitos. Os componentes principais da arquitetura e padrões são selecionados. A Figura 4-5 apresenta o fluxo das atividades e a Tabela 4-3 descreve as atividades desta fase.

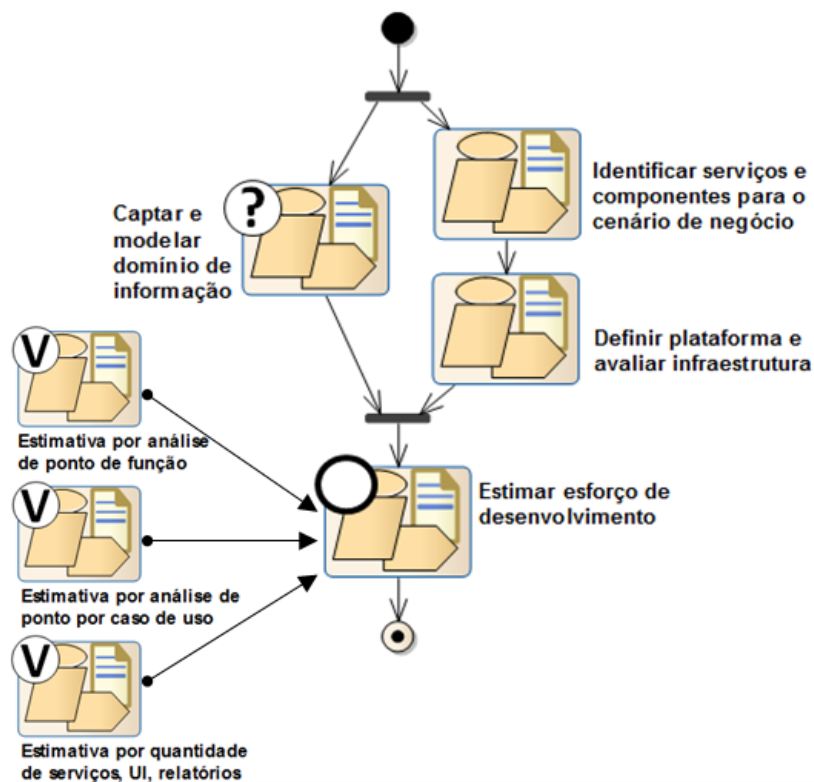


Figura 4-5. Fase de Identificação e Definição da Arquitetura. Fonte: o Autor, 2012.

Tabela 4-3. Atividades para fase de Identificação e Definição da Arquitetura

Identificador:	ATIV_IDENT_DOMINIO
Nome:	Captar e modelar domínio de informação
Descrição:	Esta atividade compreende a análise do processo e decomposição do domínio em entidades de informação e seus relacionamentos descritos nas mensagens dos serviços.
Tipo:	Atividade
Variabilidade:	Opcional
Quando:	Projeto tamanho M ou G
Tarefas:	<ul style="list-style-type: none"> - Captar vocabulário de negócio; - Consultar repositórios de serviços e mensagens existentes; - Avaliar a utilização de guia/padrões de mensagens buscando manter a semântica das mensagens; - Elaborar o contrato para as entidades de negócio; - Mapear com as tarefas do processo de negócio com a utilização dos serviços; e - Analisar dependências
Responsável:	Analista de sistemas
Participante:	Usuário chave
Entrada:	Requisitos de negócio Modelo futuro do processo de negócio Lista serviços e aderência (artefato em elaboração, não é necessária a conclusão da atividade que gera este artefato).
Saída:	Modelo de domínio (Rascunho das informações utilizadas) Glossário dos termos do negócio
Ferramenta:	Editor de XML Editor de diagramas UML Repositório de serviços
Origem:	Adaptado de SOMA (ARSANJANI et al., 2008)
Identificador:	ATIV_IDENT_SRV
Nome:	Identificar serviços e componentes para o cenário de negócio
Descrição:	Esta atividade busca identificar os serviços e componentes candidatos para a solução, quer sejam os serviços: existentes, a serem adquiridos ou desenvolvidos. Para cada serviço existente ou passível de aquisição deve-se analisar a aderência às necessidades.
Tipo:	Atividade
Tarefas:	<ul style="list-style-type: none"> - Analisar as necessidades do processo de negócio; - Identificar serviços candidatos através da base de artefatos de processos e serviços; - Consultar o repositório de serviço e listar serviços reutilizáveis; - Identificar as características e importância dos serviços; - Avaliar percentual de aderência dos serviços existentes, conforme: 1) Reutilização de serviços sem modificações, apenas incluindo um novo consumidor; 2) Criação de serviços a partir de serviços existentes combinando os seus contratos; 3) Extensão de funcionalidades e/ou inclusão de elementos, mantendo a compatibilidade com os consumidores que já utilizam a versão anterior; 4) Opção menos recomendada, onde se cria a réplica do serviço para modificar drasticamente uma funcionalidade, geralmente uma nova versão do serviço é gerada, mas sem manter compatibilidade (devem ser avaliados os mecanismos de abstração); 5) Novo serviço candidato (inexistente na base).
Responsável:	Analista de sistemas
Participante:	Analista de negócio e Arquiteto da solução
Entrada:	Requisitos de Negócio Modelo futuro do processo de negócio
Saída:	Lista de serviços e aderência
Ferramenta:	Repositório de serviços Planilha eletrônica

Origem:	Adaptado de (ARSANJANI et al., 2008)
Identificador:	ATIV_IDENT_PLATF
Nome:	Definir plataforma e avaliar infraestrutura
Descrição:	Esta atividade busca definir a(s) plataforma(s) técnica(s) apropriada(s) e os componentes de infraestrutura necessários para a solução. A organização deve possuir uma equipe que defina os padrões de arquitetura corporativa e estabeleça os modelos referenciais de arquitetura.
Tipo:	Atividade
Responsável:	Arquiteto da solução
Participante:	Arquiteto de infraestrutura Arquiteto de segurança da informação. Analista de sistemas
Tarefas:	<ul style="list-style-type: none"> - Verificar requisitos de negócio e funcionais e analisar pré-requisitos da solução; - Acordar e documentar os requisitos não funcionais; - Definir os componentes de infraestrutura do software a serem utilizados para: serviços e roteamento (ESB), controle de Jobs, gerenciamento e execução de processos (BPMS), monitoração (BAM), regras de negócio (BRM), de documentos e colaboração (ECM – Enterprise Content Management), segurança e impressão; - Avaliar a utilização de ambientes ou serviços existentes; e - Avaliar custos de contratação de SAAS – <i>Software as a Service</i>, PAAS – <i>Platform as a Service</i> e IAAS – <i>Infrastructure as a Service</i>. - Descrever decisões e recomendações; - Definir o padrão de qualidade de serviço (QoS); - Analisar capacidade dos ativos de SW e HW (QoS); - Para projetos tamanho G e críticos, deve-se submeter a proposta para avaliação da equipe de arquitetura corporativa.
Entrada:	Requisitos de negócio Modelo futuro do processo de negócio Lista de serviços e aderência
Saída:	Documento de Arquitetura de Software – Rascunho com a identificação da plataforma, infraestrutura e justificativa. Juntamente com uma matriz de oportunidade com custo e benefício das propostas.
Ferramenta:	Editor de texto Repositório de serviços (Opcional) Formulários de dimensionamento: para novas soluções ou grandes modificação no ambiente são utilizados formulários de dimensionamento do fabricante (<i>sizing</i>). (Opcional) Projeção de carga: Para melhorias pode-se monitorar o ambiente e projetar a nova carga da solução, ou ainda, utilizar ferramentas de simulação de carga, conforme atividade ATIV_DESEN_TESTE_HOMO_CARGA.
Origem	Autor
Identificador:	ATIV_IDENT_ESTIMAR
Nome:	Estimar esforço de desenvolvimento
Descrição:	Nesta atividade os artefatos gerados previamente são analisados e aplicam-se técnicas para conhecer o tamanho do produto. Através do tamanho do produto e a produtividade média da organização pode-se estimar o esforço (volume de horas) e os demais indicadores financeiro para análises de viabilidade.
Tipo:	Atividade
Variabilidade:	Ponto de variação alternativo mandatório
Tarefas:	<ul style="list-style-type: none"> - Buscar informações sobre o domínio da informação e nível de aderência dos serviços e componentes reutilizáveis; - Documentar o tamanho do software para cada serviço candidato resultante da decomposição e identificação;

	- Utilizar técnicas de estimativa para conhecer o tamanho do produto; - Calcular ajustes conforme requisitos não funcionais; e - Utilizar histórico para validação da estimativa.
Responsável:	Analista de Sistemas
Entrada:	Documento de visão Requisitos de negócio Modelo de domínio Lista de serviços e aderência
Saída:	Documento de visão
Origem:	(SEI, 2010a)(ABNT, 2008)(SOFTEX, 2011)
Identificador:	ATIV_IDENT_ESTIMAR_PF
Nome:	Estimativa por análise de ponto de função
Descrição:	Analisar e propor uma solução inicial para possibilitar a estimativa através da contagem por análise por pontos de função (FPA). Deve-se definir e aplicar os ajustes, conforme requisitos não funcionais da solução.
Variabilidade:	Variante de ATIV_IDENT_ESTIMAR
Quando:	Organização que já possuem base histórica para FPA e conhecimento no uso desta abordagem.
Ferramenta:	Base histórica de estimativas Planilha eletrônica
Identificador:	ATIV_IDENT_ESTIMAR_UCP
Nome:	Estimativa por análise de ponto por caso de uso
Descrição:	Esta atividade busca estimar o tamanho do projeto através da análise de pontos por caso de uso (UCP). Com esta análise também é proposto um modelo inicial da solução através de um rascunho dos casos de uso.
Variabilidade:	Variante de ATIV_IDENT_ESTIMAR
Quando:	Organização que já possuem base histórica para UCP e conhecimento no uso desta abordagem.
Saída:	Rascunho do modelo de caso de uso
Ferramenta:	Base histórica de estimativas Planilha eletrônica
Identificador:	ATIV_IDENT_ESTIMAR_TAB
Nome:	Estimativa por quantidade de serviços, telas, adaptações e relatórios
Descrição:	Esta atividade busca analisar e propor uma solução definindo a quantidade de serviços, telas e relatórios. Conforme a complexidade de cada um dos artefatos, identificar o item na tabela e somar o esforço médio de todos os artefatos.
Variabilidade:	Variante de ATIV_IDENT_ESTIMAR
Quando:	Organização não possui base histórica para FPA e nem para UCP.
Ferramentas:	Planilha eletrônica. Tabela com esforço conforme complexidade dos serviços, telas, adaptações e relatórios.

4.3.3 Fase de Especificação e Projeto dos Serviços

Na fase de Especificação e Projeto dos Serviços, os artefatos de negócio, juntamente com os produtos de software identificados para arquitetura, são analisados detalhadamente para projetar cada componente da solução. As principais entregas da fase são: as especificações detalhadas dos requisitos, as regras de

negócio, as regras de *user interface*, as restrições técnicas e a especificação de cada um dos componentes, as *user interfaces* e/ou serviços.

A especificação de adaptações dos serviços é uma atividade que preza por manter a compatibilidade com suas versões anteriores, de modo a evitar testes regressivos em diversos cenários de uso. Para isso, faz-se necessário o apoio do uso do repositório de serviços que mantenha esta documentação atualizada e vincule os consumidores dos diferentes cenários de uso. A alteração de um serviço com um elevado número de consumidores gera a necessidade de testes que podem até mesmo inviabilizar financeiramente o projeto. Em situações pode-se optar por manter mais versões ativas do mesmo serviço para reduzir o trabalho com testes.

A fase de Especificação e Projeto dos Serviços está detalhada na Figura 4-6 e Tabela 4-4, onde o fluxo das atividades é apresentado e as atividades e tarefas da fase são detalhadas.

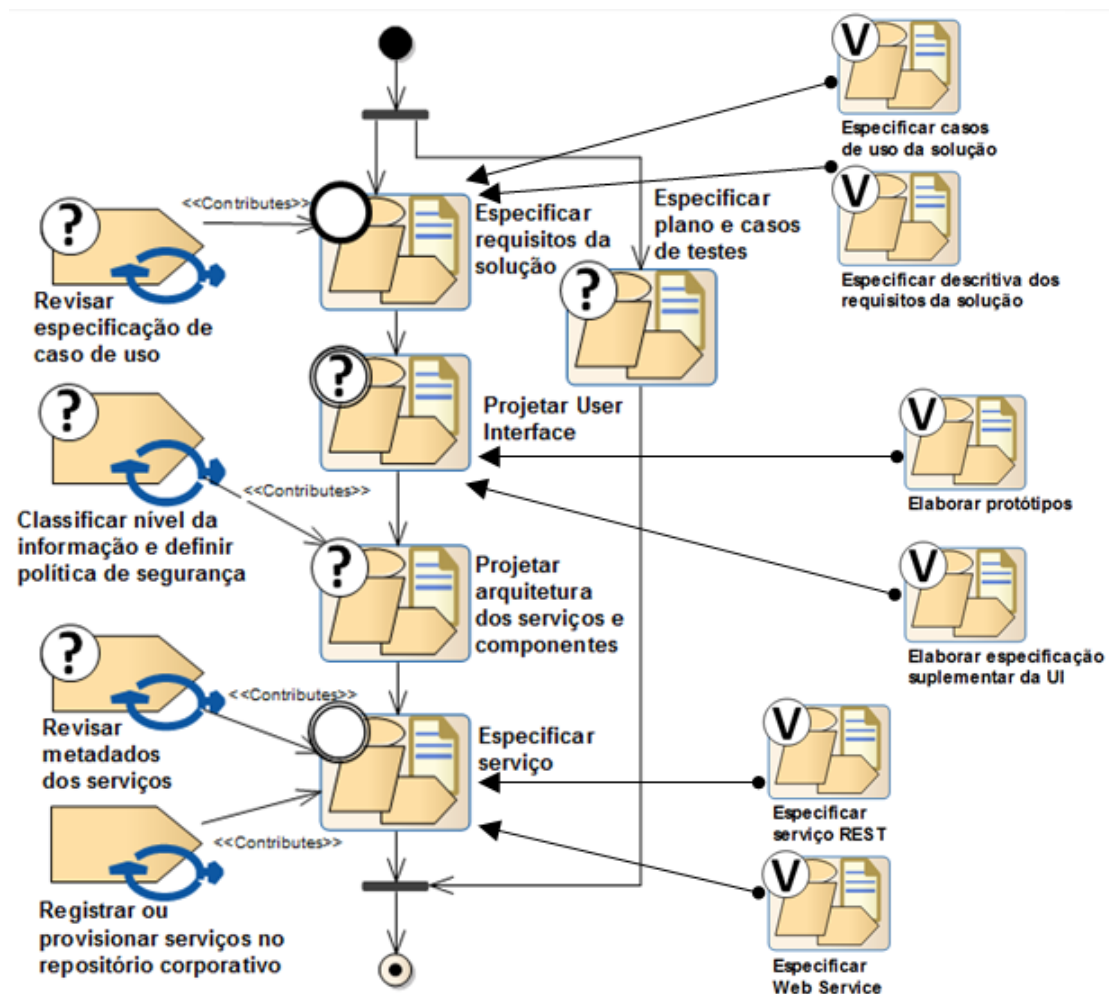


Figura 4-6. Fase de Especificação e Projeto dos Serviços. Fonte: o Autor, 2012.

Tabela 4-4. Atividades para fase de Especificação e Projeto dos Serviços

Identificador:	ATIV_ESPEC_REQ_SIST
Nome:	Especificar requisitos da solução
Descrição:	Verificar os papéis envolvidos no cenário de negócio; Identificar pontos do processo a serem realizados pela solução de modo a delinear as fronteiras.
Tipo:	Atividade
Variabilidade:	Ponto de variação alternativo mandatório
Tarefas:	- Revisar os requisitos de negócio; - Elicitar as definições pendentes, regras de negócio; - Avaliar se o escopo ou esforço foram impactados sendo necessária a execução do processo de gerenciamento de mudanças; - Detalhar cenário do sistema; - Descrever os passos de iteração com o sistema; - Descrever funcionalidades; e - Analisar dependências.
Responsável:	Analista de sistemas
Participante:	Usuário chave
Entrada:	Documento de visão Requisitos de negócio
Origem:	(OpenUP, 2011)
Identificador:	ATIV_ESPEC_REQ_SIST_UC
Nome:	Especificar casos de uso da solução
Descrição:	Elaborar modelo de casos de uso e identificar relações, comportamentos opcionais e abstratos; Detalhar cada caso de uso, especificando os fluxos principais e alternativos, assim como a pré-condição, a pós-condição e os demais requisitos especiais.
Tipo:	Atividade
Variabilidade:	Variante ATIV_ESPEC_REQ_SIST
Quando:	Estimativa for realizada por UCP e para projetos tamanho M e G
Saída:	Especificação de caso de uso
Ferramenta:	Editor de diagramas UML Editor de texto
Identificador:	ATIV_ESPEC_REQ_SIST_TXT
Nome:	Especificar descritiva dos requisitos da solução
Descrição:	Esta atividade compreende a elaboração de uma descrição textual das funcionalidades, detalhando o modo operacional da solução proposta para realização dos requisitos.
Variabilidade:	Variante de ATIV_ESPEC_REQ_SIST
Quando:	Projeto tamanho P
Saída:	Especificação descritiva dos requisitos da solução
Ferramenta:	Editor de texto
Identificador:	TAR_ESPEC_REQ_REV
Nome:	Validar especificação de requisitos
Descrição:	Esta tarefa tem como objetivo validar a especificação de requisitos e garantir sua conformidade com o escopo e expectativa do cliente.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Projeto tamanho M e G
Passos:	- O responsável pela validação recebe e avalia o documento; - Analisar conformidade conforme a visão do produto, escopo e funcionalidades esperadas; - Optar pelo aceite ou não da solução proposta; e - No caso de não aceitar, reportar a(s) não conformidade(s).
Responsável:	Usuário chave

Entrada:	Especificação de casos de uso
Saída:	Registro de aceite e lista de não conformidades
Ferramenta:	Checklist da especificação de requisitos Editor de planilha
Origem:	(SEI, 2010a)(ABNT, 2008)(SOFTEX, 2011)
Identificador:	ATIV_ESPEC_TEST
Nome:	Especificar plano e casos de testes
Descrição:	Esta atividade busca especificar detalhadamente os testes de verificação de modo a proporcionar estabilizar a solução antes dos testes de validação do usuário.
Tipo:	Atividade
Variabilidade:	Opcional
Quando:	Projeto tamanho M e G
Tarefas:	- Definir escopo, metas dos testes, abordagem, recursos e ambientes; - Descrever um fluxo lógico com os cenários a serem testados; - Preparar, para cada cenário, o conjunto de dados necessários; e - Para cada cenário, elaborar um passo a passo com as ações de entrada e condições esperadas do sistema.
Responsável:	Analista de sistemas
Participante:	Usuário chave
Entrada:	Caso de Uso
Saída:	Plano de Teste Caso de Teste
Ferramenta:	Editor de planilha
Origem:	Adaptado de (OpenUP, 2011) e (SHUJA; KREBS, 2008)
Identificador:	ATIV_ESPEC_UI
Nome:	Projetar <i>user interface</i>
Descrição:	Esta atividade compreende o projeto de uma interface de comunicação do usuário com o sistema, onde geralmente as características de usabilidade e utilidade são observadas. A saída é uma especificação como o projeto da interface (gráfica) do usuário.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU opcional
Tarefas:	- Identificar as características e necessidade dos usuários relacionados e seus meios de acesso a solução; - Estabelecer um mapa de navegação; - Agrupar informação e funcionalidade e projetar as <i>user interfaces</i> ; - Validar com o usuário que o projeto da user interface atende a sua necessidade; e - Verificar guia de usabilidade.
Responsável:	Analista UX
Participante:	Usuário chave
Entrada:	Documento de Arquitetura de Software Especificação de caso de uso Caso de Teste
Ferramenta:	Checklist de usabilidade
Origem:	Adaptado de (SHUJA; KREBS, 2008)
Identificador:	ATIV_ESPEC_UI_PROT
Nome:	Elaborar protótipos
Descrição:	Esta atividade busca definir um projeto que possibilite a avaliação do usuário chave. O projeto deste protótipo deve simular o funcionamento das <i>user interfaces</i> e a experiência de navegação no sistema.
Variabilidade:	Variante de ATIV_ESPEC_UI
Quando:	- Em situações que a aplicação deve ser intuitiva, pois não haverá treinamento, à exemplo os <i>web sites</i> . - Produto requer boa usabilidade, devido a restrições de projeto, como: sistemas embarcados, dispositivos móveis.

	- Aplicação com muitos usuários (superior a 30 usuários) de uso contínuo (mais de 1 transação por usuário a cada 5 minutos).
Saída:	Mapa de navegação Protótipo das interfaces da solução
Ferramenta:	Ferramenta de prototipação: utilizar editor de apresentação, planilha eletrônica, uma linguagem de apresentação ou ainda uma ferramenta específica para geração de protótipos.
Identificador:	ATIV_ESPEC_UI_SUPL
Nome:	Elaborar especificação suplementar
Descrição:	Esta atividade compreende a elaboração de um documento que descreve para os casos de uso um conjunto de restrições quanto às funcionalidades, usabilidades, comportamento e regras para os componentes visuais, e orientações a serem seguidas na construção da interface (gráfica).
Variabilidade:	Variante de ATIV_ESPEC_UI
Quando:	Projeto tamanho M ou G
Saída:	Especificação suplementar
Ferramenta:	Editor de texto
Identificador:	ATIV_ESPEC_ARQ
Nome:	Projetar arquitetura dos serviços e componentes para a solução
Descrição:	Esta atividade define o projeto dos elementos que irão compor a solução. Esta atividade gera um artefato que deve orientar a equipe de desenvolvimento estabelecendo uma visão geral da solução, tanto dos s serviços que serão consumidos quanto providos. Deve-se registrar as razões e escolhas arquiteturais, componentes de infraestrutura do software e mecanismos de abstração.
Tipo:	Atividade
Variabilidade:	Opcional
Quando:	Projeto tamanho M ou G
Tarefas:	- Analisar os requisitos funcionais, não funcionais e modelo Casos de Uso; - Identificar oportunidades e pontos comuns; - Propor mecanismo(s) de abstração, identificar diretrizes e direcionadores da arquitetura; - Verificar a lista de ativos já listados para reutilização; - Analisar a aplicação dos padrões de arquitetura existentes e mapear infraestrutura para: Controles para execução do fluxo de trabalho (BPEL); Regras de negócio gerenciáveis ou dinâmicas (BRM); Indicadores para monitoração de atividades de negócio (BAM); Necessidade de componentes de colaboração (ECM); e - Elaborar e documentar as definições da arquitetura para a solução.
Responsável:	Arquiteto da Solução
Participante:	Analista de sistemas Arquiteto de Infraestrutura.
Entrada:	Especificação de caso de uso ou Especificação descritiva dos req. Documento de Arquitetura de Software
Saída:	Documento de Arquitetura de Software
Ferramentas:	Editor de diagramas UML Editor de texto.
Origem:	Adaptado de (SHUJA; KREBS, 2008)(ARSANJANI et al., 2008)
Identificador:	TAR_ESPEC_POLIT_SEG
Nome:	Classificar nível da informação e definir política de segurança
Descrição:	Esta atividade busca classificar o nível de importância e segurança da informação e a seleção da política de segurança adequada.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Informação impactante ao negócio
Tarefas:	- Avaliar criticidade e riscos da informação para o negócio;

	<ul style="list-style-type: none"> - Analisar cenários de utilização e exposição das informações; - Selecionar o padrão de política de segurança conforme o nível da informação; e - Avaliar a utilização de mecanismos de proteção adicional.
Responsável:	Arquiteto de segurança da informação
Entrada:	Modelo futuro do processo de negócio Modelo de Domínio Documento de Arquitetura de Software
Saída:	Documento de Arquitetura de Software: Definição dos padrões de política de segurança para os: componente; user interfaces; e, canais de comunicação com os serviços.
Ferramentas:	Repositório de serviços
Origem:	Adaptado de (SHUJA; KREBS, 2008)(ARSANJANI et al., 2008)
Identificador:	ATIV_ESPEC_SRV
Nome:	Especificar serviço
Descrição:	<p>Esta atividade envolve as ações necessárias para a especificação do projeto de um novo serviço reutilizável, ou adaptação de um serviço existente.</p> <p>Nesta atividade deve-se definir a abordagem a ser utilizada:</p> <ol style="list-style-type: none"> 1) Não identificado o produto reutilizável no repositório: <u>Criar um novo serviço</u>, sem relação com os serviços existentes; 2) Identificado o serviço candidato que pode ser entregue através da combinação de serviços existentes: Especificar a criação de um novo serviço a partir de serviços existentes, consumindo e <u>combinando-os</u> de modo a fornecer o resultado desejado e eventuais funcionalidades adicionais; 3) Identificado o serviço candidato aderente no repositório, mas este requer ajustes: A adaptação pode incluir novas informações no serviço ou regras de negócios processadas nos eventos durante a execução do serviço. Esta atividade pode definir a inclusão de <u>variabilidades no serviço</u> (MEDEIROS; ALMEIDA; MEIRA, 2010), que pode passar a ter comportamentos complementares alternativos. Deve-se prezar por manter a compatibilidade para evitar o impacto nos demais consumidores do serviço. Para minimizar o impacto, pode ser necessário criar uma <u>nova versão</u> do serviço e manter ambas ativas.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU mandatório
Tarefas:	<ul style="list-style-type: none"> - Revisar a aderência dos serviços identificados; - Projetar o novo serviço; - Definir tipo de comunicação; - Elaborar o desenho da interface onde se define a estrutura/contrato das mensagens para entrada, saída e exceções; - Avaliar dependências, variabilidades do serviço; e - Especificar o comportamento do serviço.
Responsável:	Analista SOA
Entrada:	Modelo futuro do processo de negócio Lista de serviços e aderência Modelo de Domínio Documento de Arquitetura de Software
Saída:	Especificação do serviço
Ferramentas:	Repositório de serviços Editor de texto Editor XML
Origem:	Adaptado de (ARSANJANI et al., 2008)(THIES; VOSSEN, 2009)
Identificador:	ATIV_ESPEC_SRV_REST
Nome:	Especificar serviço REST
Descrição:	Especificar o funcionamento do serviço ReST(<i>REpresentational State Transfer</i>) definindo uma URL única para o recurso

	<p>definido pela entidade a ser acessada. O ReSTful definiu uma padronização para realizar operações de manutenção da informação (CRUDs), variando apenas o verbo HTTP.</p> <p>GET PUT POST DELETE READ UPDATE CREATE DELETE</p> <p>A descrição do serviço deve ser descrita através da linguagem WADL (<i>Web Application Description Language</i>).</p>
Variabilidade:	Variante de ATIV_ESPEC_SRV
Quando:	A solução está baseada em plataforma de integração Web e padrões leves, aplicações Web, aplicação móveis, sem necessidade de uso de SOAP ou WS-*
Identificador:	ATIV_ESPEC_SRV_WS
Nome:	Especificar Web Service
Descrição:	A especificação de Web Service deve primeiramente documentar as informações definindo um XML Schema com todas as informações e suas restrições. Após, deve-se definir nomes para as operações e suas respectivas mensagens. Depois, as operações são agrupadas em um serviço juntamente com definições de comunicação.
Variabilidade:	Variante de ATIV_ESPEC_SRV
Quando:	A solução está baseada em uma plataforma de integração Web Service SOAP, ou quando precisa utilizar padrões de integração que requeira recursos como: transação distribuída, segurança, garantia da entrega (WS-*) (WEBBER et al., 2010).
Identificador:	TAR_ESPEC_SRV_REG
Nome:	Registrar ou provisionar serviços no repositório corporativo
Descrição:	O objetivo desta tarefa é assegurar o registro do serviço no repositório tanto para as buscas em tempo de desenho da solução quanto para a execução. Todos os serviços devem ser catalogados, mesmo que o produto não utilize o repositório de serviços em tempo de execução.
Tipo:	Tarefa
Passos:	<ul style="list-style-type: none"> - Incluir ou revisar do serviço no repositório; - Incluir cenário de negócio e provisionar (ativar serviço existente); e - Definir conexões entre a aplicação provedora e a consumidora.
Responsável:	Analista de sistemas
Entrada:	Lista de serviços com informações de acesso
Saída:	Cenário composto por provedores, consumidores e serviços registrados no repositório.
Ferramenta:	Repositório de serviços
Origem:	Adaptado de (OpenGroup, 2009)
Identificador:	TAR_ESPEC_SRV_REV
Nome:	Revisar metadados dos serviços
Descrição:	Esta atividade compreende a revisão dos tipos de dados envolvidos no desenho das mensagens, buscando a padronização e reúso.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Organização deseja manter modelos canônicos que padronizam a forma como as entidades de negócio são representadas (nível OSIMM 3 ou superior).
Passos:	<ul style="list-style-type: none"> - Receber especificação dos serviços (WADL ou WSDL); - Identificar a definição dos tipos de dados e sua conformidade em relação à nomenclatura padrão das entidades de negócio; - Se entidade de negócio não estiver padronizada na organização, consultar definições das organizações padronização do área cenário de negócio, segue alguns exemplos: Indústria automotiva (AIAG); Saúde (HL7), Indústria química (CIDX), Bancos (BIAN, SWIFT, UNIFI), Comércio (RosettaNet, 1SYNC); e

	- No caso de não conformidades, reportar os problemas ao projetista do serviço.
Responsável:	Administrador de dados
Entrada:	Especificação do serviço
Saída:	Registro de aceite e lista de não conformidades
Ferramenta:	Repositório de metadados
Origem:	Adaptado de (OSIMM, 2009)

4.3.4 Fase de Construção, Composição e Testes

A fase de Construção, Composição e Testes é onde o produto é de fato criado. As fases anteriores projetam e identificam os componentes para, nesta fase, passarem pelo processo de manufatura do software. Os principais objetivos desta fase são: a construção do produto com agilidade, a montagem e integração dos componentes, a entrega de um produto de qualidade com baixos percentuais de defeito e o aceite do usuário. Ao final desta, como o modelo é iterativo e incremental, deve-se primar pela execução da fase de Liberação para disponibilizar parte do software em produção, tão logo quanto possível. Ao realizar entregas parciais a cada iteração reduz-se o risco do projeto e antecipa-se a curva de seu retorno de investimento. Inclusive, para o gerenciamento de contratos pode-se condicionar pagamentos ao aceite das entregas funcionais do produto (COCKBURN, 2006). A Figura 4-7 apresenta o fluxo das atividades e a Tabela 4-5 descreve as atividades da fase.

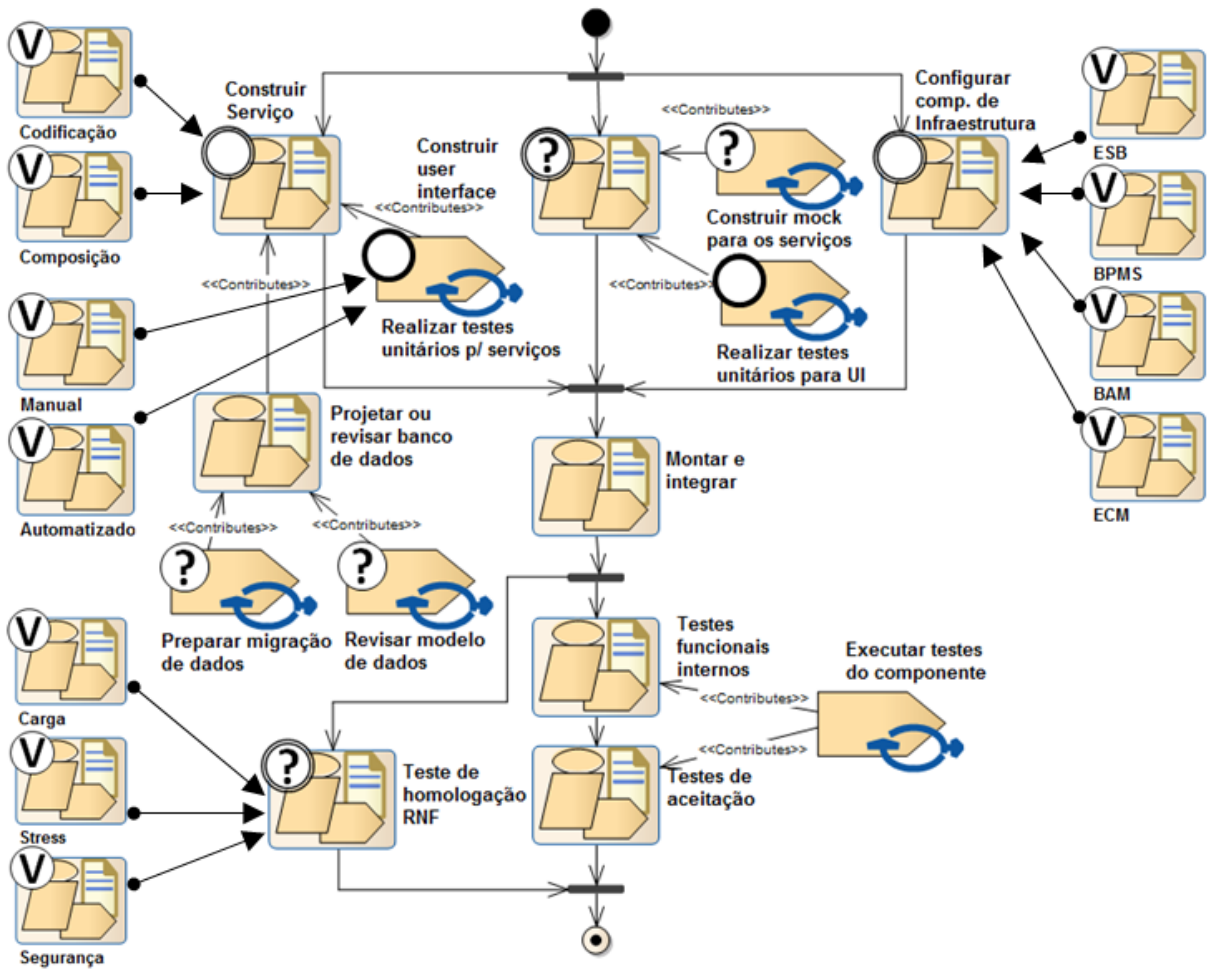


Figura 4-7. Fase de Construção, Composição e Testes. Fonte: o Autor, 2012.

Tabela 4-5. Atividades para fase de Construção, Composição e Testes.

Identificador:	ATIV_CONST_SRV
Nome:	Construir Serviço
Descrição:	Esta atividade descreve a construção ou adaptações dos serviços projetados.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU mandatório
Tarefas:	<ul style="list-style-type: none"> - Receber e verificar a especificação do serviço; - Avaliar definições no Documento de Arquitetura de Software; - Realizar a construção do serviço, seja por codificação ou ferramenta de composição; e - Avaliar o serviço e registrar as decisões significativas.
Entrada:	Especificação de requisitos do sistema Documento de Arquitetura de Software Especificação do serviço
Saída:	Componente do serviço e fontes Documento de Arquitetura de Software
Origem:	Adaptado de (OSIMM, 2009)(OpenGroup, 2009)(PAPAZOGLU; HEUVEL, 2006)
Identificador:	ATIV_CONST_SRV_CODE
Nome:	Construir serviço através de codificação
Descrição:	Esta atividade estabelece a construção dos serviços desenhados ou adaptações, incluindo os componentes, classes, e manipulações e persistências de dados.
Variabilidade:	Variante de ATIV_CONST_SRV
Quando:	Requisitos do produto não são suportados pela ferramenta de composição de serviço, por exemplo: produto requer a criação de novos repositórios de dados.
Responsável:	Programador SOA
Ferramenta:	Geradores de serviços: Para Java utilizar os utilitários wsgen.exe e wsimport.exe; Já para o Microsoft .NET o wsdl.exe; etc.
Identificador:	ATIV_CONST_SRV_COMPOSITION
Nome:	Construir Serviço através de composição
Descrição:	Esta tarefa realiza a construção do serviço utilizando a abordagem de combinação dos serviços existentes o analista pode derivar novos serviços sem a necessidade de codificação.
Variabilidade:	Variante de ATIV_CONST_SRV
Quando:	Existe plataforma para composição de serviços e o produto possui requisitos suportados pela ferramenta.
Responsável:	Analista SOA
Ferramenta:	Plataforma para composição de serviço
Identificador:	TAR_DESEN_SRV_TESTE
Nome:	Realizar testes unitários para os serviços
Descrição:	Esta tarefa compreende a realização dos testes individuais de um serviço, de modo a verificar se o serviço construído está atendendo ao que foi solicitado na especificação.
Tipo:	Tarefa
Variabilidade:	Ponto de variação alternativo mandatório
Passos:	<ul style="list-style-type: none"> - Avaliar os casos de teste; - Definir caso de teste unitário para o serviço, buscando testar suas regras de negócio e abranger os diferentes comportamentos do serviço (variabilidades); - Preparar o ambiente para a execução; - Opcionalmente, construir script para testes automatizados; - Executar testes; - Verificar se a resposta está atendendo ao que foi solicitado; e - Comunicar resultados com o registro de execução dos testes.
Responsável:	Testador

Entrada:	Especificação de requisitos do sistema Especificação do serviço Componente do serviço Caso de Teste (opcional)
Saída:	Registro de execução dos testes Script de testes (opcional)
Origem:	Adaptado de (SHUJA; KREBS, 2008)(OpenUP, 2011)
Identificador:	TAR_DESEN_SRV_TESTE_MANUAL
Nome:	Realizar testes unitários manuais para os serviços
Descrição:	Esta tarefa compreende a realização dos testes individuais de um serviço de forma manual, sem a utilização de automatização.
Variabilidade:	Variante de TAR_DESEN_SRV_TESTE
Quando:	Projetos de tamanho P.
Identificador:	TAR_DESEN_SRV_TESTE_AUTO
Nome:	Construir e realizar testes unitários automatizados para os serviços
Descrição:	Esta tarefa busca a realização dos testes individuais de um serviço de forma automatizada, através do desenvolvimento de um artefato que permita a reprodução dos casos de testes.
Variabilidade:	Variante de TAR_DESEN_SRV_TESTE
Quando:	Projetos de tamanho M e G.
Ferramenta:	Teste unitário caixa preta são facilmente produzidos por scripts na ferramenta SoapUI ¹³ . Já testes que envolvam passos intermediários realizado pelos componentes e classes envolvidas utiliza-se: JUnit, NUnit, MStest, ABAP Unit, etc.
Identificador:	ATIV_DESEN_SRV_DB
Nome:	Projetar ou revisar banco de dados
Descrição:	Deve-se projetar o banco de dados conforme as necessidades do serviço e as classes modeladas. No cenário de adaptação de serviços pode-se simplesmente expandir as entidades já existentes no banco de dados para satisfazer às novas necessidades.
Tarefas:	<ul style="list-style-type: none"> - Analisar o modelo de domínio de domínio, tipos de dados especificado nos serviços, e classes das entidades de negócio; - Opcionalmente, optar por desenvolver os modelos conceitual e lógico de dados; - Elaborar o modelo físico do banco de dados. Dependendo do componente de persistência utilizado, é possível permitir a geração automática do banco e script; e - Refinar o modelo físico conforme o guia de desempenho para banco de dados.
Responsável:	Projetista de banco de dados
Entrada:	Modelo de domínio
Saída:	Modelo de entidade e relacionamento
Ferramenta:	Editor de diagrama UML Guia de desempenho para banco de dados
Origem:	Adaptado de (SHUJA; KREBS, 2008)
Identificador:	TAR_DESEN_SRV_DB_MIGR
Nome:	Preparar migração de dados
Descrição:	Esta tarefa descreve as ações necessárias para a migração de dados, onde a conversão das informações de uma base de origem e tratada para uma base de destino.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Existe uma massa de dados a ser carregada, por exemplo:

¹³ A ferramenta de automação de testes de serviços SoapUI está disponível em <<http://www.soapui.org/>>, consultado em 01/07/2012.

	Substituição de um sistema; Revisão na base em uso a qual precisa ser recriada e reimportar os dados; etc.
Passos:	Identificar as características da base de origem e destino; Definir abrangência da migração; Selecionar estratégia de migração e avaliar janela necessária; Mapear informação da origem e de destino; e Preparar recursos necessários (scripts automatizados ou pessoas para conversão manual).
Responsável:	Projetista de banco de dados
Entrada:	Modelo de entidade e relacionamento
Saída:	Script de migração de dados
Ferramenta:	Ferramenta ETL: Para realizar a extração, transformação e carga de dados.
Origem:	Adaptado de (SHUJA; KREBS, 2008)
Identificador:	TAR_DESEN_SRV_DB_REV
Nome:	Revisar modelo de dados
Descrição:	Esta tarefa busca a revisão dos modelos para manter a uniformidade e as diretrizes da organização no gerenciamento da informação.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Organização deseja manter o banco de dados em conformidade com os metadados da organização.
Passos:	- Receber modelo de entidade e relacionamento; - Compara as entidades e atributos com o repositório corporativos dos tipos de dados; e - Realizar o aceite ou reportar as não conformidades ao projetista.
Responsável:	Administrador de dados
Entrada:	Modelo de entidade e relacionamento
Saída:	Registro de aceite e lista de não conformidades
Ferramenta:	Editor de diagrama UML.
Origem:	Adaptado de (OSIMM, 2009)
Identificador:	ATIV_DESEN_UI
Nome:	Construir <i>user interface</i>
Descrição:	Esta atividade aborda a construção da <i>user interface</i> conforme especificação de requisitos do software e demais artefatos.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU opcional
Tarefas:	- Analisar os artefatos de entrada; - Avaliar definições no Documento de Arquitetura de Software; - Verificar dependência de serviços, que requeiram a construção de simulação (mock); - Construir a <i>user interface</i> ; e - Avaliar o serviço e registrar as decisões significativas.
Entrada:	Especificação de requisitos do sistema Protótipo da interface da solução (opcional) Especificação suplementar
Saída:	Componente de <i>user interface</i>
Origem:	Autor
Identificador:	ATIV_DESEN_UI_CODE
Nome:	Construir a <i>user interface</i> através de codificação
Descrição:	Esta atividade busca construir a <i>user interface</i> conforme especificação de requisitos do software, através da codificação das telas utilizando a linguagem definida conforme detalhamento da arquitetura.
Variabilidade:	Variante de ATIV_DESEN_UI
Quando:	Requisitos de usabilidade da aplicação excedem a capacidade da ferramenta de composição.
Responsável:	Programador UX

Ferramenta:	Ambiente de programação para as tecnologias utilizadas. Geralmente utilizam-se os padrões Web (HTML, CSS e Javascript). Podem ser necessárias outras tecnologias de apresentação para dispositivos móveis ou embarcados (Objective C, Java, C, .NET).
Identificador:	ATIV_DESEN_UI_COMPOSITION
Nome:	Esta atividade busca construir User Interface através da combinação de serviço e outras user interfaces existentes, sem a necessidade de codificação.
Descrição:	Montar as user interfaces através de ferramentas de composição de telas através do consumo de fontes de informações (serviços ou mocks).
Variabilidade:	Variante de ATIV_DESEN_UI
Quando:	Ferramenta de composição atende aos requisitos de usabilidade.
Responsável:	Analista SOA
Restrição:	Serviço ou mock deve estar pronto.
Ferramenta:	Ambiente para composição de aplicações
Identificador:	TAR_DESENV_SVR MOCK
Nome:	Construir mock para os serviços
Descrição:	A partir da especificação, construir mock para os serviços permitindo o desenvolvimento da User Interface independente e paralelo ao do serviço.
Tipo:	Tarefa
Variabilidade:	Opcional
Quando:	Projeto tamanho G. Para isolar as atividades e papéis entre os desenvolvedores e UI e serviços e permitir execução paralela.
Passos:	<ul style="list-style-type: none"> - Identificar serviços e tipos de dados; - Realizar o levantamento de exemplos de registros de dados de dados para construção dos mocks, pode-se utilizar informações dos casos de teste; - Construir e testar o funcionamento do Mock;
Responsável:	Analista SOA
Entrada:	Especificação do serviço
Saída:	Mock dos serviços
Ferramenta:	Gerador do mock: Através de ferramentas como o SOAPUI, configurar uma massa de dados para simulação do serviço.
Origem:	Autor
Identificador:	TAR_DESEN_UI_TESTE
Nome:	Realizar testes unitários para UI
Descrição:	Esta tarefa compreende o teste individual da <i>user interface</i> do sistema e a verificação quanto ao atendimento às especificações e a comunicação de eventuais não conformidades.
Tipo:	Tarefa
Variabilidade:	Ponto de variação alternativo mandatório
Passos:	<ul style="list-style-type: none"> - Avaliar os casos de teste; - Definir caso de teste unitário para a <i>user interface</i>, buscando o teste de todos os fluxos e elementos apresentados; - Preparar o ambiente para a execução; - Opcionalmente, construir script para automatizar a execução dos testes na <i>user interface</i>; - Executar testes; - Verificar se a resposta está atendendo ao que foi solicitado; e - Comunicar resultados com o registro de execução dos testes.
Responsável:	Testador
Entrada:	Componente de <i>user interface</i> Especificação de requisitos do sistema Caso de Teste (Opcional)
Saída:	Registro de execução dos testes e aceite

Origem:	Adaptado de (SHUJA; KREBS, 2008)(OpenUP, 2011)
Identificador:	TAR_DESEN_UI_TESTE_MANUAL
Nome:	Realizar testes unitários manuais para UI
Descrição:	Esta tarefa compreende busca definir caso de teste unitário para a UI, executar os testes manualmente para verificar se a UI está atendendo ao que foi solicitado.
Variabilidade:	Variante de TAR_DESEN_UI_TESTE
Quando:	Projeto de tamanho P.
Identificador:	TAR_DESEN_UI_TESTE_AUTO
Nome:	Realizar testes unitários automatizados para UI
Descrição:	Esta tarefa busca definir caso de teste unitário para a UI, construir script para testes automatizados, e verificar se a UI está atendendo ao que foi solicitado.
Variabilidade:	Variante de TAR_DESEN_UI_TESTE
Quando:	Projeto de tamanho M e G.
Ferramenta:	Teste funcional automatizado: Utilizar ferramenta que gere script para replicar a execução, como Selenium ¹⁴ , HP Quality Center software ¹⁵ , IBM Rational Quality Manager ¹⁶ , entre outras.
Identificador:	ATIV_DESEN_CONFIG
Nome:	Configurar componentes de infraestrutura
Descrição:	Esta atividade abrange a configuração dos componentes de infraestrutura utilizados pela solução.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU
Tarefas:	- Responsável pelo ambiente recebe a solicitação de configuração; - Avalia a possibilidade de utilização do ambiente; - Disponibilizar o recurso através da configuração solicitada; e - Verificar se ambiente está operacional.
Responsável:	Analista SOA
Participante:	Arquiteto da solução Arquiteto de infraestrutura.
Entrada:	Modelo futuro do processo de negócio Especificação de requisitos do sistema Especificação do serviço Documento de Arquitetura de Software
Saída:	Documento de Arquitetura de Software. Ambiente configurado.
Origem:	Adaptado de (OSIMM, 2009)
Identificador:	ATIV_DESEN_CONFIG_BPMS
Nome:	Configurar BPMS
Descrição:	Esta atividade busca configurar o componente de infraestrutura BPMS para o controle da execução do processo da solução.
Variabilidade:	Variante de ATIV_DESEN_CONFIG
Quando:	Projeto requer controle de fluxo de atividades
Restrição:	Requer que o desenho do processo futuro (ATIV_PROC_TOBE_FE) seja realizado em BPMN
Tarefas:	- Configurar os componentes de infraestrutura BPMS para controlar o fluxo de execução das atividades do processo de negócio.

¹⁴ Ferramenta para automação de testes no navegador, disponível em <<http://seleniumhq.org/>>, consultado em 01/07/2012.

¹⁵ Ferramenta para gerenciamento, análise e automação de testes da HP, disponível em <<http://www8.hp.com/us/en/software-solutions/software.html?compURI=1172141>>, consultado em 01/07/2012.

¹⁶ Ferramenta para gerenciamento, análise e automação de testes da HP, disponível em <<http://www-01.ibm.com/software/rational/products/rqm/>>, consultado em 01/07/2012.

	<ul style="list-style-type: none"> - O processo a ser automatizado deve ser representado em BPMN; - As restrições da execução são representadas no BPEL; - Os papéis e executores das atividades devem ser configurados; - Limite de tempo para execução das atividades; - Preparar pacote com a configuração do BPMN e BPEL pronto para teste.
Ferramenta:	BPMS
Identificador:	ATIV_DESEN_CONFIG_BAM
Nome:	Configurar BAM
Descrição:	Esta atividade descreve a configuração do Business Business Activity Monitoring (BAM) para automatizar a monitoração dos processos de negócio.
Variabilidade:	Variante de ATIV_DESEN_CONFIG
Quando:	Projeto requer controle de indicadores automatizado
Tarefas:	<ul style="list-style-type: none"> - Buscar informações sobre os indicadores (com base nos KPIs) e metas de controles sobre os serviços e/ou processo; - Refinar as ações em relação as metas dos KPIs e identificar os responsáveis; e - Configurar na ferramenta os eventos e ações na ferramenta.
Ferramenta:	BAM
Identificador:	ATIV_DESEN_CONFIG_ESB
Nome:	Configurar ESB ou EAI
Descrição:	Esta atividade compreende a configuração do componente de infraestrutura a serviços (chamado de ESB ou EAI), o qual pode ser utilizado para: Disponibilizar serviços agrupados em um único barramento; Realizar conversões de mensagens; Roteamento de serviços; Inclusão de mecanismos de enfileiramento; Controle de mensagens assíncronas; e, Comunicação com outros protocolos.
Variabilidade:	Variante de ATIV_DESEN_CONFIG
Quando:	Projeto requer controles especiais no controle, distribuição e registro das mensagens.
Ferramenta:	ESB (EAI)
Identificador:	ATIV_DESEN_CONFIG_ECM
Nome:	Configurar ECM
Descrição:	Esta atividade busca fazer a disponibilização de uma área de armazenamento no ECM, criada, configurada e pronta para armazenar documentos não estruturados.
Variabilidade:	Variante de ATIV_DESEN_CONFIG
Quando:	Projeto requer armazenamento de informações não estruturadas e colaboração.
Tarefas:	<ul style="list-style-type: none"> - Identificar necessidade de informação não estruturada; - Avaliar e reservar capacidade de armazenamento requerida; - Configurar perfil de acesso conforme padrões de segurança definidos; - Analisar necessidades de uso e configurar regras de arquivamento; - Configurar área de armazenamento e testar.
Ferramenta:	ECM
Identificador:	ATIV_DESEN_MONTAR_INTEG
Nome:	Montar e integrar
Descrição:	Esta atividade cria uma nova versão consistente e completa do sistema, reunindo todos os artefatos de software e configuração gerados no processo. Este pacote deve ser verificado, pelo chamado de teste de fumaça, através da sua aplicação em um ambiente para verificar se o pacote permite a continuação das atividades de teste.
Tipo:	Atividade

Tarefas:	- Planejar integração, plano de liberação e definir ou revisar ordem de montagem dos pacotes; - Construir o pacote; - Aplicar o pacote na infraestrutura dedicada a testes; e - Verificar se pacote está funcional, com um conjunto mínimo de testes da aplicação.
Responsável:	Analista SOA
Participante:	Arquiteto da solução Arquiteto de infraestrutura
Entrada:	Documento de Arquitetura de Software Componentes de serviço e <i>user interface</i>
Saída:	Pacote para liberação
Ferramenta:	Build: Conforme a plataforma utiliza-se as ferramentas para geração de uma nova versão do pacote. Aconselha-se automatizar esta atividade através de ferramentas de automação juntamente com ferramentas de integração contínua.
Origem:	Adaptado de (SHUJA; KREBS, 2008)(OpenUP, 2011)
Identificador:	ATIV_DESEN_TESTE_INTERNO
Nome:	Testes funcionais internos
Descrição:	Esta atividade realiza a avaliação completa do produto com o objetivo de execução dos testes, diagnóstico e resolução das possíveis falhas. Através dos testes internos estabiliza-se a solução para avaliação do cliente.
Tipo:	Atividade
Tarefas:	- Avaliar e revisar completude dos casos de teste; - Preparar conjunto de dados necessário; - Opcionalmente, construir script para automatizar os testes funcionais da solução integrada, pode-se reaproveitar o teste unitário da <i>user interface</i> e combiná-los;
Responsável:	Testador
Entrada:	Pacote para liberação (Produto montado e integrado em ambiente para homologação) Plano de teste e casos de testes (opcional)
Saída:	Registro de execução dos testes e aceite
Ferramenta:	Editor de planilha.
Origem:	Adaptado de (SHUJA; KREBS, 2008)(OpenUP, 2011)
Identificador:	TAR_DESEN_TESTE_EXEC_REG
Nome:	Executar testes e registrar defeitos
Descrição:	Esta tarefa compreende a execução dos casos de testes, a análise dos resultados e comunicação de eventuais problemas.
Tipo:	Tarefa
Responsável:	Testador
Passos:	- Executar os casos testes; - Analisar resultados e comparar com a situação esperada; e - Registrar o defeito e solicitar a correção para equipe.
Entrada:	Caso de teste
Saída:	Registro de execução dos testes e aceite
Ferramenta:	Ferramenta para registrar os defeitos. Editor de planilha.
Origem	Adaptado de (SHUJA; KREBS, 2008)(OpenUP, 2011)
Identificador:	ATIV_DESEN_TESTE_ACEITA
Nome:	Testes de aceitação
Descrição:	Nesta atividade, os testes novamente são realizados, mas pelo usuário chave que representa o patrocinador e clientes finais da solução.
Tipo:	Atividade estende ATIV_DESEN_TESTE_INTERNOS
Responsável:	Usuário chave
Participante:	Testador

Identificador:	ATIV_DESEN_TESTE_HOMO
Nome:	Teste de homologação dos requisitos não funcionais
Descrição:	Nesta atividade é realizada a avaliação do desempenho, estabilidade e segurança do produto em relação aos seus requisitos não funcionais.
Tipo:	Atividade
Variabilidade:	Ponto de variação OU opcional
Quando:	Solução é crítica para o negócio ou pode afetar imagem da organização.
Tarefas:	<ul style="list-style-type: none"> - Analisar e selecionar os casos de testes críticos para a simulação a ser executada; - Construir o script de teste, onde deve ser avaliada a reutilização dos scripts previamente construídos; - Preparar um ambiente similar ao de produção para realizar a homologação; - Executar a simulação de carga e reportar os resultados; e - Realizar o aceite ou reportar o problema para repetir o(s) teste(s).
Responsável:	Analista de sistemas
Participante:	Arquiteto da solução Arquiteto de infraestrutura
Entrada:	Pacote para liberação.
Saída:	Registro de execução dos testes
Ferramenta:	Ferramenta para teste de carga. Editor de planilha.
Origem:	Autor
Identificador:	ATIV_DESEN_TESTE_HOMO_CARGA
Nome:	Testes de carga
Descrição:	Esta atividade simula a carga prevista para o ambiente e avaliar a resposta do sistema nesta situação
Variabilidade:	Variante ATIV_DESEN_TESTE_HOMO
Quando:	Deseja-se avaliar a capacidade de resposta do sistema no cenário planejado no ambiente de homologação.
Identificador:	ATIV_DESEN_TESTE_HOMO_STRESS
Nome:	Testes de stress
Descrição:	Esta atividade simula a capacidade da arquitetura avaliando a degradação do desempenho conforme o crescimento da carga, até a degradação máxima. A simulação de carga até a parada do sistema também é chamada de teste de ruptura.
Variabilidade:	Variante ATIV_DESEN_TESTE_HOMO
Quando:	Deseja-se avaliar a capacidade máxima suportada pela arquitetura para o ambiente de homologação.

4.3.5 Liberação, Execução e Monitoração

Após a fase de Construção, Composição e Testes, segue a fase de Liberação, Execução e Monitoração. Esta fase consiste na transição e disponibilização da nova solução ou versão da solução para o usuário final. Um pacote para transporte ou cópia do aplicativo para o ambiente de produção é realizado, registros são atualizados na base CMDB (*Configuration Management Database*) e o usuário é habilitado para utilização do software, geralmente através de treinamentos. Os indicadores de SLA são estabelecidos e monitorados para

manter o acordo de nível de serviços. A inclusão de novos consumidores a um serviço pode gerar impacto ao ambiente em execução, o que pode comprometer a realização do SLA das soluções que estão atendendo à organização. Por isso, faz-se necessário a constante monitoração, tanto dos KPIs de negócio, como dos KPIs técnicos dos serviços. Esta fase assemelha-se a fase “*Deployment, Monitoring, and Management*” do SOMA, proposta por (ARSANJANI et al., 2008).

Dentro de SOA e BPM, o componente de infraestrutura *Business Activity Monitoring* (BAM) possui um importante papel para automatização de indicadores, tanto operacionais (tempo de resposta e erros), quanto para os processos de negócio. O controle dos Indicadores chave de desempenho (KPIs) precisa ser constantemente revisado levando em consideração o cumprimento do cenário atual, assim como possíveis melhorias para atendimento de novas demandas. As ações sobre os KPI são classificadas basicamente em dois tipos, a saber:

- Ações corretivas – detecção de problemas através de eventos (gerados por indicadores fora dos padrões) para ações imediatas que buscam estabelecer a continuidade do processo;
- Ações de melhoria – estabelecem a análise de tendências através de dados históricos da execução do processo e utilização de ferramentas de *Business Intelligence* (BI). Demandará melhorias para médio ou longo prazo nos processos de negócio, podendo requerer desenvolvimento.

Contudo, o detalhamento desta fase não está no escopo do trabalho. A definição e implantação desta fase devem seguir os modelos de serviços e operações, como: ITIL (OGC, 2011), CMMI-SVC (SEI, 2010c), ISO-20000 (ABNT, 2011) ou MM-GSTI (MACHADO, 2011). A partir da proposta MM-GSTI está sendo elaborado o MR-MPS para serviços.

4.4 Exemplo de seleção das variabilidades

Para exemplificar a seleção das variabilidades, foi utilizado cenários de exemplo, como o representado pela organização ABC que solicitou a realização de um projeto. O objetivo estratégico deste projeto é reduzir os custos do departamento de vendas, melhorar a produtividade do processo e reduzir o tempo total para elaborar um pedido ou proposta comercial em pelo menos 20%. Para preparar o projeto o engenheiro de software tem a responsabilidade de analisar as

características do projeto e definir o processo utilizado, por meio da seleção das seguintes variabilidades da linha SOPLA:

- A fase de Modelagem de Negócios e Transformação será selecionada, pois o projeto precisa aprimorar o processo de negócio, analisar os impactos, e analisar e detalhar os objetivos e requisitos de negócio.
- A atividade Mapear processo atual e ferramentas precisa ser realizada, pois o processo já é executado.
- Para o ponto de variabilidade Modelar processo de negócio deve-se realizar ambas as variabilidades: a atividade baseada na técnica de Brown Paper devido ao grande impacto e a necessidade de envolver as pessoas que apoiaram na mudança, e, também, realizar a atividade com Ferramenta de Modelagem, pois a organização ABC possui controle e certificação de seus processos de negócio.
- A tarefa Simular execução do processo de negócio deve ser realizada, já que o impacto financeiro é significativo com o objetivo de redução de 20%.
- A tarefa Definir indicadores para o processo de negócio: como o objetivo do projeto possui uma grande mudança e diversos impactos, logo a definição de indicadores é fundamental para provar o valor da proposta.

Após execução da fase de Modelagem de Negócios e Transformação, identificou-se que 60% dos pedidos e ordens de venda eram produtos de prateleira e 40% eram projetos e serviços que precisavam de uma análise mais detalhada. Para 60% dos pedidos e ordens foi proposto um novo processo de negócio onde os clientes fossem motivados a utilizar canal eletrônico com autonomia e sem envolvimento do departamento de vendas. A análise do volume dos 60% dos pedidos e ordens de venda revelou que cerca de 50% das vendas, equivalente a 30% do volume total, eram realizadas para um pequeno grupo de 5 grandes clientes. Para estes clientes foi proposta um canal diferenciado permitindo a integração direta de seus processos, baseado em padrões de integração, como o RosettaNet.

As informações são novamente pelo Engenheiro de Software para definição das variabilidades da próxima fase, que é a Identificação e Definição da Arquitetura.

- A atividade opcional Captar e modelar domínio de informação será realizada devido a classificação de tamanho do projeto.
- Para o ponto de variabilidade Estimar esforço de desenvolvimento foi selecionada a variabilidade Estimativa por análise de ponto por caso de

uso, devido ao conhecimento organizacional e reuso dos casos de uso para as próximas atividades.

O Engenheiro de Software novamente avalia o processo a ser executado para a fase Especificação e Projeto dos Serviços, a qual deve ser realizada devido o tamanho do projeto e a necessidade de criar novos serviços.

- Para o ponto de variabilidade Especificar casos de uso da solução foi selecionado: Especificar casos de uso da solução devido à utilização previa do UCP e o tamanho do projeto.
- A tarefa opcional Validar especificação de requisitos: será realizada devido o tamanho do projeto.
- No ponto de variabilidade Projetar user interface serão selecionadas as atividades: Elaborar protótipos para projetar atender ao requisito de usabilidade necessário para um canal de venda fácil para os clientes, e, também Elaborar especificação suplementar, devido a classificação de tamanho do projeto.
- A atividade opcional Projetar arquitetura dos serviços e componentes para a solução: será selecionada devido a classificação de tamanho do projeto.
- A tarefa opcional Classificar nível da informação e definir política de segurança: será realizada devido à importância das informações.
- No ponto de variabilidade Especificar serviço será selecionado a variabilidade Especificar Web Service devido à necessidade de integração com os diversos clientes.
- A tarefa Revisar metadados dos serviços faz-se necessária para verificar a conformidade com padrões de troca de informação como o RosettaNet.

Na fase de Fase de Construção, Composição e Testes novamente o Engenheiro precisa analisa as características do projeto para selecionar as atividades a serem utilizadas.

- O ponto de variabilidade Construir Serviço será realizado pela variabilidade de codificação, devido à ausência de uma plataforma para combinação e o pequeno número de serviços existentes e reutilizados.
- No ponto de variabilidade Construir a user interface também será realizado por codificação para melhor atender os requisitos de usabilidade.

- Para os pontos de variabilidade nas tarefas Realizar testes unitários, tanto para serviços como para user interfaces, serão selecionadas as variabilidades de testes automatizados devido o tamanho do projeto.
- Para o ponto de variabilidade Configurar componentes de infraestrutura, conforme os requisitos não funcionais do projeto, serão selecionadas as atividades Configurar BPMS, pois o projeto requer controle de fluxo de atividades, e também a variabilidade Configurar ESB para controlar o acesso dos clientes com maior volume de vendas que se integraram diretamente com os serviços.
- A tarefa opcional Preparar migração de dados não será realizada, pois não há substituição de sistemas e massas de dados para carga.
- A tarefa opcional Revisar modelo de dados não será realizada, pois não será mantida a conformidade dos metadados de negócio até o modelo do banco de dados.
- No ponto de variabilidade do Teste de homologação dos requisitos não funcionais será realizada a variabilidade Testes de carga para avaliar a capacidade de resposta do sistema no cenário planejado.

4.5 Apoio Computacional à Linha de Processo Proposta

As ferramentas para documentação e gerenciamento dos processos de engenharia de software, como o Rational Composer e o EPF, possuem limitações na edição dos processos; na reutilização de elementos do processo, que muitas vezes é realizada por cópias; na seleção de variabilidade que é pouco eficiente, e, ainda, existe uma carência quanto às capacidades de configuração, execução e monitoração dos processos (ALEIXO et al, 2011). Tendo em vista estes problemas e as propostas já apresentadas na seção 2.4, utilizou-se parte das ferramentas para gerenciamento e customização de variabilidades em processo de software de (ALEIXO et al, 2011), porém optou-se por desenvolver e integrar a uma nova camada para suportar a execução do processo.

Para realizar o controle e a execução do processo, uma plataforma de BPM foi utilizada, realizando a construção de uma nova ferramenta de transformação entre os modelos UMA e BPMN. Após a transformação para BPMN, qualquer plataforma de BPM pode ser utilizada para a execução e monitoração dos processos

de software. As atividades e ferramentas envolvidas no apoio computacional à linha de processos estão apresentadas na Figura 4-8.

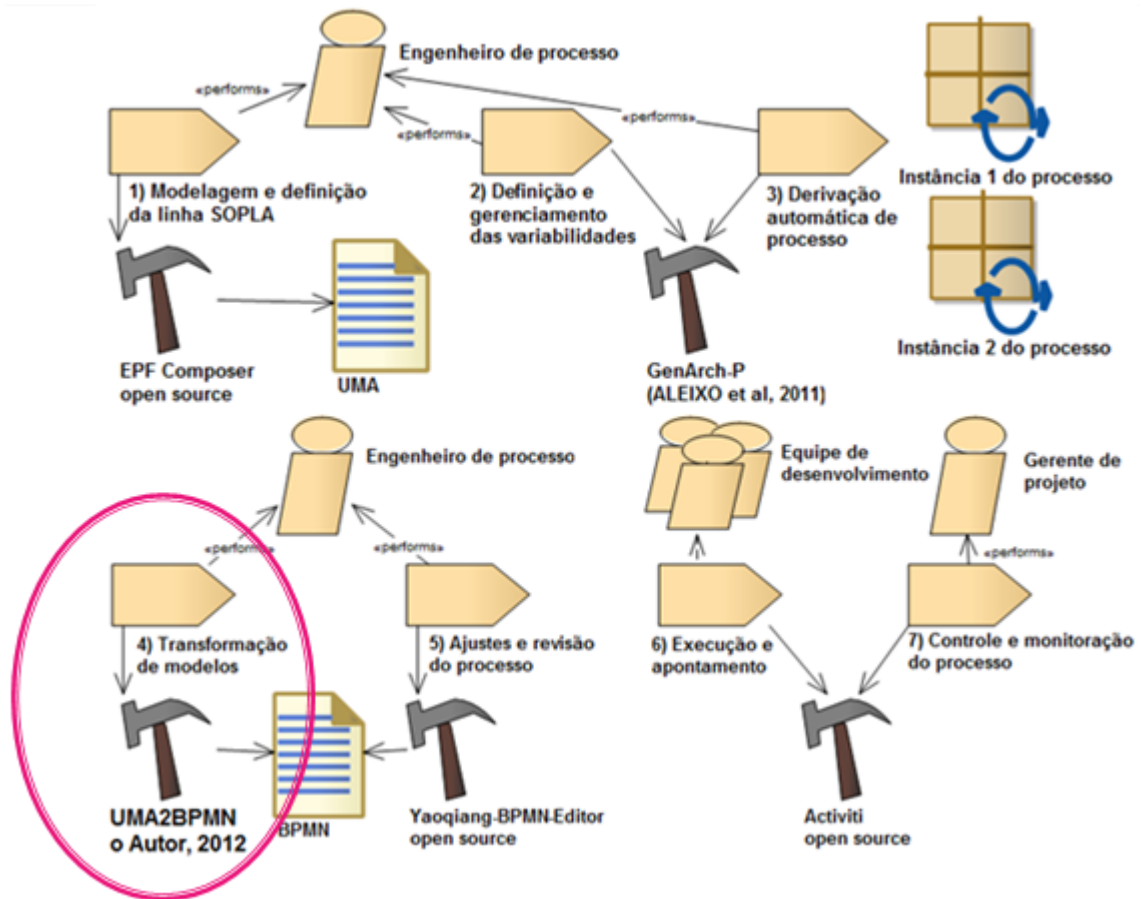


Figura 4-8. Atividades e ferramentas no apoio computacional à linha de processo, adaptado de (ALEIXO et al, 2011).

Conforme se pode observar na Figura 4-8, o apoio computacional é oferecido em 7 passos, conforme descrito a seguir:

- Passo 1 - Modelagem e definição da linha, através do EPF, ferramenta previamente apresentada na seção 2.3.2 (EPF, 2012);
- Passo 2 - Definição e gerenciamento das variabilidades, através da ferramenta GenArch-P (ALEIXO et al, 2011);
- Passo 3 - Derivação automática de processo, onde a partir da linha, um novo processo é instanciado (ALEIXO et al, 2011);
- Passo 4 – Transformação de modelos, realizada através de um plugin para o EPF desenvolvido, por meio desta dissertação, denominado UMA2BPMN;

- Passo 5 – Ajustes, revisão e preparo para implantação do processo, através da ferramenta Yaoqiang BPMN;
- Passo 6 – Execução e apontamento das atividades na ferramenta Activiti;
- Passo 7 – Controle e monitoração do processo através da ferramenta Activiti.

Esta abordagem inicia com o passo 1 que realiza a modelagem e definição dos processos, através da ferramenta EPF, a qual documenta os elementos dos processos utilizando o seu metamodelo UMA. Toda a linha de processos proposta nesta dissertação foi documentada no EPF, gerando todos os elementos e suas variabilidades.

Na Figura 4-9, as atividades da fase de Construção, Composição e Testes podem ser visualizadas no EPF. Como se pode observar, a coluna predecessores (*predecessor*) identifica o elemento que ocorre antes, e todas as informações no EPF foram definidas baseadas nos diagramas da SOPLA (da Figura 4-3 à Figura 4-7). É possível observar que o UMA utiliza o mesmo conceito das ferramentas de gerenciamento de projetos, onde as atividades são relacionadas com suas predecessoras. Esta definição facilita a exportação para um Gráfico de Gantt, para realizar o controle em ferramentas de projeto como o Microsoft Project, recurso nativo do EPF.

Presentation Name	Index	Predecessors	Model Info
Service Oriented Process-Line Approach	0		
Modelagem de negócios e transformação	1		extends 'FASE1_MO...
Definir objetivos, motivação e requisitos de negócio	2		extends 'ATIV_OBJE...
Mapear processo atual e ferramentas	4	2	extends 'ATIV_PRO...
Modelar processo de negócio	6	4	extends 'ATIV_PRO...
Definir indicadores para o processo de negócio	8	6	
Simular execução do processo de negócio	9	6	
Identificação e definição da arquitetura	10	1	extends 'FASE2_IDE...
Captar e modelar domínio de informação	11		extends 'ATIV_IDEN...
Identificar serviços e componentes para o cenário de negócio	13		extends 'ATIV_IDEN...
Definir plataforma e avaliar infraestrutura	15	13	extends 'ATIV_IDEN...
Estimar esforço de desenvolvimento	17	11,15	extends 'ATIV_IDEN...
Especificação e projeto dos serviços	21	10	extends 'FASE3_ESP...
Especificar requisitos da solução	22		extends 'ATIV_ESPE...
Especificar plano e casos de testes	26		extends 'ATIV_ESPE...
Projetar User Interface	28	22	extends 'ATIV_ESPE...
Especificar serviço	32	33	extends 'ATIV_ESPE...
Projetar arquitetura dos serviços e componentes	33	28	extends 'ATIV_ESPE...
Construção, composição e testes	36	10,21	extends 'FASE4_CO...
Construir serviço	37		extends 'ATIV_DESE...
Construir a user interface do usuário	42		extends 'ATIV_DESE...
Configurar componentes de infraestrutura	48		extends 'ATIV_DESE...
Montar e integrar	53	42,48,37	extends 'ATIV_DESE...
Testes funcionais internos	55	53	extends 'ATIV_DESE...
Testes de aceitação	58	55	extends 'ATIV_DESE...
Teste de homologação	60	53	extends 'ATIV_DESE...
Liberação, execução e monitoração	63	36	extends 'FASE5_LIB...

Figura 4-9. EPF com as atividades da fase de Construção, Composição e Testes. Fonte: o Autor, 2012.

Após esta definição, o passo 2 utiliza a ferramenta GenArch-P para a definição e gerenciamento das variabilidades do processo. O passo 3 é a análise das características necessárias ao projeto alvo, seguido pela seleção das variabilidades alternativas e opcionais para realizar a geração de uma nova instância do processo (ALEIXO et al, 2011).

O passo 4 realiza a transformação do modelo UMA para BPMN, onde foi desenvolvido um novo *plugin* para o eclipse que transforma a instância do processo derivado da linha do formato UMA para BPMN versão 2.0. Esta foi a principal contribuição realizada ao ferramental utilizado, onde se utilizou a *Extensible Style Sheet Language Transformation (XLST)*. Este *plugin* do eclipse estende as opções de exportação do EPF, conforme apresentado na Figura 4-10 e é denominado UMA2BPMN.

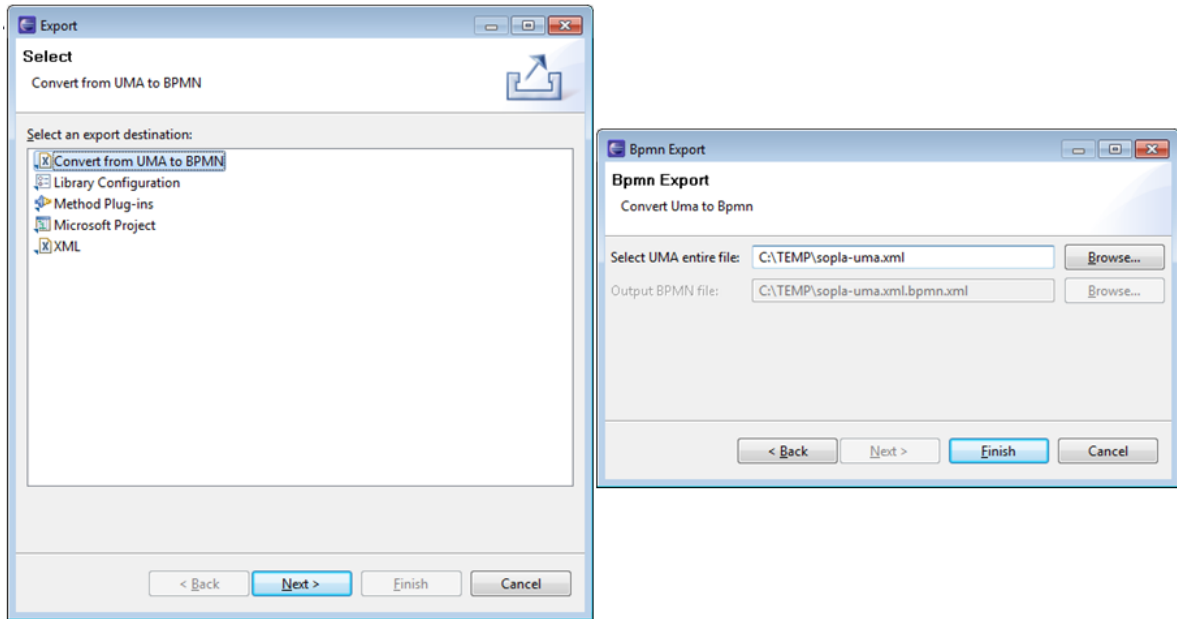


Figura 4-10. Uma2Bpmn Eclipse Plugin

Fonte: o Autor, 2012.

Para geração do arquivo UMA, utiliza-se a funcionalidade de exportar a biblioteca completa da instância do método. Após a criação do arquivo UMA da instância do processo, é realizada a transformação entre os modelos UMA e BPMN. A principal dificuldade do passo 4 está na conversão das informações dos predecessores para gerar todos os fluxos de transição existentes com a definição de origem e destino, conforme modelo apresentado na Figura 4-11.

As regras que realizam a transformação entre o UMA e o BPMN estão resumidas nas seguintes etapas:

- As atividades do UMA (Activiti) são transformadas em *userTask* no BPMN;
- Utilizando os predecessores entre as atividades do UMA foram criados os fluxos entre as atividades (*sequenceFlow* no BPMN);
- As atividades no UMA sem predecessor, no BPMN são iniciadas diretamente pelo evento de inicialização do processo (*startEvent*);
- As atividades no UMA sem sucessor, no BPMN têm um fluxo para o evento de final do processo (*endEvent*);
- As atividades que estão relacionadas com mais de um sucessor no UMA, inclusive o evento de inicialização, na transformação para o BPMN dão origem a inclusão de um *parallelGateway* que se relaciona com todos os demais fluxos que ocorrem separadamente;

- No UMA as atividades com mais de um predecessor, na transformação para o BPMN também geram um *parallelGateway* para junção das atividades inclusive antes do evento de finalização pode ocorrer esta necessidade.

Inicialmente, optou-se por transformar todas as tarefas do processo, mas esta regra foi simplificada para controlar apenas as atividades. Esta simplificação foi necessária devido à dificuldade de realizar os apontamentos e acompanhamentos sobre o fluxo de execução com todas as tarefas, as quais muitas vezes ocorrem de forma simultânea e conjunta. Não somente a SOPLA, mas também métodos clássicos como o RUP, definem apenas o fluxo de execução das atividades, sem definir os detalhes da execução e sequenciamento para as tarefas.

O resultado, após a execução da transformação, é a geração de um arquivo BPMN. A definição dos predecessores apresentados na Figura 4-9, aplicando as regras definidas anteriormente, gera um arquivo similar à Figura 4-11, o qual pode ser visualizado em um diagrama conforme Figura 4-12.

```

<process id="SOPLA" name="SOPLA">
  <startEvent id="startEvent1" />
  <userTask id="const_servico" />
  <userTask id="const_ui" />
  <userTask id="conf_compinfra" />
  <userTask id="montar_integ" />
  <!-- .. -->
  <parallelGateway id="fork_startEvent1" />
  <sequenceFlow id="startEvent1_fork_startEvent1" sourceRef="startEvent1" targetRef="fork_startEvent1" />
  <sequenceFlow id="fork_startEvent1_const_servico" sourceRef="fork_startEvent1" targetRef="const_servico" />
  <sequenceFlow id="fork_startEvent1_const_ui" sourceRef="fork_startEvent1" targetRef="const_ui" />
  <sequenceFlow id="fork_startEvent1_conf_compinfra" sourceRef="fork_startEvent1" targetRef="conf_compinfra" />
  <parallelGateway id="join_montar_integ" />
  <sequenceFlow id="const_ui_join_montar_integ" sourceRef="const_ui" targetRef="join_montar_integ" />
  <sequenceFlow id="const_servico_join_montar_integ" sourceRef="const_servico" targetRef="join_montar_integ" />
  <sequenceFlow id="conf_compinfra_join_montar_integ" sourceRef="conf_compinfra" targetRef="join_montar_integ" />
  <sequenceFlow id="join_montar_integ_montar_integ" sourceRef="join_montar_integ" targetRef="montar_integ" />
  <!-- .. -->
</process>

```

Figura 4-11. Arquivo BPMN resultante da transformação. Fonte: o Autor, 2012.

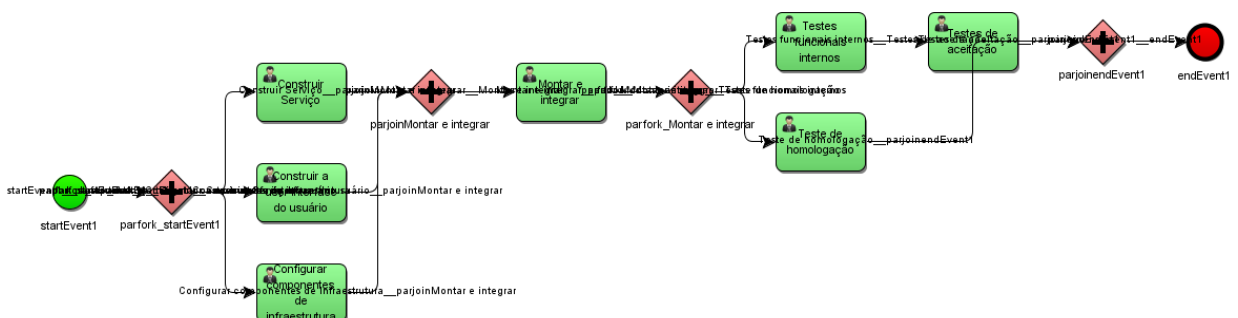


Figura 4-12. Diagrama gerado em BPMN. Fonte: o Autor, 2012.

O passo 5 da abordagem é a revisão do processo, onde utilizou-se uma ferramenta para modelagem BPMN de código aberto (licença GPL), chamada Yaoqiang BPMN. Nesta etapa pode-se visualizar e realizar ajustes visuais no diagrama gerado para o processo. Estes ajustes auxiliam os usuários no acompanhamento e visualização do fluxo de execução do processo, o que pode ser observado comparando a Figura 4-12 e Figura 4-13.

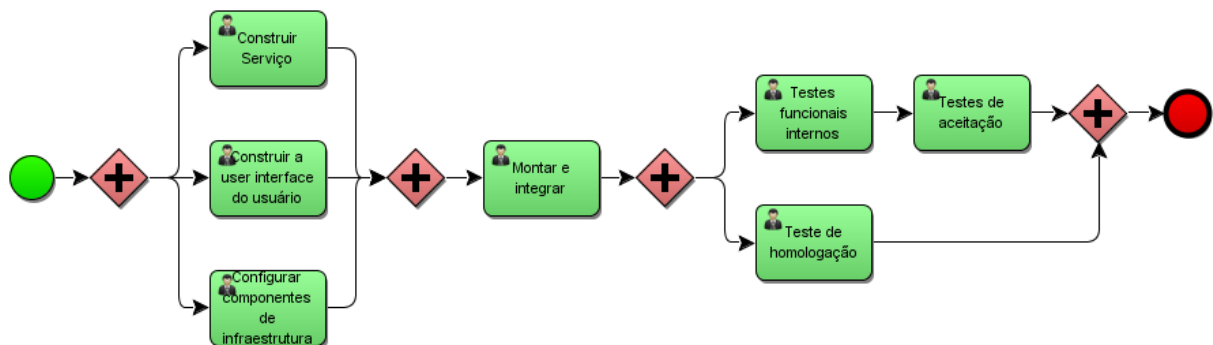


Figura 4-13. Ajustes visuais no editor de BPMN. Fonte: o Autor, 2012.

O passo 6 foi realizado com a carga do modelo BPMN na plataforma de BPM Activiti também de código aberto (licença Apache). O Activiti é uma plataforma de BPM muito enxuta que permite a definição de formulários para realização das atividades de maneira muito fácil e rápida. Para apontamento da realização de cada uma das atividades do processo utilizou-se este recurso, para construção de um formulário padrão para armazenamento das informações. Geralmente utiliza-se um arquivo do tipo .bar para carregar o processo na ferramenta, conforme apresentado na Figura 4-14. A geração deste arquivo pode ser realizada tanto pelos plug-ins do Activiti para o Eclipse, quanto diretamente pela ferramenta Yaoqiang BPMN.

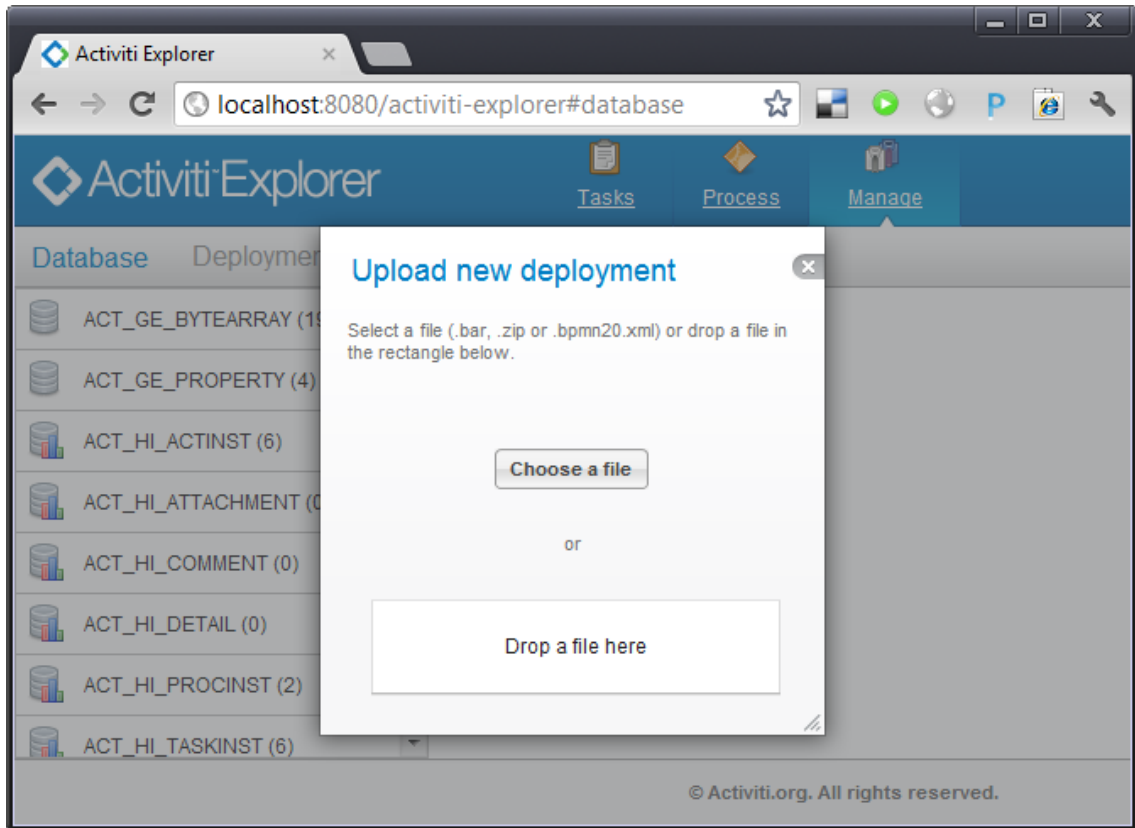


Figura 4-14. Carregar o processo na ferramenta de BPM. Fonte: o Autor, 2012.

O último passo é o controle e monitoração da execução do processo, onde o gerente do projeto utiliza a base de dados da plataforma BPM Activiti. Esta base possui todos os eventos apontados na execução do processo, os quais podem auxiliá-lo na monitoração e tomada de ações corretivas para desvios no projeto. As plataformas comerciais de BPM oferecem recursos adicionais como bases de dados analíticas e ferramentas com painéis de instrumentação (*dashboards*) que facilitam a construção e visualização gráfica das informações.

4.6 Considerações sobre o capítulo

Neste capítulo foi apresentada a proposta de linha de processo de software para BPM e SOA, denominada SOPLA. A documentação da linha ocupou-se a detalhar os papéis, produtos de trabalho, fases do ciclo de vida de desenvolvimento de software, atividades e tarefas com suas variabilidades. Além disso, foi utilizado e adaptado ferramentas para o apoio Computacional à Linha de Processo. Estas adaptações tiveram ênfase na conversão de modelos para possibilitar o controle da execução do processo de software através da plataforma de BPM. O próximo capítulo apresentará a avaliação da linha proposta.

CAPÍTULO 5 - AVALIAÇÃO DA LINHA DE PROCESSOS PROPOSTA

*É tão arriscado acreditar em tudo,
quanto não acreditar em nada. – Denis Diderot*

A avaliação de uma linha de processo de software é um trabalho difícil de ser realizado e que requer um longo período para observar todas as causas e efeitos. Por meio de uma linha de processos podem ser derivados diversos processos, onde cada um pode gerar um experimento para sua avaliação, o que demandaria um considerável custo e período de tempo. Além disso, a proposta do trabalho busca o estabelecimento de processos que alavanquem o reuso sistemático de serviços e processos de negócio. Para comprovar e medir o retorno do reuso sistemático é necessário identificar o comportamento dos ativos de processos e sistemas (artefatos de software ou documental relacionado ao processo de negócio, user interface, serviços, componentes, bases de armazenamento) ao longo do surgimento e atendimento de diferentes necessidades ou requisitos de negócio na linha do tempo. A execução de cada processo derivado da linha está exposta a diversas variáveis que influenciam a execução do processo de software, como: restrição de plataformas tecnológicas, paradigmas de desenvolvimento, experiência dos profissionais, cultura organizacional de planejamento e gerenciamento, trabalho agradável e ferramental adequado (MAFRA; TRAVASSOS, 2006). Realizar um experimento que cubra todas as possibilidades de combinações destas variáveis é uma tarefa desafiadora e que implicaria em um investimento considerável de tempo e recursos.

Devido às dificuldades de realização e aplicação de um estudo experimental ou quasi-experimental, a abordagem utilizada foi a da avaliação por especialistas, conforme descrito no Capítulo 3.

5.1 Caracterização dos especialistas para o estudo

A caracterização buscou identificar o conhecimento e a experiência do especialista em relação a cada uma das três áreas de conhecimento necessárias para avaliação da proposta, definidas como: processo de software, orientação a serviços e BPM.

Conforme se pode observar no questionário (APÊNDICE), para cada área foram identificados itens que pudessem caracterizar esta experiência do especialista. Por exemplo, no que se refere ao processo de software, buscou-se entender o grau de conhecimento do especialista sobre: modelos de processo de software, RUP, processo de software para reuso sistemático, utilização de linhas de produto de software, elaboração de linhas de produto de software, utilização de linhas de processo de software e elaboração de linhas de processo de software. Para obter a caracterização do tópico optou-se pela média aritmética simples entre os itens. O resumo que caracteriza o perfil dos especialistas está apresentado na Tabela 5-1.

Tabela 5-1. Resumos do perfil dos especialistas.

Origem	Atividade e ramo de atuação	Processo Software		Orientação a Serviços		BPM		
		Média (até 5)	Exper. (anos)	Média (até 5)	Exper. (anos)	Média (até 5)	Exper. (anos)	
A	Academia Indústria	Professor de universidade Gerente de Projetos	4,6	6,5	3,6	3	3,8	3
B	Academia	Professor de universidade	4,3	9,5	4,3	5	3,4	3,5
C	Indústria	Arquiteto de Software Indústria financeira	4,0	20	6,3	4	4,0	2
D	Academia Indústria	Professor de universidade Arquiteto Corporativo	2,1	10	4,4	8	2,2	2
E	Indústria	Desenvolvedor	3,7	5	3,7	4	3,4	4
F	Indústria	Consultor Arquiteto de Software	3,6	5	3,8	4	2,4	2
G	Academia Indústria	Professor de universidade Desenvolvedor na área de Serviços de TI e comunicação	2,4	2,5	1,9	1	2,8	1
H	Indústria	Pesquisador de TI	4,1	13	5,0	8	4,6	6
I	Indústria	Analista de Requisitos Serviços de TI e comunicação	3,3	5,5	3,0	0	2,0	1
J	Indústria	Arquiteto Corporativo Indústria de óleo e gás	3,9	12	5,6	4	3,6	3
K	Academia Indústria	Professor de universidade Arquiteto de Software	3,0	6	4,4	4	3,0	3
L	Indústria	Arquiteto de Software Desenvolvedor	3,3	4,5	4,0	6	3,8	2,5
Médias			3,5	8,3	4,2	4,3	3,3	2,8

Fonte: o Autor, 2012.

Conforme se pode observar na Tabela 5-1, o tempo médio de experiência dos entrevistados foi de 8,3 anos no que se refere a processos de software, o que indica uma certa maturidade em relação ao assunto. Somente um dos especialistas possuía experiência de apenas 1 ano. Todos os demais possuíam experiência de pelo menos 4,5 anos com processos de software.

No que diz respeito ao uso de orientação a serviços e de BPM, podemos observar que o tempo médio de experiência dos envolvidos é menor, ficando em 4,8 e 2,8 anos, respectivamente. Isto se deve, em parte, ao pouco tempo de adoção destas práticas na indústria (cerca de 5 a 8 anos, no máximo).

A distribuição geográfica destes especialistas foi a seguinte: 2 especialistas atuando no nordeste do Brasil; 2 especialistas atuando do sudeste do Brasil; 6 especialistas atuando na região sul do Brasil; e 2 especialistas atuando nos Estados Unidos da América.

5.2 Análise dos resultados

5.2.1 Análise da Visão Geral das Fases

A Figura 5-1 apresenta a distribuição das respostas quanto à concordância sobre a completude das fases. Conforme se pode observar, no que diz respeito às fases e sua completude, 10 dos 12 especialistas (83,3%) concordam que a proposta é adequada ou que necessita apenas de pequenos ajustes. As principais sugestões de melhoria quanto à completude das fases foram:

- Especialistas B e G: comentaram sobre a falta da definição clara dos marcos que finalizariam cada fase, de modo a facilitar o fechamento do projeto e reduzir as iterações de identificação, especificação, construção e liberação;
- Especialistas H e L: comentaram que a fase de Modelagem de Negócios e Transformação não deveria ser opcional, pois sem a definição de um benefício claro, nada deveria ser desenvolvido.

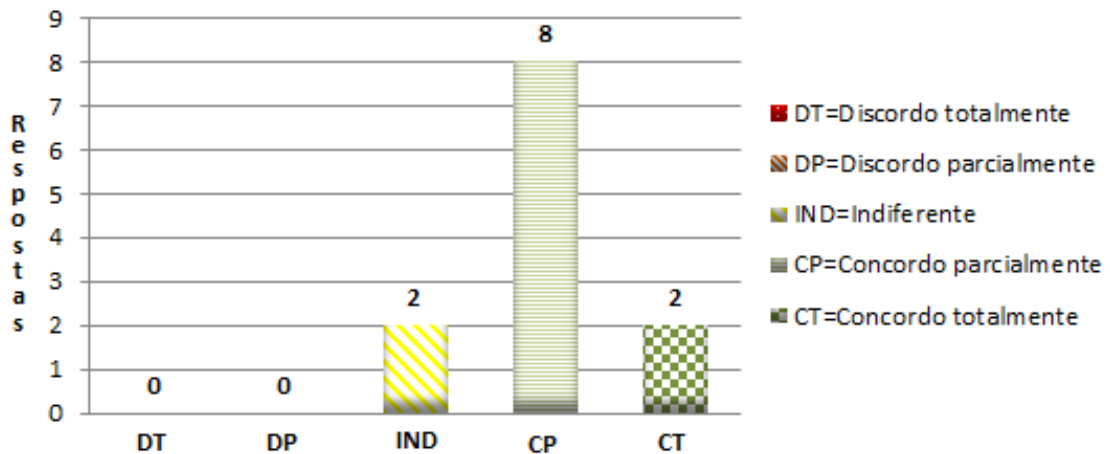


Figura 5-1. Avaliação da completude das fases do SOPLA. Fonte: o Autor, 2012.

Dentre as sugestões realizadas sobre a modificação de fases (adição ou junção), as principais contribuições foram:

- Especialistas E e G: sugeriram particionar a fase de Construção, Composição e Testes. A justificativa para esta divisão é que isto possibilitaria a utilização de fábricas de testes distribuídas geograficamente;
- Especialista A: comentou que conforme a natureza do projeto, algumas destas fases poderiam ser integradas e as variabilidades da linha deveriam contemplar esta necessidade.

Já as principais sugestões quanto às transições definidas foram:

- Especialistas C e I: comentaram que não está nítida a sobreposição das fases e sugeriram a definição de um gráfico de ondas para representar, de modo semelhante ao RUP;
- Especialista F: também fez observações sobre a sobreposição e comentou que a transição não deve acontecer de forma sequencial (cascata) e sugeriu melhorar a transição das fases para refletir melhor o ciclo iterativo e incremental.

Quanto às variabilidades das fases, as principais sugestões foram:

- Especialista A: comentou sobre a dificuldade de gerenciar um projeto e seu processo, devido às variabilidades ocorrerem em diversos momentos como na definição da linha da organização, na instanciação do processo e na execução;

- Especialista B: comentou que as transições entre as fases poderiam ocorrer de diferentes maneiras, conforme o tipo de projeto;
- Especialista C: comentou que é necessário aprimorar a definição das restrições entre as escolhas das variabilidades e dependências na documentação da linha de processo com o objetivo de melhorar a reutilização dos produtos de trabalho.

A Figura 5-2 apresenta a distribuição das respostas da avaliação quanto às sugestões de modificação de elementos no nível das fases, melhorias nas transições entre as fases e alterações nas variabilidades propostas.

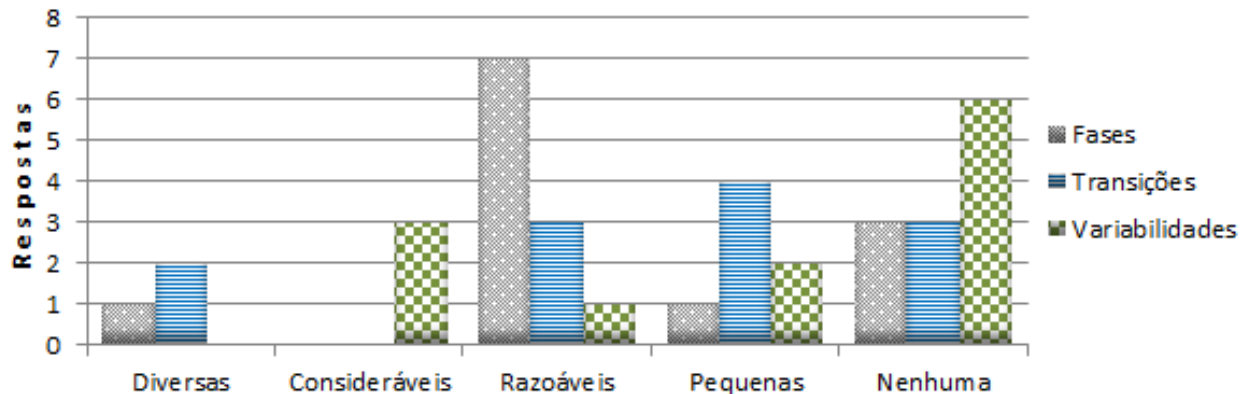


Figura 5-2. Avaliação das sugestões das Fases do SOPLA. Fonte: o Autor, 2012.

5.2.2 Análise da Fase de Modelagem de Negócios e Transformação

A Figura 5-3 apresenta a distribuição das respostas quanto à completude das atividades propostas para a fase de Modelagem de Negócio e Transformação. Nesta distribuição relacionada à completude desta fase, se pode observar: 6 especialistas (50%) acreditam que a proposta é adequada, sendo que 4 concordam totalmente e 2 mediante pequenos ajustes. Já 5 especialistas (41,7%) não concordaram e nem discordaram (tendência neutra) e um especialista (8,3%) discordou em diversos pontos. As principais sugestões de melhoria quanto à completude das fases foram:

- O especialista C: comentou sobre a importância do estabelecimento desta fase para que os processos de negócios sofram melhorias contínuas, automações, monitoração e evolução contínua. Entretanto, existem importantes melhorias no processo que devem motivar o crescimento e conduzir a organização a outro patamar. A inovação é

muitas vezes o caminho para alcance deste novo patamar, mas o fluxo de atividades proposto na linha pode dificultar estes objetivos, e impedir a criatividade;

- Especialista D: sugeriu a inclusão de algumas abordagens para a modelagem do processo futuro (TO-BE), como: engenharia de valor, benchmarking e consulta a repositórios de melhores práticas;
- Especialista H: sugeriu a elaboração de guias para as atividades e tarefas da fase;
- Especialista L: avaliou que a principal atividade ATIV_PROC_TOBE não poderia ser opcional, mas se esta realmente for opcional, então não seria necessário tornar a fase opcional.

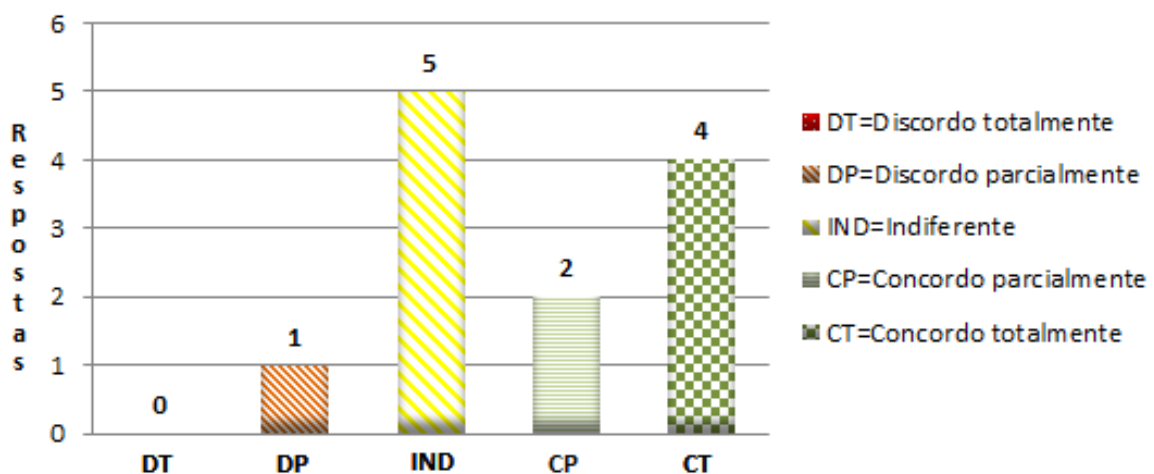


Figura 5-3. Avaliação da completude da Fase de Modelagem de Negócios e Transformação.
Fonte: o Autor, 2012.

Quanto às sugestões relacionadas à modificação de atividades (adição ou junção) ou tarefas na linha, as principais contribuições foram:

- Especialista A: sugeriu uma tarefa adicional na TAR_PROC_KPI que é a validação deste indicador com o responsável pelo gerenciamento do processo de negócio;
- Especialista D: sugeriu incluir a tarefa de “Vincular os objetivos do projeto com o plano estratégico da organização”. O especialista C ressaltou que o mais importante é conhecer o processo atual e conhecer as possibilidades reais de desenhar algo que traga a transformação, com a aplicação de novas tecnologias e inovações.

As principais sugestões de melhoria quanto à transição entre as atividades foram:

- Especialista C: acrescentou que a ATIV_PROC_TOBE precisa ter um nível de documentação adequado. Para ter o nível adequado de profundidade, o analista de negócio não deve detalhar demais, permitindo à organização e aos gestores um controle eficaz e, também, não deve ser simplória;
- Especialista D: abordou novamente o tema das sobreposições das fases, comentando que algumas atividades da fase de Modelagem de Negócios podem ocorrer em paralelo com a fase de Transformação Identificação e Definição da Arquitetura.

Quanto às variabilidades, as principais sugestões foram:

- Especialista A: sugeriu analisar o resultado da execução de alguns projetos para capturar mais atividades opcionais e alternativas. Também foi sugerido que as ferramentas da atividade ATIV_PROC_TOBE fossem definidas na linha com uma variabilidade mais fina, definida somente na ferramenta utilizada;
- Especialista L: sugeriu quebrar a atividade de mapeamento do processo atual em mais atividades. Esta divisão buscaria separar a parte de mapeamento de processo, da identificação das ferramentas utilizadas no processo. Somente a segunda atividade, de mapeamento de ferramenta, que seria a atividade opcional.

A Figura 5-4 apresenta a distribuição das respostas da avaliação quanto às sugestões de modificação dos elementos dentro da Fase de Modelagem de Negócios e Transformação, melhorias nas transições entre as atividades e alterações nas variabilidades propostas.

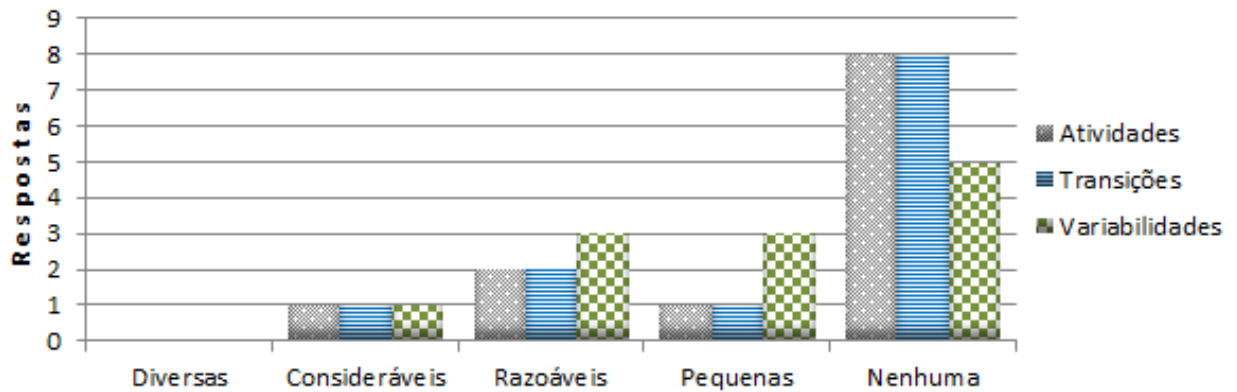


Figura 5-4. Avaliação das sugestões para Fase de Modelagem de Negócios e Transformação.
Fonte: o Autor, 2012.

5.2.3 Análise da Fase de Identificação e Definição da Arquitetura

A Figura 5-5 apresenta a distribuição das respostas quanto à completude das atividades da fase de Identificação e Definição da Arquitetura. Conforme se pode observar, no que diz respeito à completude, 8 dos 12 especialistas (66,7%) concordam que a proposta é adequada ou que necessita apenas de pequenos ajustes. Já 3 dos 12 especialistas (25%) não concordaram e nem discordaram, enquanto apenas um discordou em muitos pontos. As principais sugestões de melhoria quanto à completude das fases foram:

- Especialista C: comentou a necessidade de seguir padrões para definição do nome dos serviços e contratos baseado em um framework para garantir uma boa governança, mesmo que o contrato seja finalizado na especificação da fase seguinte, nesta fase é importante uma definição ou avaliação dos itens existentes no repositório de serviços.
- Especialista K: sugeriu que na atividade ATIV_IDENT_SRV se poderia ampliar a saída da atividade, pois durante a tarefa Avaliação de Aderência seria conveniente iniciar um rascunho da documentação dos Casos de Testes que seriam a evidência da aderência alcançada.

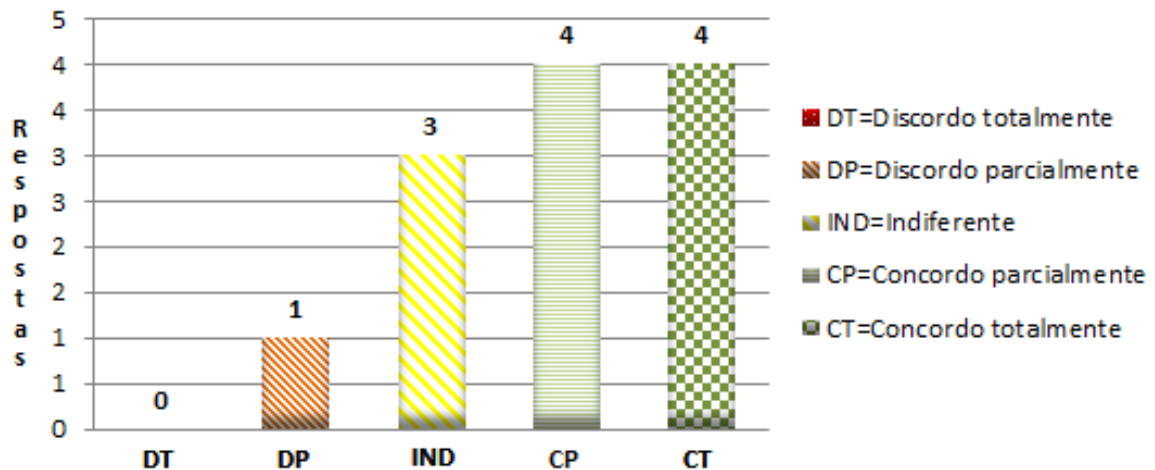


Figura 5-5. Avaliação da completude da Fase de Identificação e Definição da Arquitetura.
 Fonte: o Autor, 2012.

Dentre as sugestões realizadas sobre a modificação de atividades e tarefas, a principal contribuição foi:

- Especialistas B e H: sugeriram complementar a fase com a definição de uma atividade para definição e validação dos requisitos não funcionais. Na fase anterior os modelos e requisitos de negócio foram estabelecidos, mas antes de definir a arquitetura e estimar é fundamental a existência de um responsável pela definição e validação das restrições da solução com o cliente. Os elementos sugeridos ao processo foram: Definição dos requisitos arquiteturais para o contexto proposto; e Avaliar se a arquitetura proposta atende a cada um dos requisitos arquiteturais identificados.

Quanto à transição entre as atividades, as principais sugestões de melhorias foram:

- Especialista G: concordou parcialmente com a criação da fase Identificação e Definição da Arquitetura, entre a modelagem de negócio e a engenharia do produto e sugeriu que a definição do domínio não fosse paralela e que fosse sequencial;
- Especialistas C e D: reforçaram os comentários sobre o paralelismo com a fase anterior.

As sugestões de melhorias relacionadas às variabilidades foram:

- Especialista C: novamente ressaltou a importância das restrições entre as atividades do processo, se o caso de negócio utilizou use case,

seria prático utilizar a análise de pontos por caso de uso; mas, se a prototipação foi realizada, então a análise de pontos de função é a atividade de estimativa recomendada para melhorar a reutilização dos artefatos gerados e melhorar o desempenho do processo;

- Especialista H: comentou que a atividade ATIV_IDENT_DOMINIO não deveria ser opcional.

A Figura 5-6 apresenta a distribuição das respostas da avaliação quanto às sugestões de modificação dos elementos dentro da Fase de Identificação e Definição da Arquitetura, melhorias nas transições entre as atividades e alterações nas variabilidades propostas.

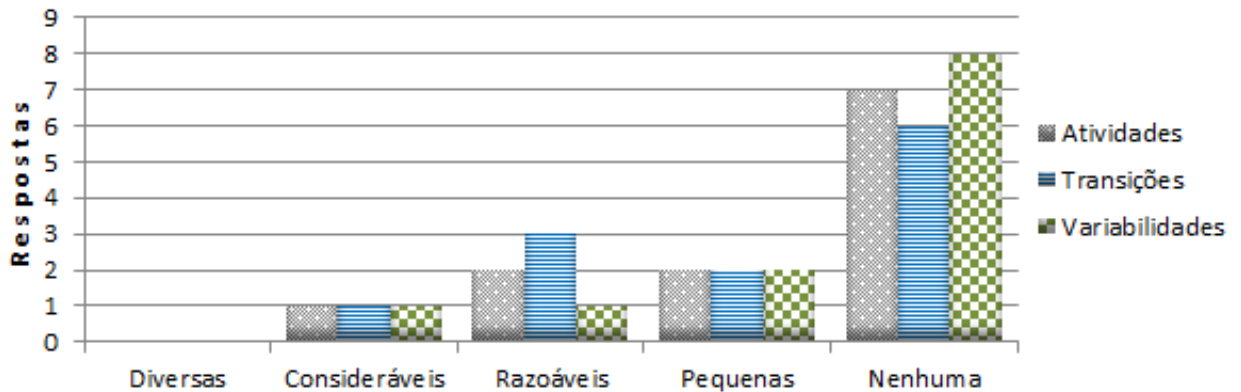


Figura 5-6. Avaliação das sugestões para Fase de Identificação e Definição da Arquitetura.
Fonte: o Autor, 2012.

5.2.4 Análise da Fase de Especificação e Projeto dos Serviços

A Figura 5-7 apresenta a distribuição das respostas quanto à completude das atividades da fase de Especificação e Projetos dos Serviços. Conforme se pode observar, no que diz respeito à completude desta fase, 7 dos 12 especialistas (58,3%) concordam que a proposta é totalmente adequada ou que necessita apenas de pequenos ajustes. As principais sugestões de melhoria quanto à completude das fases foram:

- Especialista B: sugeriu melhorar a descrição e objetivo de ter a atividade ATIV_ESPEC_ARQ, visto que na fase anterior a plataforma e arquitetura já foram estabelecidas. Outra sugestão alternativa do especialista foi a fusão das atividades ATIV_ESPEC_ARQ e ATIV_ESPEC_SRV;

- Especialista C: concordou com a descrição do processo, e destacou a importância de identificar a origem das informações, transformação dos dados, utilização do inventário, medir e controlar a granularidade dos serviços para facilitar e motivar a reutilização.

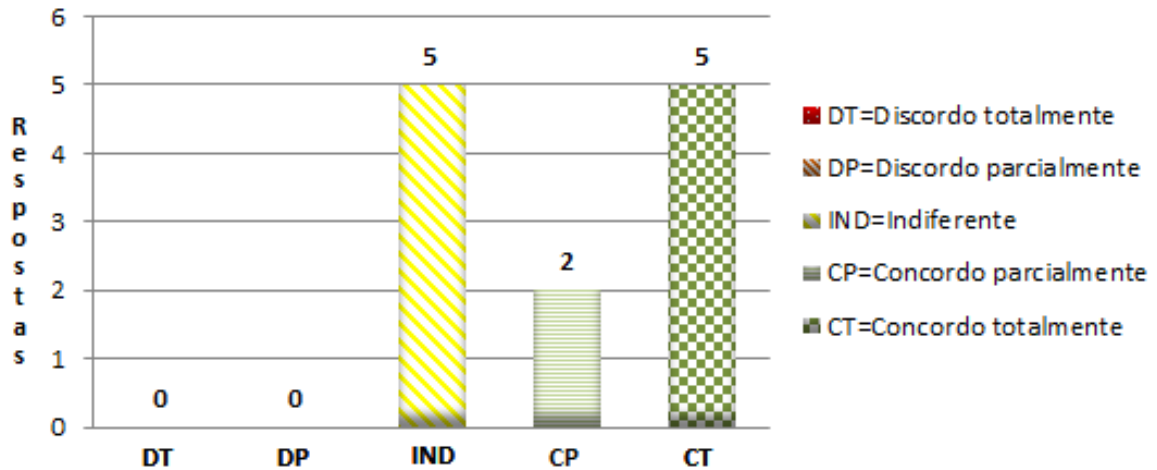


Figura 5-7. Avaliação da completude da Fase de Especificação e Projeto dos Serviços. Fonte: o Autor, 2012.

Dentre as sugestões realizadas sobre a modificação de atividades e tarefas, a principal contribuição foi:

- Especialistas A e F: sugeriram melhorar a definição da atividade ATIV_ESPEC_UI, deixando de forma explícita a validação do protótipo da user interface juntamente com a entrada e saída das informações.

As principais sugestões de melhoria quanto à transição entre as atividades foram:

- Especialista C: sugeriu, junto à tarefa TAR_ESPEC_SRV_REG, incluir a junção dos processos de revisão de nomes e governança do inventário de serviços;
- Especialistas K e G: não estavam certos da posição da atividade ATIV_ESPEC_ARQ, onde talvez a atividade ATIV_ESPEC_UI pudesse ocorrer depois. Até mesmo especialistas ainda não perceberam a importância da arquitetura no desenvolvimento das interfaces de usuário, que é uma camada onde a complexidade tem aumentado consideravelmente.

As principais sugestões de melhorias relacionadas às variabilidades apresentadas nesta fase foram:

- Especialista A: sugeriu explorar mais as variabilidades de tarefas e artefatos;
- Especialista H: não concordou com a atividade ATIV_ESPEC_TEST definida como opcional, pois ele acredita que este artefato sempre deveria ser construído.

A Figura 5-8 apresenta a distribuição das respostas da avaliação quanto às sugestões de modificação dos elementos dentro da Fase de Especificação e Projeto dos Serviços, melhorias nas transições entre as atividades e alterações nas variabilidades propostas.

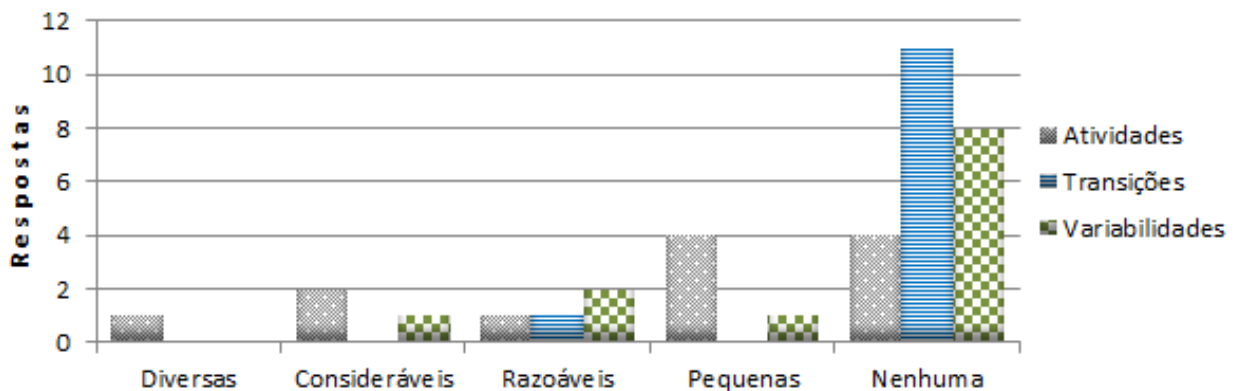


Figura 5-8. Avaliação das sugestões para Fase de Especificação e Projeto dos Serviços. Fonte: o Autor, 2012.

5.2.5 Análise da Fase de Construção, Composição e Testes

A Figura 5-9 apresenta a distribuição das respostas quanto à completude das atividades da fase. Conforme se pode observar, no que diz respeito às fases e sua completude, 9 dos 12 especialistas (75%) concordam que a proposta é adequada ou que necessita apenas de pequenos ajustes. A principal sugestão de melhoria quanto à completude das fases foi:

- O especialista C: alertou que é muito difícil adequar todos os sistemas que utilizam um serviço para uma nova versão, assim é importante que a construção contemple orientações para realizar o versionamento. Através do versionamento é possível permitir a execução de mais de uma versão do mesmo serviço em ambiente

produtivo. Por isso, ele sugere incluir guias na ATIV_CONST_SRV para detalhar as práticas relacionadas ao versionamento.

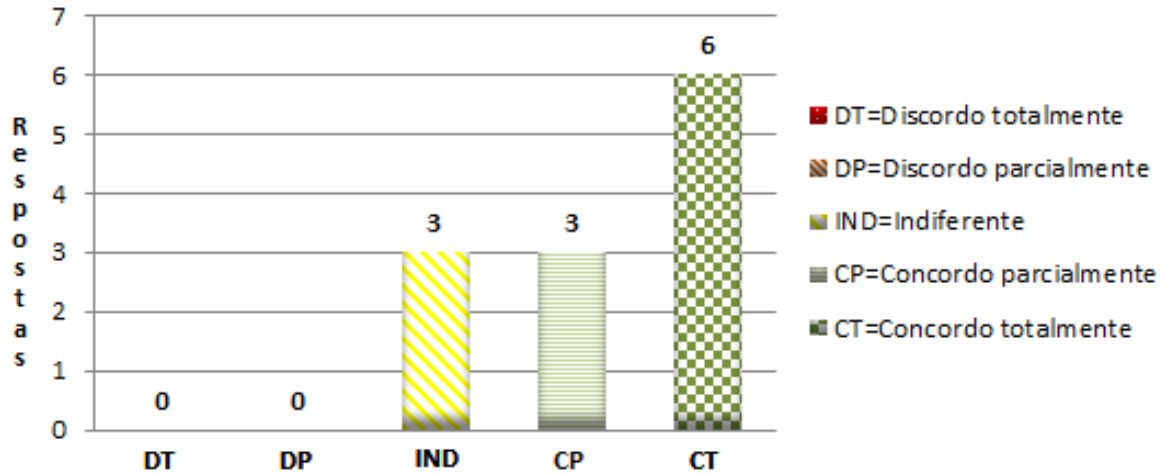


Figura 5-9. Avaliação da completude da Fase de Construção, Composição e Testes. Fonte: o Autor, 2012.

Dentre as sugestões realizadas sobre a modificação de atividades e tarefas da Fase de Construção, Composição e Testes, as principais contribuições foram:

- Especialista C: sugeriu descrever atividades ou guias na fase de construção relacionadas: à melhoria de desempenho dos serviços, à melhoria do projeto e à transformação de informações;
- Especialista H: sugeriu que o banco de dados deveria ser projetado na fase anterior.

A principal sugestão de melhoria quanto à transição entre as atividades desta Fase foi:

- Especialista E: novamente frisou a separação das fases de construção e testes, onde a transição da fase de construção poderia ou não estar ligados com a fase de testes.

A principal sugestão de melhoria relacionada às variabilidades apresentadas nesta fase foi:

- Especialista B: acredita que algumas atividades podem sofrer variações de diferentes formas, responsabilidades, profundidade e definição dos responsáveis, tais como as atividades ATIV_DESEN_TESTE_INTERNO e ATIV_DESEN_TESTE_ACEITA.

A Figura 5-10 apresenta a distribuição das respostas da avaliação quanto as sugestões de modificação dos elementos dentro da Fase de Construção, Composição e Testes, melhorias nas transições entre as atividades e alterações nas variabilidades propostas.

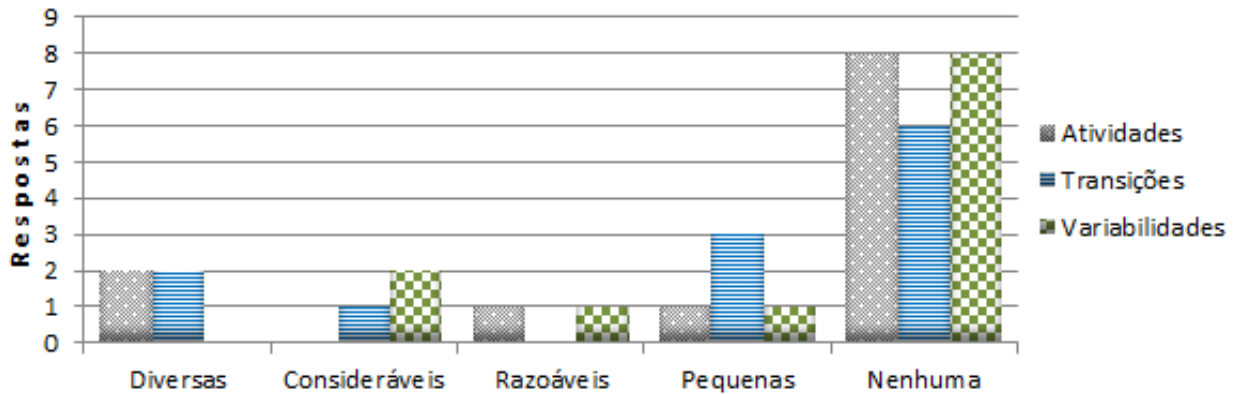


Figura 5-10. Avaliação das sugestões para Fase de Construção, Composição e Testes. Fonte: o Autor, 2012.

Dentre as observações gerais, houve diversas contribuições para a linha e também para participação na concepção e uso do produto. As principais sugestões foram:

- Especialista B: considerou a proposta como pronta para realização de estudos experimentais controlados ou não;
- Especialista E: observou as possibilidades de integração com as demais atividades dos modelos clássicos UP, Agile, XP, para complementar principalmente as questões não abordadas de planejamento. O especialista também comentou sobre a necessidade de TI aplicar estes modelos em ferramentas como de BPM, para controlar e realmente utilizar o processo de fato no dia a dia das organizações. Comentário que está em conformidade como os objetivos do trabalho desenvolvido, pois foram utilizadas ferramentas para apoiar a linha, o que não foi divulgado aos especialistas;
- Especialistas H e I: observaram a ausência de guias e modelos para apoiar a realização detalhada das atividades; e,
- Especialista L: sugeriu que a atividade ATIV_DESEN_TESTE_HOMO inicia-se depois da execução da atividade ATIV_DESEN_TESTE_INTERNO.

5.3 Sumário das Avaliações dos Especialistas

Conforme definido na seção 3.4.6, a Tabela 5-2 apresenta o resumo da análise dos resultados da avaliação dos especialistas e interpretação dos dados conforme Figura 3-3.

Tabela 5-2. Sumário dos resultados das avaliações dos especialistas.

Questão	DT	DP	IND	CP	CT	Análise
Visão geral das Fases da linha						
Completude das fases	0	0	2	8	2	Tendência de Aceitação
Modificação do processo	1	0	7	1	3	Tendência de Aceitação
Fluxo de transições	2	0	3	4	3	Tendência de Aceitação
Variabilidades	0	3	1	2	6	Tendência de Aceitação
Total	3	3	13	15	14	
Fase de Modelagem de Negócios e Transformação						
Completude	0	1	5	2	4	Tendência de Aceitação
Modificação do processo	0	1	2	1	8	Tendência de Aceitação
Fluxo de transição	0	1	2	1	8	Tendência de Aceitação
Variabilidades	0	1	3	3	5	Tendência de Aceitação
Total	0	4	12	7	25	
Fase de Identificação e Definição da Arquitetura						
Completude	0	1	3	4	4	Tendência de Aceitação
Modificação do processo	0	1	2	2	7	Tendência de Aceitação
Fluxo de transição	0	1	3	2	6	Tendência de Aceitação
Variabilidades	0	1	1	2	8	Tendência de Aceitação
Total	0	4	9	10	25	
Análise da Fase de Especificação e Projeto dos Serviços						
Completude	0	0	5	2	5	Tendência de Aceitação
Modificação do processo	1	2	1	4	4	Tendência de Aceitação
Fluxo de transição	0	0	1	0	11	Tendência de Aceitação
Variabilidades	0	1	2	1	8	Tendência de Aceitação
Total	1	3	9	7	28	
Análise da Fase de Construção, Composição e Testes						
Completude	0	0	3	3	6	Tendência de Aceitação

Modificação do processo	2	0	1	1	8	Tendência de Aceitação
Fluxo de transição	2	1	0	3	6	Tendência de Aceitação
Variabilidades	0	2	1	1	8	Tendência de Aceitação
Total	4	3	5	8	28	
Total geral	8	17	48	47	120	

Na Visão geral das Fases, 4 dos 12 especialistas selecionaram opções de discordo totalmente ou parcialmente, totalizando 6 sugestões de melhorias, onde foram destacadas: as diversas alternativas de execução das fases; múltiplas relações temporais para ocorrência das fases; e, a variabilidade opcional da fase de modelagem de negócio foi questionada. Contudo a linha já possui uma tendência de aceitação quanto às possibilidades de transições entre as fases definidas, conforme 7 concordâncias, 3 posicionamentos neutros e 2 discordância. Por isto, a sugestão de melhoria não gerou revisões da linha.

A questão relacionada ao fluxo de transição gerou descontentamento de 2 especialistas, que selecionaram a opção “Discordo totalmente”. Nesta questão foi sugerida a divisão da Fase de Construção, Composição e Testes, separando as atividades de teste e gerando uma nova fase. Através desta nova fase pode-se melhorar a integração com equipes de testes geograficamente distribuídas. Estas sugestões geraram respostas dobradas tanto na “visão geral das Fases da linha”, quanto na “Fase de Construção, Composição e Testes”, onde ainda gerou sugestão em duas situações na “modificação do processo” e no “fluxo de transição”, totalizando 6 sugestões. Contudo, muitas iniciativas para melhoria de processo têm apontado justamente para o caminho contrário a esta separação, onde os testes têm tomado a frente e dirigido o desenvolvimento. O princípio de fazer certo da primeira vez demanda um maior esforço na compreensão e definição dos testes nas diversas etapas do desenvolvimento, o que propicia a elaboração, construção e entrega de um produto com maior assertividade e qualidade. Por isto, esta sugestão não foi incluída na revisão da linha.

Também houve uma sugestão de inclusão da revisão ou da tarefa TAR_PROC_KPI que gerou outra resposta “Discordo parcialmente”. Dentro desta tarefa existe um passo de validação, contudo a sugestão foi aceita. A definição de uma atividade para validação explícita, tanto do modelo futuro, KPI e comparativos devem ser revisadas e também validadas pelo patrocinador ou grupo de governança

de processos. Contudo esta atividade permeia as atividades de planejamento que foram evitadas na concepção da linha de processos.

- Melhoria 1 – Inclusão de nova atividade: Validação das melhorias no processo de negócio. Os participantes seriam um conselho do processo de negócio alvo da melhoria e o patrocinador. No preparo do material para esta validação pode-se utilizar técnicas como variabilidades opcionais para embasar a proposta como: benchmarking e engenharia de valor.

A ausência de uma atividade explícita para a elicitación dos requisitos não funcionais e a construção de um protótipo arquitetural para avaliar a viabilidade técnica também gerou uma resposta “Discordo totalmente” e uma “Discordo parcialmente”. Contudo, os requisitos não funcionais estão contemplados na linha através de uma tarefa na atividade ATIV_IDENT_SRV. A sugestão de contemplar o modelo através de uma atividade opcional para validação da arquitetura foi aceita, mesmo esta prática podendo ser realizada com uma iteração curta inicial entre as Fases de “Identificação e Definição da Arquitetura”, “Especificação e Projeto dos Serviços” e “Construção, Composição e Testes”.

- Melhoria 2 – Inclusão de nova atividade: Validação da arquitetura proposta. Os participantes seriam o conselho de arquitetura e o patrocinador.

Outro ponto que gerou uma resposta “Discordo totalmente”, foi a Fase de Especificação e Projeto dos Serviços dentro da atividade ATIV_ESPEC_UI, na qual foi sugerido incluir uma atividade contribuinte para mapeamento e validação da entrada de dados do usuário. Outro especialista também sugeriu incluir nesta atividade uma tarefa de revisão que gerou uma resposta de “Discordo parcialmente”. A atividade ATIV_ESPEC_UI contém uma tarefa na sua descrição que contempla a validação com o usuário, já as sugestões de tornar explícito o mapeamento das informações e revisão foi aceita. Na Fase de Especificação e Projeto dos Serviços, também houve uma resposta “Discordo parcialmente” devido à sugestão de junção entre as atividades ATIV_ESPEC_ARQ e ATIV_ESPEC_SRV. Neste quesito a sugestão não foi incluída na revisão devido a natureza da atividade ATIV_ESPEC_ARQ que é direcionada ao projeto de uma solução que consome e prove serviços, enquanto a atividade ATIV_ESPEC_SRV tem o intuito de gerar um artefato de documentação para um serviço reutilizável por muitas aplicações. Para a

atividade ATIV_ESPEC_SRV também houve uma resposta “Discordo parcialmente”, onde foi sugerida melhorias para contemplar a governança de inventários de serviços, sugestão que foi aceita.

- Melhoria 3 – Na ATIV_ESPEC_UI devem ser incluídas a tarefa: Mapear as informações e validações das entradas de dados.
- Melhoria 4 – Junto à atividade ATIV_ESPEC_SRV, incluir tarefas de revisão de nomes e governança do inventário de serviços;

Um especialista selecionou 3 respostas “Discordo parcialmente” devido à ausência de granularidades de mais baixo nível no processo, onde a expectativa era definir variabilidades para as diversas tarefas da linha de processos. A análise das variabilidades de todas as tarefas foi uma restrição do escopo deste trabalho, por isso a sugestão não foi gerou novos pontos de revisão da linha. As atividades ATIV_DESEN_TESTE_INTERNO e ATIV_DESEN_TESTE_ACEITA também originaram duas respostas “Discordo parcialmente”, sendo uma na transição e outra nas variabilidades. A sugestão realizada quanto à transição não gerou novas revisões, pois o sequenciamento reduziria o desempenho médio de entrega, que é um direcionador comum dos processos para SOA e BPM reduzir o *time to market*. As sugestões quanto às variabilidades requer o detalhamento de cada tarefa e seus responsáveis, sendo esta uma restrição do escopo do trabalho.

5.4 Análise das Proposições Iniciais

Conforme os resultados da avaliação por especialistas, e analisando as respectivas proposições definidas para a pesquisa, pode-se concluir que:

- **Proposição 1:** Os elementos fundamentais da linha proposta atendem às necessidades para a derivação de novos processos.
 - Nas questões relacionadas, 40 entre as 60 questões objetivas foram marcadas com concordância e nenhum dos especialistas selecionou a escala mínima (discordo totalmente), o que indica fortes indícios quanto ao atendimento desta proposição.
- **Proposição 2:** O modelo proposto é capaz de gerenciar as variabilidades, complementando os elementos fundamentais com os elementos adicionais relativos à realidade das organizações.
 - Nas questões relacionadas, 44 entre as 60 questões objetivas foram marcadas com concordância e nenhum dos especialistas

selecionou a escala mínima (discordo totalmente). Contudo, a linha deu maior ênfase as atividades, não abordando se aprofundando no detalhe das tarefas e passos da linha.

- **Proposição 3:** A transição entre as fases e atividades da linha atendem às necessidades, sem precisar de customizações.
 - Nas questões relacionadas, também, 44 entre as 60 questões objetivas foram marcadas com concordância. Dentre as 60 questões, 7 respostas estavam em discordância devido a: falta de uma fase para estabilização e testes; e, visão mais tradicional de um fluxo de processo mais sequencial minimizando atividades paralelas. Contudo, processos dirigidos por testes e maximizar as atividades realizadas em paralelo são tendências para alcançar ganhos de produtividade nos processos.
- **Proposição 4:** A fusão das práticas de SOA e BPM em uma única linha de processos é adequada e aplicável para as atividades dos engenheiros de software.
 - A sugestão de modificação nos elementos do processo, seja adição, junção ou exclusão de elementos, para melhor atendimento do escopo da linha, obteve a concordância de 39 entre as 60 questões objetivas. Dentre as 60 questões obteve-se 8 discordâncias, composta por 5 posicionamentos com elementos complementares à linha; 1 sugestão de junção; e, 2 sugestão de exclusão da atividade. Alguns dos demais especialistas fizeram importantes contribuições para melhorar a sinergia entre os modelos, como a sugestão de aplicar de experimentos na indústria e melhorias na reutilização de artefatos, como previamente descrito.
- **Proposição 5:** O tempo para derivação de um novo processo de uma linha de processo pode ser reduzido e ou automatizado através de ferramentas adequadas.
 - O ferramental utilizado foi o EPF desenvolvido pela IBM, o GenArch-P adaptado por (ALEIXO et al, 2011), e a ferramenta desenvolvida por este trabalho, UMA2BPMN, para automatizar a transformação das instâncias de processos para BPMN. Esta ferramenta foi integrada através do desenvolvimento de um novo

plugin para o EPF. Após a exportação do BPMN, carregou-se este arquivo nas plataformas de BPM que possibilitam a execução e controle do processo. Esta proposição foi parcialmente atendida, pois este conjunto de ferramentas é experimental e precisa de maior maturidade, estabilidade e experimentos em contextos reais de utilização.

5.5 Considerações sobre o capítulo

Este capítulo apresentou o processo de avaliação por especialistas, desde seu planejamento, definição, seleção dos especialistas, coleta de dados, análise das opiniões agregadas e resultados. A proposta SOPLA foi avaliada através das sugestões de melhoria vindas dos especialistas. Os pontos destacados na avaliação contribuem para o amadurecimento do modelo para aplicação em futuros trabalhos na academia ou indústria. Por fim, as proposições definidas para a pesquisa foram avaliadas conforme o resultado da avaliação por especialistas.

CAPÍTULO 6 - CONSIDERAÇÕES FINAIS

A área de software não é uma área simples e está se tornando mais complexa em um ritmo mais rápido do que podemos colocar em ordem.

- B.W. Boehm

Após apresentação dos resultados da proposta é tempo de descrever a relevância do estudo, contribuição da pesquisa, limitações do trabalho e apontar a direção de trabalhos futuros que possam dar continuidade e amadurecer os trabalhos envolvidos.

6.1 Relevância do estudo

A orientação a serviços assim como o BPM são assuntos recentes para o mercado, e que carece de processos de desenvolvimento e governança bem definidos. A adoção de processos de desenvolvimento de software voltados ao reúso é um tema que tem sido abordado na academia, mas ainda pouco tem endereçado questões relativas ao desenvolvimento de serviços. A proposta SOLPLA explorou os modelos existentes para definição de uma linha de processos que pudesse melhorar o processo de desenvolvimento de produtos desta natureza e integrar os paradigmas, ferramentas e práticas envolvidas.

Conforme já citado, SOA já se tornou um item indispensável para as organizações e suas tecnologias são utilizadas no dia a dia dos mais diversos processos de negócio. O desenvolvimento de aplicações orientadas a serviços é um pré-requisito para alavancar 3 dos 5 assuntos mais importantes para o mercado de TI, conforme a pesquisa de prioridades dos CIOs realizada pelo (GARTNER, 2012), a saber: 1ª Entrega de soluções de negócio para dispositivos móveis; 3ª Estabelecimento de infraestrutura flexíveis e com custo reduzido utilizando a computação em nuvem (SAAS, PAAS e IAAS); e, 5ª Construção de soluções colaborativas (gerenciamento do fluxo de trabalho - BPMS).

A indústria de manufatura tem aplicado o controle de variabilidades de produto há longa data, este fato proporcionou um cenário favorável para aplicação desta abordagem para softwares embarcados. Embora a aplicação de variabilidade de produtos de software seja uma abordagem bastante estudada pela academia, grande parte da indústria de software ainda desconhece o assunto. Por isso, a aplicação desta abordagem para processo de software é um assunto pouco abordado e que torna o estudo realizado ainda mais relevante. Para evidenciar a pequena abordagem do assunto, pode-se citar o retorno das pesquisas nas bases ACM, IEEE, Science Direct, Springer e Wiley que retornam apenas 32 referências distintas na busca por *Software Process Line(s)* ou *Family(ies)*. A definição de linhas de processos de software com o gerenciamento sistemático de suas variabilidades tem sido pouco abordada na academia e indústria. Contudo, linhas de processo de software é uma abordagem relevante que pode auxiliar as organizações a alcançar ganhos de qualidade e produtividade.

6.2 Contribuições da pesquisa

A principal contribuição deste trabalho de pesquisa foi a definição de uma linha de processo de software para a engenharia de produtos orientados a serviços. A relevância para o mercado de TI é grande, pois existem poucos modelos práticos para apoiar a definição de processos de engenharia de software conforme as características dos projetos das organizações. A utilização de linhas de processos de software ainda é uma prática pouco aplicada e que carece de experimentos, práticas e ferramentas.

A segunda contribuição foi a aplicação e adaptação da abordagem de (ALEIXO et al, 2011) para definição da linha de processos. Nesta contribuição destaca-se o desenvolvimento de uma ferramenta, UMA2BPMN, para realizar a transformação do modelo UMA (utilizado pelo EPF) para a notação BPMN. Esta transformação possibilitou o controle, execução e monitoramento do processo através de plataformas de BPM, onde foi utilizada a ferramenta Activiti.

Uma terceira contribuição foi a integração de atividades de modelos de gerenciamento de processos de negócio com o processo de desenvolvimento de software. Através do BPM a TI tem buscado apoiar as organizações no crescimento sustentável, controlar e reduzir custos operacionais, lançar novos serviços, inovar e aperfeiçoar os processos de negócio para alcançar novos patamares de serviços. A

aproximação dos modelos de BPM e processo de desenvolvimento de uma solução é uma importante contribuição. Através desta integração busca-se melhorar o trabalho entre os diferentes profissionais envolvidos com suas respectivas atividades e responsabilidade em processos que precisam aprimorar sua sinergia para realizar a entrega de uma solução de qualidade e valor para o negócio.

6.3 Limitações

A principal limitação do trabalho está relacionada às atividades de planejamento, as quais não foram integradas à definição da linha de processos, devido às diversas possibilidades de variabilidades nas atividades e nas transições; além das inúmeras abordagens existentes. As atividades de planejamento são fortemente customizadas pelas organizações e também recebem grandes influências relativas à cultura organizacional de gerenciamento, controle de investimentos e orçamento. De modo semelhante, os processos de Governança BPM, monitoração e controle também não foram abordados, uma vez que o foco era no desenvolvimento de software.

A pesquisa de avaliação por especialistas ficou sujeita ao trabalho voluntário realizado pelos especialistas selecionados. Esta pesquisa exigiu um grande esforço dos especialistas participantes para análise do conteúdo da linha, avaliação do modelo através das questões e preenchimento dos diversos itens no formulário. No entanto, uma execução da linha propriamente dita não foi possível.

6.4 Trabalhos futuros

Os modelos de BPM possuem um forte relacionamento não abordado com os processos de gestão de serviços de TI, como: ITIL (OGC, 2011), CMMI-SVC (SEI, 2010c), ISO-20000 (ABNT, 2011) ou MM-GSTI (MACHADO, 2011). Os processos de monitoração, controle de indicadores e gerenciamento da capacidade carecem de melhores processos e ferramentas para integração da TI e do negócio. Certamente a linha de processos proposta pode ser adaptada para contemplar os processos relativos às fases de Liberação, Execução e Monitoração. Estas fases poderiam abranger diversos processos para automatizar o controle do desempenho do processo (em tempo de execução) e facilitar as ações corretivas.

Conforme sugestões dos especialistas, a avaliação da linha também poderia originar pesquisas experimentais na indústria. A realização de experimentos

controlados ou não controlados na indústria confirmaria na prática a aplicabilidade da proposta. Entretanto, para realizar tal pesquisa seria essencial o financiamento de empresa(s) participante(s) para viabilizar os custos e profissionais necessários.

Quanto as possíveis melhorias para a ferramenta de exportação, as fases ou iterações representadas no EPF poderiam ser transformadas em subprocessos no BPMN. Esta melhoria aprimoraria o controle e visualização do processo. Os resultados da finalização de uma fase poderiam definir automaticamente a quantidade de vezes que a próxima fase seria executada.

Ao invés de trabalhar com uma ferramenta única para controle dos processos de software utilizando BPM, o apoio computacional poderia ser realizado pelas ferramentas específicas para cada processo. Nesta abordagem, a contribuição do trabalho futuro seria melhorar a integração das atividades entre estas ferramentas, onde se destacam os trabalhos do *Open Services for Lifecycle Collaboration* (OSLC). A iniciativa OSLC tem definido padrões de integração para integração dos processos e ferramentas envolvidos na engenharia de software, baseados em XML, ReST e *Resource Description Framework* (RDF).

6.5 Mensagem final

Filho meu, não te esqueças dos meus ensinamentos, e o teu coração guarde os meus mandamentos; porque eles aumentarão os teus dias e te acrescentarão anos de vida e paz. Não te desamparem a benignidade e a fidelidade; ata-as ao teu pescoço; escreve-as na tábua do teu coração e acharás graça e boa compreensão diante de Deus e dos homens. Confia no Senhor de todo o teu coração, e não te estribes no teu próprio entendimento. Reconhece-o em todos os teus caminhos, e ele endireitará as tuas veredas. Não sejas sábio aos teus próprios olhos; teme ao Senhor e aparta-te do mal;
- Provérbios 3:1-7

REFERÊNCIAS BIBLIOGRÁFICAS

- (ABES, 2011) ASSOCIAÇÃO BRASILEIRA DE EMPRESAS DE SOFTWARE. **Mercado Brasileiro de Software: panorama e tendências, 2011 - Brazilian Software Market: scenario and trends, 2011.** São Paulo: ABES, 2011, 28 p.
- (ABNT, 2004) ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 15504: Tecnologia da informação - Avaliação de processo.** Rio de Janeiro, 2004. 35 p.
- (ABNT, 2008) ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12207: Tecnologia de Informação – Processos de ciclo de vida de software.** Rio de Janeiro, 2009. 122 p.
- (ABNT, 2011) ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 20000-1: Tecnologia de informação – Gerenciamento de Serviços.** Rio de Janeiro, 2011. 30 p.
- (ALEIXO et al, 2011) ALEIXO, F.A., FREIRE, M.A., SANTOS, W.C., KULESZA, U. **Automating the Variability Management, Customization and Deployment of Software Processes: A Model-Driven Approach.** In: INTERNATIONAL CONFERENCE ENTERPRISE INFORMATION SYSTEMS (ICEIS), 13., 2011, Beijing, China. **Anais...** Berlin: Springer Heidelberg, 2011, p.372-387.
- (ALEIXO et al, 2012) ALEIXO, F.A.; FREIRE, M.A.; ALENCAR, D.; CAMPOS, E.; KULESZA, U. **A Comparative Study of Compositional and Annotational Modelling Approaches for Software Process Lines.** In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE (SBES), 14., 2012, Natal, Brasil. **Anais...** p.1-10.
- (ALEGRÍA, 2011) ALEGRÍA, J.A.H.; BASTARRICA, M.C.; QUISPE, A.; OCHOA, S.F. **An MDE approach to software process tailoring.** International Conference on Software and Systems Process (ICSSP '11). New York, USA: ACM, 2011, p. 43-52.
- (AMBLER et al., 2005) AMBLER, S.W.; NALBONE, J.; VIZDOS, M.J. **The Enterprise Unified Process: Extending the Rational Unified Process.** Prentice Hall, 2005. 408p.
- (ARMBRUST et al., 2008) ARMBRUST, O.; KATAHIRA, M.; MIYAMOTO, Y.; MÜNCH, J.; NAKAO, H.; OCAMPO, A. **Scoping software process models: initial concepts and experience from defining space standards.** International Conference Software Process: Improvement and Practice. Heidelberg: Springer-Verlag, 2008, p. 160-172.
- (ARMBRUST et al., 2009) ARMBRUST, O.; KATAHIRA, M.; MIYAMOTO, Y.; MÜNCH, J.; NAKAO, H.; OCAMPO, A. **Scoping Software Process Lines.** International Conference Software Process: Improvement and Practice. Nova Jersey: John Wiley & Sons, 2009, p. 181-197.

(ARSANJANI et al., 2008) ARSANJANI, A.; GHOSH, S.; ALLAM, A.; ABDOLLAH, T.; GANAPATHY, S.; HOLLEY, K. **SOMA: a method for developing service-oriented solutions**. IBM Systems Journal, v. 47, n. 3, p. 377-396, 2008.

(ASAP, 2011) ASAP Methodology for Implementation and Business add-ons to ASAP. Disponível em: <<http://www.sdn.sap.com/irj/bpx/asap>>. Acesso em 01 jun. 2011.

(BARRETO et al., 2010) BARRETO, A.; NUNES, E.; ROCHA, A.R.; MURTA, L. **Supporting the Definition of Software Processes at Consulting Organizations via Software Process Lines**. International Conference on the Quality of Information and Communications Technology, 7 ed., Porto, Portugal. Washington, DC: IEEE Computer Society, 2010, p. 15-24.

(BIANCO et al, 2011) BIANCO, P.; LEWIS, G. A.; MERSON, P.; SIMANTA, S. **Architecting Service-Oriented Systems (CMU/SEI-2011-TN-008)**. Pittsburgh: Software Engineering Institute, 2011. 46 p.

(BIEBERSTEIN et al., 2008) BIEBERSTEIN, N.; LAIRD, R.; JONES, K.; MITRA, T. **Executing SOA: A Practical Guide for the Service-Oriented Architect**. Indianapolis, USA: IBM Press, 2008. 240 p.

(BENDRAOU et al., 2007) BENDRAOU, R.; SADOVYKH, A.; GERVAIS, M.-P.; BLANC, X. **Software Process Modeling and Execution: The UML4SPM to WS-BPEL Approach**. In: CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS - EUROMICRO, 33., **Anais...** IEEE Computer Society, 2007, p. 314-321.

(BENDRAOU et al., 2010) BENDRAOU, R.; JÉZÉQUEL, J.-M.; GERVAIS, M.-P.; BLANC, X. **A Comparison of Six UML-Based Languages for Software Process Modeling**. IEEE Transactions on Software Engineering, v. 36, n. 5, p. 662-675, 2010.

(BOTTO, 2004) BOTTO, R. **Arquitetura Corporativa de Tecnologia da informação**. Rio de Janeiro: Brasport, 2004. 248p.

(BOECKLE et al., 2004) BOECKLE, G.; KNAUBER, P.; POHL, K.; SCHMID, K. **Software-Produktlinien – Methoden, Einführung und Praxis**. Heidelberg: Dpunkt, 2004. 320 p.

(BPM-CBOK, 2009) Antonucci, Y. L.; Bariff, M.; Benedict, T.; Champlin, B.; Downing, B. D.; Franzen, J.; Madison, D.J.; Lusk, S.; Spanyol, A.; Treat, M.; Zhao, J. L. ; Raschke, R. L. **Business Process Management Common Body Of Knowledge**. Chicago: CreateSpace, 2009. 234 p.

(BROWN et al., 2009) BROWN, W. A.; LAIRD, R. G.; GEE, C.; MITRA, T. **SOA Governance: Achieving and Sustaining Business and IT Agility**. Indianapolis, USA: IBM Press, 2009. 416p.

(CAFÉ, 2011) CAFÉ - From Concepts to Application in System-Family Engineering. Espanha. Apresentada descrição do projeto CAFÉ (ITEA Project ip00004) e seus resultados. Disponível em: <<http://www.esi.es/Cafe>>. Acesso em 10 jun. 2011.

(CARTER, 2007) CARTER, S. **The New Language of Business: SOA & Web 2.0**. New Jersey: Prentice Hall, 2007. 320p.

(CHANNABASAVAI AH; HOLLEY; TUGGLE, 2004) CHANNABASAVAI AH, K.; HOLLEY, K.; TUGGLE, E. **Migration to a Service-oriented architecture**. New York: IBM Software Group, 2004.

(CHOU, 2002) CHOU, S.-C. **A process modeling language consisting of high level UML diagrams and low level process language**. Journal of Object-Oriented Programming, v. 1, n. 4, p. 137-163, 2002.

(CIRILO; KULESZA; LUCENA, 2008) CIRILO, E.; KULESZA, U.; LUCENA, C. A. **Product Derivation Tool Based on Model-Driven Techniques and Annotations**. Journal of Universal Computer Science, v. 14, n. 18, p. 1344-1367, 2008.

(CLEMENTS; NORTHROP, 2002) CLEMENTS, P. C.; NORTHROP, L. **Software Product Lines: Practices and Patterns**. Reading, MA: Addison-Wesley Publishing Company, 2002. 563 p.

(COCKBURN, 2006) COCKBURN, A. **Agile Software Development: The Cooperative Game**. 2 ed. Boston, MA: Addison-Wesley Professional, 2006. 504 p.

(COOKE, 1991) COOKE, R. M. **Experts in Uncertainty: Opinion and Subjective Probability in Science**. New York: Oxford University Press, 1991. 321 p.

(DI NITTO et al., 2002) DI NITTO, E.; LAVAZZA, L.; SCHIAVONI, M.; TRACANELLA, E.; TROMBETTA, M. **Deriving executable process descriptions from UML**. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING – ICSE, 24., Orlando, Florida, **Anais...** New York: ACM, 2002, p. 155-165.

(ELLNER et al., 2010) ELLNER, R.; AL-HILANK, S.; DREXLER, J.; JUNG, M.; KIPS, D.; PHILIPPSEN, M. **eSPEM - A SPEM Extension for Enactable Behavior Modeling**. In: EUROPEAN CONFERENCE ON MODEL DRIVEN ARCHITECTURE - FOUNDATIONS AND APPLICATIONS (ECMFA'10), 6., 2010, Paris, France. **Anais...** Heidelberg: Springer-Verlag, 2010, p. 116-131.

(ENDRES; ROMBACH, 2003) ENDRES, A.; ROMBACH, D. **A Handbook of Software and Systems Engineering: Empirical Observations, Laws, and Theories**. Londres: Fraunhofer IESE e Pearson Addison-Wesley, 2003. 327 p.

(EPF, 2012) EPF - Eclipse Process Framework Project. Disponível em: <<http://epf.eclipse.org/>>. Acesso em 10 jun. 2012.

(ERL, 2004) ERL, T. **Service-Oriented Architecture: A Field Guide to Integration XML and Web Services**. New Jersey: Prentice Hall, 2004. 536 p.

(ERL, 2008) ERL, T. **SOA: Principles of Service Design**. Vancouver: Prentice Hall, 2008. 608 p.

(ERL, 2009) ERL, T. **SOA Design Patterns**. Vancouver: Prentice Hall, 2009. 800 p.

(ESTEVEES; JORGE, 2001) ESTEVEES, J.; JORGE, J. **Análise Comparativa de Metodologias de Implementação de SAP**. In: CONFERENCE OF ASSOCIAÇÃO PORTUGUESA DE SISTEMAS DE INFORMAÇÃO (APSI). **Anais...** Évora, Portugal: 2001, p.1-13.

(FEIN et al., 2011) FEIN, E.; RAZINKOV, N.; SHACHOR, S.; MAZZOLENI, P.; GOH, S.; GOODWIN, R.; BHAND, M.; CHEN, S.; LEE, J.; SINHA, V. S.; MANI, S.; MUKHERJEE, D.; SRIVASTAVA, B.; DHOOLIA, P. **Using MATCON to generate CASE tools that guide deployment of pre-packaged applications**. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE). **Anais...** New York: ACM, 2011, p. 1016-1018.

(FRANCH et al, 1997) FRANCH, X.; BOTELLA, P.; BURGUS, X.; RIBÓ, J.M. **ComProLab: A Component Programming Laboratory**. In: SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING (SEKE), 9. **Anais...** Madri, 1997, pp. 397-406.

(FUGGETTA, 2000) FUGGETTA, A. **Software Process: A Roadmap**. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), Limerick, Ireland. **Anais...** New York: ACM, 2000, p. 25-34

(FURLAN, 1998) FURLAN, J. D. **Modelagem de Objetos através da UML : The Unified Modeling Language**. São Paulo: Makron Books, 1998. 329 p.

(GARTNER, 2011) Gartner Inc., “Plateau of Productivity”, Disponível em: < <http://www.gartner.com/>>. Acesso em 02 out. 2011.

(GARTNER, 2012) Gartner Inc., “2012 CIO Agenda Survey”, Disponível em: < <http://www.gartner.com/>>. Acesso em 03 ago. 2012.

(GIL, 2002) GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2006. 175 p.

(GOMAA, 2004) GOMAA, Hassan. **Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures**. Boston: Pearson Education, 2004. 736p.

(GU; LAGO, 2011) GU, Q.; LAGO, P. **Guiding the selection of service-oriented software engineering methodologies**. Service Oriented Computing and Applications, p. 1-21, 2011.

(HASAN, 2004) HASAN, J. **Expert Service-Oriented Architecture in C#: Using the Web Services Enhancements 2.0**. New York: Apress, 2004. 305p.

(HAVEY, 2005) HAVEY, M. 2005. **Essential Business Process Modelling**. Sebastopol, CA: O’Reilly Media, Inc., 2005. 352 p.

(HOHPE; WOOLF, 2004) HOHPE, G.; WOOLF, B. **Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions**. Boston: Pearson Education, 2002. 480p.

(HUMPHREY; KELLNER, 1989) Humphrey, W., S.; Kellner, M. **Software Process Modeling: Principles of Entity Process Models**. SEI Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, (CMU/SEI-89-TR-2), 1989.

(IEEE, 2004). IEEE Computer Society. **Guide to the Software Engineering Body of Knowledge - SWEBOK**. Los Alamitos: CS Press, 2004. 204 p.

(IEEE, 1990) INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE 610.12: Standard Glossary of Software Engineering Terminology**. New York: IEEE Software & Systems Engineering Standards Committee, 1990, 83 p..

(IEEE, 1987) INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE 1002: Standard Taxonomy for Software Engineering Standards**. New York: IEEE Software & Systems Engineering Standards Committee, 1987, 30 p.

(ISO/IEC, 2008) INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC 12207 – Information technology – Software life cycle processes**. ISO/IEC, 2008, 123 p.

(ISO/IEC, 2011) INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTROTECHNICAL COMMISSION. **ISO/IEC TR 15504 1-9 – Information technology – Software Process Assessment**. ISO/IEC, 2011.

(JESTON; NELIS, 2008) JESTON, J.; NELIS, J. **Business Process Management - Practical Guidelines to Successful Implementations**. 2 ed. Oxford: Butterworth-Heinemann, 2008. 469 p.

(KALE, 2010) KALE, V. **Implementing SAP R/3: The Guide for Business and Technology Managers**. Indianapolis, USA: SAMS - A Division of Macmillan USA. 503 p.

(KANG, 1990) KANG, K.; COHEN, S.; HESS, J.; NOVAK, W.; PETERSON, S. (1990) **Feature-Oriented Domain Analysis (FODA) Feasibility Study** (Software Engineering Institute CMU/SEI-90-TR-21). Pittsburg: Carnegie Mellon University, 1990, 161 p.

(KO, 2009) KO, R. K. L. **A computer scientist's introductory guide to business process management (BPM)**. Crossroads 15, 4, Article 4. 2009 8 p.

(KROLL; KRUCHTEN, 2003) KROLL, P.; KRUCHTEN, P.. **Rational Unified Process Made Easy: A Practitioner's Guide to the RUP**. Harlow: Addison-Wesley Publishing Company, 2003. 464 p.

(KRUCHTEN, 2003) KRUCHTEN, P. **The Rational Unified Process: An Introduction**. Reading, MA: Addison-Wesley Longman, Inc, 2003. 336 p.

(LI, SMIDTS, 2003) Li, M.; SMIDTS, C. **A ranking of software engineering measures based on expert opinion**. IEEE Transactions on Software Engineering, v. 29, n. 9, p. 811-824, set 2003.

(LINDEN, 2002) LINDEN, F.V.D. **Engineering Software Arquitectures, Processes and Plataforms - ESAPS Overview**. In: SOFTWARE PRODUCT LINES

INTERNATIONAL CONFERENCE (SPLC), 2., San Diego, CA. **Anais...** Berlin: Springer-Verlang, 2002, p. 381-397 (Lecture Notes in Computer Science 2379).

(LINDEN; SCHMID; ROMMES, 2007) LINDEN, F.V.D.; SCHMID, K.; ROMMES, E. **Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering.** Berlin: Springer-Verlang, 2007, 333 p.

(MACHADO, 2011) Machado, R. F. **MM-GSTI: Proposta de um Modelo de Maturidade em Gerenciamento de Serviços de TI com foco nas pequenas e médias empresas.** 2011. 255 p. Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná, Curitiba, 2011.

(MAFRA; TRAVASSOS, 2006) MAFRA, S. N.; TRAVASSOS, G. H. **Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software.** Relatório Técnico - ES 687/06. Rio de Janeiro, 2006. 33 p.

(MAHMOOD; HILL, 2011) MAHMOOD, Z.; HILL, R. **Cloud Computing for Enterprise Architectures.** Springer-Verlag London Limited: 2011. 327 p.

(MARCONI; LAKATOS, 2010) MARCONI, M. A.; LAKATOS, E. M. **Fundamentos da Metodologia Científica.** 7. ed. São Paulo: Atlas, 2010. 320 p.

(MARTÍNEZ-RUIZ et al., 2011) MARTIINEZ-RUIZ, T.; GARCIA, F.; PIATTINI, M.; MÜNCH, J. **Modelling software process variability: an empirical study.** Institution of Engineering and Technology, IET Software, v. 5, n. 2, p. 172-187, abril 2011.

(MARTINS, 2008) MARTINS, G. A. **Estudo de Caso: Uma Estratégia de Pesquisa.** 2. ed. São Paulo: Atlas, 2006. 101 p.

(MCILROY,1968) MCILROY, M. D. **Mass Produced Software Components.** In: NATO SOFTWARE ENGINEERING CONFERENCE, 1968, Garmisch, Alemanha. **Anais...** Brussels: Scientific Affairs Division, NATO, 1969, p. 79-85.

(MEDEIROS; ALMEIDA; MEIRA, 2010) MEDEIROS, F. M.; ALMEIDA, E. S.; MEIRA S. R. de L. **Designing a Set of Service-Oriented Systems as a Software Product Line.** In: SOFTWARE COMPONENTS, ARCHITECTURES AND REUSE, BRAZILIAN SYMPOSIUM (SBCARS), 4., 2010, Bahia, Brasil. **Anais...**, 2010, p. 70-79.

(MINOLI, 2008) MINOLI, D. **Enterprise Architecture A to Z: Frameworks, Business Process Modeling, SOA, and Infrastructure Technology.** Londres: Taylor & Francis 2008. 504 p.

(MOSLEH; BIER; APOSTOLAKIS, 1987) MOSLEH, A.; BIER, V.M.; APOSTOLAKIS, G. **Methods for the Elicitation and Use of Expert Opinion in Risk Assessment,** Technical Report NUREG/CR-4962. Newport Beach: Pickard, Lowe and Garrick, Inc., 1987, 84 p.

(NAUR; RANDELL, 1969) NAUR, P., Randell, B. In: NATO CONFERENCE ON SOFTWARE ENGINEERING. Garmisch, Germany: NATO Science Committee, 1968. 136 p.

(NEIGHBORS, 1989) NEIGHBORS, J. M. **Draco: A method for engineering reusable software systems**. New York: Addison-Wesley, Software reusability, v. 1, concepts and models, 295–319, 1989.

(OGC, 2011) OGC - Office of Government Commerce. **Introduction to the ITIL Service Lifecycle**. The Stationary Office. Belfast, United Kingdom, 2011. 273 p.

(OLIVEIRA, 2005) OLIVEIRA, A. Ferramenta multiuso para TI. **Info Coporate**, São Paulo, p.59-67, abril. 2005.

(OMG, 2008a) OMG - Object Management Group. Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0. Disponível em: <<http://www.omg.org/spec/SPEM/2.0/>>. Acesso em 01 out. 2011.

(OMG, 2008b) OMG - Object Management Group. Business Process Maturity Model. Disponível em <<http://www.omg.org/spec/BPMM/1.0/>>. Acesso em 01 jul. 2012.

(OpenUP, 2011) OpenUP - Open Unified Process. Disponível em: <<http://epf.eclipse.org/wikis/openup/>>. Acesso em 30 out. 2011.

(OpenGroup, 2009) The Open Group - SOA Governance Framework. Technical Standard. Berkshire, United Kingdom: The Open Group, 2009, 96 p.

(OpenGroup, 2011) TOGAF 9.1 Enterprise - Framework for Enterprise Architect. Relatório Técnico. Berkshire, United Kingdom: The Open Group, 2011, 654 p.
(OSIMM, 2009) OSIMM – Open Group Service Integration Maturity Model. Service Integration Maturity Model Technical Standard: The SOA Source Book. Disponível em: <<http://www.opengroup.org/soa/source-book/osimm/model.htm>>. Acesso em 10 jan. 2011.

(PALMER; MCMENAMIN, 1991) PALMER, J.; MCMENAMIN, S. **Análise Essencial de Sistemas**. São Paulo: McGraw Hill, 1991. 567 p.

(PAPAZOGLU; HEUVEL, 2006) PAPAZOGLU, M. P.; HEUVEL, van den W. J. **Service-Oriented Design and Development Methodology**. International Journal of Web Engineering and Technology (IJWET), v. 2, n. 4, p. 412-446, 2006.

(PARNAS, 1976) PARNAS, D.L. **On the Design and Development of Program Families**. IEEE Transactions on Software Engineering, v.SE-2, n.1, p.1-9, março 1976.

(PARNAS, 1979) PARNAS, D.L. **Designing Software for Ease of Extension and Contractions**. IEEE Transactions on Software Engineering, v.SE-5, n.1, p.128-137, março 1979.

(PERROUIN, 2011) PERROUIN, G.; OSTER, S.; SEN, S.; KLEIN, J.; BAUDRY, B.; LE TRAON, Y. **Pairwise testing for software product lines: comparison of two approaches**. Software Quality Journal, Springer Netherlands, p. 1-39, agosto 2011.

(POHL; BÖCKLE; LINDEN, 2005) POHL, K.; BÖCKLE, G.; LINDEN, F.V.D.L. **Software Product Line Engineering: Foundations, Principles and Techniques**. Berlin, Heidenberg: Springer-Verlang, 2005. 467 p.

(PRIETO-DIAZ; 1991) Prieto-Diaz, R. Making Software Reuse Work: An Implementation Model. ACM SIGSOFT Software Engineering Notes, v. 16, n. 3, p. 61-68, julho 1991.

(SQI, 2011) SQI - Software Quality Institute. Software Process Improvement and Capability dEtermination. Disponível em < <http://www.sqi.gu.edu.au/spice/>>. Acesso em 06 nov. 2011.

(RATHFELDER; GROENDA, 2008) RATHFELDER C.; GROENDA, H. **iSOAMM: an independent SOA maturity model**. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED APPLICATIONS AND INTEROPERABLE SYSTEMS (IFIP WG 6.1), 8., 2008, Berlin, Alemanha. **Anais...** Heidelberg: Springer-Verlag; 2008, p. 1-15. Disponível em: <<http://portal.acm.org/citation.cfm?id=1789074.1789076>>.

(REINEHR, 2008) Reinehr, S. S. **Reuso Sistematizado de Software e Linhas de Produto de Software no Setor Financeiro: Estudos de Caso no Brasil**. 2008. 327 p. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2008.

(ROMBACH, 2005) ROMBACH, DIETER. **Integrated Software Process and Product Lines**. Unifying the Software Process Spectrum. In: INTERNATIONAL SOFTWARE, PROCESS WORKSHOP (ISPW), 2005, Beijing. **Anais...** Heidelberg: Springer-Verlag, 2005, p. 83-90.

(ROSEN et al., 2008) ROSEN, M.; LUBLINSKY, B., SMITH, K.T.; BALCER, M. J. **Applied SOA: Service-Oriented Architecture and Design Strategies**. Indianapolis: Wiley Publishing Inc., 2008. 696p.

(SCOTT, 2003) SCOTT, K. **Processo Unificado Explicado – UML**. São Paulo: Bookman, 2003. 160 p.

(SEI, 2010a) SOFTWARE ENGINEERING INSTITUTE. CMMI® for Development (CMU/SEI-2010-TR-033), Version 1.3. Pittsburg: Carnegie Mellon University, 2010, 482 p.

(SEI, 2010b) SOFTWARE ENGINEERING INSTITUTE. CMMI® for Acquisition (ESC-TR-2010-032), Version 1.3. Pittsburg: Carnegie Mellon University, 2010, 438 p.

(SEI, 2010c) SOFTWARE ENGINEERING INSTITUTE. CMMI® for Services (CMU/SEI-2010-TR-034), Version 1.3. Pittsburg: Carnegie Mellon University, 2010, 520 p.

(SHUJA; KREBS, 2008) SHUJA, A. K.; KREBS, J. **IBM Rational Unified Process Reference and Certification Guide**. Indianapolis, USA: IBM Press, 2008. 336 p.

(SILVA; MENEZES, 2005) SILVA, E. L. da; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. rev. atual. Florianópolis: UFSC, 2005. 138 p. Disponível em: <http://tccbiblio.paginas.ufsc.br/files/2010/09/024_Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes1.pdf>

(SILVIA; JURISTO, 2005) Silvia, A. T.; JURISTO, N. **Software Process Modeling**. Madri: Springer, v. 10, 2005. 208 p.

(SOAMM, 2005) SOAMM – Sonic Software, Systinet, AmberPoint, and BearingPoint. **A New Service-Oriented Architecture (SOA) Maturity Model**. OMG SOA SIG. Disponível em: <http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf>. Acesso em 10 jan. 2011.

(SÖDESTRÖM; MEIER, 2007) SÖDERSTRÖM, E.; MEIER, F. **Combined SOA Maturity Model : Towards a Guide for SOA Adoption**. Enterprise Interoperability II. Londres: Springer, 2007, p. 389-399.

(SOFTEX, 2011) SOCIEDADE PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO (SOFTEX). MR-MPS.BR – **Guia Geral:2011**. Disponível em: <<http://www.softex.br/mpsbr/guias/>>. Acessado em 06 out. 2011.

(SOMMERVILLE, 2007) SOMMERVILLE, I.. **Software engineering**. 8 ed. Harlow: Addison-Wesley Publishing Company, 2007. 840 p.

(SPANOUKAKIS, 2005) SPANOUKAKIS, G.; ZISMAN, A.; KOZLENKOV, A. **A service discovery framework for service centric systems**. In: SERVICES COMPUTING, IEEE INTERNATIONAL CONFERENCE. **Anais...** Orlando, USA: 2005, p. 251- 259.

(STANDISH GROUP, 2009) STANDISH GROUP INTERNATIONAL, INC. Estados Unidos. **2009 Research Report**. Disponível on-line em: <<http://www.standishgroup.com/>>. Acesso em 24 de ago. 2011.

(SUTTON; OSTERWEIL, 1996) SUTTON, S.M.; OSTERWEIL, L.J. **Product families and process families**. In: SOFTWARE PROCESS WORKSHOP. PROCESS SUPPORT OF SOFTWARE PRODUCT LINES. **Anais...** 1996, p. 109 – 111.

(SZYPERSKI, 2002) SZYPERSKI, C. **Component Software - Beyond Object-Oriented Programming**. 2. ed. Londres: Addison-Wesley Publishing Company, 2002, 589 p.

(THIES; VOSSEN, 2009) THIES, G.; VOSSEN, G.. **Modelling web-oriented architectures**. Sixth Asia-Pacific Conference on Conceptual Modeling (APCCM '09). Darlinghurst, Australia: Australian Computer Society, Inc., 2009, p. 97-106.

(THIOLLENT, 2005) THIOLLENT, M. Metodologia da Pesquisa-Ação. 14. ed. Editora Cortez. São Paulo, 2005, 132 p.

(TURNER, 2006) TURNER, M. S. V. **Microsoft Solutions Framework Essentials: Building Successful Technology Solutions**. Redmond, WA: Microsoft Press, 2006. 340 p.

(VAN DER AALST et al., 2003) VAN DER AALST, W.M.P.; TER HOFSTEDÉ, A.H.M.; WESKE, M. 2003. **Business process management: a survey**. In: INTERNATIONAL CONFERENCE ON BUSINESS PROCESS MANAGEMENT (BPM'03). **Anais...** Berlin, Heidelberg: Springer-Verlag , 2003, p. 1-12.

(W3C, 2011) W3C – World Wide WEB Consortium. 1997-2005. Disponível em: <<http://www.w3.org>>. Acesso em 20 de mai. 2011.

(WEBBER et al., 2010) WEBBER, J.; PARASTATIDIS, S.; ROBINSON, I.; FOWLER, M. **REST in Practice Hypermedia and Systems**. Sebastopol, CA: O'Reilly Media, Inc., 2010. 352 p.

(WEISS; LAI, 1999) WEISS, D.; LAI, C.T.T. **Software Product-Line Engineering: a Family-Based Software Development Process**. Reading, MA: Addison-Wesley Publishing Company, 1999. 426 p.

(YOURDON, 1992) YOURDON, E. **Análise Estruturada Moderna**. Rio de Janeiro: Campus, 1992. 835 p.

(ZACHOS et al., 2006) ZACHOS, K.; MAIDEN, N.; ZHU, X.; JONES, S. **Does Service Discovery Enhance Requirements Specification? A Preliminary Empirical Investigation**. In: IEEE INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE, SERVICE-ORIENTED COMPUTING: Consequences for Engineering Requirements (SOCCER). **Anais...** Minneapolis/St. Paul, Minnesota, USA: 2006, p. 4-11.

(ZACHOS et al., 2007) ZACHOS, K.; MAIDEN, N.; ZHU, X.; JONES, S. **Discovering web services to specify more complete system requirements**. *Advanced Information Systems Engineering*, v. 4495, Berlin: Springer-Verlag, p. 142-147, 2007.

(ZHANG et al., 2010) Zhang, L.; Ge, M.; Bi, X.; Chen, S. **A SOA-BPM-Based Architecture for Intelligent Power Dispatching System**. *Power and Energy Engineering Conference (APPEEC)*. Asia-Pacific, 2010, p.1-4.

GLOSSÁRIO

BAM	Esta ferramenta fica no nível mais alto da pilha e proporciona os mais altos níveis de integração de serviços de negócio. Responsável por monitorar a saúde (desempenho) do negócio. Eles fornecem a capacidade de monitoração dos processos de negócios e transações, e automatizar gatilhos com ações corretivas (MINOLI, 2008).
BPEL	<i>Business Process Execution Language</i> é a linguagem para execução que orquestra os processos de negócio usando os serviços, e permite que várias aplicações BPM sejam integradas (JESTON; NELIS, 2008).
BPM	Abordagem disciplinada para identificar, desenhar, executar, documentar, medir, monitorar, controlar e melhorar processos de negócio automatizados ou não a fim de atingir resultados consistentes com metas estratégicas da organização (BPM-CBOK, 2009).
BPMN	O grupo <i>Business Process Management Initiative</i> (BPMI) desenvolveu a notação padrão para modelagem chamada <i>Business Process Modeling Notation</i> (BPMN). Esta especificação estabelece uma notação gráfica para expressar os processos de negócio em uma linguagem XML que pode ser visualizada em diagramas (MINOLI, 2008).
BRM	<i>Business Rules Management</i> controla de forma separa as regras de negócio ou decisões, de modo isolado do resto da lógica da aplicação. Estas regras são executadas, modificadas, gerenciadas em um ambiente especializado que facilita a inclusão de novos requisitos de negócio (ROSEN et al., 2008).
Composições	<p><i>Composite Service</i> permite que um novo serviço seja construído por composição de serviços ou grupo de serviços primitivos. Esta definição permite aprimorar o reuso dos serviços mais básicos e a construção de serviços mais complexos usando somente os serviços necessários (MINOLI, 2008).</p> <p>É possível construir processos de negócio por meio da iteração de um conjunto de serviços, sem a necessidade de codificação, mas utilizando ferramentas, como a linguagem BPEL. A composição permite a entrega rápida das soluções sem uma significativa construção de código. A abordagem de <i>Composite Application</i> também permite a construção de aplicações (user interfaces) através de ferramentas declarativas (OSIMM, 2009).</p>
Computação em nuvem	Define um modelo computacional baseado em utilitários (commodities ou serviços) comuns, independente de localização, on-line que estão disponíveis sob demanda. A computação em nuvem é dividida em três camadas (IAAS, PAAS, SAAS) (MAHMOOD; HILL, 2011).
ESB	O núcleo de SOA é tipicamente construído sobre um <i>middleware</i> chamado de <i>Enterprise Service Bus</i> . Uma das principais funções

	do ESB é oferecida todas as capacidades de interconectividade necessárias para alavancar os serviços em toda a arquitetura, incluindo os serviços de transporte, controle de eventos, serviços de mediação e roteamento (MINOLI, 2008)
IaaS	A primeira camada ou fundação da computação em nuvem é a <i>Infrastructure as a Service</i> responsável por isolar a necessidade de infraestrutura, a qual é composta pelo hardware e serviços relacionados, exemplo: datacenter (dispositivos elétricos e mecânicos), redes, firewalls, servidores físicos e camada de virtualização (MAHMOOD; HILL, 2011).
Método	Descreve as características de um processo ordenado ou procedimento usados na engenharia de um produto ou na realização de um serviço (IEEE, 1987). Conteúdo do Método especifica explicações passo-a-passo textuais, descrevendo como objetivos específicos de granularidade fina de desenvolvimento são alcançados, através dos papéis com os recursos e os resultados, independente da colocação destes (no Processo) passos dentro do ciclo de vida de desenvolvimento específico (OMG, 2008a). Método de engenharia de software: está dividido em três tópicos: métodos heurísticos que lidam com abordagens informais, métodos formais que lidam com abordagens matemáticas; e métodos de prototipagem que lida com abordagens da engenharia de software com baseadas em diversas formas de prototipagem (IEEE, 2004). No contexto geral é uma abordagem definida e repetível para abordar um tipo particular de problema (OpenGroup, 2011).
Metodologia	Uma série definida e repetível de passos para abordar um tipo particular de problema, que aborda tipicamente um processo definido, mas pode também incluir a definição do conteúdo (OpenGroup, 2011).
Orientação a Serviços	Um modo de pensar (paradigma) em termos de serviços, realizando o desenvolvimento baseada em serviços e produzindo novos serviços (OSIMM, 2009).
PaaS	<i>Platform as a Service</i> é camada intermediária da computação em nuvem, na qual residem os sistemas operacionais e a infraestrutura do software, exemplo: servidores de aplicação, banco de dados, etc (MAHMOOD; HILL, 2011).
Ponto de Variabilidade	Ponto no qual uma variação ocorre em um ativo de domínio, isto é, neste ponto de fato uma seleção necessita ser feita para chegar ao ativo instanciado (LINDEN; SCHMID; ROMMES, 2007).
Processo	Sequenciamento de fases e marcos expressando um ciclo de vida do produto em desenvolvimento. Define como, a partir de um marco, alcançar o próximo marco, através da definição de sequências de trabalho, operações, ou eventos que demandam tempo, perícia (papéis), ou outro recurso, e que produzem determinado resultado (OMG, 2008a). Uma sequência de passos realizados para um determinado fim,

por exemplo, o processo de desenvolvimento do software (IEEE, 1990).

ReST	Conjunto de tecnologia para a construção de serviços que se popularizaram devido sua simplicidade. O REST ou Transferência de Estado Representacional é um estilo arquitetural definido por princípios, os quais foram elaborados na tese do pós-doutorado de Roy Fielding's (WEBBER et al., 2010).
Serviço	Um serviço é uma tarefa de negócio com uma descrição do serviço publicada que geralmente representa um contrato entre um fornecedor e um consumidor (OSIMM, 2009).
SAAS	Software as a Service é a camada superior da computação em nuvem que estabelece um modelo de entrega de software sob demanda na web, isolando toda a complexidade. Este modelo é uma alternativa ao modelo tradicional de licenciamentos (MAHMOOD; HILL, 2011).
SOA	Service-Oriented Architecture (SOA) é um estilo arquitetural que suporta a orientação a serviços (OSIMM, 2009). Vide conceito de Serviço.
SOAP	Protocolo leve para troca de informações em ambientes distribuídos e descentralizados (MINOLI, 2008).
Variabilidade	<p>É uma premissa sobre como os membros de uma família podem diferir um do outro (WEISS; LAI, 1999).</p> <p>Variabilidade é qualquer aspecto onde as características de uma linha de produto (respectivamente em seus ativos) diferem para diferentes produtos (LINDEN; SCHMID; ROMMES, 2007).</p>
Web Service	Conjunto de tecnologias definida por padrões de indústria que são suportados por diversos fornecedores. A primeira geração ficou conhecida por seguir somente tecnologias e especificações abertas, como: WSDL - Web Services Description Language, XSD - XML Schema Definition Language, SOAP - Simple Object Access Protocol, UDDI - Universal Description, Discovery, and Integration, e o WS-I Basic Profile (ERL, 2008).
WOA	Especialização da SOA com ênfase no simples uso das tecnologias e padrões da estabelecidos pela Web 2.0 (THIES; VOSSEN, 2009), Prima pela utilização de ReST, ao invés de Web Services.

APÊNDICE A – PROTOCOLO DE PESQUISA – CARTA DE APRESENTAÇÃO

PESQUISA ACADÊMICA

Esta pesquisa tem como objetivo a avaliação de uma linha de processos, e faz parte da dissertação de mestrado de Cleiton dos Santos Garcia, que está sendo desenvolvida sob orientação da Doutora Sheila Reinehr, no Programa de Pós-Graduação em Informática (PPGI) da Escola Politécnica da Pontifícia Universidade Católica do Paraná (PUCPR). O título deste trabalho é: "PROPOSTA DE UMA LINHA DE PROCESSO DE SOFTWARE PARA DESENVOLVIMENTO DE APLICAÇÕES USANDO SOA E BPM".

Desde já agradeço pela sua atenção e colaboração.

Cleiton dos Santos Garcia
cleitonsg@gmail.com
Jaraguá do Sul, 27 de maio de 2012.

ESCLARECIMENTOS:

a) O tempo total previsto para responder esta pesquisa é de 30 minutos. A pesquisa está organizada da seguinte forma:

- Parte 1 - Caracterização do Respondente
- Parte 2 – Visão Geral das fases da linha
- Parte 3 – Análise da fase de Modelagem de negócios
- Parte 4 – Análise da fase de Identificação
- Parte 5 – Análise da fase de Especificação
- Parte 6 – Análise da fase de Construção
- Parte 7 – Sugestões de melhoria

b) O Apêndice A apresenta os detalhes dos processos propostos no modelo.

Fases do SOPLA

- Fase de Modelagem de Negócios e Transformação
- Fase de Identificação e Definição da Arquitetura
- Fase de Especificação e Projeto dos Serviços
- Fase de Construção, Composição e Testes

c) O Apêndice B apresenta o Acordo de Confidencialidade.

APÊNDICE B – PROTOCOLO DE PESQUISA – TERMO DE CONFIDENCIALIDADE

TERMO DE CONFIDENCIALIDADE

Este Termo de Confidencialidade visa estabelecer um acordo entre os pesquisadores Cleiton dos Santos Garcia e Sheila Reinehr, doravante denominados Pesquisadores e o especialista em SOA e processo de software doravante denominado Participante, a respeito da confidencialidade das informações coletadas durante o processo de pesquisa da dissertação de mestrado do primeiro, sob orientação do segundo, intitulado: **“PROPOSTA DE UMA LINHA DE PROCESSO DE SOFTWARE PARA DESENVOLVIMENTO DE APLICAÇÕES USANDO SOA E BPM”**.

Por meio deste Termo de Confidencialidade, os Pesquisadores se comprometem a:

- portar-se com discrição em todos os momentos da pesquisa acadêmica, não comentando ou divulgando qualquer tipo de informação que tenha sido repassada de forma oral ou escrita;
- não divulgar o nome do Participante, em qualquer meio, a menos que expressamente autorizado por este;
- não divulgar, em qualquer meio, os dados e informações individualizadas coletados durante o processo de pesquisa com o Participante;
- divulgar, em formato de dissertação, artigos e apresentações, apenas os dados agregados, dos quais não se possa retirar ou inferir a identificação do Participante;
- retornar para o Participante as informações coletadas e analisadas, em formato individualizado dos seus próprios dados e em formato agregado com os dados de todos os demais Participantes.

As assinaturas abaixo expressam a concordância quanto ao cumprimento deste Termo de Confidencialidade, por prazo indeterminado.

Jaraguá do Sul, 27 de maio de 2012.

Cleiton dos Santos Garcia

Sheila Reinehr

APÊNDICE C – FORMULÁRIO DE CARACTERIZAÇÃO DO ESPECIALISTA

Formulário da Parte 1 - Caracterização do Respondente

*Obrigatório

1.1) Nome completo *

1.2) e-mail *

1.3) Idade *

1.4) Nome da empresa(s) ou instituição(ões) *

1.5) Que papel mais se aproxima das suas atribuições profissionais: *

- Gerente de Projetos
- Arquiteto de Software
- Desenvolvedor
- Outro:

Processo de software

1.6) Qual o seu nível de conhecimento relacionado a cada um dos assuntos abaixo? *

	Desconheço o assunto	Superficial	Razoável	Bom	Excelente
a) Modelos de Processos de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) RUP – Rational Unified Process	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Processos de software para o reuso sistemático	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) Utilização de Linha de Produto de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) Elaboração de Linha de Produto de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f) Utilização de Linha de Processos de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
g) Elaboração de Linha de Processos de Software	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.7) Qual o seu tempo total de experiência relacionado a Processos de Software (___ anos e ___ meses)? *

SOA

1.8) Qual o seu nível de conhecimento relacionado a cada um dos assuntos abaixo? *

	Desconheço o assunto	Superficial	Razoável	Bom	Excelente
a) Metodologias de Governança para SOA	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) Desenvolvimento (projeto, construção e testes) de serviços (Web Services , SOAP ou Rest)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Desenvolvimento de aplicações orientadas a serviços (SOAD - Service-oriented development of applications)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) Implantação de barramentos de serviços (ESB – Enterprise Service Bus ou EAI – Enterprise Application Integration)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) Desenvolvimento baseado em aplicações compostas (Composite Applications e Mashups)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.9) Qual o tempo total de experiência relacionado a SOA (___ anos e ___ meses)? *

BPM

1.10) Qual o seu nível de conhecimento relacionado a cada um dos assuntos abaixo? *

	Desconheço o assunto	Superficial	Razoável	Bom	Excelente
a) Metodologias de governança para os processos de negócios	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) Implantação de melhorias nos processos de negócio	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Desenvolvimento de aplicações baseadas no controle da execução dos processos de negócio (BPMS - Business Process Management Systems ou WfMS – Workflow Management Systems)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) Controle e monitoração de processo de negócio	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) Controle e monitoração do processo de negócio automatizado (como BAM - Business activity monitoring)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1.11) Qual o seu tempo total de experiência com BPM (___ anos e ___ meses)? *

Geral

1.12) Já desenvolveu algum projeto ou publicação na indústria relacionado com as áreas de Linha de processo de Software (LPS), SOA ou BPM? Se sim, cite este(s) trabalho(s).

1.13) Já desenvolveu algum trabalho acadêmico relacionado com as áreas de Linha de processo de Software (LPS), SOA ou BPM: *

- Sim, desenvolvi um TCC de graduação
- Sim, desenvolvi uma monografia de especialização
- Sim, desenvolvi uma dissertação de mestrado
- Sim, desenvolvi uma tese de doutorado
- Não desenvolvi trabalho relacionado aos assuntos

1.14) Já orientou algum trabalho acadêmico relacionado com as áreas de Linha de processo de Software (LPS), SOA ou BPM: *

- Sim, orientei um TCC de graduação
- Sim, orientei uma monografia de especialização
- Sim, orientei uma dissertação de mestrado
- Sim, orientei tese de doutorado
- Não orientei trabalho relacionado aos assuntos

1.15) Título do(s) trabalho(s), instituição, ano de conclusão:

1.16) Outras informações que julgar necessárias que ajudem a caracterizar sua experiência como especialista do assunto

APÊNDICE D – FORMULÁRIO DE AVALIAÇÃO DAS FASES DA LINHA

Formulário da Parte 2 – Visão geral das fases da linha

2.1) Quanto às fases da linha de processo, você as considera completas e suficientes?*

1 2 3 4 5

Disordo totalmente Conordo totalmente

2.1) Justificativa:

2.2) Você sugere a adição ou junção de algumas destas fases? *

1 2 3 4 5

Nenhuma adição Diversas adições

2.2) Justificativa:

2.3) Você sugere mudanças nas transições entre as fases?*

1 2 3 4 5

Nenhuma mudança Diversas mudanças

2.3) Justificativa:

2.4) Você sugere alguma nova variabilidade de fase dentro do escopo que caracteriza esta linha de processo?*

1 2 3 4 5

Nenhuma variabilidade Diversas variabilidades

2.4) Justificativa:

Formulário da Parte 3 – Análise da fase de Modelagem de Negócios e Transformação**3.1) Quanto às atividades da linha de processo, você as considera completas e suficientes? ***

1 2 3 4 5

Nenhuma adição Concordo totalmente**3.1) Justificativa:****3.2) Você sugere a adição ou junção de algumas destas atividades? ***

1 2 3 4 5

Nenhuma adição Diversas adições**3.2) Justificativa:****3.3) Você sugere mudanças nas transições entre as atividades? ***

1 2 3 4 5

Nenhuma mudança Diversas mudanças**3.3) Justificativa:****3.4) Você sugere alguma nova variabilidade de atividades dentro do escopo que caracteriza esta linha de processo? ***

1 2 3 4 5

Nenhuma variabilidade Diversas variabilidades**3.4) Justificativa:**

Formulário da Parte 4 – Análise da fase de Identificação e Definição da Arquitetura**4.1) Quanto às atividades da linha de processo, você as considera completas e suficientes? ***

1 2 3 4 5

Discordo totalmente Concordo totalmente**4.1) Justificativa:****4.2) Você sugere a adição ou junção de algumas destas atividades? ***

1 2 3 4 5

Nenhuma adição Diversas adições**4.2) Justificativa:****4.3) Você sugere mudanças nas transições entre as atividades? ***

1 2 3 4 5

Nenhuma mudança Diversas mudanças**4.3) Justificativa:****4.4) Você sugere alguma nova variabilidade de atividades dentro do escopo que caracteriza esta linha de processo? ***

1 2 3 4 5

Nenhuma variabilidade Diversas variabilidades**4.4) Justificativa:**

Formulário da Parte 5 – Análise da fase de Especificação e Projeto dos Serviços**5.1) Quanto às atividades da linha de processo, você as considera completas e suficientes?***

1 2 3 4 5

Discordo totalmente Concordo totalmente**5.1) Justificativa:****5.2) Você sugere a adição ou junção de algumas destas atividades?***

1 2 3 4 5

Nenhuma adição Diversas adições**5.2) Justificativa:****5.3) Você sugere mudanças nas transições entre as atividades?***

1 2 3 4 5

Nenhuma mudança Diversa mudanças**5.3) Justificativa:****5.4) Você sugere alguma nova variabilidade de atividades dentro do escopo que caracteriza esta linha de processo?***

1 2 3 4 5

Nenhuma variabilidade Diversas variabilidades**5.4) Justificativa:**

Formulário da Parte 6 – Análise da fase de Construção, Composição e Testes**6.1) Quanto às atividades da linha de processo, você as considera completas e suficientes?***

1 2 3 4 5

Disordo totalmente Concordo totalmente**6.1) Justificativa:****6.2) Você sugere a adição ou junção de algumas destas atividades?***

1 2 3 4 5

Nenhuma adição Diversas adições**6.2) Justificativa:****6.3) Você sugere mudanças nas transições entre as atividades?***

1 2 3 4 5

Nenhuma mudança Diversa mudanças**6.3) Justificativa:****6.4) Você sugere alguma nova variabilidade de atividades dentro do escopo que caracteriza esta linha de processo?***

1 2 3 4 5

Nenhuma variabilidade Diversas variabilidades**6.4) Justificativa:**