



**METODOLOGIA PARA DESENVOLVIMENTO DE SOLUÇÕES
BASEADAS EM ARQUITETURA ORIENTADA A SERVIÇOS PARA
FORÇA DE TRABALHO TERCEIRIZADA**

Armando Amorim Ferreira Vidigal Junior

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**METODOLOGIA PARA DESENVOLVIMENTO DE
SOLUÇÕES BASEADAS EM ARQUITETURA ORIENTADA
A SERVIÇOS PARA FORÇA DE TRABALHO
TERCEIRIZADA**

ARMANDO AMORIM FERREIRA VIDIGAL JUNIOR

ORIENTADOR: RICARDO STARCIARINI PUTTINI

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: 533/2013 DM

BRASÍLIA / DF, 05 de Agosto de 2013.

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**METODOLOGIA PARA DESENVOLVIMENTO DE SOLUÇÕES BASEADAS EM
ARQUITETURA ORIENTADA A SERVIÇOS PARA FORÇA DE TRABALHO
TERCEIRIZADA**

ARMANDO AMORIM FERREIRA VIDIGAL JUNIOR

**DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE
BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU
DE MESTRE.**

APROVADA POR:

**RICARDO STARCIARINI PUTTINI, Dr., ENE/UNB
(ORIENTADOR)**

**ANDERSON CLAYTON ALVES NASCIMENTO, Dr., ENE/UNB
(EXAMINADOR INTERNO)**

**RICARDO MATOS CHAIM, Dr., FGA/UNB
(EXAMINADOR EXTERNO)**

BRASÍLIA, 05 DE AGOSTO DE 2013.

FICHA CATALOGRÁFICA

VIDIGAL JR, ARMANDO A. F.

Metodologia para Desenvolvimento de Soluções Baseadas em Arquitetura Orientada a Serviços para Força de Trabalho Terceirizada. [Distrito Federal] 2013.

533/2013 DM, 172p, 210x297 mm (ENE/FT/UnB, MESTRE, Engenharia Elétrica, 2013).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Arquitetura Orientada a Serviços

2. Metodologia SOA

3. Engenharia de Software

4. Fábrica de Software

I. ENE/FT/UnB.

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

VIDIGAL JR, Armando A. F. (2013). Metodologia para Desenvolvimento de Soluções Baseadas em Arquitetura Orientada a Serviços para Força de Trabalho Terceirizada. Dissertação de Mestrado em Engenharia Elétrica, Publicação 533/2013 DM PPGEE, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF.

CESSÃO DE DIREITOS

AUTOR: ARMANDO AMORIM FERREIRA VIDIGAL JUNIOR

TÍTULO: METODOLOGIA PARA DESENVOLVIMENTO DE SOLUÇÕES BASEADAS EM ARQUITETURA ORIENTADA A SERVIÇOS PARA FORÇA DE TRABALHO TERCEIRIZADA.

GRAU: Mestre

ANO: 2013

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

ARMANDO AMORIM FERREIRA VIDIGAL JUNIOR

AGRADECIMENTOS

Ao orientador e amigo professor Dr. Ricardo Starciarini Puttini que, pacientemente, dedicou várias de suas horas em discussões, me trazendo por muitas vezes à realidade prática de nossos objetivos e direcionando meus esforços sempre para o que de fato era importante para nossa pesquisa. Apesar de sua complexa e atribulada agenda, sempre esteve presente em meus momentos de maior angústia, norteando de forma simples e direta os caminhos que devia seguir. Com sua visão, tornava grandes problemas por mim criados, em simples obstáculos que deviam ser analisados e contornados através de pesquisas acadêmicas e experiência prática de mercado. A você professor, os mais sinceros agradecimentos e reconhecimento que o sucesso deste trabalho é tanto seu quanto meu.

Aos professores e colegas da UNB com os quais tive o prazer de conviver, quero dizer que cada um isoladamente e todos como um grupo, contribuíram de forma efetiva para o refinamento e aprimoramento desta. Em especial gostaria de direcionar minha gratidão aos professores Dr. Rafael Timóteo e Dr. Mamede Lima-Marques que, embora não tenham trabalhado diretamente com a metodologia proposta, em muito contribuíram para o seu desenvolvimento, seja através das disciplinas ministradas ou das conversas no corredor. Meu muito obrigado.

Ao Sr. Janduy Coutinho, Presidente da Unimix Tecnologia, onde trabalhei por nove maravilhosos anos, que me proporcionou a oportunidade de me aprofundar em SOA em cursos e congressos nos EUA, na Alemanha, no Reino Unido e também no Brasil, onde tive a oportunidade de ouvir e interagir com renomados especialistas da tecnologia, dentre os quais destaco o Sr. Thomas Erl, além da oportunidade também de aplicar os conhecimentos adquiridos em seus clientes.

Aos colegas da Agência Nacional de Vigilância Sanitária, da Agência Nacional de Águas e do Ministério da Saúde, onde pude discutir, validar, aprimorar e testar, em parte ou em todo, os conceitos e processos propostos na metodologia.

Agradeço ainda aos colegas e amigos do IBTI – Instituto Brasília de Tecnologia e Inovação, onde atuei como consultor e que foi terreno fértil para a evolução e consolidação de meus estudos e pesquisas. Sem querer me esquecer dos demais, gostaria de enaltecer a ajuda e colaboração decisiva do Sr. Claynor Mazzarolo, Diretor Presidente do IBTI, que me proporcionou toda a infraestrutura necessária e à equipe de mestres que me auxiliou de forma efetiva neste trabalho, composta pelo Sr. André Toffanello, Sra. Gabriela Alves e o Sr. Valério Martins, obrigado a todos.

DEDICATÓRIA

Para:

Meire,

Minha querida esposa e meus filhos Gustavo, Tiago e Vanderlene.

Para minha família querida, que nas horas de folga e prazer, me proporcionou momentos de alegria e felicidade, tão necessários para amenizar as horas de estudo e pesquisa. Pela força e motivação que, mesmo em pequenas frases ou gestos, me levaram a insistir nessa minha caminhada.

Todo o meu amor e reconhecimento a vocês!

Ao meu pai e minha mãe, que mesmo já tendo partido de nosso convívio, tenho certeza de estarem orgulhosos de mais esse desafio cumprido de seu filho.

RESUMO

METODOLOGIA PARA DESENVOLVIMENTO DE SOLUÇÕES BASEADAS EM ARQUITETURA ORIENTADA A SERVIÇOS PARA FORÇA DE TRABALHO TERCEIRIZADA.

Autor: Armando Amorim Ferreira Vidigal Junior

Orientador: Ricardo Starciarini Puttini

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Agosto de 2013.

Para que os sistemas baseados em SOA obtenham os benefícios esperados, os princípios conhecidos da engenharia de software tradicional precisam ser adaptados para desenvolvimento orientado a serviços. São necessárias abordagens sistemáticas para a concepção, desenvolvimento e manutenção de sistemas orientados a serviços. Essa abordagem é conhecida como Engenharia de Software Orientada a Serviços - SOSE.

A impossibilidade de utilização de metodologias de desenvolvimento SOA existentes, quer seja pela ausência de informações detalhadas ou, principalmente, pela desconsideração de forças de trabalho terceirizadas no processo, foi fator determinante para a elaboração deste trabalho.

Para atingir o objetivo de criar uma metodologia de desenvolvimento de soluções baseadas em SOA, foram realizadas pesquisas bibliográficas e estudos de caso, tanto para sua elaboração como para sua validação em situação real de execução.

A base da metodologia proposta é constituída de três pilares: a definição de unidades funcionais, onde são agrupadas tarefas e atividades técnicas relacionadas, definindo escopos claros de terceirização e responsabilidades; perfis profissionais, que explicitam as competências necessárias requeridas em cada etapa e; os processos e artefatos, que definem as atividades que devem ser realizadas e seus entregáveis.

Palavras-chave: Computação Orientada a Serviços, SOA, Metodologia de desenvolvimento SOA, Engenharia de *software*, SOSE, fábrica de *software*.

ABSTRACT

A SERVICE-ORIENTED ARCHITECTURE SOLUTION DEVELOPMENT METHODOLOGY FOR OUTSOURCED WORKFORCE.

Author: Armando Amorim Ferreira Vidigal Junior

Supervisor: Ricardo Starciarini Puttini

Pos-Graduation Program of Electric Engineering

Brasília, august 2013

For systems based on SOA obtain the expected benefits, the known principles of traditional software engineering need to be adapted to service-oriented development. Systematic approaches are needed for the design, development and maintenance of service-oriented systems. This approach is known as Software Engineering Service Oriented - SOSE.

The impossibility of using existing SOA development methodologies, either by lack of detailed information or mainly by disregard of outsourced workforces in the process, was the determining factor for the development of this work.

To achieve the objective of creating a development methodology for SOA-based solutions, literature searches and case studies were conducted, for both its development and validation in the real implementation.

The basis of the proposed methodology consists of three pillars: the definition of functional units, where tasks and technical related activities are grouped, defining outsourcing and responsibilities clear scopes; professional profiles, that explain the skills required at each stage and; processes and artifacts, that define the activities that must be performed and their deliverables.

Keywords: Service Oriented Computing, SOA, SOA development methodology, Software Engineering, SOSE, software factory.

SUMÁRIO

1 - INTRODUÇÃO	18
1.1 Motivação	19
1.2 Objetivo	22
1.2.1 Objetivo Geral	22
1.2.2 Objetivos Específicos.....	23
1.3 caracterização da pesquisa	23
1.4 Organização do Trabalho	24
2 - ANÁLISE DAS METODOLOGIAS EXISTENTES	26
2.1 IBM Analysis Service-Oriented and Design (SOAD).....	27
2.2 IBM Service Oriented Modeling and Architecture (SOMA)	30
2.3 Oracle Unified Method (OUM)	32
2.4 Outras Metodologias	35
2.4.1 Processo CBDI-SAE	35
2.4.2 Service Oriented Architecture Framework (SOAF)	35
2.4.3 Service Oriented Unified Process (SOUP).....	36
2.4.4 Metodologia proposta por Papazoglou	36
2.4.5 Metodologia proposta por Thomas Erl	36
2.5 Resumo Comparativo entre as Metodologias	37
3 - ESTRUTURA DA METODOLOGIA	38
3.1 Unidades Funcionais	39
3.2 Ciclo de Vida de Desenvolvimento de Software.....	41
3.3 Perfis Profissionais	44
3.4 Visão Geral da Metodologia	46
4 - ESTRUTURA FUNCIONAL	49
4.1 Unidades Corporativas	51
4.1.1 Escritório de Gestão de Processos de Negócio	51
4.1.2 Escritório de Administração de Dados	53
4.1.3 Escritório de Infraestrutura de TI.....	54
4.1.4 Escritório de Arquitetura Corporativa	56
4.1.5 Escritório de Gestão de Projetos	59
4.2 Unidades Dedicadas.....	60
4.2.1 Centro de Excelência SOA.....	60

4.2.2	Fábrica de Serviços.....	61
4.2.3	Fábrica de Software	63
4.2.4	Fábrica de Testes	64
5 -	PERFIS PROFISSIONAIS.....	65
5.1	Descrição dos Perfis Profissionais.....	65
5.1.1	Administrador de Dados.....	65
5.1.2	Administrador de Infraestrutura SOA	66
5.1.3	Analista de Negócio	66
5.1.4	Analista de Processo	67
5.1.5	Analista SOA.....	68
5.1.6	Analista de Testes	69
5.1.7	Arquiteto Corporativo	70
5.1.8	Arquiteto SOA	70
5.1.9	Auditor SOA	71
5.1.10	Desenvolvedor	72
5.1.11	Especialista em Governança SOA.....	72
5.1.12	Gerente de Projeto.....	73
5.1.13	Testador.....	74
6 -	CICLO DE VIDA DE PROJETOS DE SOLUÇÕES EM SOA.....	75
6.1	Modelagem de Negócio Corporativo	81
6.1.1	Processos e Tarefas da Modelagem de Negócio Corporativo.....	83
6.1.1.1	Tarefa 1 – Modelar Processos Empresariais.....	83
6.1.1.2	Tarefa 2 – Modelar Processos de Negócio (1ª visão – Visão de Atividades)	84
6.1.1.3	Tarefa 3 – Desenhar Processos de Negócio (2ª visão - Visão de Tarefas).....	85
6.1.1.4	Tarefa 4 – Modelar Estrutura da Informação	86
6.1.2	Visibilidade do Processo de Desenvolvimento	86
6.2	Análise de Inventário de Serviço.....	87
6.2.1	Processos e Tarefas da Análise de Inventário de Serviço	88
6.2.1.1	Tarefa 1 – Executar Análise Orientada a Serviço.....	88
6.2.1.2	Tarefa 2 – Analisar Modelos Corporativos de Negócio.....	88
6.2.1.3	Tarefa 3 – Definir Arquitetura Tecnológica.....	89
6.2.1.4	Tarefa 4 – Modelar Serviços Candidatos	89
6.2.2	Heurísticas para Análise de Inventário de Serviço	89

6.2.2.1	Princípio da Capacidade de Composição de Serviço	89
6.2.2.2	Princípio do Reuso de Serviço.....	90
6.2.3	Padrões de Projeto para Análise de Inventário de Serviço	90
6.2.4	Visibilidade do Processo de Desenvolvimento	91
6.3	Análise Orientada a Serviço	92
6.3.1	Processos e Tarefas da Análise Orientada a Serviço.....	93
6.3.1.1	Tarefa 1 – Definir Escopo da Análise	93
6.3.1.2	Tarefa 2 – Identificar Sistemas Existentes	93
6.3.1.3	Tarefa 3 – Identificar Entidades de Negócio e Fluxo de Informação.....	94
6.3.1.4	Processo 1 – Modelar Serviços Candidatos.....	94
6.3.2	Heurísticas para Análise Orientada a Serviço	97
6.3.2.1	Princípio da Capacidade de Composição de Serviço	98
6.3.3	Padrões de Projeto para Análise Orientada a Serviço.....	99
6.3.4	Visibilidade do Processo de Desenvolvimento	100
6.4	Projeto Orientado a Serviço	101
6.4.1	Processos e Tarefas do Projeto Orientado a Serviço.....	102
6.4.1.1	Processo 1 – Projetar Serviço de Entidade	103
6.4.1.2	Processo 2 – Projetar Serviço Utilitário	107
6.4.1.3	Processo 3 – Projetar Serviço de Tarefa.....	110
6.4.1.4	Processo 4 – Projetar Serviço de Tarefa Orquestrada	112
6.4.2	Heurísticas para Projeto Orientado a Serviço	116
6.4.2.1	Princípio da Padronização de Contrato de Serviço	116
6.4.2.2	Princípio do Baixo Acoplamento de Serviço	118
6.4.2.3	Princípio da Abstração de Serviço.....	120
6.4.2.4	Princípio da Visibilidade de Serviço	121
6.4.3	Padrões de Projeto para Projeto Orientado a Serviço.....	122
6.4.4	Visibilidade do Processo de Desenvolvimento	122
6.5	Projeto de Lógica de Serviço	124
6.5.1	Processos e Tarefas do Projeto de Lógica Orientada a Serviço	125
6.5.1.1	Tarefa 1 – Decompor Lógica de Serviço.....	125
6.5.1.2	Tarefa 2 – Projetar Lógica Principal de Serviço.....	126
6.5.1.3	Tarefa 3 – Projetar Lógica de Composição e Orquestração	126
6.5.1.4	Tarefa 4 – Projetar Lógica de Barramento	127
6.5.1.5	Tarefa 5 – Projetar Lógica de Pré e Pós Processamento	127

6.5.1.6	Tarefa 6 – Projetar Lógica de Monitoração de Processos (BAM)	127
6.5.2	Heurísticas para o Projeto de Lógica de Serviço	127
6.5.2.1	Princípio da Padronização de Contrato de Serviço	127
6.5.2.2	Princípio do Baixo Acoplamento de Serviço	129
6.5.2.3	Princípio da Autonomia de Serviço.....	132
6.5.2.4	Princípio da Ausência de Estado de Serviço	133
6.5.3	Padrões de Projeto para Projeto de Lógica de Serviço	134
6.5.4	Visibilidade do Processo de Desenvolvimento	135
6.6	Desenvolvimento de Serviço	136
6.6.1	Processos e Tarefas do Desenvolvimento de Serviço	136
6.6.1.1	Processo 1 – Implementar Serviço	136
6.6.2	Visibilidade do Processo de Desenvolvimento	137
6.7	Teste de Serviço	138
6.7.1	Processos e Tarefas do Teste de Serviço	138
6.7.1.1	Processo 1 – Testar Serviço.....	138
6.7.1.2	Processo 2 – Homologar Serviço.....	140
6.7.2	Visibilidade do Processo de Desenvolvimento	140
6.8	Disponibilização e Publicação de Serviço	142
6.8.1	Processos e Tarefas da Disponibilização e Publicação de Serviço	142
6.8.1.1	Processo 1 – Publicar de Serviço.....	142
6.8.1.2	Processo 2 – Catalogar de Serviço	143
6.8.2	Visibilidade do Processo de Desenvolvimento	144
6.9	Implementação de Aplicação Composta.....	146
6.9.1	Processos e Tarefas da Implementação de Aplicação Composta	146
6.9.1.1	Processo 1 – Implementar Aplicação Composta	147
6.9.2	Visibilidade do Processo de Desenvolvimento	147
6.10	Teste da aplicação composta.....	148
6.10.1	Processos e Tarefas do Teste da Aplicação Composta	148
6.10.1.1	Processo 1 – Testar Solução	148
6.10.1.2	Processo 2 – Homologar Solução.....	149
6.10.2	Visibilidade do Processo de Desenvolvimento	149
6.11	Disponibilização de Aplicação Composta.....	150
6.11.1	Processos e Tarefas da Disponibilização de Aplicação Composta	150
6.11.1.1	Processo 1 – Implantar Solução	150

6.11.2	Visibilidade do Processo de Desenvolvimento	150
7 -	IMPLEMENTAÇÃO E APLICAÇÃO DA METODOLOGIA	152
7.1	Agência Nacional de Vigilância Sanitária – ANVISA.....	152
7.2	Agência Nacional de Águas – ANA	153
7.3	Ministério da Saúde – MS.....	154
7.3.1	CADSUS	154
7.3.2	CNES	155
7.3.3	Contratação de Fábrica de Serviço Terceirizada e Consultoria para Centro de Excelência SOA.....	156
7.4	Outros Trabalhos já Iniciados.....	156
8 -	CONCLUSÕES.....	158
	REFERÊNCIAS BIBLIOGRÁFICAS	162
	APÊNDICES	170
	Treinamentos Sugeridos.....	170

ÍNDICE DE TABELAS

Tabela 2-1 – Análise Metodologias	37
Tabela 3-1 – Lista de Unidades Funcionais	41
Tabela 3-2 - Ciclo de Vida de Desenvolvimento de Software	42
Tabela 3-3 – Perfis Profissionais	45

ÍNDICE DE FIGURAS

Figura 2-1 – Visão Geral dos Passos da <i>Framework</i> SOAD	29
Figura 2-2 – Visão Geral da Metodologia SOMA	31
Figura 2-3 – Visão Geral do Processo da SOMA	32
Figura 2-4 – Visão Geral da OUM.....	33
Figura 3-1 – Visão Geral da Metodologia SOA.....	48
Figura 4-1 – Estrutura Funcional	50
Figura 4-2 – Processos do Escritório de Gestão de Processos de Negócio.....	52
Figura 4-3 – Processos do Escritório de Administração de Dados	54
Figura 4-4 – Processos do Escritório de Infraestrutura de TI	55
Figura 4-5 – Tipos de arquitetura que compõem uma Arquitetura Empresarial (conforme o modelo TOGAF®)	57
Figura 4-6 – Processos do Escritório de Arquitetura Corporativa.....	58
Figura 4-7 – Processos do Escritório de Gestão de Projetos	59
Figura 4-8 – Processos do Centro de Excelência SOA	60
Figura 4-9 – Processos da Fábrica de Serviços.....	62
Figura 4-10 – Processos da Fábrica de <i>Software</i>	63
Figura 4-11 – Processos da Fábrica de Testes	64
Figura 6-1 – Quadro Resumo de Informações da Etapa	76
Figura 6-2 – Quadro de Artefatos da Etapa	77
Figura 6-3 – Definição de Processo Empresarial.....	82
Figura 6-4 – Processo Modelar Negócio Corporativo.....	83
Figura 6-5 – Entradas e Saídas da Etapa de Modelagem de Negócio Corporativo.....	86
Figura 6-6 – Processo Analisar Inventário de Serviços	88
Figura 6-7 – Entradas e Saídas da Etapa de Análise de Inventário de Serviço.....	91
Figura 6-8 – Processo Analisar com Orientação a Serviço.....	93
Figura 6-9 – Subprocesso Modelar Serviços Candidatos	94
Figura 6-10 – Entradas e Saídas da Etapa Análise Orientada a Serviço	100
Figura 6-11 – Etapa Projeto Orientado a Serviço	103
Figura 6-12 – Subprocesso Projetar Serviço de Entidade	103
Figura 6-13 – Subprocesso Projetar Serviço Utilitário.....	107
Figura 6-14 – Projetar Serviços de Tarefa.....	110
Figura 6-15 – Subprocesso Projetar Serviço de Tarefa Orquestrada	113
Figura 6-16 – Entradas e Saídas da Etapa de Projeto Orientado a Serviço	123

Figura 6-17 – Processo Projetar Lógica de Serviço.....	125
Figura 6-18 – Entradas e Saídas da Etapa de Projeto de Lógica de Serviço	135
Figura 6-19 – Entradas e Saídas da Etapa de Desenvolvimento de Serviço	137
Figura 6-20 – Entradas e Saídas da Etapa de Teste de Serviço	141
Figura 6-21 – Entradas e Saídas da Etapa de Disponibilização e Publicação de Serviço.....	145
Figura 6-22 – Entradas e Saídas da Etapa de Implementação de Aplicação Composta.....	147
Figura 6-23 – Entradas e Saídas da Etapa de Teste da Aplicação Composta.....	149
Figura 6-24 – Entradas e Saídas da Etapa de Disponibilização de Aplicação Composta.....	151

ACRÔNIMOS

2PC	Two-phase Commit
B2B	Business to Business
BAM	Business Activity Monitoring
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BPMO	Business Process Management Office
BPMS	Business Process Management System
CBD	Component-Based Development
CBDI-SAE	Component-Based Development and Integration – Service Architecture and Engineering
CBOK	Common Body of Knowledge
CEP	Complex Event Processing
CMM	Capability Maturity Model
COS	Computação Orientada a Serviços
DAO	Data Administration Office
EA	Enterprise Architecture
EAO	Enterprise Architecture Office
EAP	Estrutura Analítica do Projeto
ESB	Enterprise Service Bus
IASA	International Association of Software Architects
ITIO	Information Technology Infrastructure Office
ISO	International Organization for Standardization
ISTQB	International Software Testing Qualifications Board
KPI	Key Performance Indicator
OASIS	Organization for the Advancement of Structured Information Standards
OOAD	Object Oriented Analysis and Design
PMBok	Project Management Body of Knowledge
PMI	Project Management Institute
PMO	Project Management Office
PMP	Project Management Professional

QoS	Quality of Service
RF	Reference Framework
RFP	Request for Propose
RQ	Repeatable Quality
RUP	Rational Unified Process
SE	Software Engineering
SLA	Service Level Agreement
SOA	Service-Oriented Architecture
SOA CoE	SOA Center of Excellence
SOAD	Service-Oriented Analysis and Design
SOAP	Simple Object Access Protocol
SOMA	Service-Oriented Modeling and Architecture
SOSE	Service-Oriented Software Engineering
SOUP	Service-Oriented Unified Process
SOX	Sarbanes-Oxley
TI	Tecnologia da Informação
TOGAF	The Open Group Architecture Framework
OUM	Oracle Unified Method
UDDI	Universal Description, Discovery and Integration
UI	User Interface
WBS	Work Breakdown Structure
WS-BPEL	Web Services Business Process Execution Language
WSDL	Web Service Description Language
XML	Extensible Markup Language
XP	Extreme Programming
XSD	XML Schema Definition

1 - INTRODUÇÃO

Empresas modernas encontram diversos desafios de gestão de TI relacionados à restrição cada vez maior de recursos, à competitividade do mercado e à volatilidade dos cenários de negócio. Em resposta, devem projetar, desenvolver e implantar sistemas complexos com custo e tempo adequados (BREIVOLD e LARSSON, 2007). Ao longo dos últimos anos diversas técnicas, metodologias e diferentes abordagens de análises de negócio têm surgido com o objetivo de melhorar o desenvolvimento de aplicações voltadas ao negócio (BRABEC e BUCHALCEVOVÁ, 2007).

Arsanjani (2004) descreve o alinhamento da filosofia orientada a serviços com os objetivos e estratégias de negócio das organizações, como um conjunto de padrões e diretrizes para a criação de serviços de baixo acoplamento alinhados ao negócio, fornecendo uma flexibilidade sem precedentes na capacidade de resposta a desafios de novos negócios e oportunidades, sendo fator crítico de sucesso.

SOA (*Service Oriented Architecture*) é um estilo de arquitetura de software que possibilita que as funções de negócios sejam expostas como serviços reutilizáveis e autônomos, permitindo que parceiros de negócios pesquisem e descubram os serviços sob demanda (CHERBAKOV et al., 2005). Uma aplicação construída com base na orientação a serviços compõe serviços em tempo de execução, para atender requisitos de negócios com potencial de constante mudança. A agilidade, reutilização e flexibilidade que SOA oferece são motivadores para um número crescente de empresas adotem SOA como seu estilo arquitetônico no desenvolvimento, manutenção e evolução de sistemas de informação (PAPAZOGLU, 2007).

O paradigma de “orientação a serviço” preconiza que a estratégia necessária para resolver um grande problema possa ser construída, implementada e gerenciada, através da decomposição da lógica de negócio em uma coleção de funções mais simples e relacionadas, cada uma abordando parte do problema específico, e estimula a existência de unidades individuais autônomas de lógica, relacionadas uma às outras (ERL, 2008). Tais unidades são chamadas “serviços”. A composição de serviços facilita a solução de problemas complexos agregando serviços, que são organizados de forma a atender requisitos de negócio ou funções computacionais bem definidas. Como consequência, além

de promover o reuso de software, a composição de serviços dá maior agilidade e proporciona redução de custos no processo de desenvolvimento de software (MILANOVIC; MALEK, 2004).

A Arquitetura Orientada a Serviço pode ser definida como “uma estrutura (*framework*) para integrar processos de negócio e a infraestrutura de TI que o suporta, na forma de componentes seguros e padronizados – serviços – que podem ser reutilizados e combinados para endereçar mudanças de prioridade do negócio” (BIEBERSTEIN, 2006).

Para reforçar esta visão, Papazoglou (2007) cita que normalmente a execução de operações de negócio compreende uma série de invocações de diferentes componentes, possivelmente com protocolos distintos de computação distribuída, muitas vezes de forma orientada a eventos ou assíncrona, refletindo e atendendo às necessidades dos processos de negócios subjacentes.

Os principais benefícios de arquiteturas baseadas em SOA são facilitar o crescimento gerenciável de sistemas empresariais de grande porte, facilitar o fornecimento escalável da internet por meio do uso de serviços e redução de custos para a colaboração na organização (OASIS, 2006).

1.1 MOTIVAÇÃO

SOA vem atraindo uma enorme atenção do mercado e sua adoção se expande rapidamente. Segundo recente pesquisa (WINTERGREEN, 2012) o mercado de SOA cresceu 38% em 2011, atingindo a cifra de \$5.5 bilhões de dólares americanos frente aos \$3.9 bilhões de dólares em 2010, representando um crescimento de 24% acima das previsões anteriores. Este relatório indica ainda um crescimento anual estimado de 11,5% até 2014, apenas no mercado americano. Em sua atualização (WINTERGREEN, 2013), o relatório aponta que, em 2012, o mercado atingiu os \$7,1 bilhões de dólares, um crescimento de cerca de 30%, indicando ainda que até 2019 deve superar a marca dos \$15 bilhões de dólares.

Toda essa expansão tem sido relacionada à necessidade de rápida adaptação das organizações às constantes mudanças de cenário do mercado, porque os sistemas que são construídos com base em SOA prometem trazer benefícios comerciais para os clientes, na sua flexibilidade na mudança e na sua estreita vinculação ao negócio, além de benefícios

tecnológicos para os desenvolvedores. Outra forte tendência é o crescimento de outras tecnologias/estratégias que são fortalecidas e fortalecem a adoção de SOA, como computação em nuvem (*Cloud Computing*) e *Software* como Serviço (SaaS – *Software as a Service*) (WANG et al., 2010).

Para que os sistemas orientados a serviços obtenham os benefícios esperados, os princípios conhecidos da engenharia de software tradicional (SE - *Software Engineering*) precisam ser adaptados para desenvolvimento orientado a serviços. São necessárias abordagens sistemáticas para a concepção, desenvolvimento e manutenção de sistemas orientados a serviços. Essa nova abordagem é conhecida como Engenharia de *Software* Orientada a Serviços (SOSE - *Service-oriented System Engineering*) (TSAI et al., 2007).

Diversos trabalhos, tais como *roadmaps* de pesquisa (PAPAZOGLU et al., 2008), metodologias (PAPAZOGLU e YANG, 2002), projetos de engenharia de serviços (ZIRPINS et al., 2003), foram propostos por pesquisadores para investigar e detalhar os desafios de SOSE. Em particular, Tsai (2005), aponta problemas colaborativos e os causados pela natureza dinâmica dos sistemas orientados a serviços. É possível obter uma boa visão geral das diferenças entre a tradicional engenharia de software (SE) e a orientada a serviços (SOSE) no trabalho de Tsai et al. (2007). A agenda de pesquisa proposta pela Networked European Software & Services Initiative (NESSI, 2008) aborda oito diferentes áreas de pesquisa. Todos esses trabalhos endereçam a apenas parte dos reais desafios da SOSE.

Fundamentalmente, trabalhos e pesquisas analisados apontam para elementos essenciais ao sucesso de uma iniciativa SOA, sendo os principais: a existência e observância de uma metodologia de desenvolvimento SOA adequada às características da organização; a definição de uma arquitetura de referência SOA aderente a arquitetura corporativa; a definição e obediência das estratégias de Governança SOA; e a aplicação das melhores práticas de mercado já consolidadas para a abordagem orientada a serviços.

Uma metodologia de desenvolvimento voltada para forças de trabalho terceirizadas deve levar em consideração todos os princípios e melhores práticas de fábrica de software (HOTLE e LANDRY, 2009; CLEMENTS e NORTHROP, 2001; BROWN et al., 2011),

procedimentos de certa forma já maduros no mercado nacional em processos de terceirização.

Dois casos específicos serviram como motivadores determinantes para a criação e aprofundamento deste trabalho, as experiências vividas na ANVISA – Agência Nacional de Vigilância Sanitária e ANA – Agência Nacional de Águas. Embora estes casos sejam melhor detalhados no Capítulo 7, cabe aqui um breve resumo. Na ANVISA, com uma Arquitetura de Referência SOA definida, porém, com baixa maturidade, sem governança e sem uma metodologia formalmente constituída, mesmo com a concordância da fábrica de software terceirizada, o processo de desenvolvimento não atingiu os objetivos esperados, culminando com a desistência da fábrica de continuar sua participação no processo.

Na ANA, com um melhor nível de maturidade em processos de desenvolvimento de soluções, mesmo que tradicionais, com uma Arquitetura de Referência SOA homologada pela instituição, com vistas a experiência da ANVISA, foi iniciada a construção deste trabalho, a primeira versão desta metodologia. Após sua homologação, a empresa terceirizada como fábrica de software foi chamada para tomar conhecimento dos novos rumos na Agência e utilizar a metodologia para o desenvolvimento de um novo projeto. Houve recusa imediata por parte da terceirizada, sob a alegação que o contrato em vigor não contemplava SOA e nem uma metodologia específica de trabalho.

Outro fator motivador determinante de iniciar o desenvolvimento desta metodologia, foram as tentativas de se utilizar metodologias já existentes no mercado. Muitos problemas foram enfrentados, da dificuldade de obter informações precisas e detalhadas das mesmas, como principalmente das de domínio de empresas como a IBM e a Oracle, até a não consideração da utilização de forças de trabalho terceirizadas para a execução de seus processos e tarefas.

A introdução de equipes terceirizadas para a execução de uma metodologia de desenvolvimento de soluções baseadas em SOA traz uma complexidade ao cenário que não era tratada em nenhuma das metodologias analisadas. Esta complexidade se traduz principalmente na ausência de definições claras das responsabilidades de cada unidade da estrutura da organização que participa na execução da metodologia, da descrição detalhada dos processos e tarefas que a compõe e da discriminação das habilidades e capacidades

profissionais de seus atores, justamente os três pilares que estruturam a metodologia proposta, além do apontamento dos artefatos necessários em todas suas fases. Deve-se ainda levar em consideração que estes fatores possibilitam a utilização de mais de uma empresa terceirizada em todo o processo metodológico.

Focando nestes itens, este trabalho foi desenvolvido levando em consideração as metodologias já existentes, as características do mercado brasileiro e especificidades das empresas e órgãos governamentais, culminando com o estabelecimento de uma metodologia para desenvolvimento de soluções baseadas em SOA, flexível e voltada para a utilização de forças de trabalho terceirizadas.

A experiência do autor na atuação como consultor especialista promovendo iniciativas em SOA no Brasil, tem mostrado a grande dificuldade que empresas e, principalmente, órgãos governamentais enfrentam na efetividade de seu plano de adoção.

Entre as dificuldades observadas destacam-se:

- Contratar empresas prestadoras de serviço aptas a trabalhar com orientação a serviço;
- Definir uma metodologia de desenvolvimento para soluções SOA que seja suficientemente flexível para ser usada em um ambiente onde muitas das atividades são de responsabilidade de empresas terceiras e, por muitas vezes, mais de uma empresa;
- Gerenciar e controlar todo o processo de desenvolvimento e os ativos gerados.

Logo, de forma a garantir o sucesso de suas iniciativas em SOA, as empresas carecem de métodos, padrões, processos e governança adaptados a essa realidade, necessidade que este trabalho pretende suprir.

1.2 OBJETIVO

1.2.1 Objetivo Geral

Este trabalho objetiva especificar e detalhar uma metodologia para desenvolvimento de soluções baseadas em SOA adaptada às necessidades e características do mercado, das

empresas e organizações governamentais que promovem a iniciativa de adoção de uma arquitetura orientada a serviço, habitualmente terceirizando sua execução, seja em sua totalidade ou fases/etapas específicas do processo de desenvolvimento de software.

1.2.2 Objetivos Específicos

- OE1: Definir a estrutura funcional necessária à metodologia, de forma a agrupar tarefas e atividades tecnicamente relacionadas;
- OE2: Definir os perfis profissionais para a metodologia, tornando clara a divisão de responsabilidades de tarefas e capacidades técnicas necessárias para sua execução;
- OE3: Definir processos de modo a especificar as atividades requeridas no desenvolvimento de software orientado a serviço;
- OE4: Validar e testar a metodologia através de estudo de caso, onde a metodologia definida é aplicada no desenvolvimento de software com orientação a serviço em organizações que estejam iniciando a adoção em SOA, com uso de algum tipo de força de trabalho terceirizada.

1.3 CARACTERIZAÇÃO DA PESQUISA

Os tipos de pesquisa empregados na elaboração desta metodologia foram:

Descritiva: teve um cunho qualitativo em função da necessidade de se realizar um estudo exploratório, compreendendo a realização de grupos de foco e entrevista com especialistas, buscando assim descobrir o que está faltando em relação ao problema de pesquisa (Richardson, 1999).

A observação foi o instrumento principal para a interpretação da realidade, contribuindo para sua descrição, classificação e interpretação dos fatos e fenômenos.

Exploratória: Estas pesquisas têm como objetivo proporcionar maior familiaridade com o problema, com vistas a tomá-lo mais explícito ou a constituir hipóteses. Pode-se dizer que estas pesquisas têm como objetivo principal o aprimoramento de idéias ou a descoberta de intuições. Seu planejamento é, portanto, bastante flexível, de modo que possibilite a

consideração dos mais variados aspectos relativos ao fato estudado. Na maioria dos casos, essas pesquisas envolvem: (a) levantamento bibliográfico; (b) entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado; e (c) análise de exemplos que “estimulem a compreensão” (SELLTIZ et al., 1967, p.63).

Nesta perspectiva da pesquisa, os dois instrumentos mais utilizados foram a pesquisa bibliográfica e, principalmente, o estudo de casos, no caso, da Agência Nacional de Vigilância Sanitária, da Agência Nacional de Águas e do Ministério da Saúde, que por vezes funcionaram como instrumentos de pesquisa, mas principalmente no caso do Ministério da Saúde, como validador da metodologia proposta.

Participante: A pesquisa participante caracteriza-se pela interação entre pesquisadores e membros das situações investigadas (BARBIER, 1996). Isto se aplicou principalmente no caso da Agência Nacional de Águas e no Ministério da Saúde, onde a interação entre o pesquisador e os profissionais da instituição ocorreu de forma ativa, de onde surgiu o embrião da metodologia proposta.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho é dividido nos seguintes Capítulos:

No Capítulo 2 são apresentadas algumas das principais metodologias orientadas a serviço existentes no âmbito internacional e conceitos de fábrica de software envolvidos.

No Capítulo 3 é apresentada a estrutura e uma visão geral da metodologia proposta.

No Capítulo 4 é apresentada a estrutura funcional associada à metodologia, detalhando unidades corporativas e unidades dedicadas como agrupamentos lógicos de atividades e responsabilidades.

No Capítulo 5 são definidos os perfis profissionais necessários para executar as atividades previstas na metodologia.

No Capítulo 6, o ciclo de vida de projetos de soluções baseadas em SOA, ou seja, a metodologia em si, é apresentado.

No Capítulo 7 são apresentados os resultados e aplicações da metodologia.

No Capítulo 8 são apresentadas as conclusões do trabalho.

2 - ANÁLISE DAS METODOLOGIAS EXISTENTES

Além das atividades de engenharia de software tradicionais, como codificação, testes e implantação, o desenvolvimento de aplicações SOA requer a identificação, descoberta e composição de serviços. Assim, as metodologias existentes de desenvolvimento de software já não satisfazem as necessidades para o desenvolvimento SOA. São necessárias abordagens sistemáticas, disciplinadas e quantificáveis para as fases de concepção, desenvolvimento e manutenção. Tsai et al. (2007) generaliza estas abordagens como metodologias de engenharia de software orientada a serviços (SOSE - *Service-oriented System Engineering*). Diferentemente de algumas abordagens orientadas a serviço dirigidas a atividades específicas, tais como métodos de identificação de serviços, metodologias SOSE muitas vezes fornecem orientações sobre múltiplas atividades e visam à engenharia de aplicações orientadas a serviço e não apenas à concepção de serviços atômicos ou a composições de serviços.

Muitas metodologias SOSE foram propostas por fornecedores, tais como IBM, que apresentou SOAD – SOA Decision Modeling (ZIMMERMANN et al., 2004), SOMA – Service Oriented Modeling e Architecture (ARSANJANI, 2008) e SOUP – Service-Oriented Unified Process (MITTAL, 2006); ORACLE, Unified Method (OUM) Oracle's Full Lifecycle Method for Deploying Oracle-Based Business Solutions (2012); OASIS, uma metodologia para arquiteturas de serviços (JONES; MORRIS, 2006); e Everware-CBDI, um processo orientado a serviço (ALLEN, 2007).

Na SOSE, métodos e processos para o desenvolvimento de aplicações orientadas a serviço estão se tornando um tema de pesquisa importante e recorrente. Metodologias SOSE propostas por Papazoglou et al. (2006) e Chang (2007) são exemplos bem conhecidas do meio acadêmico.

As metodologias variam muito em sua abrangência, complexidade, disciplinas e fases contempladas. Como então analisá-las, compará-las, obter o que de melhor cada uma oferece e adaptá-las às características do mercado brasileiro de forma a melhor instrumentar as iniciativas de adoção SOA nas empresas e órgãos governamentais? Foi utilizado, como base para a nossa análise, o trabalho por pesquisadores (GU; LAGO, 2011), que desenvolveram um método de avaliação de metodologias de engenharia de

software orientadas a serviço, bem como um estudo comparativo elaborado por um grupo de pesquisadores (RAMOLLARI et al, 2007).

Análise de recursos é um método de avaliação qualitativa, oferecendo uma forma de verificar e avaliar um grande número de metodologias por meio da avaliação das suas características ou funcionalidades (KITCHENHAM, 1996). Ao realizar a análise de recursos pode-se obter uma compreensão de aspectos específicos de uma metodologia (JAYARATNA, 1994), podendo-se sistematicamente selecionar ou descartar uma metodologia para atingir um objetivo específico (SIAU, 1998).

Diferentemente dos quadros de comparação já existentes, como, por exemplo, o encontrado em Kontogogos e Avgeriou (2009), a nossa abordagem será mais descritiva, destacando as características de algumas das metodologias de maior prestígio no mercado. Nela, a importância das características será ressaltada por meio de comentários relativos à sua relevância no contexto de SOA.

Para a apresentação das metodologias, não foi feito um comparativo detalhado entre elas, até mesmo porque os trabalhos citados já o fizeram, mas destacamos algumas características que consideramos mais relevantes.

2.1 IBM ANALYSIS SERVICE-ORIENTED AND DESIGN (SOAD)

IBM Analysis Service-Oriented and Design (SOAD): (ZIMMERMANN, 2004; CHANG, 2007 (2)). SOAD propõe elementos que devem fazer parte de uma análise orientada a serviços e metodologia de projeto, portanto, é um *framework* abstrato e não uma metodologia holística. Concentra suas atividades apenas nas fases de análise e projeto de arquitetura de soluções baseadas em serviços. SOAD baseia-se em técnicas existentes comprovadas, como OOAD – *Object-Oriented Analysis and Design*, CBD – *Component-Based Development*, e BPM – *Business Process Management*. Além disso, introduz técnicas específicas de SOA, tais como:

- Categorização e agregação de serviço - Serviços têm diferentes usos e finalidades, por exemplo, serviços de software podem ser distinguidos de serviços comerciais. Além disso, serviços atômicos podem ser orquestrados (compostos) em serviços de nível superior, completos. Composição de serviços é simplificada por modelos

executáveis (como os modelados em BPEL), o que é algo com que métodos e ferramentas de modelagem tradicionais não lidam;

- Políticas e aspectos - Um serviço tem sintaxe, semântica e características de QoS (*Quality of Service*) que devem todas ser modeladas; contratos de interface formais têm que cobrir mais do que o Web Services Description Language (WSDL) faz. Portanto, a estrutura do WS-Policy é uma especificação relacionada importante. Outra característica importante da metodologia é a Rastreabilidade de Negócios, pois esta é uma qualidade desejável, além do princípio bem estabelecido de rastreabilidade arquitetônica: deve ser possível ligar diretamente todos os artefatos de execução para uma linguagem que um especialista de domínio não técnico possa entender. Isto é particularmente importante para abstrações diretamente expostas na camada de negócios e de serviço. A Lei Sarbanes-Oxley (SOX) é um exemplo de diretriz de negócio que exige este tipo de rastreabilidade;
- Processo de análise “*meet-in-the-middle*” - Não existem projetos sem obstáculos no mundo real, aplicações legadas sempre têm que ser levadas em conta. Portanto, idealmente, é necessária uma abordagem *meet-in-the-middle* (encontro no meio), que combina o uso de abordagens *top-down* (de cima para baixo) e *bottom-up* (de baixo para cima), ao invés de usar exclusivamente uma ou outra. A abordagem *bottom-up* tende a levar a pobres abstrações de serviço do negócio, caso o projeto seja ditado pela TI, ao invés de necessidades de negócios atuais e futuras. Por outro lado, o processamento *top-down* pode causar características insuficientes de requisito não funcional e comprometer outros fatores de qualidade de arquitetura (por exemplo, problemas de desempenho devido à falta de normalização no modelo de domínio), bem como gerar incompatibilidades na camada de serviço e de componente;
- “*Semantic Brokering*” - Em qualquer contexto SOA, um contrato formal de interface para a invocação por meio de sintaxe é importante. A questão semântica (o significado de parâmetros e assim por diante) também tem que ser resolvida (modelagem de domínio). Isso é fundamental em qualquer cenário *business-to-business* (B2B) e de invocação dinâmica. Tais cenários são pilares da visão SOA de

ser flexível e ágil na resposta às novas necessidades de negócios em um mundo de fusões e aquisições, transformação do negócio, globalização e assim por diante;

- “*Harvesting*” (colheita ou descoberta) de serviço - Esta é a gestão do conhecimento e a questão do ciclo de vida. Os serviços devem ser identificados e definidos para reuso (e descoberta) como um dos principais critérios da condução de SOA. Se um componente (ou serviço) não tem potencial para ser reutilizado, então provavelmente não deve ser implantado como um serviço. Ele pode ser conectado a outro serviço associado com a arquitetura corporativa, mas não vai ser um serviço em seu próprio direito. No entanto, mesmo se a reutilização está prevista para logo desde o início, o processo ainda tem que formalizar o processo de descoberta do serviço. Utilização do serviço por mais de um consumidor é um objetivo explícito do projeto SOA. Um registro de serviço (*service registry*) e um diretório UDDI em toda a empresa podem, por exemplo, ser parte da resposta.

Uma Visão Geral do Modelo de Decisão SOA (SOAD) é apresentada na Figura 2-1 a seguir:

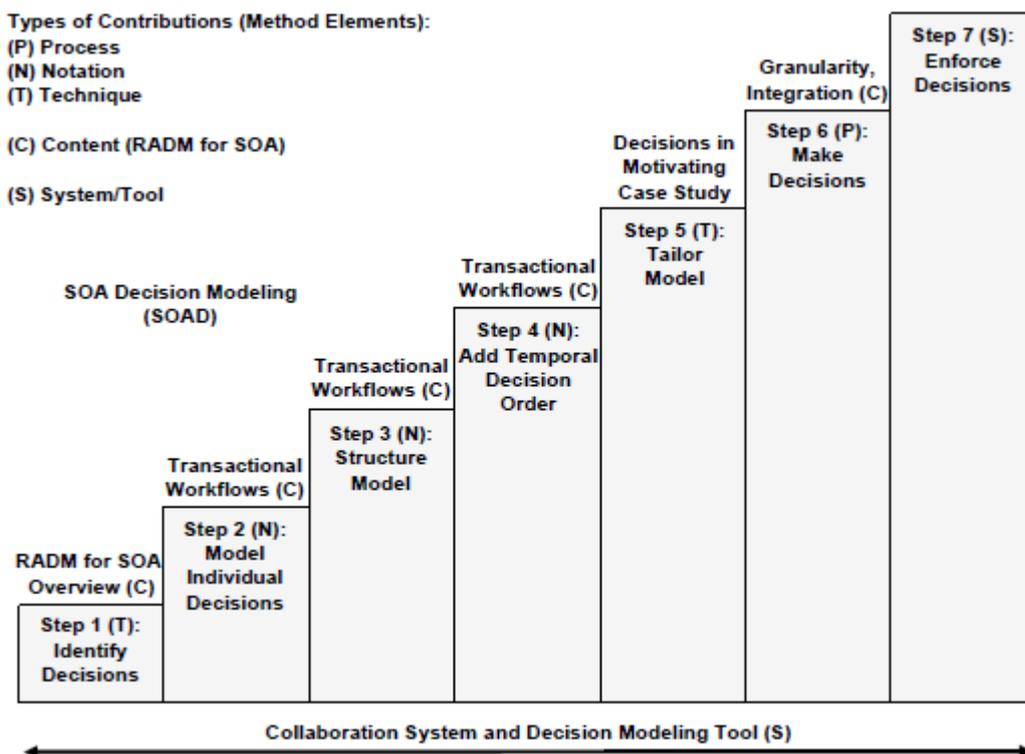


Figura 2-1 – Visão Geral dos Passos da Framework SOAD
(Fonte: Zimmermann, 2009)

2.2 IBM SERVICE ORIENTED MODELING AND ARCHITECTURE (SOMA)

IBM *Service Oriented Modeling and Architecture (SOMA)*: (ARSANJANI, 2004; ARSANJANI, 2008) SOMA é uma metodologia madura de modelagem desenvolvida pela IBM que consiste em três etapas: identificação, especificação e realização de serviços, fluxos (processos de negócios) e os componentes que realizam serviços. O processo é altamente iterativo e incremental.

Como um método de ciclo de vida de desenvolvimento de software para desenvolvimento de soluções baseadas em SOA ou qualquer solução usando princípios orientados a serviços, SOMA define as principais técnicas e descreve os papéis em um projeto de SOA e uma estrutura de divisão de trabalho (WBS - *Work Breakdown Structure*). A WBS inclui tarefas, os produtos de trabalho de entrada e saída para as tarefas, bem como a orientação normativa necessária para uma análise detalhada, projeto, implementação e implantação de serviços, componentes e fluxos necessários para construir um ambiente SOA robusto e reutilizável.

O método inclui as sete fases principais mostrados na Figura 2-2. As fases “fractais” contêm recursos que podem ser aproveitados quando necessário em diferentes sequências. Em um modelo fractal, etapas consistem de recursos que podem ser utilizadas por outras fases. Governança serve como contexto e para o ciclo de vida orientada a serviços. SOMA fornece apoio e vínculos com os dois aspectos principais de governança: em tempo de projeto e gestão de tempo de execução.

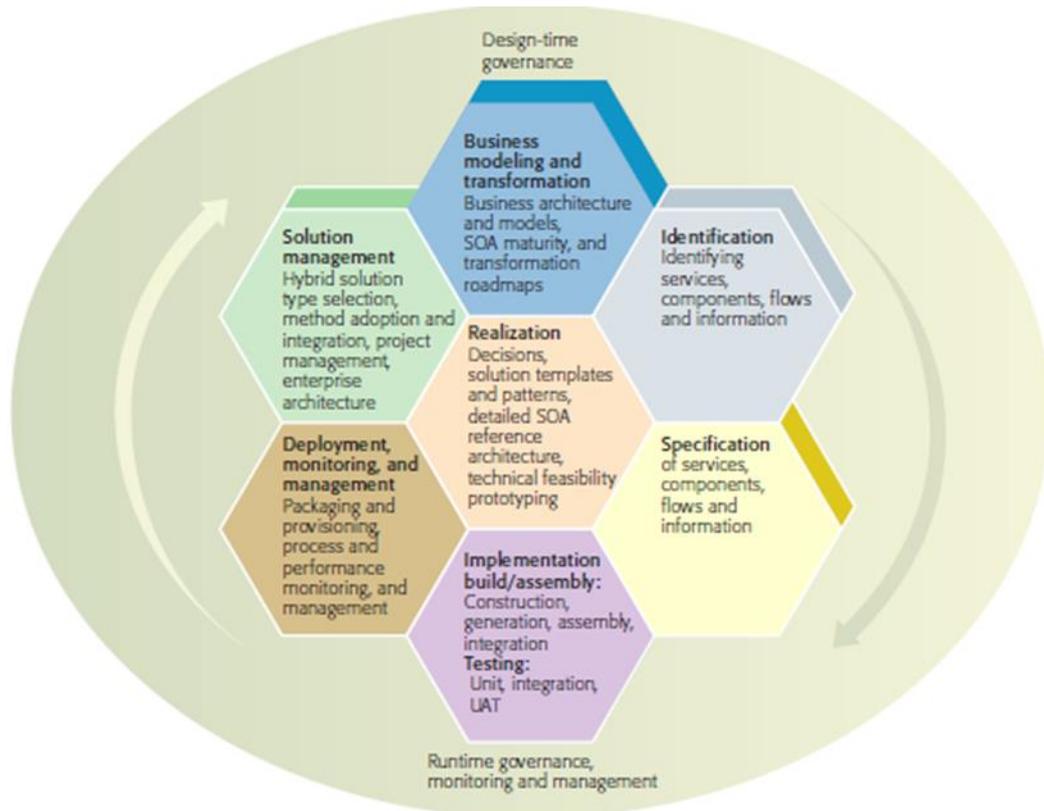


Figura 2-2 – Visão Geral da Metodologia SOMA
(Fonte: Arsanjani, 2009)

O processo apresentado na Figura 2-3 é uma descrição sequencial do fluxo global fractal a um nível elevado. Considerando a Figura 2-2, ilustra os padrões de capacidade e natureza fractal do método SOMA.

A Figura 2-3 ilustra um típico fluxo de processo de execução do método SOMA.

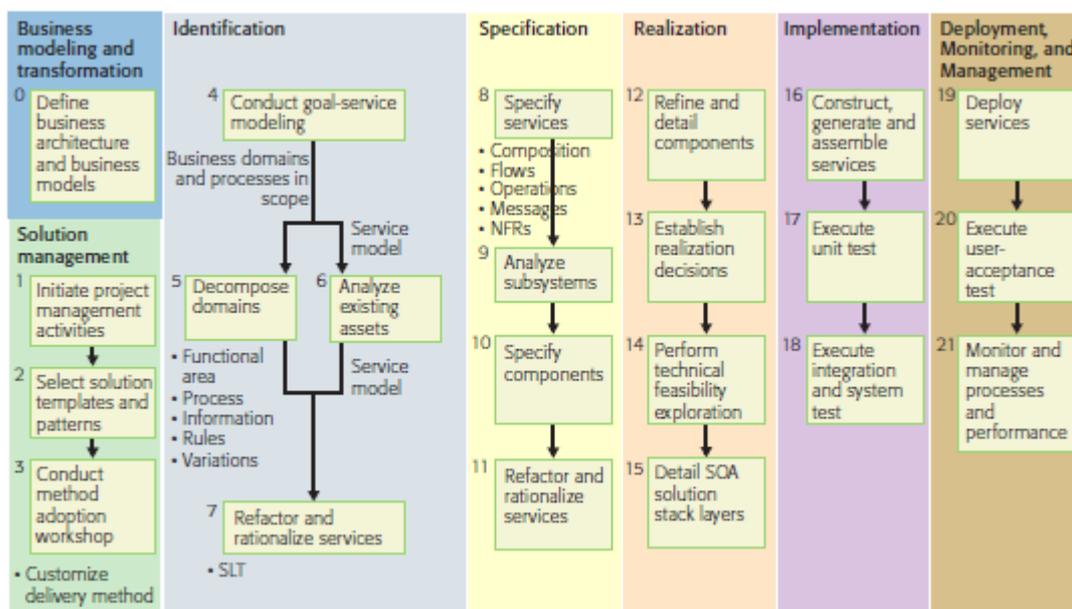


Figura 2-3 – Visão Geral do Processo da SOMA
(Fonte: Arsanjani, 2009)

SOMA é uma metodologia completa, muito sofisticada e, como toda metodologia proprietária, utiliza características específicas da plataforma tecnológica que comercializa.

2.3 ORACLE UNIFIED METHOD (OUM)

Oracle Unified Method (OUM): (ORACLE, 2012) A metodologia da Oracle evoluiu bastante após a aquisição da Sun Microsystems, que possuía a **SOA Repeatable Quality (SOA RQ)**, uma metodologia baseada em um RUP (*Rational Unified Process*) como processo incremental e iterativo composto por cinco fases: iniciação, elaboração, construção, transição e concepção. UMLs compatíveis com os artefatos são utilizados para documentar diversos subprodutos dessas fases.

A área de foco fornece uma estrutura para desenvolver e implementar soluções de negócios baseadas em Oracle. OUM usa as fases e processos do projeto para incluir qualidade, postos de controle e permitir a coordenação das atividades ao longo do projeto. Durante uma fase do projeto, a equipe do projeto executa tarefas em vários processos.

A Figura 2-4 ilustra o método.

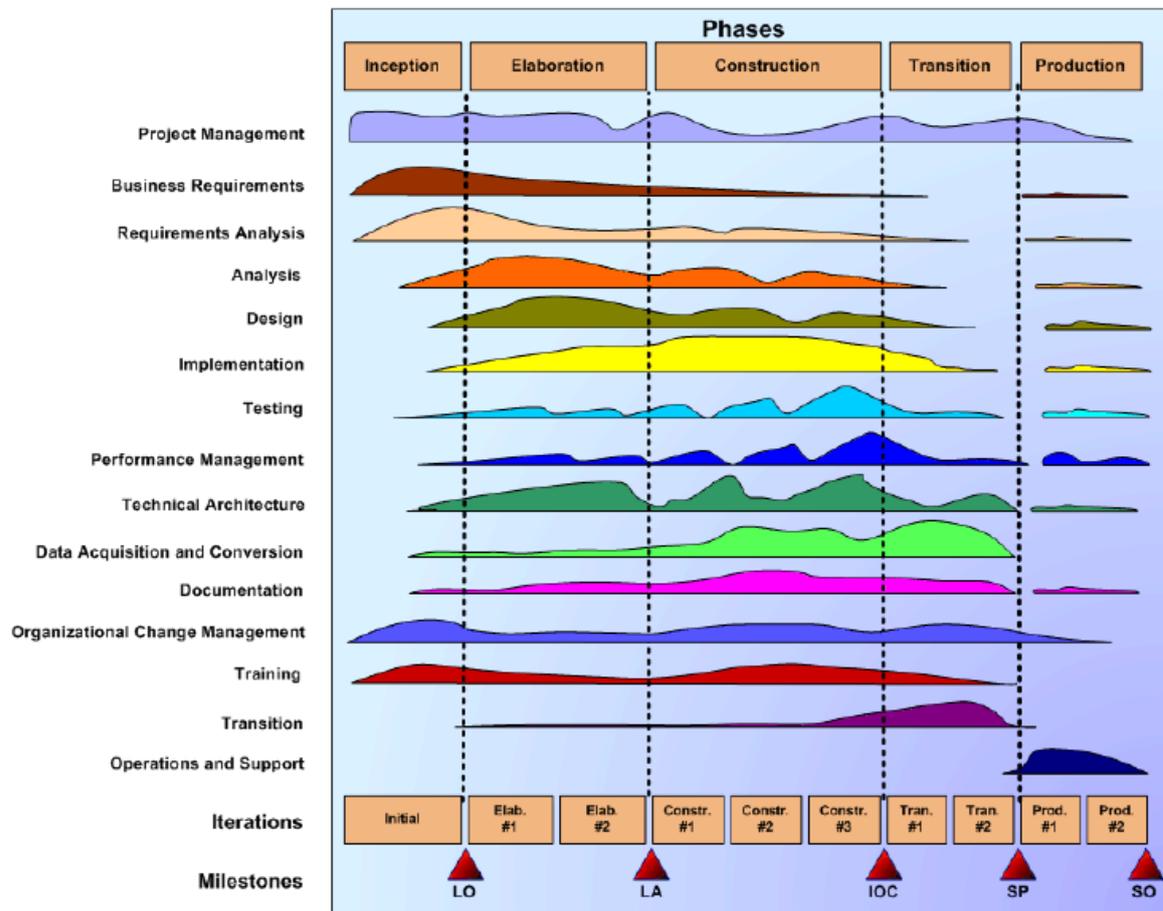


Figura 2-4 – Visão Geral da OUM
(Fonte: Oracle, 2012)

Na metodologia, projetos são entregues por etapa, agrupamento cronológico de tarefas é uma abordagem utilizada com o intuito de reduzir o risco. Cada fase permite que haja um posto de controle em relação aos objetivos do projeto e medição com base em critérios de qualidade. OUM inclui as seguintes fases:

Iniciação (*Inception*) - O objetivo primordial da fase de Iniciação é obter o consenso entre todas as partes interessadas sobre os objetivos do ciclo de vida do projeto. A fase de Iniciação é usada para lançar um projeto; rever o direcionamento estratégico da empresa e confirmar, documentar e priorizar as necessidades de negócios de alto nível para o projeto. É também a hora de começar a montagem e integração da equipe do projeto, o escopo e desenvolver o plano inicial de projeto;

Elaboração (*Elaboration*) - Seu objetivo é desenvolver os requisitos detalhados, efetuar a partição da solução e criar quaisquer protótipos necessários e básicos da arquitetura do sistema. Este esforço resulta em uma base estável para o projeto e diminui o esforço de implementação em fase de construção. A arquitetura se desenvolve a partir dos requisitos mais importantes, aqueles que têm o maior impacto sobre a arquitetura do sistema, e de uma avaliação do risco. Durante a fase de elaboração, o entendimento da equipe do projeto de requisitos de negócio do cliente é verificado para reduzir o risco do desenvolvimento;

Construção (*Construction*) - O objetivo da fase de Construção é construir a solução de modelos de requisitos detalhados, por meio da configuração de pacotes de funcionalidade de software padrão, desenvolvimento e teste de componentes customizados e integração de um sistema que está pronto para uma primeira versão que irá para a produção. Em suma, para concluir o desenvolvimento da aplicação, deve-se certificar que todos os componentes se encaixam e preparar o sistema para o teste de aceitação e implantação. A aplicação deve ser concluída dentro de um número pré-definido de iterações. As atualizações são feitas a cada um dos modelos (Modelo de Caso de Uso, Modelo de *Design*, Implementação Arquitetural, etc), enquanto os requisitos são progressivamente refinados. Quando todas as iterações previstas foram estabelecidas para cada partição, a aplicação completa é testada.

Transição (*Transition*) - O objetivo da fase de Transição é instalar a solução para o sistema de produção, realizar testes de aceitação e iniciar a vida da aplicação. Durante esta fase, o novo sistema é aceito pela organização, a organização está preparada para o novo sistema e o sistema é colocado em produção. Se o novo sistema substitui um velho, uma transição suave para o novo aplicativo deve ser fornecida. A fase de Transição pode atravessar várias iterações e inclui testar o sistema em preparação para *release* e pequenos ajustes com base no *feedback* do usuário.

Produção (*Production*) - A meta da fase de produção é operar o sistema recém-desenvolvido, avaliar seu sucesso e apoiar os usuários. Isto inclui: o monitoramento do sistema atuando de forma apropriada para permitir operação contínua, o desempenho do sistema de medição, operação e manutenção de sistemas de apoio; responder a pedidos de ajuda, relatórios de erros e solicitações de recursos por utilizadores e a gestão do processo de controle de mudança aplicável, de modo que os defeitos e novos recursos sejam priorizados e atribuídos a futuros lançamentos.

Todas as tarefas da OUM também são organizadas em processos que agrupam tarefas relacionadas. Membros da equipe do projeto são atribuídos a esses grupos, de acordo com sua especialização e experiência. OUM inclui os processos de Requisitos de Negócio, Análise de Requisitos, Análise, Projeto, Implementação, Testes, Gestão de Desempenho, Arquitetura Técnica, Aquisição e Conversão de Dados, Documentação, Gestão de Mudança Organizacional, Treinamento, Transição, Operação e Suporte.

É metodologia completa, complexa e totalmente orientada a construção de soluções em plataforma Oracle.

2.4 OUTRAS METODOLOGIAS

As demais metodologias analisadas serão descritas de forma mais superficial, não porque tenham uma menor importância, mas pelo fato de que suas descrições detalhadas seriam muito repetitivas vis-à-vis as já descritas. Desta forma, são destacados apenas alguns de seus diferenciais ou suas definições mais estratégicas.

2.4.1 Processo CBDI-SAE

Processo CBDI-SAE: (ALLEN, 2007; LARRUCEA, 2008; TANG et al., 2008) O Fórum CBDI desenvolveu uma metodologia SOA como parte de sua CBDI-SAE (*Component-based Development and Integration - Service Architecture and Engineering*) *Framework* de Referência (RF - *Reference Framework*). As quatro áreas-chave de disciplina do processo são: consumir, fornecer, gerenciar e ativar. Cada grupo de área de disciplinas similares é subdividido em unidades de processo e, em seguida, subdivisão é feita para as tarefas. Esta metodologia tem como objetivo a integração de negócios à TI por meio da análise *top-down* de requisitos de negócios, bem como *bottom-up* de integração do sistema legado. O processo CBDI-SAE pretende cobrir todo o ciclo de vida SOA, incluindo atividades de implantação, monitoramento e governança.

2.4.2 Service Oriented Architecture Framework (SOAF)

Service Oriented Architecture Framework (SOAF): (ERRADI et al., 2006; BOUKADI, 2010) SOAF consiste em cinco fases principais: elicitação de informações, identificação de serviços, definição de serviço, realização de serviços e elaboração de *roadmaps* e

planejamento. Utiliza, simultaneamente, dois tipos de análise de modelagem: modelagem "To-Be", que é a abordagem *top-down* orientada para negócios descrevendo os processos de negócios necessários, e a modelagem "AS-IS", que é a abordagem *bottom-up* descrevendo processos de negócios como eles são moldados pelas aplicações existentes.

2.4.3 Service Oriented Unified Process (SOUP)

Service Oriented Unified Process (SOUP): (MITTAL, 2006) Como o nome sugere, esta abordagem por K. Mittal baseia-se principalmente no Rational Unified Process. Seu ciclo de vida é composto por seis fases: Iniciação, definir, projetar, construir, implantar e apoio. No entanto, a SOUP não tem documentação detalhada e deixa espaço para a adaptação. Ele é usado em duas variações ligeiramente diferentes: uma adotando RUP para projetos iniciais de SOA e outra adotando uma mistura de RUP e XP para a manutenção de implementações SOA existentes (KIM, 2006).

2.4.4 Metodologia proposta por Papazoglou

Metodologia proposta por Papazoglou: (PAPAZOGLU; HEUVEL, 2006) Propõe examinar uma metodologia de desenvolvimento de serviço do ponto de vista dos fornecedores e consumidores, que tenta cobrir todo o ciclo de vida SOA. Ela é parcialmente baseada em modelos de desenvolvimento bem estabelecidas, como o RUP, CBD e BPM. A metodologia utiliza um processo iterativo e incremental que compreende uma preparação e oito fases principais distintas: Planejamento, Análise, Projeto de Serviço, Construção de Serviço, Teste de Serviço, Provisionamento de Serviço, Implantação de Serviço e Execução de Serviço.

2.4.5 Metodologia proposta por Thomas Erl

Metodologia proposta por Thomas Erl: (ERL, 2008) A análise orientada a serviços e metodologia de projeto documentado no livro de Thomas Erl é considerada a primeira a ser publicada que é agnóstica com relação ao fornecedor. Esta metodologia é um guia passo a passo através das duas fases principais: análise e projeto. As atividades na fase de análise tem uma visão de negócios *top-down* onde os candidatos a serviço são identificados. Estes servem como entrada para a próxima fase, projeto orientado a serviços, onde os candidatos a serviço são especificados em pormenor e depois disponibilizados como *Web Services*.

2.5 RESUMO COMPARATIVO ENTRE AS METODOLOGIAS

A Tabela 2-1 a seguir apresenta, de forma resumida, as principais características das metodologias analisadas.

Tabela 2-1 – Análise Metodologias

Item Avaliado	Metodologias							
	SOAD ⁱ	SOMA ⁱⁱ	OUM ⁱⁱⁱ	CBDI-SAE ^{iv}	SOAF ^v	SOUP ^{vi}	Papazoglou ^{vii}	Thomas Erl ^{viii}
Estratégia de entrega	M	M	M	M	M	M	M	T
Cobertura ciclo de vida	A&P	Completa	Completa	Completa	A&P	Completa	Completa	A&P&I
Proprietária	Sim	Sim	Sim	Não	Não	Não	Não	Não
Ágil	n/a	3	4	2	2	5	3	1
Processo existente	Não	RUP	RUP	?	Não	RUP, XP	RUP	Não
Técnicas existentes	OOAD, BPM	?	?	?	Não	Não	CBD, BPM	BPM
UML	Sim	?	Sim	?	?	Não	Não	Não
Visão do consumidor	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Visão do provedor	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Prevê terceirização	Não	Não	Não	Não	Não	Não	Não	Não
Define estrutura funcional	Não	Parcial	Parcial	Parcial	Não	Parcial	Parcial	Parcial
Define perfis e papéis	Não	Parcial	Parcial	Parcial	Não	Parcial	Parcial	Parcial

Observações:

- 1- As análises das ferramentas até o item “Visão do provedor”, foram retiradas de Ramollari (2007), com exceção da metodologia OUM;
- 2- Para o item “Estratégia de entrega” → M = *Meet-in-the-middle*, T = *Top-down*, B = *Bottom-up* e ? = sem informação;
- 3- Para o item “Cobertura ciclo de vida” → A = Análise, P = Projeto e I = Implementação;
- 4- Para o item “Ágil” a escala utilizada foi de 1 a 5, sendo 1 a menos ágil.

i - (ZIMMERMANN, 2004; CHANG, 2007 (2)); ii - (ARSANJANI, 2004; ARSANJANI, 2008); iii - (ORACLE, 2012); iv - (ALLEN, 2007; LARRUCEA, 2008; TANG et al., 2008); v - (ERRADI et al., 2006; BOUKADI, 2010); vi - (MITTAL, 2006); vii - (PAPAZOGLU ET AL., 2006); viii - (ERL, 2008).

3 - ESTRUTURA DA METODOLOGIA

Para estruturar a metodologia proposta e torna-la flexível para a terceirização de sua execução, seja total ou parcialmente, foi necessário definir de forma clara as fronteiras de responsabilidade e partição de cada ator no processo de desenvolvimento. Para isso foram criados dois conceitos estruturantes, o de unidades funcionais e o de perfis profissionais.

As unidades funcionais têm como objetivo agrupar tarefas e atividades afins, de modo a definir estruturas e seu papel no processo de desenvolvimento. Estes macropapéis definem seu escopo de atuação na metodologia proposta, facilitando desta maneira sua terceirização.

Com a constatada ausência de profissionais especializados em SOA no mercado nacional, os perfis profissionais foram criados para diferentes propósitos, seja para facilitar a especificação do profissional a ser contratado, seja para a elaboração de um plano de formação e capacitação técnica. Além destes, e talvez o mais importante propósito, seja definir e explicitar a responsabilidade de cada um, seu papel específico na metodologia e as capacidades técnicas necessárias para a execução das tarefas associadas.

Com estes dois conceitos estruturados, o terceiro pilar desta metodologia foi elaborado, o ciclo de vida de desenvolvimento de software, definindo seus processos, as respectivas tarefas que deverão ser executadas e os artefatos de entrada e saída de cada etapa que deverão ser criados ao longo de todo o processo de desenvolvimento de soluções orientadas a serviço.

Estas definições permitirão não apenas a terceirização das unidades funcionais, mas também o acompanhamento da execução de seus contratos, através da delimitação clara das obrigações e atividades da contratada, bem como do controle e aceitação dos entregáveis previstos.

3.1 UNIDADES FUNCIONAIS

Essa estrutura é composta por unidades funcionais que representam agrupamentos lógicos de atividades e responsabilidades dentro da Metodologia (funções). Essa estrutura não reflete, necessariamente, a estrutura organizacional formal vigente.

Unidades funcionais são unidades organizacionais lógicas que fornecem força de trabalho com competências profissionais adequadas necessárias para executar atividades técnicas relacionadas. Atividades técnicas são agrupadas em *unidades de trabalho*, que geralmente correspondem a funções claramente definidas no *ciclo de vida de desenvolvimento de software*. Cada unidade de trabalho produz entregáveis formalmente definidos (artefatos de software). Competências profissionais são definidas por *perfis profissionais*, que estabelecem um tipo de profissional que é capaz de executar um conjunto selecionado de atividades técnicas. Unidades funcionais são classificadas como *unidades corporativas e unidades dedicadas*, de acordo com o tipo de competência (jurisdição) do trabalho a ser realizado.

Unidades corporativas executam atividades e responsabilidades de nível corporativo (ou departamental), que não são exclusivas para o processo de desenvolvimento de software orientado a serviços. Estas unidades podem estar previamente estabelecidas na organização.

Desenvolvimento SOA requer diversas interações com estruturas de nível corporativo, ou seja, que executam tarefas e têm responsabilidades que não são exclusivamente relacionadas com projetos SOA. Unidades corporativas são usadas para modelar estas estruturas e interações. Na maioria das organizações, deve haver estruturas organizacionais semelhantes que têm responsabilidades equivalentes. Note que, com exceção da unidade Escritório de Gestão de Projetos, todas as outras unidades corporativas têm responsabilidades que não são consideradas como serviços terceirizados nesta metodologia. Estratégias de terceirização de serviços executados por unidades corporativas estão fora do escopo deste artigo (consulte a Tabela 3-1 e nota 1).

Unidades dedicadas executam atividades e responsabilidades que são exclusivamente relacionadas com os processos de desenvolvimento de software orientados a serviços. Estas unidades existem apenas para o propósito de desenvolvimento SOA na organização.

Unidade funcional estabelece pontos concretos de gestão para a força de trabalho e unidades de trabalho relacionadas. Portanto, eles são usados para gerenciar e controlar os recursos de terceiros e responsabilidades no processo de desenvolvimento de software, no sentido de que uma unidade funcional completa pode ser fornecida por uma força terceirizada de trabalho. Esta noção também é útil do ponto de vista de aquisição, contratação e gestão de terceirização. Assim, os contratos podem ser estabelecidos com prestadores de serviços para um escopo de serviço de uma unidade funcional inteira (ou seja, um conjunto de unidades de trabalho). Esses contratos são regulados por acordos de nível de serviço, que especificam o perfil profissional exigido em cada tipo de unidade de trabalho, em conjunto com outros atributos de qualidade de serviço específicos para serviços (ex: tempo para concluir, número de erros no software, etc.).

Unidades dedicadas são estruturas importantes na metodologia enquanto definem a organização de um conjunto de recursos humanos que são atribuídos através de contratos de terceirização.

Na metodologia proposta, cada unidade funcional pode ser terceirizada para um prestador de serviços diferente. No entanto, uma opção comum é a de terceirizar as unidades Fábrica de Serviço e de Fábrica de Software para o mesmo prestador de serviço. Esta abordagem geralmente reduz a carga de governança relacionada com o retorno entre diferentes prestadores de serviços. Cada provedor de serviços deve ter um contrato *master* com a organização do cliente, que especifica termos e condições gerais, especificação de serviços e SLAs de acordo com o escopo dos serviços terceirizados. Esses contratos são complementados por projetos específicos de subcontratos, que são especificados em artefatos chamados Ordens de Serviço, que estabelecem termos reais do projeto, incluindo seu escopo, programação, estimativas e medições, e define as unidades de trabalho necessárias a cada projeto orientado a serviços específico.

A maioria do trabalho técnico realizado pelas unidades dedicadas está organizada com um modelo de serviço de fábrica (HOTLE e LANDRY, 2009; CLEMENTS e NORTHROP, 2001; BROWN et al., 2011). As unidades de fábrica são concebidas como uma evolução dos métodos e técnicas tradicionais de fábrica de software, utilizando técnicas e princípios de fabricação para oferecer um serviço de desenvolvimento de software orientado a serviços.

Terceirização é usada para unidades de trabalho claramente definidas que exigem uma formação técnica, mas não representam o conhecimento estratégico para a organização. Nesse sentido, nem todas as unidades funcionais são consideradas como objetivo para a terceirização, como todas as atividades estratégicas e muitas de gestão são geralmente atribuídas a profissionais com forte comprometimento com a organização (empregados, por exemplo). Embora as unidades funcionais possam ser comparadas com as unidades funcionais da organização, de fato não existem na realidade como parte da estrutura organizacional formal. A Tabela 3-1 – Lista de Unidades Funcionais apresenta as unidades funcionais definidas nesta metodologia, sua classificação e elegibilidade para a terceirização.

Tabela 3-1 – Lista de Unidades Funcionais

Unidade Funcional	Tipo	Terceirização¹
Gestão Executiva do Programa de Adoção SOA	Corporativa	Não
Escritório de Gestão de Processos de Negócio	Corporativa	Não
Escritório de Administração de Dados	Corporativa	Não
Escritório de Infraestrutura de TI	Corporativa	Não
Escritório de Arquitetura Corporativa	Corporativa	Não
Escritório de Gestão de Projetos	Corporativa	Sim
Centro de Excelência SOA	Dedicada	Sim
Fábrica de Serviços	Dedicada	Sim
Fábrica de Software	Dedicada	Sim
Fábrica de Testes de Software	Dedicada	Sim

3.2 CICLO DE VIDA DE DESENVOLVIMENTO DE SOFTWARE

O paradigma de desenvolvimento de software orientado a serviço requer tarefas específicas. Estas são definidas em um ciclo de vida de desenvolvimento de software orientado a serviço. Nessa metodologia, o ciclo de vida é definido de forma a favorecer a divisão do trabalho técnico exigido no processo entre as diferentes unidades funcionais que atuam como prestadores de serviços. O trabalho é dividido em fases, que define as etapas no processo de desenvolvimento com entregáveis bem definidos. Cada fase é ainda

¹ Este é o escopo de terceirização considerado nessa metodologia. Aplicabilidade do modelo de terceirização para unidades funcionais marcadas como “não” não são discutidas nessa dissertação.

classificada de acordo com a disciplina de desenvolvimento de software envolvida no trabalho que está sendo realizado.

A unidade de trabalho realizada em cada fase é definida, em processos formais, que podem ser adicionalmente decompostos em subprocessos e tarefas. A

Tabela 3-2 - Ciclo de Vida de Desenvolvimento de Software apresenta a visão geral das fases do ciclo de vida de desenvolvimento de software, juntamente com a respectiva disciplina de desenvolvimento de software e subprocessos/tarefas.

A fim de assegurar a visibilidade dos processos e a correta interação entre os prestadores de serviços, cada processo do ciclo de vida tem um conjunto claramente definido de entradas e saídas. Esses são classificados como artefatos de software (SA), artefatos de processo de desenvolvimento (DPA), artefatos de gerenciamento de projetos (PMA) ou artefatos corporativos (CA).

Tabela 3-2 - Ciclo de Vida de Desenvolvimento de Software

Fase	Disciplina	Processos
Modelagem de Negócio Corporativo	Modelagem de Negócio	Modelar Negócio Corporativo Modelar Processos Empresariais Modelar Processos de Negócio – 1ª Visão Modelar Processos de Negócio – 2ª Visão Modelar Estrutura da Informação
Análise de Inventário de Serviço	Análise	Analisar Inventário de Serviços Executar Análise Orientada a Serviço Analisar Modelos Corporativos de Negócio Definir Arquitetura Tecnológica Modelar Serviços Candidatos
Análise Orientada a Serviço	Análise	Analisar com Orientação a Serviço Definir Escopo da Análise Identificar Sistemas Existentes Identificar Entidades de Negócio e Fluxos de Informação Modelar Serviços Candidatos
Projeto Orientado a Serviço	Projeto	Projetar Serviço de Entidade Projetar Serviço Utilitário Projetar Serviço de Tarefa Projetar Serviço de Tarefa Orquestrada
Projeto de Lógica de Serviço	Projeto	Projetar Lógica de Serviço Decompor Lógica de Serviço Projetar Lógica Principal de Serviço Projetar Lógica de Composição e Orq. Projetar Lógica de Barramento Projetar Lógica de Pré e Pós Processam. Projetar Lógica de Monit. de Processos
Desenvolvimento de Serviço	Implementação	Implementar Serviço
Teste de Serviço	Teste	Testar Serviço

Disponibilização e Publicação de Serviço	Disponibilização	Homologar Serviço Publicar Serviço Catalogar Serviço
Implementação de Aplicação Composta	Implementação	Implementar Aplicação Composta
Teste de Aplicação Composta	Teste	Testar Solução Homologar Solução
Disponibilização de Aplicação Composta	Disponibilização	Implantar Solução

3.3 PERFIS PROFISSIONAIS

A metodologia não define *atores* (i.e. recursos humanos) específicos para realização ou gerenciamento de cada tarefa ou processo. A atribuição de recursos humanos específicos deve ser realizada projeto a projeto, como parte do planejamento e constituição/alocação da *equipe de projeto*. Entretanto, a metodologia define *perfis profissionais* associados a uma unidade de trabalho definida. Cada *perfil profissional* definido na metodologia está associado a um conjunto de *competências profissionais*, que indicam as capacidades e habilidades recomendadas e desejáveis para o profissional ao qual o perfil é atribuído. Essas definições são formalizadas a seguir.

- a. *Ator*: recurso humano que executa determinada tarefa ou processo. Atores são instâncias de um perfil profissional requerido em determinada parte do processo.
- b. *Equipe de Projeto*: conjunto de atores que atuam em um determinado projeto.
- c. *Perfil Profissional*: define um tipo de profissional, com suas respectivas *competências profissionais* associadas a um conjunto de atividades de projeto atribuídas a *atores*.
- d. *Competência Profissional*: Define um conjunto de habilidades e capacidades que serão demandas em um determinado perfil. Essas podem ser *recomendadas* (capacidades normalmente requeridas para o desempenho da atividade profissional associada ao perfil) ou *desejáveis* (capacidades complementares que agregam valor à atividade profissional do perfil).

A Tabela 3-3 – Perfis Profissionais apresenta a lista de *perfis profissionais* referenciados na metodologia, com suas respectivas *competências profissionais*, as quais deverão ser objeto de capacitação profissional continuada da equipe.

Tabela 3-3 – Perfis Profissionais

Perfil Profissional	Competência Profissional
Administrador de Dados	Especialista em administração de dados
Administrador de Infraestrutura SOA	Especialista em <i>middleware</i> e infraestrutura SOA
Analista de Negócio	Profissional de Gestão de Processos de Negócio Especialista BPMS (desejável)
Analista de Processo	Profissional especializado em modelagem de processos Especialista BPMS
Analista SOA	Profissional Analista especializado em SOA
Analista de Testes	Profissional Analista especializado em testes Conhecimento em SOA (desejável)
Arquiteto Corporativo	Especialista em Arquitetura Corporativa
Arquiteto SOA	Profissional especializado em arquitetura SOA Especialista de Infraestrutura SOA
Auditor SOA	Especialista/consultor sênior em SOA Formação de Arquiteto SOA (desejável)
Desenvolvedor	Desenvolvedor/Programador (dependente de plataforma) Especialização em SOA (desejável)
Especialista em Governança SOA	Profissional especialista em Governança SOA
Gerente de Projeto	Profissional de Gerência de Projetos
Testador	Testador de software Especialização em SOA (desejável)

3.4 VISÃO GERAL DA METODOLOGIA

Para um melhor entendimento de como surgiu o modelo proposto, principalmente os três pilares desta metodologia, as Unidades Funcionais, os Perfis Profissionais e o Ciclo de Vida de Desenvolvimento, cabe uma breve descrição de como os mesmos surgiram ao longo de sua elaboração.

Na Agência Nacional de Vigilância Sanitária (ANVISA), em seu projeto de automação de processos utilizando alguns princípios de SOA, não havia nenhuma metodologia de desenvolvimento apoiando a iniciativa. As dificuldades enfrentadas foram muitas mas, o principal problema enfrentado foi quando a empresa terceirizada responsável pelo papel de fabrica de software, assumiu a tarefa do desenvolvimento dos web services necessários para a implementação dos processos.

Ficou latente que, além da falta de experiência e conhecimento técnico da empresa nos conceitos SOA, a ausência de definições claras e precisas das responsabilidades dos atores envolvidos no processo de desenvolvimento, da empresa e também de setores da ANVISA, inviabilizou uma comunicação efetiva entre as partes, causando hiatos de atuação, dificuldade de monitoramento e gestão das ações e, até, questões contratuais de mensuração e pagamento dos serviços efetuados.

Somada a esta experiência da ANVISA se juntou a da Agência Nacional de Águas (ANA), onde havia mais de uma empresa terceirizada no processo e profissionais distribuídos entre diversas áreas de atuação.

Estas duas experiências demonstraram claramente a necessidade da presença de uma metodologia que gerisse todo este contexto e, também, da existência de definições claras de papéis e responsabilidade dentro do processo de desenvolvimento SOA. Maiores detalhes da experiência vivida nos estudos de caso nos quais a metodologia foi concebida podem ser conhecidas no Capítulo 7 - IMPLEMENTAÇÃO E APLICAÇÃO DA METODOLOGIA.

Neste momento foi iniciada a pesquisa no mercado das metodologias já existentes, que foi detalhada no Capítulo 2 - ANÁLISE DAS METODOLOGIAS EXISTENTES, onde surgiram as dificuldades já mencionadas como a dificuldade de se obter informações detalhadas das metodologias de fornecedores de soluções SOA e, principalmente, da não

consideração da participação de empresas terceirizadas no processo e da não definição precisa dos papéis e responsabilidades dos atores envolvidos.

Estes fatores deixaram clara a impossibilidade de se utilizar ou mesmo adaptar alguma das metodologias existentes e a necessidade de se criar uma nova, atendendo aos requisitos levantados e criando a conceituação das Unidades Funcionais e os Perfis Profissionais, identificando claramente os papéis e responsabilidades distribuídos pelas unidades funcionais e, dentro delas, os profissionais que deveriam executar as atividades, apontando seu perfil, suas habilidades e capacidades profissionais.

Desta forma, a metodologia proposta apresenta exatamente o cruzamento entre os processos que compõem o ciclo de vida do desenvolvimento para aplicações baseadas em orientação a serviço, as unidades funcionais responsáveis pela execução de cada fase e, os profissionais associados necessários a conclusão das atividades. Este cruzamento é representado pela Figura 3-1, a seguir:

A Figura 3-1 apresenta a visão geral da metodologia, identificando as suas etapas, perfis principais e respectivas unidades funcionais associadas.

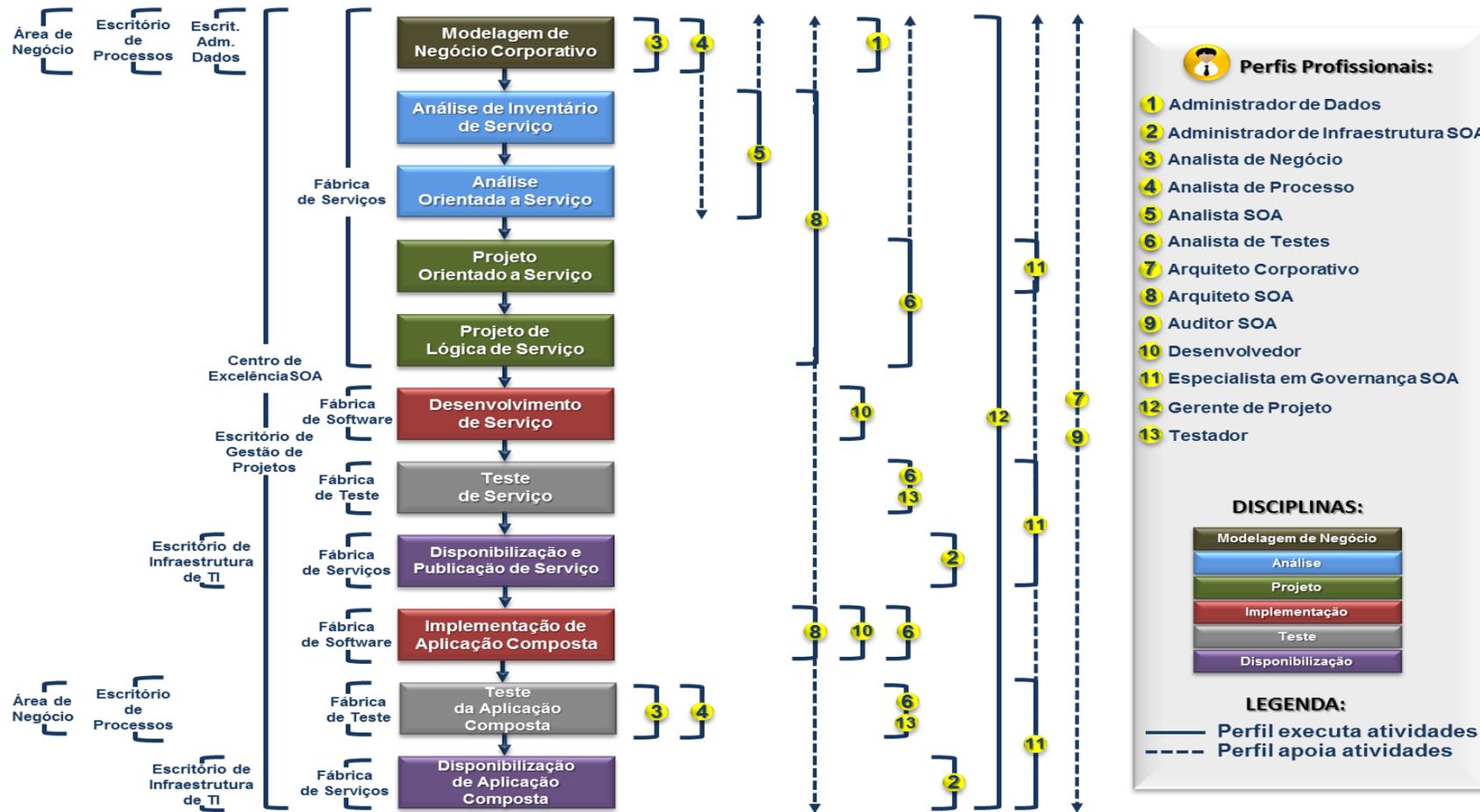


Figura 3-1 – Visão Geral da Metodologia SOA

4 - ESTRUTURA FUNCIONAL

Para o contexto da Metodologia de Desenvolvimento SOA, define-se uma estrutura funcional e suas respectivas unidades funcionais com objetivo de identificar funções típicas do processo de desenvolvimento de soluções orientadas a serviço. É importante compreender que essas estruturas não refletem, obrigatoriamente, a estrutura organizacional formal vigente. Desse modo, a “Estrutura Funcional” representa, na realidade, um agrupamento lógico de atividades e responsabilidades dentro da Metodologia.

Algumas dessas estruturas podem ter seus papéis executados por equipes terceirizadas. A Metodologia proposta é flexível o suficiente para incorporar essa realidade de mercado. Mais especificamente, a estrutura pode ser usada para auxiliar o processo de contratação das equipes terceirizadas.

A estrutura funcional definida é ilustrada na Figura 4-1 e é composta por 10 (dez) Unidades Funcionais, assim distribuídas:

- **Gestão Executiva do Programa de Adoção SOA:** Unidade gestora da iniciativa SOA, responsável pela coordenação e acompanhamento de todo o projeto.
- **Unidades Corporativas:** Grupo de unidades representado pela cor azul na Figura 4-1, atuam não somente nos projetos SOA, mas, em toda estrutura corporativa de TI. Possuem atuação gerencial e normativa no processo SOA, com exceção da Unidade Gestão de Processos, que pode atuar tanto na gerência quanto na execução de tarefas operacionais ligadas ao processo SOA.
- **Unidades Dedicadas:** Grupo de unidades representadas em cor verde na Figura 4-1, juntamente com o Centro de Excelência SOA. São aquelas que de fato terão participação dedicada na execução dos processos de desenvolvimento SOA.

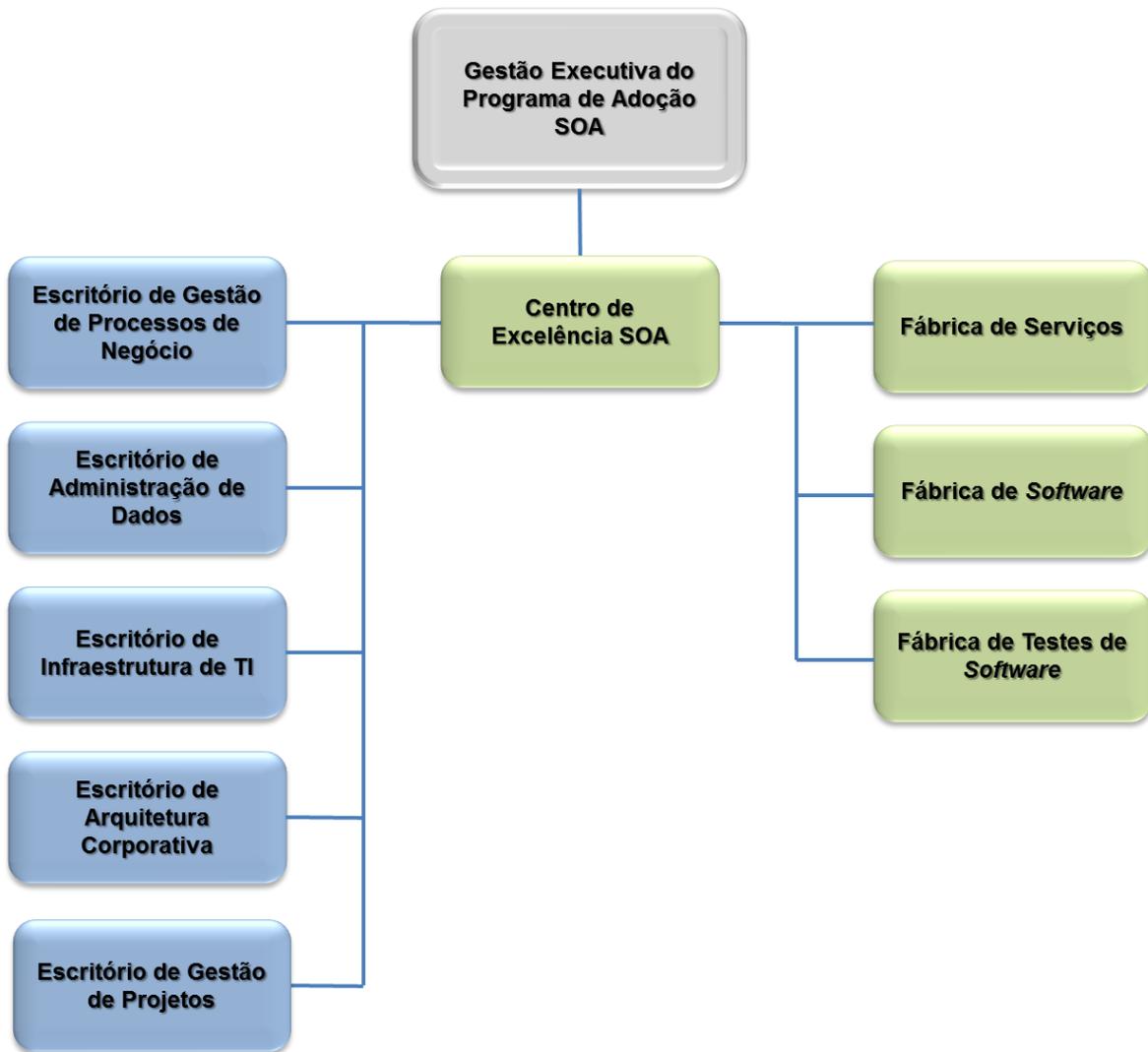


Figura 4-1 – Estrutura Funcional

A seguir são descritas as principais características das unidades funcionais e os principais processos executados sob a responsabilidade de cada unidade, no contexto do desenvolvimento da SOA.

Por convenção, nas seções seguintes, os processos anotados na cor azul indicam responsabilidades diretamente relacionadas com o desenvolvimento SOA, enquanto os processos anotados em verde indicam processos que possuem influência indireta na metodologia.

O detalhamento de processos com influência indireta na metodologia não está faz parte do escopo dessa dissertação. Entretanto, um pequeno descritivo de cada um dos processos

elencados é apresentado nesta seção. No que se refere a processos diretamente relacionados à metodologia, estes serão detalhados, conforme a necessidade, na Seção 6.

4.1 UNIDADES CORPORATIVAS

4.1.1 Escritório de Gestão de Processos de Negócio

A unidade funcional Escritório de Gestão de Processos de Negócio, também referenciada como BPMO (*Business Process Management Office*) tem como responsabilidade promover a melhoria continuada dos processos de negócio da organização. Suas responsabilidades compreendem o projeto, modelagem e otimização dos processos de negócio, visando trazer efetividade e eficiência ao negócio através da melhoria da execução e monitoração desses processos.

Essa unidade pode ser centralizada no Departamento de Tecnologia da Informação ou ainda como unidade ligada diretamente ao alto corpo gestor, visando modernização, padronização de processos ou normatização. Outra opção comum consiste na estruturação dessas funções diretamente em cada grande área de negócio da organização (i.e. Secretarias ou Departamentos). Neste caso, é comum se estruturar uma unidade adicional que assume o papel normativo e promotor de padrões e boas práticas, muitas vezes denominada *Escritório de Processos*.

Além da realização de diversas atividades relacionadas à modelagem e projeto de processos de negócio, BPMO normalmente tem um papel normativo, estabelecendo práticas, políticas e normas relacionadas com a gestão de processos de negócios (BPM). Note que existem várias tecnologias de BPM que têm suporte direto em plataformas SOA. Estabelecer um claro alinhamento de práticas e atividades entre BPM e SOA resultam em benefícios diretos para o desenvolvimento de soluções de TI orientadas a serviços (JESTON, NELIS, 2008). Esse alinhamento deve ser estendido, com maiores benefícios, ao nível de plataforma e tecnologia (por exemplo, middleware compatível de BPM e SOA).

A Figura 4-2, a seguir, apresenta os principais processos executados sob a responsabilidade desta unidade que estão relacionados com a metodologia.

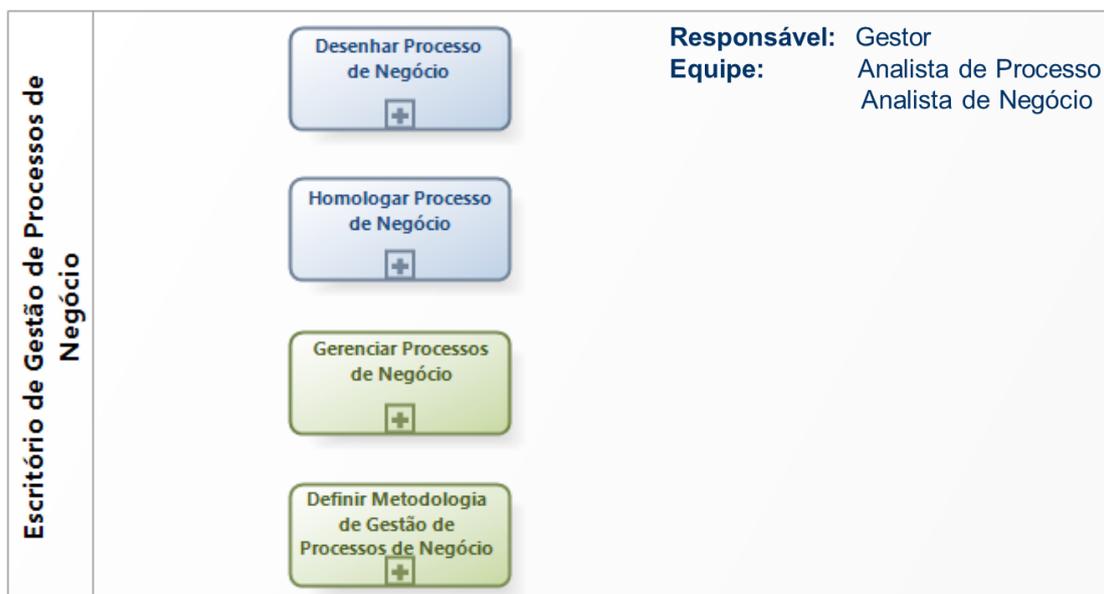


Figura 4-2 – Processos do Escritório de Gestão de Processos de Negócio

- Desenhar Processo de Negócio:** Envolve a criação de especificações para processos de negócio novos ou modificados dentro do contexto dos objetivos de negócio, objetivos de desempenho de processo, fluxo de trabalho, aplicações de negócio, plataformas tecnológicas, recursos de dados, controles financeiros e operacionais, e integração com outros processos internos e externos. Tanto um desenho lógico de (quais atividades são realizadas) como um desenho físico (como atividades são executadas) são incluídos como entregáveis. Esse processo pressupõe a execução de atividades de mapeamento/modelagem e análise de processos de negócio². Essas atividades estão diretamente relacionadas à etapa “Modelagem de Negócio Corporativo”. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.
- Homologar Processos de Negócio:** Refere-se à homologação e validação dos modelos gerados junto às áreas de negócios relacionadas. Essas atividades estão diretamente relacionadas à etapa “Modelagem de Negócio Corporativo”. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.

² Na metodologia de *Business Process Management* (BPM), esse processo é normalmente precedido pelas etapas de modelagem e análise de processos de negócio. Por simplicidade, essas fases são omitidas neste documento, ficando as mesmas subentendidas no desenho de processos de negócio aqui anotado.

- **Gerenciar Processos de Negócio:** Esse processo estabelece a gerência dos processos de negócio, de forma integrada e articulada, em nível corporativo, promovendo a governança de processos e o alinhamento dos processos de negócio à estratégia organizacional³.
- **Definir Metodologia de Gestão de Processos de Negócio:** Atividades relacionadas ao estudo, definição, normatização e uso de *Metodologia de Gestão de Processos de Negócio*.

4.1.2 Escritório de Administração de Dados

Também nominado como DAO (*Data Administration Office*), é responsável por planejar, organizar, documentar, manter, integrar e controlar recursos de dados e da informação corporativa. Entre suas atribuições destacam-se a modelagem lógica de dados, através de um processo de planejamento, coordenação e articulação de dados corporativos transversais a áreas de negócio distintas, e o gerenciamento de repositório de metadados corporativos, que constituem representação física dos modelos de dados corporativos.

Esse papel também pode existir dentro de outras unidades que são responsáveis pela arquitetura da informação corporativa (por exemplo, Escritório de Arquitetura da Informação), que são unidades comuns em estruturas de organizações modernas. SOA alavanca em nível corporativo, dados padronizados e modelos de informação (ERL, 2008; ERL, 2009) e estes são os cruzamentos diretos entre a jurisdição do DAO e requisitos de projetos SOA.

Entre os seus principais processos, ilustra-se, na Figura 4-3, aqueles diretamente relacionados ao desenvolvimento SOA, cuja descrição é brevemente apresentada a seguir.

³ A metodologia BPM define formalmente um ciclo de evolução continuada dos processos de negócio que compreende as etapas de modelagem, análise, desenho, monitoramento (de desempenho) e transformação (melhoria) de processos de negócio.

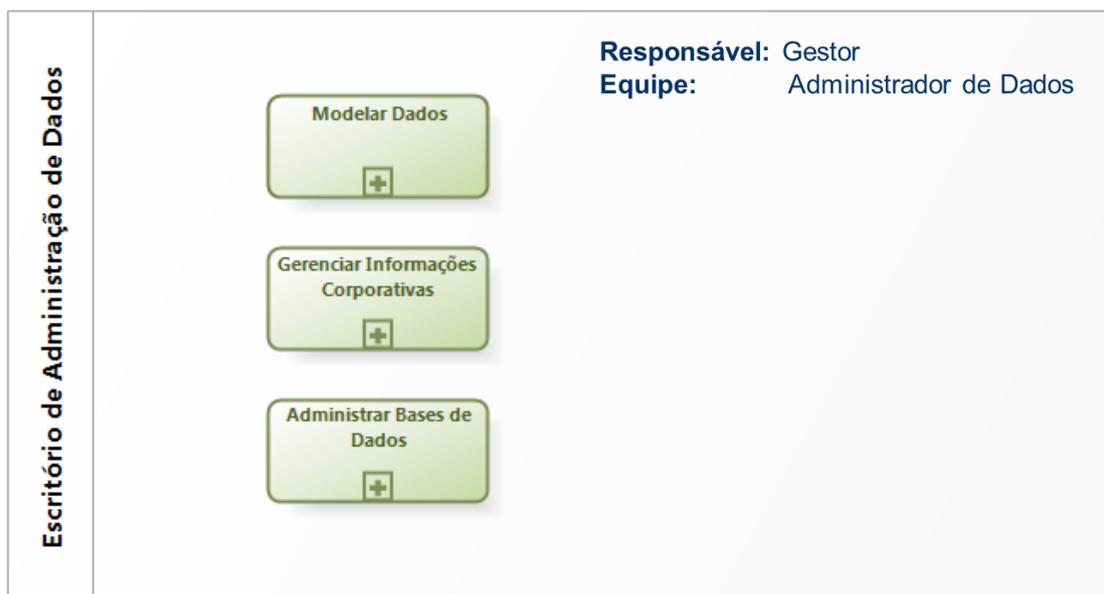


Figura 4-3 – Processos do Escritório de Administração de Dados

- **Modelar Dados:** Consiste na modelagem e homologação de modelos de dados lógicos e físicos, bem como de seus respectivos metadados, em alinhamento com modelos de informação. Utiliza técnicas e ferramental apropriado para criação, documentação e controle desses modelos. Promove e garante ainda que padrões de informação e dados sejam seguidos e que os modelos estejam em conformidade com as regras de negócio corporativas.
- **Gerenciar Informações Corporativas:** Consiste no planejamento, coordenação e gestão de dados e da informação como um recurso corporativo, promovendo e garantindo a segurança (integridade/autenticidade, confidencialidade e disponibilidade), inteligibilidade, acessibilidade e relevância de dados, de maneira transversal em toda organização.
- **Administrar Bases de Dados:** Compreende o planejamento, implantação, otimização (*tuning*), evolução e gerenciamento das bases de dados corporativas e sistemas gerenciadores de bancos de dados associados, em suporte continuado à promoção e garantia do desempenho, acesso e segurança dos dados.

4.1.3 Escritório de Infraestrutura de TI

Conhecida como ITIO (*Information Technology Infrastructure Office*), essa unidade tem entre suas atribuições a disponibilização de recursos de infraestrutura computacional,

necessários à execução das soluções tecnológicas corporativas. Na maioria das vezes, esta unidade também opera os sistemas em produção, incluindo sistemas de informação baseados em SOA. ITIO tem um papel importante nas etapas de implantação do desenvolvimento orientado a serviços.

Os principais processos sob a sua responsabilidade que estão diretamente relacionados ao ciclo de desenvolvimento SOA são ilustrados na Figura 4-4.



Figura 4-4 – Processos do Escritório de Infraestrutura de TI

- Implantar Solução/Serviço:** Atividade de efetivar a disponibilização de uma solução ou serviço em ambiente de produção ou homologação, providenciando a configuração e o ajuste do ambiente para atender aos requisitos de uso da solução/serviço. Essas atividades estão diretamente relacionadas às etapas “Disponibilização e Publicação de Serviço” e “Disponibilização de Aplicação Composta” da metodologia. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.
- Disponibilizar Infraestrutura:** Consiste no processo de disponibilização dos recursos de infraestrutura para cada fase do processo de automação. Envolve a gestão de ambientes de desenvolvimento, testes, homologação, treinamento e de produção, com capacidades e configurações adequadas e em conformidade com a arquitetura tecnológica corporativa.

- **Administrar Ambiente/Serviço/Solução:** Engloba a administração do ambiente (i.e. infraestrutura tecnológica), bem como a configuração dos serviços e soluções que nele residem, visando garantir um nível de serviço adequado.

4.1.4 Escritório de Arquitetura Corporativa

Arquitetura Corporativa, ou Empresarial (EA), é o processo de traduzir a visão e estratégia de negócios em efetiva mudança na empresa através da criação, comunicação e melhoria dos requisitos chaves, princípios e modelos operacionais que descrevem o estado atual e futuro da empresa, permitindo sua evolução (Gartner Group). Até à data atual, existem muitos modelos de EA. No framework TOGAF, a EA abrange negócio, informação, arquiteturas de aplicações e tecnológicas e destaca os relacionamentos e dependências entre eles (OPEN GROUP, 2011).

A unidade EAO (*Enterprise Architecture Office*) é responsável pela concepção, desenvolvimento, criação, evolução e gestão da EA. Este papel tem uma relação mais direta com arquiteturas específicas de TI (tais como SOA), que devem ser mantidos em sincronia com os objetivos de negócio e requisitos. Portanto, o processo EA supervisiona e limita todas as outras arquiteturas, incluindo aquelas relacionadas a soluções de software (isto é, SOA). Modelos SOA são geralmente usados para descrever arquiteturas de software e, por conseguinte, SOA foi posicionado no interior do quadrante arquitetura de aplicação. No entanto, tem implicações diretas e dependências com todas as outras arquiteturas no âmbito da EA.

Embora não haja um processo diretamente relacionado com desenvolvimento orientado a serviços que seja executado com a participação direta da EAO, arquitetos corporativos supervisionam o desenvolvimento de negócios específicos, dados, informações e arquiteturas tecnológicas relacionadas a soluções orientadas a serviços, que devem ser apoiadas e delimitadas pela arquitetura corporativa (OASIS, 2006).

Diversos modelos, metodologias, processos e práticas de arquitetura empresarial existem atualmente. Dentre esses, o modelo TOGAF® (*The Open Group Architecture Framework*), padrão desenvolvido pelo grupo de trabalho de arquitetura do Open Group, tem recebido relevante visibilidade e experimentação. Esse modelo define quatro tipos de arquitetura que compõem uma arquitetura empresarial, conforme ilustrado na Figura 4-5:

- **Arquitetura de Negócio:** descreve os modelos de negócio corporativo, sua estratégia, governança, organização e processos de negócio chaves.
- **Arquitetura de Dados (Informação):** descreve os modelos lógicos de dados (e informação) corporativos, mapeando-os a modelos físicos (i.e. bases de dados) e seus recursos de gerenciamento.
- **Arquitetura de Aplicações (Soluções):** descreve as aplicações e sistemas de informação que suportam o negócio corporativo, bem como sua estrutura, definições de serviços, funcionalidades, componentes e suas relações com os processos de negócio.
- **Arquitetura tecnológica:** descreve os componentes da infraestrutura tecnológica da organização, evidenciando recursos de hardware, software e redes de comunicação corporativos.

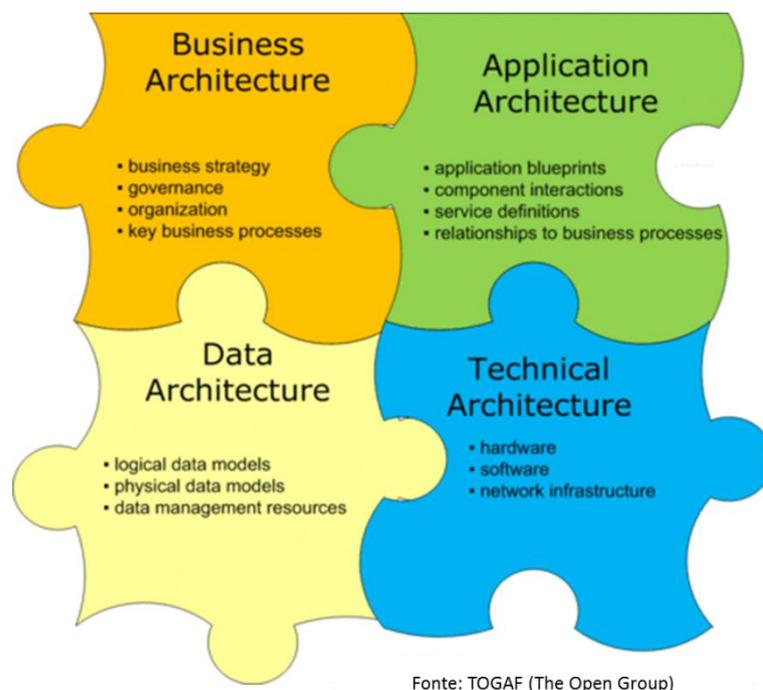


Figura 4-5 – Tipos de arquitetura que compõem uma Arquitetura Empresarial (conforme o modelo TOGAF® (OPEN GROUP, 2011))

A unidade funcional Escritório de Arquitetura Corporativa tem como principal responsabilidade projetar, desenvolver, estabelecer e evoluir a Arquitetura Corporativa, além de assegurar que os diversos modelos arquiteturais de TI, estejam em alinhamento com as necessidades de negócio.

Nesse sentido, a gestão da arquitetura corporativa pode ser entendida como um processo *master* que influencia e supervisiona todos os demais processos de gestão de arquiteturas, incluindo aquelas que utilizam orientação a serviço. Em especial, a arquitetura corporativa tem especializações importantes relacionadas ao desenvolvimento SOA que indicam modelos arquiteturais corporativos para as soluções orientadas a serviço (arquitetura de aplicações), bem como para a arquitetura tecnológica de suporte (plataformas tecnológicas de *middleware*).

Entre os processos e atividades desta unidade, são apresentados na Figura 4-6 aqueles que têm influência no desenvolvimento de SOA. Tais processos são brevemente descritos a seguir.

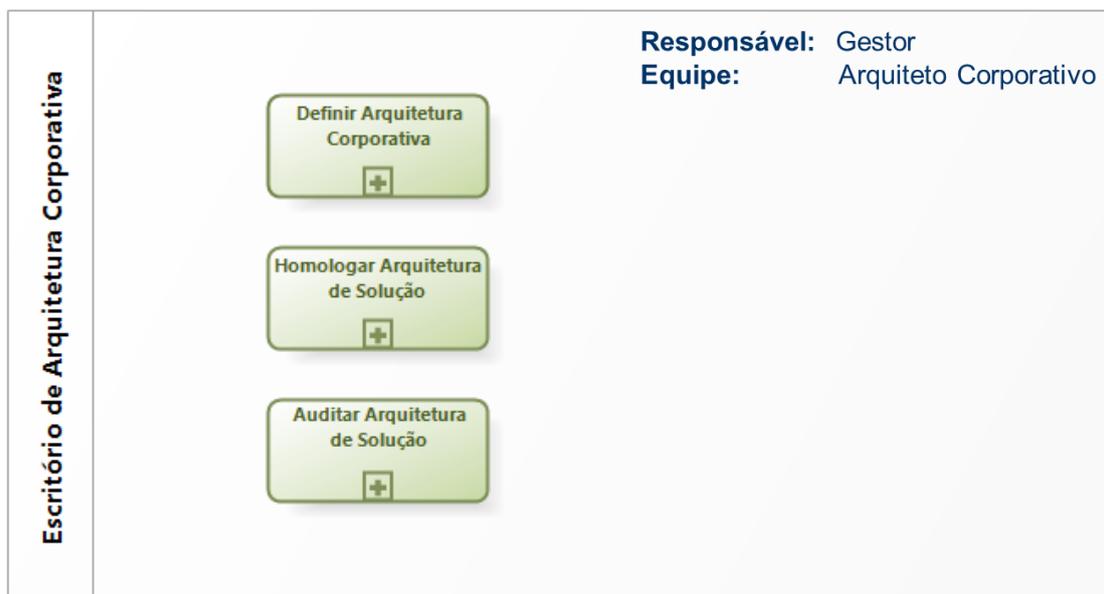


Figura 4-6 – Processos do Escritório de Arquitetura Corporativa

- **Definir Arquitetura Corporativa:** Consiste na definição e evolução de toda a Arquitetura Corporativa. Inclui tanto a definição de arquitetura corporativa como visão (i.e. cenário projetado futuro) quanto a modelagem e documentação da arquitetura corporativa em seu estado atual.
- **Homologar Arquitetura de Solução:** Compreende a verificação da arquitetura projetada de uma solução com vistas a assegurar a conformidade e aderência desta com as diretrizes da arquitetura corporativa.
- **Auditar Arquitetura de Solução:** Consiste na verificação da arquitetura de soluções existentes ou em projeto, visando identificar fatores de conformidade e

inconformidade com as diretrizes da arquitetura corporativa, indicando ainda adequações necessárias para corrigir eventuais distorções.

4.1.5 Escritório de Gestão de Projetos

Esta unidade, também referenciada como PMO (*Project Management Office*), é uma estrutura comum em TI corporativa, como a maioria das atividades de desenvolvimento é organizada como projetos. PMO é responsável pelas metodologias de gerenciamento de projetos e melhores práticas (PMI, 2008). Também gerencia o portfólio de projetos da organização e, muitas vezes, abriga o “*pool*” de gerentes de projeto. PMO supervisiona os processos de gerenciamento de projetos, que é transversal a todas as outras atividades técnicas na metodologia de desenvolvimento orientada a serviços.

Normalmente, já existe uma estrutura nas empresas, que exerce essa atividade, inclusive com metodologia específica de gerenciamento de projetos definida, comumente baseada no *Project Management Body of Knowledge* (PMBOK®).

Os processos sob a responsabilidade da Gestão de Projetos são ilustrados na Figura 4-7.



Figura 4-7 – Processos do Escritório de Gestão de Projetos

- **Gerenciar Projetos:** Este *macroprocesso* inclui processos e atividades de gerenciamento de projetos, seguindo a metodologia definida. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.

- **Definir Metodologia de Gerenciamento de Projetos:** Consiste das atividades relacionadas à definição e atualização da metodologia de gerenciamento de projetos.

4.2 UNIDADES DEDICADAS

4.2.1 Centro de Excelência SOA

SOA *Center of Excellence* (SOA CoE), tem por objetivo agrupar os especialistas em arquitetura orientada a serviços e, promover e executar a governança de SOA (HOLLEY, 2006; WALKER, 2007). Desse modo, o gerenciamento, controle, a garantia de qualidade e auditoria de todo o processo SOA, é de sua responsabilidade, além de projetar e manter atualizada a arquitetura orientada a serviços. Sua atuação deve garantir que todos os padrões, arquitetura e processos definidos sejam obedecidos em toda a Estrutura Funcional e em cada projeto de desenvolvimento SOA (JOACHIM, 2011).

As atividades associadas a essa unidade devem ter participação efetiva da equipe técnica permanente. Não obstante, parte de suas atividades podem ser reforçadas por times de profissionais de terceiros, em especial através de contratos de consultoria, *mentoring* e transferência de tecnologia.

Os principais processos sob a responsabilidade desta unidade estão na Figura 4-8. Processos detalhados no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.



Figura 4-8 – Processos do Centro de Excelência SOA

- **Definir Arquitetura e Metodologia SOA:** Consiste na definição e constante atualização da Arquitetura de Referência e da Metodologia de Desenvolvimento SOA, bem como de procedimentos e melhores práticas envolvidos no processo de adoção SOA. O perfil Arquiteto SOA participa apenas desse processo no Centro de Excelência SOA, ficando suas demais responsabilidades na metodologia vinculadas a unidade Fábrica de Serviços.
- **Garantir Aplicação de Padrões:** Constituído de atividades relacionadas à garantia do uso apropriado dos padrões corporativos de SOA, com especial foco no uso da Arquitetura de Referência e da Metodologia de Desenvolvimento SOA.
- **Garantir Qualidade de Soluções e Serviços SOA:** Contempla as atividades de controle de qualidade das soluções, serviços, demais produtos e artefatos associados, gerados nos projetos de desenvolvimento SOA. Inclui atividades de acompanhamento do desenvolvimento e revisão de qualidade de produtos.
- **Auditar Processo de Desenvolvimento SOA:** Engloba as atividades de auditoria dos processos de desenvolvimento SOA, com objetivo de verificar a adoção e a conformidade do uso dos modelos definidos.
- **Gerenciar Governança SOA:** Esse *macroprocesso* contempla as atividades relacionadas à governança de SOA, incluindo o estabelecimento dos mecanismos de governança.
- **Gerenciar Fábricas:** Compreende as atividades de controle e fiscalização das atividades e funções desempenhadas pelas Fábricas de Serviço, Software e Testes. Abrange desde os artefatos gerados para encaminhamento às fábricas até o retorno dos respectivos entregáveis.

4.2.2 Fábrica de Serviços

Unidade funcional responsável pelos processos das disciplinas de *modelagem de processos de negócio, análise e projeto dos serviços*, bem como participação nos processos da disciplina de implantação, com aplicação de orientação a serviço.

Os processos sob responsabilidade ou com participação direta desta unidade estão ilustrados na Figura 4-9. Processos detalhados no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.



Figura 4-9 – Processos da Fábrica de Serviços

- **Desenhar Processos de Negócio:** Processo relacionado à etapa “Modelagem de Negócio Corporativo”. Realizado em conjunto com a unidade de Gestão de Processos de Negócio.
- **Realizar Análise Orientada a Serviço:** Processos relacionados às etapas de “Análise de Inventário de Serviço” e “Análise Orientada a Serviço”.
- **Realizar Projeto Orientado a Serviço:** Processos relacionados às etapas “Projeto Orientado a Serviço” e “Projeto de Lógica de Serviço”.
- **Catalogar e Publicar Serviços:** Processos relacionados à etapa “Catalogação e Publicação”. Realizado em conjunto com a unidade de Gestão de Infraestrutura.
- **Integrar Serviços e Interfaces de Usuário:** Processos relacionados à etapa “Integração de Serviços e Interfaces de Usuário”.
- **Implantar Solução:** Acompanhamento do Processo relacionado à etapa “Implantação da Solução”. Realizado em conjunto com a unidade de Gestão de Infraestrutura.

4.2.3 Fábrica de Software

Unidade funcional especializada nas atividades da disciplina de implementação de software. Responsável pela implementação (codificação) dos serviços e demais componentes da solução orientada a serviço. Existem várias organizações que usam o modelo de desenvolvimento de fábrica de software e contratos *master* convencionais de terceirização geralmente são adequados para a alocação de programadores e desenvolvedores. Note, no entanto, que a implementação pode estar relacionada a um *middleware* SOA específico e experiência no desenvolvimento para essas plataformas é necessária. Em tais casos, os contratos especiais de terceirização podem ser aplicados. (HOTLE e LANDRY, 2009; CLEMENTS e NORTHROP, 2001; BROWN et al., 2011).

Os processos sob responsabilidade desta unidade estão ilustrados na Figura 4-10.



Figura 4-10 – Processos da Fábrica de Software

- **Implementar Serviços:** Processos relacionados à etapa “Implementação de Serviço”. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.
- **Implementar Interfaces de Usuário:** Consiste na produção de código fonte de interfaces de usuário (camada de apresentação) não orientadas a serviço.

4.2.4 Fábrica de Testes

Unidade funcional especializada na disciplina de testes de software. Responsável pela execução dos processos de teste, conforme planos de testes definidos, para serviços e soluções orientadas a serviço.

Os processos sob responsabilidade desta unidade estão ilustrados na Figura 4-11.

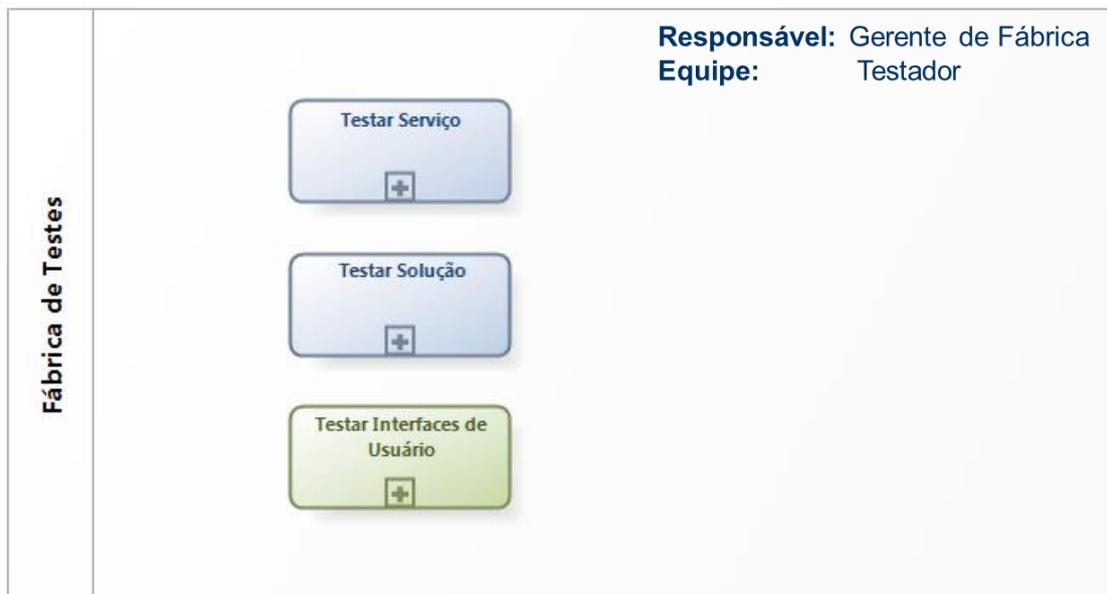


Figura 4-11 – Processos da Fábrica de Testes

- **Testar Serviço:** Processos relacionados à etapa “Teste de Serviço”. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.
- **Testar Solução:** Processos relacionados à etapa “Teste de Solução”. Processo detalhado no Capítulo 6 – Ciclo de Vida de Projetos de Soluções em SOA.
- **Testar Interfaces de Usuário:** Consiste na realização de testes unitários em interfaces de usuário (camada de apresentação) não orientadas a serviço.

5 - PERFIS PROFISSIONAIS

5.1 DESCRIÇÃO DOS PERFIS PROFISSIONAIS

Esta seção descreve os perfis profissionais utilizados no desenvolvimento das atividades da metodologia de desenvolvimento SOA.

É importante salientar que os perfis descritos neste capítulo não, necessariamente, representam profissionais distintos para cada um. Embora isso seja recomendável, para garantir que todas as atividades sejam executadas com seu devido foco, entende-se que se terá que adaptar esses perfis a sua realidade de disponibilidade de recursos humanos com suas empresas terceirizadas, se houver. Isso pode ser feito, alocando mais de um perfil a um mesmo profissional. Ressalta-se, porém, que isso deve ser feito com prudência, deixando claro, dentro do processo de desenvolvimento da nova arquitetura, o papel de cada profissional e suas responsabilidades.

Para cada perfil definido serão discriminadas suas responsabilidades, capacidades profissionais e habilidades requeridas. Uma relação de treinamentos, reconhecidos internacionalmente, sugeridos para cada perfil encontra-se no Apêndice

5.1.1 Administrador de Dados

Descrição: Trabalha ativamente junto à equipe de desenvolvimento no intuito de garantir a padronização de todos os requisitos dos dados como modelos, nomenclatura, acesso etc.

Responsabilidades: Prover suporte na criação do modelo lógico de dados; documentar em dicionário de dados os modelos lógicos e físicos de processos, dados e funções; documentar em dicionário de dados dos objetos de base de dados tais como: *database*, *tablespaces*, *storagespaces*, *constraints*, *tables*, *views*, *triggers*, *roles*, *database links*, *procedures*, *packages*, *functions*, *profiles* e *synonyms* e levantar e gerar a documentação de todos os componentes do sistema (modelos lógicos e físicos, objetos de base de dados, esquemas lógicos, regras de negócio e a sua interface com os aplicativos).

Capacidades Profissionais:

- Administrador de Dados.

Habilidades Requeridas:

- Domínio nos modelos de Banco de Dados existentes;
- Sólidos conhecimentos e experiência em modelagem de dados;
- Bons conhecimentos do negócio da organização.

5.1.2 Administrador de Infraestrutura SOA

Descrição: Gerencia e disponibiliza recursos de hardware e software para a disponibilização do ambiente SOA, incluindo Servidor de Aplicação, *Middleware* SOA e demais plataformas associadas à arquitetura tecnológica.

Responsabilidades: Implantar e prestar suporte em toda a infraestrutura de software da arquitetura SOA; estudar, propor e adotar solução para melhoria contínua no desempenho da arquitetura; monitorar o ambiente e intervir em casos de exceção e garantir a disponibilidade dos recursos da arquitetura.

Capacidades Profissionais:

- Especialista em Gerenciamento de Infraestrutura SOA;
- Arquiteto de Infraestrutura (desejável).

Habilidades Requeridas:

- Sólidos conhecimentos e experiência na plataforma de gerenciamento do middleware SOA definido na Arquitetura de Referência SOA;
- Conhecimentos em arquitetura de infraestrutura (desejável).

5.1.3 Analista de Negócio

Descrição: Sua principal característica é o conhecimento sobre o negócio da organização, suas diretrizes e objetivos, não sendo necessário, em princípio, nenhum conhecimento específico de SOA. Este perfil muitas vezes não se encontra na área de TI.

Responsabilidades: Esse grupo de profissionais deverá ser responsável pela implantação da cultura da Gestão por Processos; mapeamento e melhoria contínua dos processos da organização; definição das métricas e níveis de serviço dos processos (KPIs); implementação de metodologia e melhores práticas para levantamento; redesenho e acompanhamento dos processos; padronização e documentação de métodos e processos;

desenvolvimento e acompanhamento da Governança de Processos e planejamento estratégico da organização no que diz respeito à otimização da utilização de recursos e processos, dentre outras.

Capacidades Profissionais:

- Analista de Negócio;
- Analista BPM;
- Especialista BPMS (desejável).

Habilidades Requeridas:

- Domínio do negócio da organização;
- Acesso ao corpo gestor;
- Comunicação e liderança;
- Sólidos conhecimentos e experiência em modelos de negócio empresariais;
- Uso da ferramenta BPM definida na Arquitetura de Referência SOA (desejável).

5.1.4 Analista de Processo

Descrição: Técnicos da área de TI com perfil semelhante ao de um Analista de Negócios, porém, com uma visão mais operacional e menos estratégica. Visualiza a automação de processos como parte das atividades de análise, desenho e modelagem de processos.

Responsabilidades: Esse grupo de profissionais deverá ser responsável, juntamente com o Analista de Negócio, pela implantação da cultura da Gestão por Processos; mapeamento e melhoria contínua dos processos da organização; definição das métricas e níveis de serviço dos processos (KPIs); implementação de metodologia e melhores práticas para levantamento; redesenho e acompanhamento dos processos; padronização e documentação de métodos e processos e desenvolvimento e acompanhamento da Governança de Processos.

Realiza acompanhamento da modelagem dos processos, especificando os requisitos de negócio, documentando-os e incorporando os mesmos na ferramenta de modelagem. Elabora glossário de negócio, protótipo de interface gráfica e atua na construção de modelo de dados, validando-os junto ao Administrador de Dados.

Capacidades Profissionais:

- Analista BPM;
- Especialista BPMS;
- Analista SOA (desejável).

Habilidades Requeridas:

- Conhecimento da operação do negócio da organização;
- Boa comunicação e facilidade para trabalhar em equipe;
- Experiência em análise e levantamento de requisitos;
- Sólidos conhecimentos e experiência em modelagem de processos de negócio visando automação;
- Conhecimentos e experiência em modelagem de dados;
- Domínio da ferramenta de BPM definida na Arquitetura de Referência SOA.

5.1.5 Analista SOA

Descrição: O Analista SOA funciona como um elo ou canal de comunicação entre a área de negócio da organização e a equipe técnica da Fábrica de Serviços. Esse perfil deve possuir um profundo conhecimento da arquitetura, padrões e ferramental SOA e agir como um “tradutor” da parte negocial para um modelo de implementação. Ele deve trabalhar em sintonia com o Analista de Negócio e de Processos.

Responsabilidades: Além de sua função básica de definir os componentes dos processos de negócio em uma visão mais técnica/operacional, possui uma gama extensa de responsabilidades no processo de implantação da tecnologia. Pode-se destacar, dentre elas, a função de identificar os serviços candidatos de acordo com os princípios básicos de SOA, sua granularidade, níveis de serviço, segurança etc. Ele é diretamente responsável pelo nível de reuso de componentes dentro da arquitetura e, conseqüentemente, pela aferição de um dos principais benefícios da adoção de SOA: a redução do ciclo de desenvolvimento com o reuso de componentes.

Outra importante responsabilidade é a de praticar a Governança em SOA e analisar profundamente o inventário de serviços, garantindo a reutilização dos componentes, sua rastreabilidade, sua vinculação com as regras de negócio e diretrizes da organização, além de conhecer todo o controle de seu “ciclo de vida” e seus aspectos de segurança.

Capacidades Profissionais:

- Analista SOA;
- Especialista BPMS (desejável).

Habilidades Requeridas:

- Conhecimento da operação do negócio da organização;
- Boa comunicação e facilidade para trabalhar em equipe;
- Sólidos conhecimento e experiência em análise orientada a serviços;
- Uso da ferramenta de BPM definida na Arquitetura de Referência SOA (desejável);
- Uso das ferramentas SOA e de Governança definidas na Arquitetura de Referência SOA (desejável).

5.1.6 Analista de Testes

Descrição: Este perfil está associado com a definição da estratégia de testes de software.

Responsabilidades: Elaborar a estratégia de testes de software, acompanhar a execução de testes e validar seus resultados; elaborar casos e roteiro de testes; aplicar teste no processo e componentes; elaborar plano de testes; planejar atividades de teste; apresentar relatórios de gerenciamento; determinar os resultados dos testes; avaliar a eficiência do esforço de teste e avaliar a qualidade do produto final. Também é sua responsabilidade atualizar as ferramentas de governança com relação aos planos de testes.

Capacidades Profissionais:

- Analista de Testes;
- Especialista em Infraestrutura SOA;
- Analista SOA (desejável).

Habilidades Requeridas:

- Sólidos conhecimento e experiência em metodologias de testes de *software*;
- Conhecimento da plataforma de middleware SOA definida;
- Uso das ferramentas SOA e Governança definidas (desejável).

5.1.7 Arquiteto Corporativo

Descrição: É o profissional responsável pelo desenvolvimento e evolução da arquitetura corporativa.

Responsabilidades: Compor ou contribuir com padronizações de projeto corporativas; envolver-se em projetos de entrega de serviços para assegurar que serviços agnósticos sejam posicionados apropriadamente; avaliar o uso de serviços em tempo de execução e determinar a infraestrutura requerida; avaliar ou identificar preocupações de segurança e disponibilidade de capacidades de serviço individuais; ajudar a definir e, potencialmente, gerenciar esquemas de inventários de serviços.

Capacidades Profissionais:

- Arquiteto Corporativo;
- Arquiteto SOA;
- Especialista em Infraestrutura SOA;
- Especialista em Gerenciamento de Infraestrutura SOA (desejável).

Habilidades Requeridas:

- Bom conhecimento do negócio e da operação da organização;
- Comunicação e liderança;
- Sólidos conhecimentos e experiência em arquitetura corporativa;
- Conhecimento da plataforma de *middleware* SOA definida;
- Uso das ferramentas SOA e de Governança definidas.

5.1.8 Arquiteto SOA

Descrição: O Arquiteto SOA é o projetista da arquitetura dos serviços. Trabalha em sintonia com o Analista SOA, inclusive na manutenção da governança.

Responsabilidades: Além de sua função básica de definir a arquitetura e infraestrutura física e lógica para suportar os processos de negócio e componentes, possui uma gama extensa de responsabilidades no processo de implantação da tecnologia. Elabora o contrato de serviços, a especificação técnica do serviço e a arquitetura de composição. É responsável pela aplicação de *design patterns* de SOA. É, também, de responsabilidade do Arquiteto definir o modelo de interoperação de serviços com soluções legadas.

Capacidades Profissionais:

- Arquiteto SOA;
- Especialista em Infraestrutura SOA.

Habilidades Requeridas:

- Sólidos conhecimentos e experiência em arquitetura SOA;
- Sólidos conhecimentos e experiência na plataforma de *middleware* SOA definida;
- Uso das ferramentas SOA e de Governança definidas.

5.1.9 Auditor SOA

Descrição: Funciona como um *controlador* de todo o processo SOA. Ele não é o responsável por definir os procedimentos e metodologias, nem de testes e validação de cada processo e/ou componente automatizado, mas sim, garantir que os mesmos sejam executados segundo as regras estabelecidas. Sua visão é global, incluindo a fase de levantamento/mapeamento dos processos e sua validação pela área de Gestão de Processos, passando por sua automatização, até sua entrada em operação pela Gestão de Infraestrutura. Para que esse perfil tenha uma atuação mais isenta, recomenda-se que os profissionais com este perfil estejam alocados no Centro de Excelência SOA.

Responsabilidades: Garantir que todo o processo de desenvolvimento SOA ocorra segundo as normas e metodologias definidas; verificar se todos os procedimentos determinados para cada fase do processo estejam sendo rigorosamente seguidos, se todos os artefatos gerados estão dentro dos padrões homologados e verificar se os princípios de governança em SOA definidos estão sendo obedecidos.

Capacidades Profissionais:

- Especialista de Governança SOA;
- Analista SOA (desejável);
- Arquiteto SOA (desejável).

Habilidades Requeridas:

- Sólidos conhecimentos e experiência em orientação a serviço.
- Conhecimento da plataforma de *middleware* SOA definida;

- Uso das ferramentas SOA e de Governança definidas.

5.1.10 Desenvolvedor

Descrição: Profissional desenvolvedor na linguagem definida na Arquitetura de Referência SOA com conhecimento em tecnologia de implementação de *web services* e também, para interface, conhecimento no *framework* definido. Não necessita ser um especialista SOA mas é desejável.

Responsabilidades: Desenvolver código e encapsular no componente *web service* definido e gerado pelo Arquiteto SOA; desenvolver no ambiente definido, os componentes de interface especificados.

Capacidades Profissionais:

- Desenvolvedor especialista na linguagem definida;
- Desenvolvedor *Web Service*;
- Desenvolvedor na *framework* definida.

Habilidades Requeridas:

- Sólidos conhecimentos na linguagem definida;
- Conhecimento do *framework* definido;
- Conhecimento em *web services*.

5.1.11 Especialista em Governança SOA

Descrição: Esse especialista é o projetista e guardião da governança dos processos e serviços/componentes. Trabalha em sintonia com o Analista SOA, com o Arquiteto SOA e também com o Auditor SOA na manutenção e respeito aos quesitos de governança definidos.

Responsabilidades: Atua em todo o processo de desenvolvimento SOA. É o responsável pela definição e manutenção dos princípios de governança. Certifica-se do cumprimento, por todos os atores do processo de desenvolvimento, das diretrizes de governança definidas.

Normalmente não detém o poder de intervenção, mas sim, o de identificar e relatar as não conformidades do processo em relação à governança.

Capacidades Profissionais:

- Especialista de Governança SOA;
- Especialista nas ferramentas de Governança SOA definidas.

Habilidades Requeridas:

- Sólidos conhecimentos e experiência em orientação a serviço.
- Conhecimento da plataforma de *middleware* SOA definida;
- Domínio das ferramentas SOA e de Governança definidas.

5.1.12 Gerente de Projeto

Descrição: A gerência de projetos não muda com SOA, seu papel continua sendo o mesmo, só que na visão SOA, cada processo é um pequeno projeto. Normalmente já se adota metodologia baseada no PMBOK®, mas deve atentar-se ao fato que sua adoção plena pode acabar causando a não observação de um dos principais benefícios de SOA, sua agilidade e rapidez na disponibilização de novos instrumentos.

Responsabilidades: Apoiar a definição de escopo de cada projeto e promover o mapeamento dos requisitos; promover toda documentação pertinente e aderente aos padrões de melhores práticas; auxiliar a definição das atividades necessárias para a correta execução do projeto; especificar o sequenciamento das atividades do projeto; auxiliar na estimativa de duração das atividades; promover a construção de EAP (Estrutura Analítica do Projeto); definir os recursos essenciais para o projeto; definir e monitorar o cronograma de cada projeto; delimitar e especificar os meios de comunicação entre os membros da equipe; promover o planejamento de riscos; promover o planejamento de respostas a riscos; promover o planejamento de aquisições; conceber o Plano de Projeto; definir o controle de mudanças no projeto; definir o processo de verificação constante do escopo do projeto; promover o controle de qualidade do projeto; auxiliar a produção dos relatórios de aceitação/aprovação; elaborar e manter atualizados os planejamentos e cronogramas de atividades de cada projeto; acompanhar a execução dos projetos conforme os planos e cronogramas e acompanhar para que os projetos sejam executados conforme as melhores práticas especificadas no PMBOK®.

Capacidades Profissionais:

- Gerente de Projeto.

Habilidades Requeridas:

- Sólidos conhecimentos e experiência em gerenciamento de projetos;
- Liderança e comunicação;
- Organização e comprometimento.

5.1.13 Testador

Descrição: Profissional alocado na Fábrica de Testes. Este perfil atua na execução dos testes dos componentes.

Responsabilidades: Executar os Planos Testes elaborados e gerar seus respectivos relatórios.

Capacidades Profissionais:

- Testador

Habilidades Requeridas:

- Conhecimento em Metodologias de Testes;
- Noções em SOA.

6 - CICLO DE VIDA DE PROJETOS DE SOLUÇÕES EM SOA

Esta seção descreve a Metodologia de Desenvolvimento SOA proposta, apresentando as atividades necessárias para um processo de desenvolvimento SOA e relacionando cada uma às suas específicas Unidades Funcionais e aos seus respectivos Perfis Profissionais.

O ciclo de vida da metodologia é apresentado na Figura 3-1. Embora este seja representado como um ciclo de vida de desenvolvimento de software em “cascata”, ou seja, os processos executados sequencialmente, os mesmos podem ser utilizados em um modelo iterativo de desenvolvimento, conforme a necessidade independentemente da sequência definida.

Dependendo dos requisitos especificados na Ordem de Serviço de Fábrica de Serviço (origem da demanda), diferentes demandas podem surgir, como o simples desenvolvimento de um serviço, a automatização de um processo com serviços e interfaces ou um serviço com uma interface de interação para o usuário, dentre outras combinações possíveis. Dentre estas, destacamos os três tipos de elementos que, se levantados, necessitam ser desenvolvidos/automatizados para integrar uma solução SOA:

- Processos;
- Interfaces de Usuário e;
- Serviços SOA.

As próximas seções detalham as etapas da Metodologia proposta, com seus respectivos processos e atividades.

Adicionalmente à descrição dos processos e atividades, são apresentados ainda conjuntos de heurísticas e de padrões de projeto aplicáveis como parte do processo de desenvolvimento SOA:

- Heurísticas: O uso de heurísticas estabelece um conjunto de considerações práticas para aplicação dos princípios de orientação a serviço que devem ser consideradas pelo analista/arquiteto ao executar tarefas em um projeto de desenvolvimento SOA. Estas heurísticas são adaptadas dos princípios de orientação a serviço, extraídos de Erl (2008, 2009); Josuttis (2007); Portier (2007) e Hewitt (2009).
- Padrões de Projeto: A aplicação sistemática de padrões de projeto (*design patterns*) de SOA resulta em melhores soluções orientadas a serviço.

Recomenda-se que se desenvolva e mantenha um catálogo de padrões (*design patterns catalog*) com documentação acessível e regras de aplicação claramente estabelecidas. O desenvolvimento deste catálogo é igualmente indicado na Arquitetura de Referência SOA, mas sua construção não faz parte do escopo desta metodologia. Esse catálogo deverá estar sob responsabilidade e gestão do Centro de Excelência SOA. Não obstante, alguns padrões de projeto com aplicabilidade reconhecida são apontados nas seções seguintes. Os padrões de projeto aqui elencados estão documentados em Erl (2008) e <http://www.soapatterns.org>.

Esses elementos adicionais da metodologia aparecem especialmente nas atividades das disciplinas de análise e projeto com orientação a serviços.

No início de cada etapa, é apresentado um quadro resumo das principais informações sobre a etapa, como os artefatos de entrada e saída, atividades e perfis profissionais, executores e envolvidos, mais relevantes. Conforme ilustrado na Figura 6-1, a seguir.

Entradas de Processo	•
Saídas de Processo	•
Atividades	•
Perfis Executores	•
Perfis Envolvidos	•

Figura 6-1 – Quadro Resumo de Informações da Etapa

Foi optado por introduzir apenas as informações mais relevantes apenas para não poluir o quadro e torna-lo de fácil leitura. Por exemplo, embora o perfil Gerente de Projeto execute atividades pertinentes a sua área de atuação em todas as etapas, este perfil não aparece em nenhum dos quadros resumos, o mesmo acontecendo, por exemplo, para os perfis de Auditor SOA e Especialista em Governança SOA, que se envolvem com suas atividades de acompanhamento e controle em todas as etapas, porém, não são destacados nos perfis envolvidos. Nas Entradas e Saídas de Processo, os artefatos especificamente relacionados a esta metodologia SOA são representados em negrito.

Ao final de cada etapa, é apresentado também um quadro que demonstra todos os artefatos de entrada e de saída daquela etapa. A esquerda do quadro, a etapa em questão é destacada para fins informativos. Os artefatos tratados pela etapa aparecerão em negrito e símbolos representarão se o artefato é parcial (em atualização) ou concluído na etapa, conforme ilustrado na Figura 6-2, a seguir.



Figura 6-2 – Quadro de Artefatos da Etapa

Discriminação dos Artefatos:

- Documento de Modelagem de Processo de Negócio

Composto pelos desenhos de mapeamento dos processos, definição das regras de negócio, interfaces com os sistemas legados e todas as informações alinhadas com os processos corporativos, diretrizes e estratégias empresariais. São incluídos também os aspectos relacionados com sua automatização, como definição das interfaces do usuário, integração com os sistemas de apoio e bases de dados, entre outros;

- Modelo de Informação

Conforme padrão XSD. Define todo o modelo de informação e estrutura de dados relacionados aos processos de negócio modelados;

- Documento de Definição de Candidatos a Serviços

Contém a especificação preliminar dos serviços candidatos, ou seja, grupamentos de lógica e capacidades que podem vir a ser elencados para se tornarem serviços de fato;

- Perfil de Serviço

Definido na Arquitetura de Referência SOA. Contém todas as informações do serviço, suas especificações técnicas, suas capacidades e definições necessárias para seu efetivo desenvolvimento.

- Contrato de Serviço

Definido na Arquitetura de Referência SOA. É o WSDL do serviço, contendo todos os dados que o contrato deve abranger segundo as melhores práticas, inclusive alguns de governança, níveis de serviço, descritivos, etc.

- Documento de Arquitetura de Serviço

Definido na Arquitetura de Referência SOA. Contém as definições e especificações da arquitetura do serviço;

- Documento de Arquitetura da Solução

Definido na Arquitetura de Referência SOA. Contém as definições e especificações da solução;

- Plano de Testes de Serviço

Possui todas as etapas de testes, as funcionalidades e regras a serem testadas e como, especificação da base de testes, procedimentos e resultados esperados para os testes do serviço;

- Plano de Testes da Solução

Possui todas as etapas de testes, as funcionalidades e regras a serem testadas e como, especificação da base de testes, procedimentos e resultados esperados para os testes da solução;

- Ordem de Serviço de Fabrica de Serviço

Conforme contrato administrativo;

- Ordem de Serviço de Fabrica de Software

Conforme contrato administrativo;

- Ordem de Serviço de Fábrica de Testes

Conforme contrato administrativo;

- Documento de Projeto de Serviço

Documento final do serviço, após a realização dos testes, contendo todas as informações do projeto do serviço;

- Documento de Projeto de Solução

Documento final da solução, após a realização dos testes, contendo todas as informações do projeto da solução;

- Código Fonte de Serviço

Conjunto do(s) arquivo(s) que compõem a totalidade do código do serviço;

- Código Fonte da Solução

Conjunto do(s) arquivo(s) que compõem a totalidade do código do serviço;

- Relatório de Testes de Serviço

Relatório contendo os resultados dos testes, suas inconformidades e situações de erro;

- Relatório de Testes da Solução

Relatório contendo os resultados dos testes, suas inconformidades e situações de erro;

- Manual de Uso de Serviço

Contém informações sobre suas políticas de uso, métodos de acesso, segurança, etc.;

- Manual de Uso da Solução

Contém informações sobre suas políticas de uso, métodos de acesso, segurança, etc.;

- Documento de Configuração de Serviço

Contém informações e dados de configuração do serviço na infraestrutura SOA;

- Documento de Configuração da Solução

Contém informações e dados de configuração da solução na infraestrutura SOA.

Para maior clareza da descrição da metodologia proposta, optou-se por não sobrecarregar o texto com as devidas citações dos autores que serviram de base para sua elaboração. Desta forma, preferimos nominá-los aqui, com destaque para Thomas ERL, Ali ARSANJANI e Norbert BIEBERSTEIN.

6.1 MODELAGEM DE NEGÓCIO CORPORATIVO

Entradas de Processo	<ul style="list-style-type: none"> • Ordem de Serviço da Fábrica de Serviço • Arquitetura Empresarial • Modelo de Dados Corporativo • Arquitetura de Referência SOA • Normas internas • Legislação aplicável • Informações colhidas com partes interessadas (<i>stakeholders</i>) • Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none"> • Documento de Modelagem de Processos <incluindo requisitos funcionais e não funcionais> • Modelo de Informação (Estruturada) • Artefatos de Arquitetura Empresarial (Arquitetura de Negócio e Arquitetura de Informação) atualizados • Artefatos de Administração de Dados (Modelo de Dados Corporativo) atualizados • Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none"> • Levantamento dos Processos Empresariais • Decomposição e Desenho dos Processos Empresariais (até se chegar ao nível de Processos de Negócio) • Modelagem de (Estrutura da) Informação
Perfis Executores	<ul style="list-style-type: none"> • Administrador de Dados • Analista de Negócio • Analista de Processo
Perfis Envolvidos	<ul style="list-style-type: none"> • Analista SOA • Arquiteto Corporativo • Arquiteto SOA • Especialista em Governança SOA

O conceito de Processo de Negócio pode não ter o mesmo significado para um profissional da área de Negócio que para um profissional da área de TI. Por isso, dividimos esse conceito em dois tipos de processos, os Processos Empresariais e os Processos de Negócio.

Os Processos Empresariais podem ser entendidos como um conjunto de atividades, recursos materiais, humanos, tecnológicos e financeiros que agreguem valor a insumos recebidos de fornecedores internos ou externos e gerando produtos a clientes, conforme ilustrado na Figura 6-3.

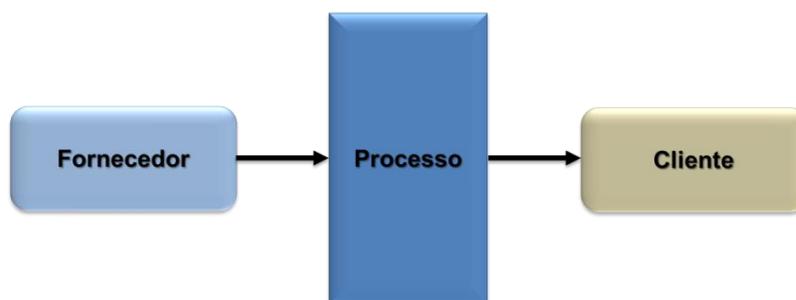


Figura 6-3 – Definição de Processo Empresarial

A partir desse conceito, entende-se que a organização é um processo, porque atende plenamente o conceito à medida que ela recebe insumos de fornecedores externos, agrega valor a esses insumos utilizando todos os recursos disponíveis e gera produtos a clientes, no sentido de satisfazer às necessidades destes.

Indo para dentro da Organização, pode-se visualizar que, a mesma é formada por um conjunto de processos que atendem o conceito apresentado e analisando-se cada um desses processos pode-se identificar que são formados, também, por outro conjunto de processos.

Os Processos Empresariais mostram “o que” a organização gera e como está decomposta essa geração.

A decomposição, para o entendimento dos processos empresariais, pode se repetir até chegarmos ao detalhamento da atividade, momento em que identificamos os “Processos de Negócio”.

Processo de Negócio pode ser conceituado como um grupo de atividades realizadas numa sequência determinada, que gera uma saída, a qual pode ser o produto ou ser parte integrante do produto. Pode ser encarado, também, como qualquer atividade ou conjunto de atividades onde há uma entrada, uma transformação e uma saída.

Os Processos de Negócios estão focados no “como” a Organização opera.

Como já mencionado anteriormente, não entraremos em maiores detalhes sobre essa atividade, como, por exemplo, descrever como levantar ou mapear um processo de negócio, pois isso está relacionado a uma metodologia BPM específica, não fazendo parte do escopo deste trabalho. Porém, como essa fase serve de insumo para o desenvolvimento SOA em si, é importante definir alguns passos que deverão ser seguidos e artefatos que deverão ser elaborados.

Também nesta fase deverá ser gerado um artefato, que não é foco deste trabalho, de definição das interfaces do usuário, deve ser detalhado o suficiente para seguir para desenvolvimento.

Em um processo de desenvolvimento tradicional existe uma etapa específica para o levantamento de requisitos, funcionais e não funcionais. No caso de SOA, porém, por ter seu ciclo altamente acoplado a processos de negócio, é nesta fase de modelagem em que tanto os requisitos mais macros do negócio (visão empresarial) como também os dos processos de negócio em si (visão de atividades e tarefas), são levantados e documentados.

6.1.1 Processos e Tarefas da Modelagem de Negócio Corporativo

A Figura 6-4 ilustra o processo Modelar Negócio Corporativo e suas tarefas e subprocessos:

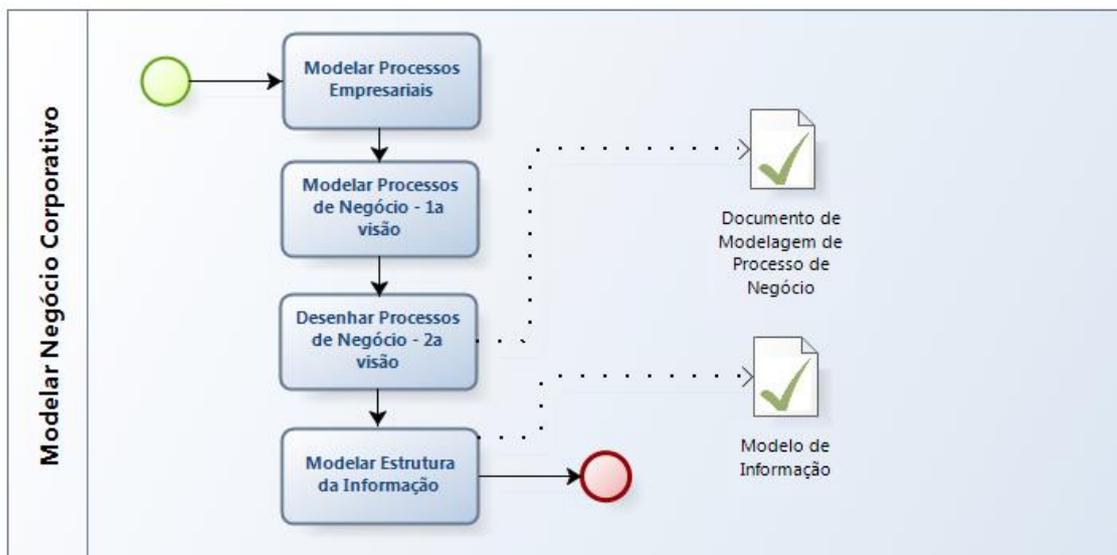


Figura 6-4 – Processo Modelar Negócio Corporativo

6.1.1.1 Tarefa 1 – Modelar Processos Empresariais

O esforço realizado para a identificação dos processos empresariais deve ser documentado. Ele servirá para vários fins na Organização, principalmente porque é uma maneira fácil de entender a organização, podendo ser utilizado, por exemplo, em programas de integração de novos colaboradores.

O Processo Empresarial maior (a própria empresa) deve ser decomposto em Processos Empresariais mais detalhados e assim, sucessivamente, até se chegar ao nível de Processos de Negócio.

Os artefatos gerados desse esforço deve ser um conjunto de diagramas contendo todas as decomposições do processo maior, que é a própria organização, e um documento descritivo do processo.

6.1.1.2 Tarefa 2 – Modelar Processos de Negócio (1ª visão – Visão de Atividades)

Modelar processos de negócio significa entender a sequencia (fluxo) das atividades que são executadas em um processo a partir de uma “entrada”, que é o início do processo, até a “saída” produzida pelo mesmo.

Não obstante que se encontre ainda afastado das necessidades da transformação BPM para SOA, o levantamento, identificação, mapeamento, modelagem e documentação dos processos de negócio, seus requisitos funcionais e não funcionais, em seus respectivos artefatos, são necessários.

Esta atividade inclui o mapeamento e desenho do fluxo de processos de negócio em nível de atividades. Isso permite a visão do processo de negócio e o envolvimento dos perfis no fluxo principal de atividades (1ª visão – Visão de Atividades). Esta se justifica principalmente como elemento de apresentação visual para fácil entendimento da Governança Corporativa. Ao mesmo tempo, permite um melhor desenvolvimento do fluxo de delegações que se encontra internalizado nos modelos de processos de negócio das ferramentas de mercado.

Processos mapeados devem ser detalhados em um nível que se possa observar todos os processos de negócio de alto nível (atividades) envolvidos no projeto.

Técnicas descritivas, quadros de fornecedores e clientes (como mostrado anteriormente) e outros artefatos (tal como o IDEF0) são importantes na representação do negócio e são insumos necessários para todo processo de mapeamento que é discutido no decorrer deste documento.

6.1.1.3 Tarefa 3 – Desenhar Processos de Negócio (2ª visão - Visão de Tarefas)

Consiste em detalhar/decompor o modelo de fluxo de processos do passo anterior. Deste deve ser derivado um segundo nível de mapeamento que observe as tarefas como atividades autocontidas de serviços, identificando os elementos essenciais na ótica de tarefas sistêmicas, manuais ou processos de negócio informatizados, com todos seus requisitos, funcionais e não funcionais levantados (2ª visão - Visão de Tarefas).

É importante que em cada processo no BPMN já fique explícito o tipo de processo, por exemplo, processos não sistematizados são “*Manual Task*”, processos que envolvem interface humana são “*User Task*”, tarefas de processamento (sistêmicos ou humanos) são “*Service Task*”.

Para um mapeamento puro dos processos sistêmicos na forma de atuações isoladas junto a um sistema - cada ator sistêmico atua unicamente em uma atividade, que pode ser compostas ou não de várias tarefas – da 1ª visão para a 2ª visão algumas regras implícitas de transformações são necessárias no modelo em BPMN:

- Todo processo de negócio que envolva mais de um ator deve ser isolado em atividades de negócio da forma como o são realizados sistemicamente, isto é, com somente um ator envolvido. Isto é, cada “pool” de atividades do processo deve ser apresentado com uma (e apenas uma) “lane”;
- A atividade do processo é a “pool” BPMN e o ator envolvido a “lane” BPMN;
- Cada um dos processos/subprocessos isolados deve conter apenas um elemento de início e fim, porém, exceções (*triggers*) de qualquer tipo (ex.: mensagens) e mensagens de colaboração (BPMN *collaboration*) podem existir;
- Antes de cada ponto de decisão se deve garantir existir um processo de negócio ou um processo de interface humana BPM para que se ative no fluxo de informações a variável que será tratada;
- Sequências de tarefas em cada atividade que são claramente autocontidas (aquelas que se assemelham a subprocessos ou, no caso mais claro, tarefas paralelas) são redefinidas em atividades isoladas (seu grupo de tarefas que é autocontido).

6.1.1.4 Tarefa 4 – Modelar Estrutura da Informação

Para completar todas as informações necessárias para a implementação, é necessário mapear e documentar todo o fluxo de informação, e seus respectivos modelos de dados, que ocorrerá ao longo do processo.

Para que esta modelagem seja completa, é necessário identificar todas as fontes de informações, como sistemas legados, bases corporativas e departamentais, que estejam inseridas dentro do escopo em análise.

6.1.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-5 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-5 – Entradas e Saídas da Etapa de Modelagem de Negócio Corporativo

6.2 ANÁLISE DE INVENTÁRIO DE SERVIÇO

Entradas de Processo	<ul style="list-style-type: none">• Ordem de Serviço da Fábrica de Serviço• Documento de Modelagem de Processos de Negócio• Modelo de Informação (Estruturada)• Arquitetura de Referência SOA• Artefatos de Governança de Serviços Corporativos (i.e. Contratos de Serviços disponíveis no Inventário de Serviços)• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Documento de Definição de Candidatos a Serviços (preliminar)• Perfil de Serviço (preliminar)• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Executar Análise Orientada a Serviço• Definir Esquema de Inventário de Serviços
Perfis Executores	<ul style="list-style-type: none">• Analista SOA• Arquiteto SOA
Perfis Envolvidos	<ul style="list-style-type: none">• Analista de Processo• Especialista em Governança SOA

Há duas fases de análise primárias em um projeto de SOA típico: a análise de serviços individuais e a análise coletiva de um inventário de serviços.

A fase de Análise Orientada a Serviço, descrita na seção 6.3, é dedicada a produzir definições conceituais de serviços chamados candidatos a serviços.

A Análise de Inventário de Serviços estabelece um ciclo onde o processo de análise orientada a serviço é executado iterativamente (junto com outros processos).

Um projeto de SOA é frequentemente, de fato, um projeto de entrega de serviços que é parte de uma iniciativa SOA maior. O escopo de uma iniciativa e o tamanho de um inventário de serviços alvo tende a determinar a quantidade de esforço antecipado.

Uma característica fundamental de projetos de SOA consiste em enfatizar a necessidade de aumentar o esforço de análise antecipada. Maior esforço de análise antecipada resulta em um melhor esquema conceitual definido, o que é desejado para levar à criação de um inventário de serviços de melhor qualidade.

Esse esforço de análise antecipada é realizado iterativamente como parte do ciclo de análise de inventário de serviços. Essa etapa é composta de um ciclo iterativo durante o

qual o esquema do inventário de serviços é definido de modo incremental como resultado de repetidas iterações de tarefas incluídas na análise orientada a serviço.

6.2.1 Processos e Tarefas da Análise de Inventário de Serviço

A Figura 6-6 ilustra o processo Analisar Inventário de Serviços e suas tarefas e subprocessos:

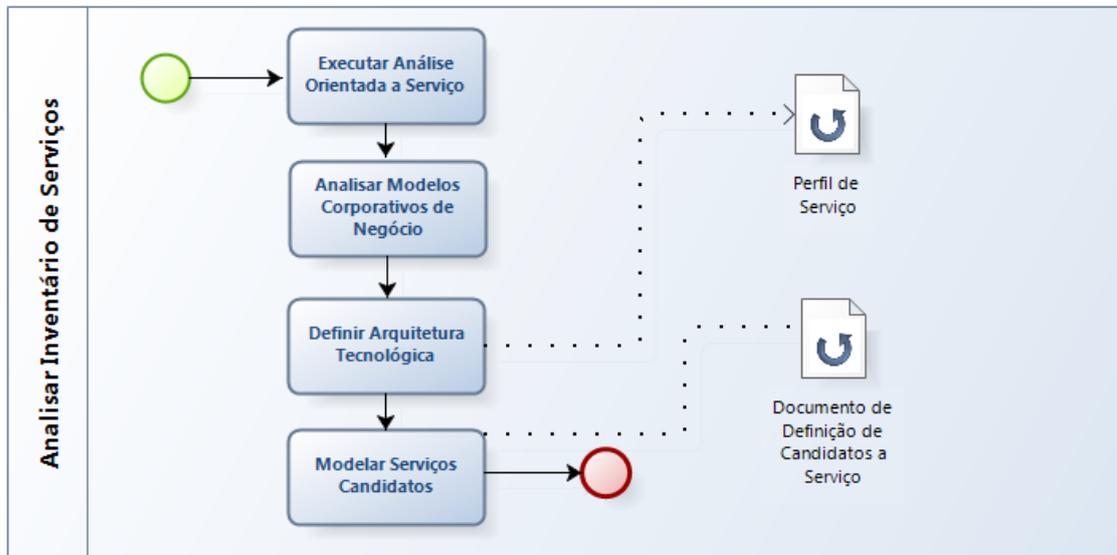


Figura 6-6 – Processo Analisar Inventário de Serviços

6.2.1.1 Tarefa 1 – Executar Análise Orientada a Serviço

Constituído de duas tarefas:

- Coletar informações; e
- Modelar serviços.

6.2.1.2 Tarefa 2 – Analisar Modelos Corporativos de Negócio

Este processo representa a identificação e definição de documentos de negócio relevantes, incluindo definições de processos de negócio, modelos de entidades de negócio e modelos lógicos de dados, dentre outros.

6.2.1.3 Tarefa 3 – Definir Arquitetura Tecnológica

Na medida em que as iterações são completadas, existe a oportunidade de se refinar a arquitetura tecnológica, em qualquer extensão viável, em resposta à evolução do esquema do inventário de serviços.

6.2.1.4 Tarefa 4 – Modelar Serviços Candidatos

Na medida em que as tarefas precedentes são executadas iterativamente, candidatos a serviços cada vez mais refinados são produzidos e começam a popular o esquema do inventário de serviços.

6.2.2 Heurísticas para Análise de Inventário de Serviço

Esta seção apresenta heurísticas para análise de serviços que devem ser consideradas pelo analista ao analisar inventário de serviços para uma iniciativa SOA.

6.2.2.1 Princípio da Capacidade de Composição de Serviço

6.2.2.1.1 Composições de Serviços

Heurística 1: *Composições de serviço são usadas para evitar a construção de lógica de solução redundante.*

Heurística 2: *Serviços são projetados para serem membros efetivos de composição, independente deles precisarem ser imediatamente listados em uma composição.*

Um inventário de serviços idealmente terminará constituído por uma alta percentagem de serviços agnósticos. Esses serviços são posicionados como meio primário (ou único) pelo qual a lógica de solução que eles representam deva e possa ser acessada. Isso resulta em uma mudança onde mais e mais requisitos de negócio não são atendidos pela construção ou extensão de aplicações. Ao contrário, eles são atendidos por composição de serviços existentes em novas configurações de composição.

A habilidade de compor serviços com efetividade é requisito crítico para atingir alguns dos objetivos fundamentais de computação orientada a serviço.

NOTA: Composições complexas de serviços colocam demandas significantes no projeto de serviço, bem como no ambiente de *runtime*.

6.2.2.2 Princípio do Reuso de Serviço

6.2.2.2.1 Serviços como Recursos Corporativos

Heurística: *Soluções individuais devem usar (compartilhar) lógicas de solução encapsuladas por serviços (agnósticos), posicionados como recursos corporativos (em um inventário de serviços).*

Heurística: *Projetar serviços que possam ter uma demanda maior que a esperada.*

Heurística: *Serviços semelhantes devem ser consolidados em um único serviço. A assinatura do serviço deve ser definida de acordo com a função mais genérica. Se as funções consideram diferentes requisitos não funcionais, então deve ser considerado o requisito não funcional mais restritivo.*

Um dos objetivos mais fundamentais de orientação a serviço consiste em estabelecer serviços como recursos corporativos. Um recurso corporativo é algo que pode ser usado por mais de uma parte da corporação. Um contexto de serviço que posiciona a lógica de serviço como um recurso corporativo é agnóstico por natureza. Um contexto agnóstico não tem propósito específico e, portanto, é potencialmente útil para muitos.

O reuso é o princípio mais importante no projeto de serviços. Normalmente, têm-se requisitos específicos de consumidores específicos. Contudo, quando se quer ter todas as vantagens de SOA, é necessário que os serviços que foram projetados sejam reusados por outros consumidores com requisitos ligeiramente diferentes. Ou seja, não se conhece de antemão todos os consumidores de um serviço e seus requisitos durante o projeto do serviço, o que torna a tarefa muito difícil.

6.2.3 Padrões de Projeto para Análise de Inventário de Serviço

Os seguintes padrões de projeto devem ser observados como parte do processo de análise de inventário de serviços:

- Normalização de Serviços
- Centralização de Lógica

- Centralização de Esquemas
- Camadas de Serviços

6.2.4 Visibilidade do Processo de Desenvolvimento

A Figura 6-7 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-7 – Entradas e Saídas da Etapa de Análise de Inventário de Serviço

6.3 ANÁLISE ORIENTADA A SERVIÇO

Entradas de Processo	<ul style="list-style-type: none">• Ordem de Serviço da Fábrica de Serviço• Documento de Modelagem de Processos de Negócio• Modelo de Informação (Estruturada)• Documento de Definição de Candidatos a Serviços• Perfil de Serviço• Arquitetura de Referência SOA• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Documento de Definição de Candidatos a Serviços (final)• Perfil de Serviço (atualizado)• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Executar Análise Orientada a Serviço
Perfis Executores	<ul style="list-style-type: none">• Analista SOA• Arquiteto SOA
Perfis Envolvidos	<ul style="list-style-type: none">• Analista de Processo• Especialista em Governança SOA

Na análise orientada a serviço são produzidas definições conceituais de serviços chamados de candidatos a serviço.

Em tempo de análise de serviços é possível identificar a construção de vários tipos de serviços, estes serviços podem ser categorizados dependendo:

- Do tipo de lógica a ser encapsulada;
- Da extensão do potencial de reuso que a lógica possui;
- De como a lógica de serviço se relaciona com os domínios existentes.

Neste contexto é possível identificar três classificações comuns de serviços que representam o modelo de serviço utilizado em Projetos de SOA:

- Serviço de Entidade;
- Serviço de Tarefa;
- Serviço Utilitário.

6.3.1 Processos e Tarefas da Análise Orientada a Serviço

A Figura 6-8 ilustra o processo Analisar com Orientação a Serviço e suas tarefas e subprocessos:

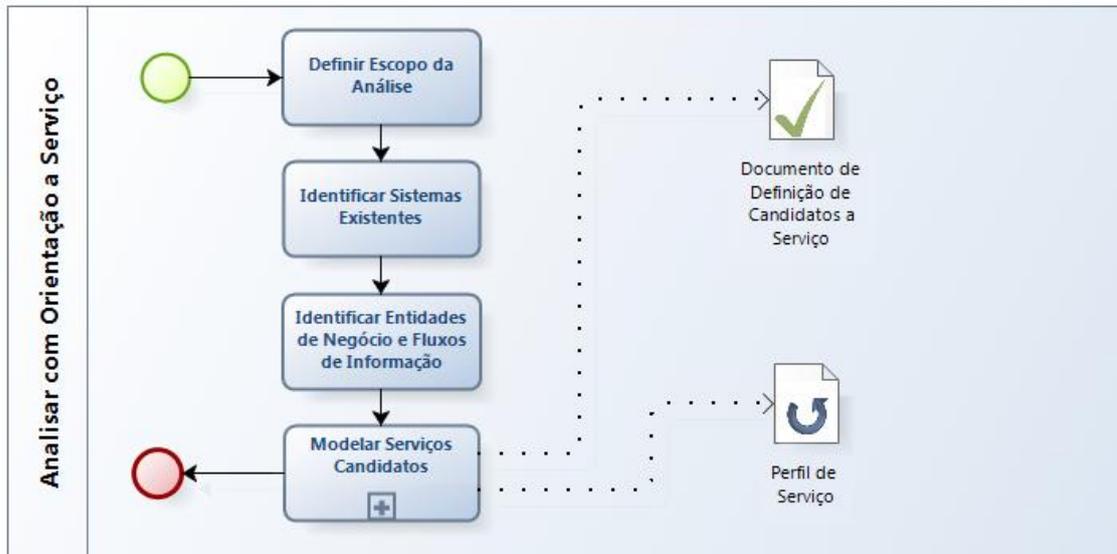


Figura 6-8 – Processo Analisar com Orientação a Serviço

6.3.1.1 Tarefa 1 – Definir Escopo da Análise

Para estabelecer apropriadamente o escopo da análise é necessário que os requisitos de negócio que se deseja atender estejam completamente definidos e maduros.

Isso significa uma clara definição dos processos de negócio (e todos os subprocessos) que devem estar documentados com detalhes suficientes.

Essa documentação de processos de negócio será usada como ponto de partida do processo descrito no subprocesso “Modelar Serviços Candidatos”.

6.3.1.2 Tarefa 2 – Identificar Sistemas Existentes

Sistemas existentes que serão afetados pelo escopo desta análise (o escopo do processo de negócio pai) precisam ser identificados.

A análise orientada a serviço não determina como exatamente os serviços irão encapsular a lógica legada.

Entretanto, o conhecimento de sistemas afetados provê pistas que podem ajudar a levantar considerações práticas (restrições, autonomia, etc.).

6.3.1.3 Tarefa 3 – Identificar Entidades de Negócio e Fluxo de Informação

Para estabelecer apropriadamente o escopo da análise é necessário, em complemento à Tarefa “Definir Escopo da Análise”, que os requisitos de negócio que se deseja atender possuam modelo de dados definidos e maduros.

Isso significa uma clara definição do fluxo de informação (e todos os subprocessos) que devem estar documentados com detalhes suficientes.

Essa documentação de fluxo de informação será usada como ponto de partida do processo de modelagem de serviços no subprocesso “Modelar Serviços Candidatos”.

6.3.1.4 Processo 1 – Modelar Serviços Candidatos

Esse subprocesso contém as seguintes tarefas, ilustradas na Figura 6-9:

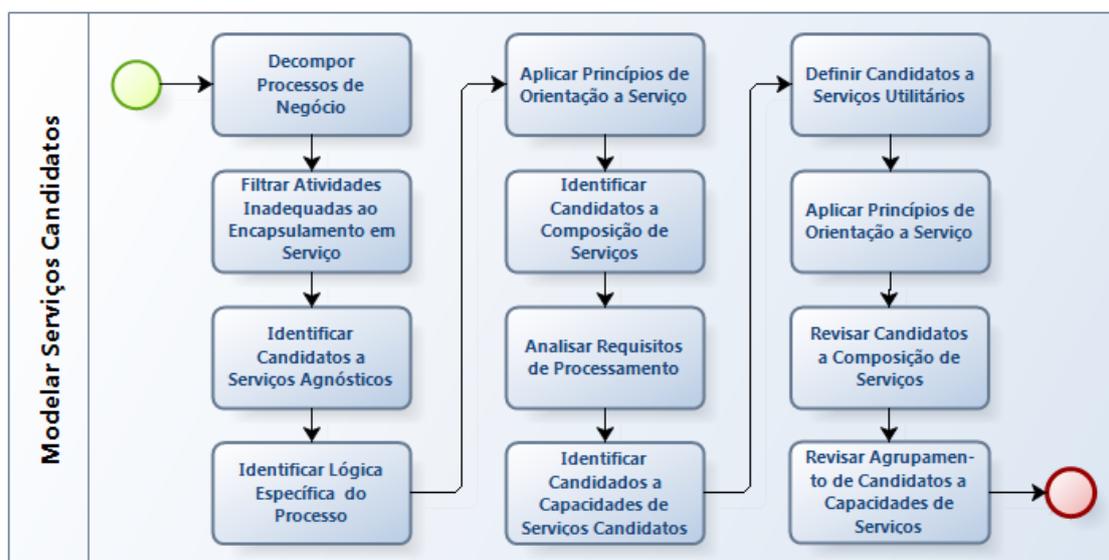


Figura 6-9 – Subprocesso Modelar Serviços Candidatos

6.3.1.4.1 Tarefa 1 – Decompor Processos de Negócio

Nesta tarefa, o processo de negócio documentado é decomposto em uma série de passos de processo granulares. É importante que a lógica de fluxo de trabalho seja decomposta na

representação de menor granularidade possível. Isso frequentemente difere do nível de granularidade no qual os passos de processo são inicialmente documentados.

6.3.1.4.2 Tarefa 2 – Filtrar Atividades Inadequadas ao Encapsulamento em Serviços

Algumas tarefas dentro de um processo de negócio podem ser facilmente identificadas como pertencentes à lógica que não deve ser encapsulada em um candidato a serviço. Exemplos incluem:

- Tarefas manuais de processo que não podem, ou não devem, ser automatizados.
- Tarefas de processos executados por lógica legada existente para a qual o encapsulamento por um candidato a serviço não é uma opção.

6.3.1.4.3 Tarefa 3 – Identificar Candidatos a Serviços Agnósticos

Dependendo da natureza da funcionalidade representada em cada passo, defina alguns grupos preliminares de passos relacionados.

Usando modelos de serviço estabelecidos, tente definir grupos que representem contextos funcionais agnósticos ao processo de negócio.

Por exemplo, para contextos de serviço de entidade, use modelos de negócio corporativo existentes, tais como modelos de entidades de negócio ou modelos lógicos de dados.

Cada contexto representa um candidato a serviço agnóstico em potencial.

Os passos são agrupados para encapsulamento potencial dentro desses candidatos a serviços. Cada passo representa um candidato a capacidade de serviço em potencial. Note que essas definições são ainda preliminares e estão sujeitas a mudanças.

6.3.1.4.4 Tarefa 4 – Identificar Lógica Específica do Processo

Os passos restantes devem representar lógica que é específica a um processo de negócio pai (entre entidades) e, portanto, adequados para encapsulamento por serviços de tarefa. Exemplos comuns incluem:

- Regras de negócio específicas do processo;
- Lógica condicional;
- Lógica de exceção;

- Lógica de sequência.

Note que esses passos tendem a não se tornar candidatos a capacidades de serviço.

6.3.1.4.5 Tarefa 5 – Aplicar Princípios de Orientação a Serviço

Os princípios de SOA que têm significância especial, durante a modelagem de serviços, são os de Capacidade de Reuso de Serviço e de Autonomia de Serviço:

- Considerações de reuso ajudam a tornar os passos dentro de contextos agnósticos em ações mais genéricas.
- Considerações de autonomia relacionam-se com os sistemas legados afetados, identificados anteriormente.
- Visibilidade de Serviço (capacidade de descoberta) também é levada em consideração para propósitos de meta anotação.

6.3.1.4.6 Tarefa 6 – Identificar Candidatos a Composições de Serviços

Identifique um conjunto de cenários mais comuns que podem ocorrer dentro das fronteiras do processo de negócio. Para cada cenário, siga os passos de processamento requeridos, como eles existem correntemente. Deve-se assegurar que estejam incluídas condições de falha que envolva lógica de tratamento de exceção, como parte dos cenários escolhidos.

6.3.1.4.7 Tarefa 7 – Analisar Requisitos de Processamento

O foco até agora esteve na lógica do processo de negócio, de modo que existe uma tendência natural a enfatizar as definições de candidatos a serviços de negócio. Esse passo requer que o estudo dos requisitos de processamento subjacentes de cada capacidade de serviço definida até agora. Isso pode revelar a necessidade por candidatos a capacidades de serviço não negociais adicionais.

6.3.1.4.8 Tarefa 8 – Identificar Candidatos a Capacidades de Serviços Utilitários

Essa atividade consiste em decompor cada requisito de processamento identificado em uma série de passos de menor granularidade. Esses passos tornam-se a base para candidatos a capacidades de serviço em potencial. Deve-se ser explícito sobre como se denomina esses passos, de modo que eles referenciem a função que estão executando. O

passo do processo de negócio para o qual essa função está sendo identificada idealmente não é referenciado.

6.3.1.4.9 Tarefa 9 - Definir Candidatos a Serviços Utilitários

Nessa tarefa, os passos de processamento são agrupados de acordo com um contexto de processamento predefinido. Com candidatos a serviços não comerciais, o contexto primário é uma relação lógica entre candidatos a capacidades de serviço. O resultado é um conjunto de candidatos a serviços utilitários.

6.3.1.4.10 Tarefa 10 – Aplicar Princípios de Orientação a Serviço

Repete-se a Tarefa “Aplicar Princípios de Orientação a Serviço”, avaliando considerações de orientação a serviço a cada novo candidato a serviço utilitário. A ênfase no reuso e na autonomia é tão importante para entrega de serviços utilitários quanto para a entrega de serviços de negócio. Autonomia, em particular, é uma consideração chave para serviços utilitários que atuam como serviços cápsula para ambientes legados.

6.3.1.4.11 Tarefa 11 – Revisar Candidatos a Composição de Serviços

Os cenários originais que foram identificados na Tarefa “Identificar Candidatos a Composições de Serviço” são revisitados. Desta vez, entretanto, incorpore também os novos candidatos a serviços utilitários. Isso pode resultar no mapeamento de atividades de serviços elaboradas. Deve-se manter o histórico de como candidatos a serviços de negócio relacionam-se com candidatos a serviços utilitários subjacentes durante este exercício.

6.3.1.4.12 Tarefa 12 – Revisar Agrupamento de Candidatos a Capacidades de Serviços

O mapeamento de cenários de atividades na Tarefa “Revisar Candidatos a Composição de Serviços” geralmente levantará novas considerações. Isso pode resultar na necessidade de refinar e alterar o agrupamento e definição dos candidatos a capacidades de serviço que foram previamente definidos. Isso também revelará a quantidade de serviços que precisam participar em um candidato a composição de serviços.

6.3.2 Heurísticas para Análise Orientada a Serviço

Esta seção apresenta heurísticas para análise de serviços que devem ser consideradas pelo analista ao identificar e modelar serviços para uma iniciativa SOA.

6.3.2.1 Princípio da Capacidade de Composição de Serviço

6.3.2.1.1 Serviço Autocontido

Heurística: *Priorizar a definição de serviços autocontidos. Caso seja necessário que um serviço invoque outro serviço, considerar o impacto da realização de composição de serviços, bem como o uso de arquiteturas e tecnologias específicas para realização dessas composições (p.e. mediadores (ESB), orquestração de serviços).*

O objetivo, enfim, é minimizar as dependências. Quanto menos dependências, modificações ou erros em um sistema, menos consequências serão produzidas em outro sistema.

6.3.2.1.2 Granularidade

Heurística: *A granularidade definida é influenciada pela capacidade de se realizar composições complexas, segundo critérios de complexidade de projeto e de desempenho.*

Heurística: *Projetar serviços com granularidade baixa para disponibilizar serviços com alto valor ao negócio.*

Serviços são abstrações que escondem dos consumidores os detalhes de implementação. Entretanto, uma granularidade mais fina acarreta normalmente em um tempo de execução maior. Por essa razão, geralmente é melhor ter uma chamada de serviço transferindo todos os dados necessários entre um provedor e seu consumidor ao invés de ter múltiplas chamadas de serviço processando a mesma quantidade de dados.

Serviços com granularidade grossa ajudam a separar a estrutura interna de um provedor de serviço da sua interface externa. Ter um serviço para cada um acessar um atributo de uma entidade (por exemplo, um serviço para cada *setter* e *getter*) pode resultar em objetos distribuídos que aumentam as dependências entre os sistemas distribuídos.

Em termos de necessidades e capacidades, a granularidade geralmente está associada ao detalhamento do nível do problema que está sendo resolvido, ou seja, projetar num nível mais estratégico versus num nível algorítmico, e definir o nível ótimo não está relacionado em contar o número de interfaces ou o número de tipos de trocas de informação que estão conectadas a uma interface.

Outro problema com a recomendação de serviços grossos tem a ver com considerações de desempenho. Processar uma quantidade de dados demanda tempo e, se o consumidor não precisa desses dados, o tempo empregado para o processamento será em vão.

Para o projeto orientado a serviço, deve-se pensar na granularidade no nível do provedor (quantos serviços são providos) ou no nível da especificação do serviço. Uma especificação de serviço é um agrupamento lógico das operações do serviço ao se considerar o negócio.

Quando as operações dos serviços são projetadas, considere colaborações, cenários de uso e mensagens (número e tamanho delas) que irão trafegar entre provedores e consumidores de serviços. Operações com granularidade grossa fornecem mais valor de negócio e permitem modificações mais fáceis no código. Os parâmetros desse tipo de operação (que usam mensagens para entrada, saída e erro) são menos suscetíveis a erro. Usar parâmetros com granularidade fina ao invés de mensagens, contudo, é geralmente mais descritivo.

Para a granularidade do serviço, deve-se considerar que serviços podem ser compostos. Isso está relacionado com a reutilização e a habilidade de prover novas funcionalidades baseadas em funcionalidades existentes. Um conjunto de serviços em certo nível de granularidade pode ser orquestrado/coreografado e o resultado pode ser outro serviço com granularidade mais grossa. Pode-se, contudo, ter-se tarefas de negócio fornecidas pelos serviços e então ter uma sequência dessas tarefas (um processo de negócio) fornecida também como um serviço.

Por fim, tomar decisões de projeto sobre granularidade de serviço é sempre um *tradeoff* entre desempenho (por exemplo, tamanho e número de mensagens sendo trocadas e a eficiência da implementação) e potencial de reuso. Além disso, SOA contém tanto serviços com granularidade fina (serviços atômicos) quanto grossa (serviços compostos).

6.3.3 Padrões de Projeto para Análise Orientada a Serviço

Os seguintes padrões de projeto devem ser observados como parte do processo de análise orientada a serviço:

- Abstração de Utilitário: corresponde ao modelo serviço utilitário.
- Abstração de Entidade: corresponde ao modelo serviço de entidade.

- Abstração de Processo: corresponde aos modelos serviço de tarefa e serviço de tarefa orquestrada.
- Encapsulamento em Serviço
- Contexto Agnóstico
- Contexto Não Agnóstico
- Capacidade Agnóstica
- Composição de Capacidade
- Recomposição de Capacidade
- Orquestração

6.3.4 Visibilidade do Processo de Desenvolvimento

A Figura 6-10 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-10 – Entradas e Saídas da Etapa Análise Orientada a Serviço

6.4 PROJETO ORIENTADO A SERVIÇO

Entradas de Processo	<ul style="list-style-type: none">• Ordem de Serviço da Fábrica de Serviço• Documento de Definição de Candidatos a Serviço• Documento de Modelagem de Processos de Negócio• Modelo de Informação (Estruturada)• Perfil de Serviço• Arquitetura de Referência SOA• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Contrato de Serviço• Perfil de Serviço (atualizado)• Plano de Testes de Serviço (preliminar)• Manual de Uso de Serviço• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Projetar Serviços (contratos)
Perfis Executores	<ul style="list-style-type: none">• Analista de Testes• Arquiteto SOA• Especialista em Governança SOA
Perfis Envolvidos	<ul style="list-style-type: none">• -

A etapa de projeto em Projetos de SOA tipicamente começa quando uma extensão significativa da análise foi completada. O termo “projeto orientado a serviço” é mais comumente associado com um processo formal para projetar contratos de serviço. Entretanto, esse termo é, também, comumente usado em um sentido mais amplo que engloba o projeto da lógica de serviço.

Em geral, é importante reconhecer que uma característica distinta primária do projeto orientado a serviço é uma abordagem “contrato primeiro” para o projeto de serviços.

Todo o esforço colocado nos processos de análise e modelagem de serviços resulta em uma coleção de candidatos a serviços que estabelecem o ponto de partida para o projeto de serviços. Cada candidato a serviço pode ser usado como entrada para uma instância do processo de projeto orientado a serviço.

Existe um processo diferente para cada um dos modelos de serviço primário. Todos os processos são moldados e estruturados em torno da aplicação de princípios de projeto de orientação a serviço.

Ao contrário do processo de análise orientada a serviço, que usualmente resulta na criação de múltiplos candidatos a serviços, uma iteração do processo de projeto orientado a serviço usualmente resulta na criação de um único contrato de serviço.

Cada um dos seguintes processos de projeto consiste em conjuntos genéricos de tarefas que evidenciam as considerações primárias da criação de contratos de serviço.

Ao projetar serviços como Serviços Web, o resultado final de cada processo de projeto de contrato é uma definição WSDL abstrata.

Como parte de cada definição abstrata de serviço, as seguintes partes são formalmente definidas:

- Definição dos tipos (XML *Schema*) usados para representar as cargas de mensagens (*payloads*);
- Definição de todas as operações (capacidades) do serviço;
- Definição de mensagens de entrada e saída de cada operação;
- Definição dos *namespaces* requeridos;
- Projeto Orientado a Serviço (contrato).

Essa etapa engloba ainda a elaboração do manual de uso de serviços, que, conforme o princípio da abstração de serviço, deverá estar completamente definido a partir do Contrato de Serviço e de sua especificação técnica.

6.4.1 Processos e Tarefas do Projeto Orientado a Serviço

As seções seguintes introduzem os subprocessos da etapa Projeto Orientado a Serviço, estruturados conforme o modelo de serviço utilizado:

- Projetar Serviço de Entidade;
- Projetar Serviço Utilitário;
- Projetar Serviço de Tarefa;
- Projetar Serviço de Tarefa Orquestrada.

Representado na Figura 6-11 a seguir:

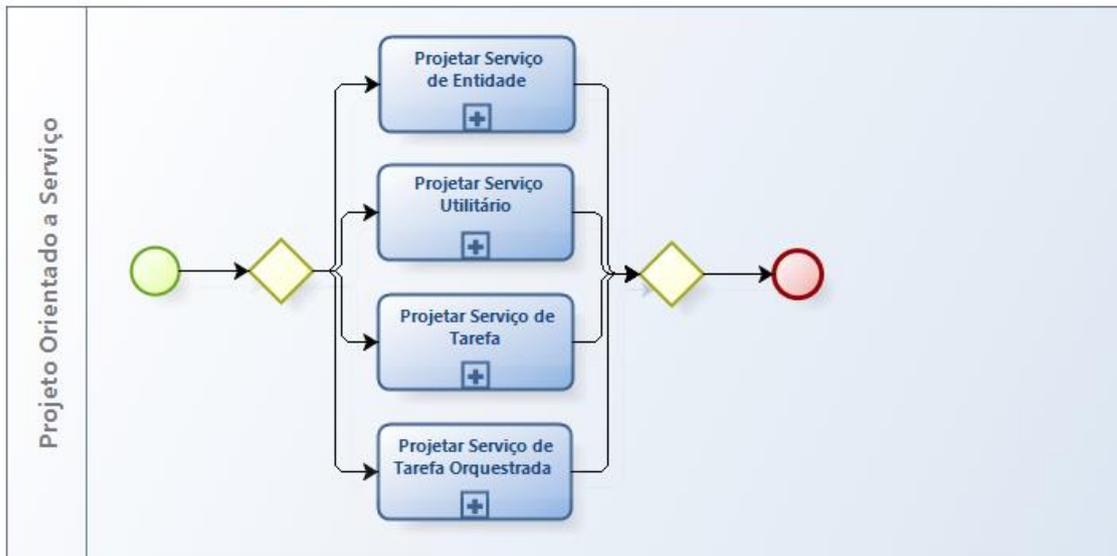


Figura 6-11 – Etapa Projeto Orientado a Serviço

6.4.1.1 Processo 1 – Projetar Serviço de Entidade

Um serviço de entidade é baseado em um contexto funcional que corresponde a uma ou mais entidades de negócio dentro de uma organização.

Serviços de entidade são agnósticos em relação a processos de negócio e são construídos para reuso e composição. Eles são os tipos de serviços menos influenciados pelos demais. Para esses serviços, determinar o correto nível de granularidade e serviço é crítico.

As tarefas deste subprocesso estão ilustradas na Figura 6-12.

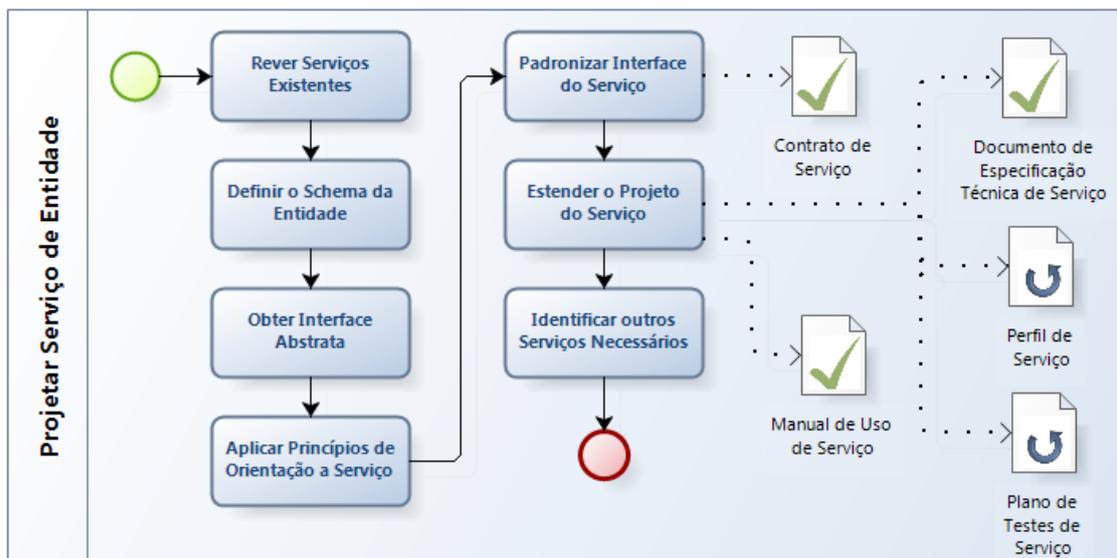


Figura 6-12 – Subprocesso Projetar Serviço de Entidade

6.4.1.1.1 Tarefa 1 – Rever Serviços Existentes

O primeiro passo na concepção de um novo serviço é confirmar se ele é mesmo necessário, pois outros serviços podem já estar proporcionando alguma ou todas as funcionalidades identificadas. Eles podem, também, já ter estabelecido um contexto adequado no qual esses novos candidatos a operação podem ser implementados (como novas operações em serviços existentes).

6.4.1.1.2 Tarefa 2 – Definir o Schema da Entidade

É recomendado começar o projeto de interface de serviço com uma definição formal das mensagens que o serviço necessita processar. Para isso, é necessário formalizar as estruturas de mensagem que são definidas dentro da área “*types*” do WSDL.

Mensagens SOAP transportam dados de *payload* dentro do corpo (seção “*Body*”) do envelope SOAP. Esses dados precisam ser organizados e tipados em esquemas XSD.

6.4.1.1.3 Tarefa 3 – Obter Interface Abstrata

Em seguida, o candidato a operação de serviço proposto deve ser analisado. Para definir uma interface inicial do serviço, as seguintes etapas devem ser seguidas:

- Assegurar que cada operação candidata seja adequadamente genérica e reusável, garantindo que a granularidade da lógica encapsulada seja apropriada. Estudar as estruturas de dados definidas no Passo 2 – Definir o *Schema* da Entidade e estabelecer o conjunto de nomes das operações;
- Criar a área “*portType*” (ou “*interface*”) dentro do documento WSDL e preenchê-la com as construções das operações que correspondam a operação candidata;
- Formalizar a lista de valores de entrada e saída necessários para acomodar o processamento da lógica de cada operação. Isto é feito definindo-se as construções de mensagens que fazem referência aos tipos de esquema XSD dentro dos elementos “*part*”.

6.4.1.1.4 Tarefa 4 – Aplicar Princípios de Orientação a Serviço

Os princípios de reuso e autonomia são, de certa forma, partes naturais do modelo de projeto de entidade, no qual as operações expostas pelos serviços de entidade pretendem

ser inerentemente genéricas e reusáveis e, também, porque o uso da declaração de importação é incentivada para o reuso de esquemas e criação de definições de WSDL modulares.

Como serviços de entidade, muitas vezes, precisam ser compostos por uma camada de serviço pai e por eles contarem com a camada de aplicação para realizar sua lógica de negócios, a sua autonomia é geralmente bem definida. A menos que esses serviços, governados por um controlador de entidade, tenham requisitos de processamento incomuns ou imponham algum tipo de dependências, mas, em geral, os serviços de entidade mantêm sua autonomia.

Por razões semelhantes às descritas acima, a independência de estado também é relativamente controlável.

Capacidade de descoberta do serviço é importante tanto para o projeto de serviços de entidade quanto para sua utilização pós-implantação. Como mencionado na tarefa “Rever Serviços Existentes”, é preciso garantir que um projeto de serviço não implemente lógica já existente. Um mecanismo eficaz de descoberta tornará essa atividade muito mais simples. Da mesma forma, uma medida que pode ser tomada para fazer um serviço mais fácil de ser descoberto é alimentá-lo com detalhes de metadados usando o elemento *documentation*, informando clara e objetivamente as características do serviço.

6.4.1.1.5 Tarefa 5 – Padronizar a Interface do Serviço

Dependendo das necessidades, esse passo pode envolver uma série de tarefas de projeto. A seguir, são apresentadas algumas ações que podem ser usadas para conseguir um projeto de serviço padronizado e simplificado:

- Rever e aplicar *guidelines* e padrões de projeto;
- Rever o projeto de serviço para suportar novas características de SOA.

6.4.1.1.6 Tarefa 6 – Estender o Projeto do Serviço

O processo de modelagem de serviços tende a concentrar-se nos requisitos de negócios mais evidentes. Como promover o reuso é sempre incentivado, muitas vezes fica para o processo de projeto, a tarefa de garantir que uma quantidade suficiente de funcionalidades

reusáveis seja construída dentro de cada serviço. Isto é especialmente importante para os serviços de entidade, como um conjunto completo de suas operações mais comuns, normalmente é esperado pelos consumidores potenciais desses serviços.

Esta etapa envolve a realização de uma análise especulativa de quais outros tipos de funcionalidades esse serviço, dentro de seu contexto funcional pré-definido, deve oferecer.

Existem duas maneiras comuns para implementar uma nova funcionalidade:

- Adicionar novas operações;
- Adicionar novos parâmetros para operações existentes.

Embora a última opção possa simplificar interfaces de serviço, ela também pode se tornar não intuitiva, na medida em que muitos parâmetros associados a uma operação podem aumentar a complexidade de compreensão de suas capacidades e, dessa forma, exigir que os potenciais consumidores do serviço precisem conhecer, de forma mais detalhada, o serviço para efetivamente usá-lo.

Adicionar operações é um meio simples e direto de fornecer funções evidentes associadas à entidade. O conjunto clássico de operações para um serviço de entidade inclui inserir, atualizar, excluir e buscar.

Quando tarefas completamente novas são definidas, elas podem ser incorporadas por novas operações que seguem os mesmos padrões de projeto das já existentes. Se novos requisitos funcionais, que se relacionem com operações existentes, são identificados, um método comum de extensão dessas operações é o de adicionar valores de entrada e de saída. Isto permite a uma operação, receber e transmitir um conjunto de combinações de mensagens.

Porém, cuidados devem ser tomados, para não complicar muito as operações visando potencial reuso. Em geral, é aconselhável submeter qualquer nova funcionalidade proposta para um processo de análise separado.

Observe que, após a identificação de novas operações, as tarefas de 1 a 5 necessitam ser repetidas para moldar e padronizar adequadamente as extensões adicionadas.

6.4.1.1.7 Tarefa 7 – Identificar outros Serviços Necessários

Na análise orientada a serviços, o processo de modelagem de serviços deve ter identificado alguns serviços utilitários chaves, mas pode não ter sido possível identificar todos.

Agora que há um projeto real para esse novo serviço, onde cada uma de suas operações pode e deve ser estudada mais detalhadamente. Fazendo isso, é possível determinar se serviços utilitários adicionais são necessários para executar a funcionalidade exposta. Se for necessário criar novos serviços utilitários, deve-se verificar se já existem ou se devem ser adicionados à lista de serviços que deverão ser entregues como parte da solução.

6.4.1.2 Processo 2 – Projetar Serviço Utilitário

Serviços utilitários encapsulam lógica agnóstica não negocial que é, idealmente, altamente reutilizável e com alta capacidade de recomposição. Ao contrário do projeto de serviços de negócio, o projeto de serviços utilitários geralmente não requer *expertise* de análise de negócio.

Devido a considerações práticas e especificidades tecnológicas que precisam ser levadas em consideração, serviços utilitários podem ser os serviços mais difíceis de projetar.

Serviços Utilitários estabelecem uma camada não negocial que representa vários recursos tecnológicos. As tarefas deste subprocesso estão ilustradas na Figura 6-13.

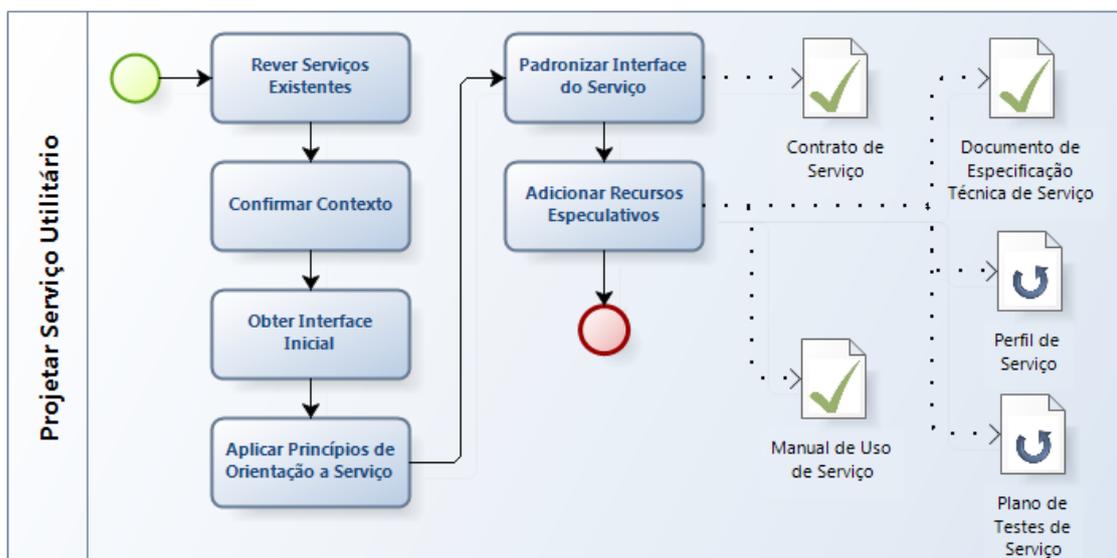


Figura 6-13 – Subprocesso Projetar Serviço Utilitário

6.4.1.2.1 Tarefa 1 – Rever Serviços Existentes

Evitar redundância em serviços utilitários é uma tarefa mais complexa do que em outros tipos de serviços reutilizáveis. Os serviços de entidade se beneficiam de documentação clara e estruturada, como por exemplo, modelos de negócios, o que não acontece para serviços utilitários. Com isso, esforços extras de descoberta são essenciais, como rever o inventário existente em busca de qualquer lógica similar.

6.4.1.2.2 Tarefa 2 – Confirmar Contexto

É importante que o agrupamento de operações candidatas proposto pelos candidatos a serviço sejam reavaliados e comparados com projetos de serviços utilitários existentes. Ao reavaliar o contexto do serviço, pode-se descobrir que uma ou mais operações, na verdade, pertencem a outros serviços utilitários.

6.4.1.2.3 Tarefa 3 – Obter Interface Inicial

Avaliar os candidatos a operação do serviço, seguindo os seguintes passos para definir o primeiro corte de sua interface:

- Garantir que a granularidade das partições lógicas, representadas pelos candidatos a operação de um determinado candidato a serviço utilitário, seja adequadamente genérica e reutilizável;
- Documentar os valores de entrada e de saída necessários para o processamento de cada operação candidata e definir as estruturas de mensagem usando construtores de esquemas XSD (que essencialmente estabelecem os construtores “*types*” do WSDL);
- Completar a definição abstrata de serviço adicionando a área “*portType*” (ou “interface”), juntamente com seus construtores “*operation*”, e os construtores “*message*” necessários contendo os elementos “*part*” que fazem referência aos tipos de esquema apropriados.

6.4.1.2.4 Tarefa 4 – Aplicar Princípios de Orientação a Serviço

Este passo enfatiza os quatro princípios de orientação a serviços que não são fornecidos pelo conjunto de tecnologias de Serviços Web: capacidade de reuso de serviço, autonomia de serviço, independência de estado de serviço e capacidade de descoberta de serviço.

Reuso foi discutido no processo de modelagem de serviços e é tratado diretamente na Tarefa 5, onde trabalhamos para tornar o serviço utilitário o mais útil possível para potenciais solicitantes do serviço. No entanto, os candidatos à operação existentes também devem ser revisados para garantir que eles estejam projetados para serem genéricos e reutilizáveis.

Autonomia é a principal preocupação na concepção de serviços utilitários. Deve-se garantir que a lógica base da aplicação, responsável pela execução das operações do serviço, não imponha dependências ao serviço, ou não tenha dependências nelas mesmas.

Independência de estado também pode ser mais difícil de conseguir com os serviços utilitários. Devido a possibilidade de interagir com uma variedade de diferentes plataformas de aplicações, esses serviços estão sujeitos a ambientes de implementação altamente imprevisíveis. A melhor maneira de promover um projeto de serviço utilitário independente de estado é realizar o máximo de análise o mais antecipadamente possível.

A descoberta é importante para garantir que o projeto em questão não se sobreponha com lógica já fornecida por outros serviços utilitários, dessa forma, um estudo cuidadoso dos inventários de serviços existentes se torna crítico.

6.4.1.2.5 Tarefa 5 – Padronizar a Interface do Serviço

É importante que todos os tipos de serviços sejam projetados da mesma maneira. Para isso, deve-se garantir que o resultado da definição WSDL do serviço utilitário baseie-se nos mesmos padrões e convenções utilizadas e definidas nos demais projetos.

6.4.1.2.6 Tarefa 6 – Adicionar Recursos Especulativos

Para entregar serviços utilitários altamente reutilizáveis, deve-se analisar a possibilidade de acrescentar funcionalidades ao projeto de serviço em questão, mesmo não fazendo parte da

primeira relação de requisitos. Essas novas funcionalidades podem afetar as operações existentes ou pode resultar na adição de novas operações.

6.4.1.3 Processo 3 – Projetar Serviço de Tarefa

Projetar contratos de serviços de tarefas requer significativamente menos esforço que os dois processos de projeto anteriores. Não se espera obrigatoriamente que serviços de tarefa de sejam reutilizáveis e os seus contratos de serviço são, frequentemente, bem menores do que os demais tipos de serviço.

Serviços de tarefa são usualmente projetados por último por causa da sua dependência com outros serviços para executar lógica de composição de serviços.

Serviços de Tarefa estabelecem uma camada de processos de negócio responsável por compor outros serviços de modo a executar uma determinada tarefa.

O subprocesso de projetar um serviço de tarefa é ilustrado na Figura 6-14.

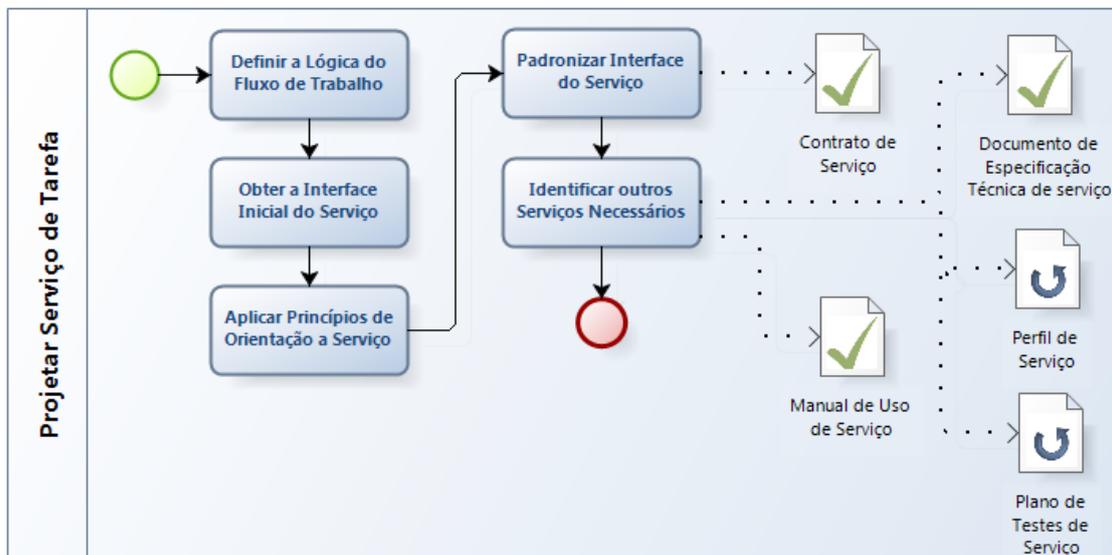


Figura 6-14 – Projetar Serviços de Tarefa

6.4.1.3.1 Tarefa 1 – Definir a Lógica do Fluxo de Trabalho

Serviços de tarefa, normalmente, contém a lógica de fluxo de trabalho incorporada que é utilizada para coordenar uma composição de serviço subjacente. Assim, o primeiro passo, é definir essa lógica para cada cenário possível de interação que possa ser imaginado.

Como o serviço de tarefa é projetado depois dos projetos de serviços de entidade e utilitário terem sido concluídos, esses documentos de cenário devem ser revistos e transformados em modelos concretos de interação de serviços. Isso irá prover informações que auxiliarão na determinação de tipos, operações e formatos de mensagens.

6.4.1.3.2 Tarefa 2 – Obter a Interface Inicial do Serviço

Os candidatos a operação dos serviços são utilizados para obter um conjunto de operações correspondentes.

Ao contrário dos processos anteriores de projeto, a fonte da qual a interface de serviço foi extraída, inclui também os diagramas de atividade e da lógica de fluxo de trabalho documentados na Tarefa 1 – Definir a Lógica do Fluxo de Trabalho. Esta informação nos dá uma boa idéia sobre quais operações adicionais o serviço de tarefa poderá exigir.

Documente ainda os valores de entrada e saída requeridos para o processamento de cada operação e preencha a seção “*types*” com os tipos de esquema XSD necessários para processar as operações.

Construa, finalmente, a definição do WSDL, criando a área “*portType*” (ou “*interface*”), inserindo as construções de operações identificadas. Em seguida, adicione as construções de mensagens necessárias contendo os elementos que fazem referência aos tipos de esquema apropriados.

6.4.1.3.3 Tarefa 3 – Aplicar Princípios de Orientação a Serviço

Embora serviços de tarefa tenham menores probabilidades de reuso que os de entidade e utilitário, por usualmente terem encapsuladas lógicas de fluxo de trabalho específicas para um processo de negócio, a possibilidade ainda existe. Considere o potencial de reutilização da lógica a ser encapsulada em outros processos correlatos e dentro do próprio processo.

Como quase sempre agem como controlador (*parente controller*) de serviços em composições, a autonomia de serviços de tarefa geralmente é dependente da autonomia dos serviços subjacentes. Um estado consistente de autonomia pode, portanto, ser um desafio para manter, mas deve sempre ser analisado.

Serviços de tarefa contêm a lógica de fluxo de trabalho que pode impor dependências de processamento na composição de serviços. Isso pode levar à necessidade de gestão do “estado”. No entanto, a utilização de mensagens de estilo de documento SOAP permite ao serviço delegar a persistência de algumas ou todas essas informações de “estado” para a própria mensagem.

É sempre importante que serviços possam ser descobertos, mas essa necessidade para serviços centrados em tarefa não é tão premente como os demais serviços mais genericamente reutilizáveis. Independentemente disso, serviços centrados em tarefa podem ser reutilizados e, sua existência e características, devem ser passíveis de descoberta para os potenciais consumidores e solicitadores.

6.4.1.3.4 Tarefa 4 – Padronizar Interface do Serviço

Nesta tarefa, deve se incorporar os padrões de projeto e orientações existentes. Deve-se assegurar que quaisquer características de SOA definidas sejam totalmente suportadas pelo projeto de interface do serviço.

6.4.1.3.5 Tarefa 5 – Identificar outros Serviços Necessários

Para executar a sua parte na lógica de processo de uma solução, serviços de tarefa podem compor uma aplicação. Portanto, a implementação de uma interface de serviço de tarefa, exige que qualquer necessidade das camadas de serviços subjacentes esteja disponível para apoiar as necessidades de processamento de suas operações.

Como este é o último dos subprocessos de projeto de serviços, todos os serviços de apoio necessários precisam ser identificados. Podem consistir em serviços já existentes e/ou serviços que acabamos de projetar durante o processo. O projeto da lógica do processo inserido no serviço centrado em tarefa pode revelar a necessidade de serviços adicionais que ainda não haviam sido considerados.

6.4.1.4 Processo 4 – Projetar Serviço de Tarefa Orquestrada

O projeto de serviços de tarefa orquestrada é também referido como “projeto orientado a serviço de processos de negócio” e serviços de tarefa orquestrada são também chamados de “serviços de processos (de negócio)”.

Uma distinção primária entre um serviço de tarefa orquestrada e um serviço de tarefa regular é o escopo da lógica de processo que está sendo encapsulada.

Serviços de tarefa orquestrada tendem a conter lógica de composição de serviços mais complexas e, às vezes, de longo tempo de execução.

É um serviço de tarefa que reside em uma plataforma de orquestração, o que significa que a lógica de serviços de tarefa para múltiplos processos de negócio estará provavelmente centralizada em uma localização.

Existem ferramentas de modelagem que permite que analistas técnicos e arquitetos criem, graficamente, os diagramas de processos de negócios que representam os seus requisitos lógicos de fluxo de trabalho, ao mesmo tempo em que faz autogeração de código.

O resultado é um diagrama que expressa a visão dos analistas do processo e seu código executável, que já pode ser encaminhado aos arquitetos e desenvolvedores. Na Figura 6-15 esse subprocesso é representado:

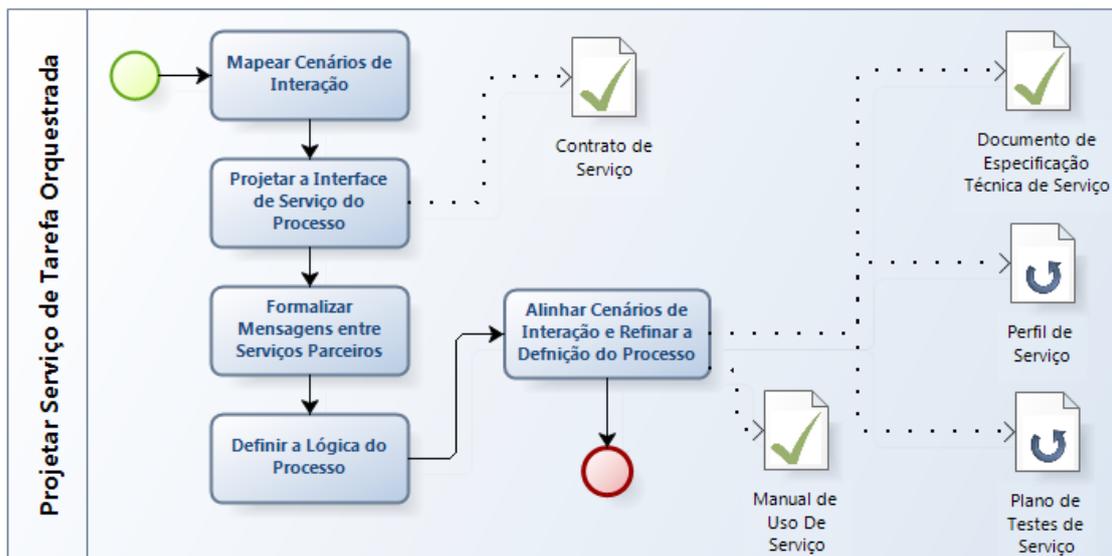


Figura 6-15 – Subprocesso Projetar Serviço de Tarefa Orquestrada

6.4.1.4.1 Tarefa 1 – Mapear Cenários de Interação

Usando as informações reunidas até o momento, podemos definir os requisitos de troca de mensagens do serviço de tarefa orquestrada:

- A lógica de fluxo de trabalho disponível, produzida durante o processo de modelagem do serviço;
- O serviço de tarefa orquestrada candidato;
- Os projetos de serviço existentes.

Essas informações são usadas para formar a base de uma análise, durante a qual, todos os cenários possíveis de interação entre serviços de tarefa orquestrada e serviços parceiros são mapeados. O resultado é uma série de requisitos de processamento que irão formar a base do projeto de serviço de tarefa orquestrada.

6.4.1.4.2 Tarefa 2 – Projetar a Interface de Serviço de Tarefa Orquestrada

Agora que compreendemos os requisitos de troca de mensagens, podemos proceder para a definição de serviço para o serviço de tarefa orquestrada. Quando trabalhando com ferramentas de modelagem de processos, a definição do WSDL do serviço de tarefa orquestrada é tipicamente gerada automaticamente, porém sendo possível editar o código fonte de marcação ou até mesmo importar a própria definição WSDL.

De qualquer forma, é melhor rever a definição WSDL que está sendo usada e alterá-la conforme necessário, o que pode ser feito usando-se o seguinte roteiro:

- Documentar os valores de entrada e saída requeridos para processamento de cada operação e alimentar a seção “*types*” com os tipos de esquemas XSD requeridos para processar as operações. Mover as informações do esquema XSD para um arquivo separado, se necessário;
- Construir a definição WSDL com a criação da área “*portType*” (ou “*interface*”), inserindo as construções “*operation*” identificadas. Adicionar, então, as construções “*message*” necessárias contendo os elementos “*part*” que referenciam os tipos de esquemas apropriados. Adicionar as convenções de nomenclatura alinhadas com as utilizadas pelas definições de outros WSDL’s;
- Adicionar meta informação via o elemento “*documentation*”;
- Aplicar padrões adicionais de projeto.

6.4.1.4.3 Tarefa 3 – Formalizar Mensagens entre Serviços Parceiros

Agora se inicia a definição do processo WS-BPEL, ou BPMN 2.0, estabelecendo detalhes dos serviços com os quais o serviço de tarefa orquestrada irá interagir.

6.4.1.4.4 Tarefa 4 – Definir a Lógica do Processo

Essa etapa é o processo em si, que exige que toda a inteligência do fluxo de trabalho existente seja transposta e implementada através da definição do processo WS-BPEL, ou BPMN 2.0.

6.4.1.4.5 Tarefa 5 – Alinhar Cenários de Interação e Refinar a Definição do Processo

Esse passo é opcional e sugere a execução de duas tarefas específicas: revisitar os cenários originais de interação criados na Tarefa 1 – Mapear Cenários de Interação e rever a definição do processo WS-BPEL, ou BPMN 2.0, para verificar oportunidades de otimização.

Trazer os cenários de interação em alinhamento com a lógica do processo expressa na definição do processo WS-BPEL, ou BPMN 2.0, fornece uma série de benefícios, incluindo:

- Os mapas de interação de serviços (como diagramas de atividades ou em qualquer formato que os tenha criado) são uma parte importante da documentação da solução e é útil para futuras manutenções e transferência de conhecimento;
- Os mapas de interação de serviços fazem bons casos de teste e podem poupar testadores de ter que realizar uma análise especulativa;
- A implementação da lógica original de fluxo de trabalho como uma série de atividades WS-BPEL, ou BPMN 2.0, pode ter introduzido lógica nova ou expandida no processo. Uma vez comparada com os cenários de interação existentes, a necessidade de interações adicionais de serviços pode surgir, levando a descoberta de novas condições de falha ou exceção que, então, podem ser abordadas na definição do processo WS-BPEL, ou BPMN 2.0.

Refinando a definição do processo, faz-se possível:

- Consolidar ou reestruturar atividades para atingir melhorias de desempenho;

- Agilizar o código de marcação para facilitar a manutenção;
- Descobrir recursos que anteriormente não eram considerados.

6.4.2 Heurísticas para Projeto Orientado a Serviço

Esta seção apresenta heurísticas para projeto orientado a serviços (projeto de contratos) que devem ser consideradas pelo projetista ao projetar contratos de serviços para uma iniciativa SOA.

6.4.2.1 Princípio da Padronização de Contrato de Serviço

6.4.2.1.1 Abordagem “Contrato Primeiro”

Heurística: *Serviços dentro de um mesmo inventário de serviços estão em conformidade com os mesmos padrões para projeto de contrato.*

Heurística: *Evitar o uso de ferramentas de geração automática de contrato a partir de lógica implementada (legada).*

Serviços expressam seus propósitos e capacidades através de um serviço contrato padronizado. Isso implica em grande ênfase no projeto de contratos de serviços advoga o princípio de projeto “contrato primeiro”. Seu propósito é assegurar que a maneira na qual os serviços expressam funcionalidades e representam tipos de dados é mantida relativamente alinhada e independente (desacoplada) do projeto da lógica de solução.

6.4.2.1.2 Padronizações de Projeto

Heurística: *A arquitetura de referência SOA deve estabelecer diferentes padronizações de projeto (design standards), com objetivo de garantir a uniformidade dos produtos de um projeto de SOA.*

Heurística: *Serviços dentro de um mesmo inventário de serviços devem estar em conformidade com os mesmos padrões para projeto de contrato (Princípio da Padronização de Contratos).*

Há dois tipos de padronização de contratos:

- Padronização de Representação de Dados (ou Modelo de Dados).

- Padronização da Expressão Funcional.

6.4.2.1.3 Modelos de dados padronizados (modelo canônico)

Heurística: *Definições WSDL devem compartilhar XML Schemas comuns. Esses XML Schemas definem a estrutura de dados de entrada e/ou saída para cada capacidade de serviço.*

Heurística: *Antes de criar um tipo de dado no projeto do serviço, verificar os tipos de dados existentes no modelo canônico.*

Um objetivo fundamental dessa heurística consiste em evitar transformações através da padronização da representação de dados entre contratos de serviços.

Recomenda-se ter um conjunto básico de tipos fundamentais que possam ser compostos com outros tipos (estruturas, *records*, etc.) e sequências (*arrays*).

É necessário tomar cuidado com enumerações, herança e polimorfismo (mesmo quando XML suportar isto). Em resumo, deve-se ser conservador com os tipos de dados.

A harmonização dos tipos de dados é um grande problema em ambientes distribuídos. Não há dúvida que tudo se torna muito mais fácil se os tipos de dados estão compartilhados em todos os sistemas. Segundo Josuttis (2007), quando a orientação a objetos se tornou famosa, ter um modelo de objeto de negócio comum se tornou um objetivo geral. Entretanto, essa abordagem se tornou um problema quando aplicada a grandes sistemas. A primeira razão é organizacional: é muito difícil entrar em acordo sobre tipos de dados comuns. Existem visões e interesses que variam muito de sistema pra sistema, uma vez que os diversos sistemas possuem diferentes proprietários. Cedo ou tarde o preço da harmonização se torna muito alto, pois é muito caro manter todos os sistemas sincronizados e, quando um novo serviço for adicionado, a heterogeneidade será introduzida novamente. Se tipos de dados não estão harmonizados, então é preciso criar mapeamentos entre eles.

Apesar da utilização de tipos de dados heterogêneos diminuir o acoplamento, a existência de uma diretriz organizacional para definir um modelo canônico de dados pode ser útil para manter a reusabilidade. Para a definição do contrato dos serviços, contudo, a

homogeneização dos tipos de dados dificulta a criação de visão sobre os tipos de dados e pode aumentar desnecessariamente o tamanho das mensagens entre os serviços.

6.4.2.1.4 Expressão Funcional padronizada (convenções de nomenclatura)

Heurística: *Serviços de entidade serão nomeados em acordo com as entidades de negócio correspondentes.*

Heurística: *Os nomes de serviços de tarefas serão baseados no processo de negócio que o serviço é responsável por automatizar.*

Heurística: *Nomes de capacidades de serviço incluíram um verbo seguido por um nome.*

Heurística: *Os nomes das capacidades de serviço não podem repetir o nome do serviço.*

Padronização da Expressão Funcional refere-se a ter cada serviço expressando detalhes de seu respectivo contexto funcional usando convenções comuns, o que facilita a descoberta e o reuso de serviços.

6.4.2.2 Princípio do Baixo Acoplamento de Serviço

6.4.2.2.1 Acoplamento Consumidor-para-Contrato

Heurística: *Reduzir o acoplamento entre o contrato de serviço e o consumidor de serviço.*

Um provedor não deve ter conhecimento sobre uma instância específica de um consumidor e vice-versa. Trocar um provedor ou a implementação de um serviço deve ter efeitos mínimos nos consumidores.

Uma consideração central de projeto para o estabelecimento de relações entre consumidor e serviço é se o consumidor de serviço de fato se acopla ao contrato de serviço. Deve-se enfatizar fortemente que um consumidor de serviço apenas se acople ao contrato de serviço (acoplamento Consumidor-para-Contrato).

A forma negativa fundamental de acoplamento de consumidor ocorre quando o programa consumidor é projetado para ignorar o contrato de serviço, de modo a acessar partes da implementação subjacente do serviço diretamente (acoplamento Consumidor-para-Implementação).

6.4.2.2.2 Acoplamento Lógica-para-Contrato

Heurística: *Separar a especificação do serviço da sua implementação, descrevendo um modelo técnico (contrato) e um modelo conceitual (requisitos, restrições etc.).*

Heurística: *Reduzir o acoplamento entre o contrato de serviço e sua lógica subjacente.*

Ao projetar serviços, os seguintes tipos de acoplamento devem ser levados em consideração de modo a atingir o nível apropriado de acoplamento global:

- Lógica-para-Contrato (positivo)
- Contrato-para-Lógica (negativo)
- Contrato-para-Funcionalidade (negativo)
- Contrato-para-Implementação (negativo)
- Contrato-para-Tecnologia (negativo)

Tipos negativos de acoplamento de contrato são evitados de modo a maximizar a independência de governança do contrato de serviço. Um contrato de serviço completamente desacoplado pode evitar impor tipos de acoplamento negativos aos consumidores de serviço.

Em última análise, qualquer forma indesejável de acoplamento permitida no projeto do contrato de serviço termina sendo imposta e proliferada através de todos os consumidores de serviço. Isso causa sérios problemas quando serviços agnósticos são largamente reusados.

É importante ter todas as especificações dos serviços de forma independente de uma implementação particular, promovendo o projeto e a evolução independente da lógica de serviço.

Heurística: *Descrever formalmente a especificação do serviço.*

Heurística: *Atualizar especificação do serviço com detalhes da fase de projeto.*

A especificação do serviço deve ser descrita formalmente, tanto em um formato que seja compreensível por máquinas, para que os geradores de código possam ajudar com a implementação do serviço, como também por humanos.

O contrato do serviço deve estar no nível de especificação (não no nível de implementação). Para garantir uma rastreabilidade através dos processos de negócio, é interessante relacionar a especificação do serviço com a modelagem do processo de negócio o qual o serviço apoia.

6.4.2.3 Princípio da Abstração de Serviço

Heurística: Contratos de Serviços contêm apenas informações essenciais e informações sobre os serviços são limitadas ao que está publicado nos contratos de serviços.

Em um nível fundamental, esse princípio enfatiza a necessidade de esconder tanto quanto possível os detalhes subjacentes de um serviço. A informação sobre um serviço é, portanto, limitada ao que está publicado no contrato de serviço. Entretanto, abstração de serviço influencia o conteúdo do contrato de serviço.

Existem quatro tipos de meta abstração:

- Funcional: relaciona-se com metadados que descrevem o que o serviço é capaz de fazer.
- Tecnológica: relaciona-se com metadados que descrevem a implementação técnica da lógica de serviço subjacente.
- Programática: relaciona-se com metadados que descrevem como o serviço realiza suas capacidades.
- Qualidade de serviço: relaciona-se com metadados que descrevem o comportamento, limitações e requisitos de integração do serviço.

A aplicação desse princípio envolve a necessidade de introduzir medidas de controle de acesso às especificações de projeto do serviço e ao código fonte. Isso protege tanto o serviço quanto seus consumidores de formas indesejáveis de acoplamento.

O conteúdo de contratos de serviços pode ser regulado por este princípio, pois este encoraja a racionalização de meta informações e restrições. O propósito por de trás dessas limitações é maximizar a longevidade do contrato de serviço e dar liberdade de evolução da implementação subjacente aos proprietários de serviço.

A aplicação deste princípio introduz a questão da “composição escondida”, onde programas consumidores de serviço não estão cientes que o serviço que eles estão invocando, de fato, encapsula uma composição inteira.

6.4.2.4 Princípio da Visibilidade de Serviço

Heurística: Prover nomes significativos e bem descritivos do serviço e suas operações baseado no domínio do negócio, não na tecnologia. Definir padrões de nomes e descrições de acordo com as diretrizes da organização.

Heurística: Publicar descrição dos serviços em um repositório de contratos.

Heurística: Suplementar as especificações de serviços com metadados comunicativos pelos quais eles podem ser eficientemente descobertos e interpretados.

É desejável que serviços (especialmente serviços agnósticos) possam ser facilmente localizados e entendidos de modo que eles possam ser apropriadamente reusados e recompostos.

Existem dois conceitos relevantes para se obter essa característica:

- Capacidade de descoberta: O processo de buscar e encontrar uma lógica de solução dentro de um ambiente específico é referido como descoberta. Para que algo dentro de uma corporação possa ter capacidade de descoberta, ele precisa ser equipado com meta informação que permitirá que ele seja incluído no escopo dos mecanismos de busca da descoberta. Informação de capacidade de descoberta é essencialmente uma combinação do conteúdo de um contrato de serviço e metadados no registro do diretório correspondente. Algo que possa ser corretamente descoberto é considerado possuidor de uma medida de capacidade de descoberta.

- Capacidade de ser interpretado: Uma vez encontrados por um humano, é importante que o propósito e as capacidades do que foi descoberto possam ser claramente entendidos. Este nível de clareza ou “qualidade de comunicações” é referido como capacidade de ser interpretado.

Os passos envolvidos por um humano para avaliar os resultados de uma consulta de um processo de descoberta e, então, escolher um serviço como sendo capaz de atender requisitos de automação desejados é parte do processo de interpretação.

6.4.3 Padrões de Projeto para Projeto Orientado a Serviço

Os seguintes padrões de projeto devem ser observados como parte do processo de projeto orientado a serviço:

- Contrato Desacoplado
- Centralização de Contratos
- Terminal Oficial

6.4.4 Visibilidade do Processo de Desenvolvimento

A Figura 6-16 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-16 – Entradas e Saídas da Etapa de Projeto Orientado a Serviço

6.5 PROJETO DE LÓGICA DE SERVIÇO

Entradas de Processo	<ul style="list-style-type: none"> • Ordem de Serviço da Fábrica de Serviço • Documento de Modelagem de Processos de Negócio • Modelo de Informação (Estruturada) • Contrato de Serviço • Perfil de Serviço • Plano de Testes de Serviço • Manual de Uso de Serviço • Arquitetura de Referência SOA • Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none"> • Documento de Arquitetura de Serviço • Documento de Arquitetura da Solução (preliminar) • Plano de Testes de Serviço (final) • Plano de Testes da Solução (preliminar) • Perfil de Serviço (atualizado) • Ordem de Serviço da Fábrica de Software • Manual de Uso da Solução (preliminar) • Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none"> • Projetar Lógica Orientada a Serviço
Perfis Executorores	<ul style="list-style-type: none"> • Analista de Testes • Arquiteto SOA
Perfis Envolvidos	<ul style="list-style-type: none"> • Especialista de Governança SOA

Devido à ênfase em análise antecipada, modelagem, projeto e no uso de padronizações de projeto, o estágio de desenvolvimento de um projeto de SOA é muito mais controlado que em projetos tradicionais.

Também, devido à oportunidade de abstração de características proprietárias de serviços, uma dada composição de serviços pode ser composta de serviços construídos usando-se linguagens (de programação) diferentes.

O projeto da lógica de serviço tem por finalidade definir as escolhas de projeto de arquitetura de serviço (e de solução) que qualificam, enquadram e restringem os processos de implementação (codificação) de serviços. De maneira similar aos processos de análise e projeto de serviços, o projeto da lógica de serviço deve ter características distintas para cada tipo de serviço. Vale ressaltar que, em um projeto de solução orientada a serviços, o projeto da solução como um todo se concretiza a partir de composições de serviços individuais e de sua integração com a camada de interface de usuário (camada de

apresentação). Desse modo, esse processo de projeto de lógica (de serviço e de solução) orientada a serviço deve enfatizar o uso e aplicação dos princípios de orientação a serviço.

6.5.1 Processos e Tarefas do Projeto de Lógica Orientada a Serviço

A Figura 6-17 ilustra o processo Analisar com Orientação a Serviço e suas tarefas e subprocessos:

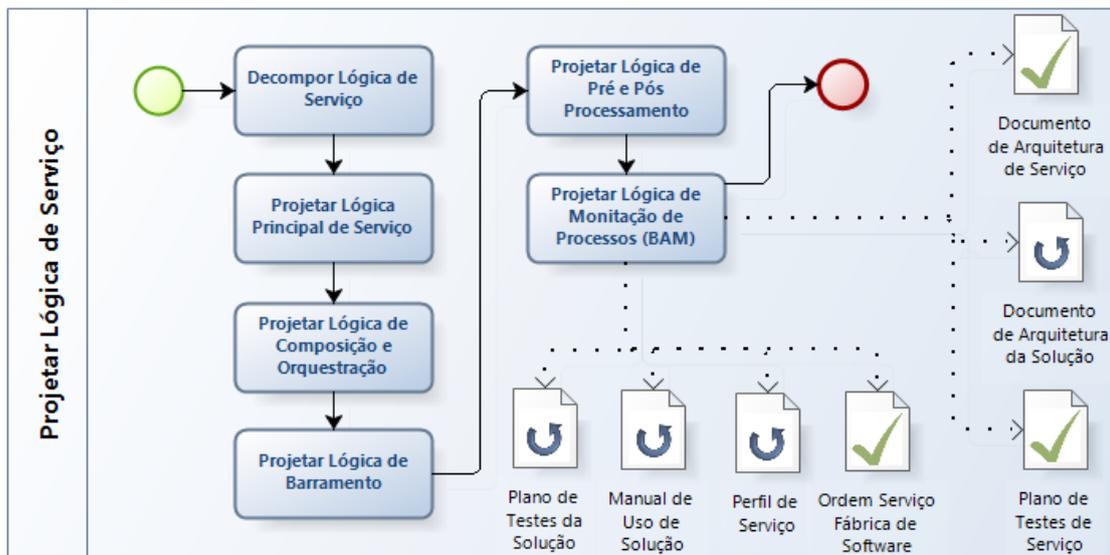


Figura 6-17 – Processo Projetar Lógica de Serviço

6.5.1.1 Tarefa 1 – Decompor Lógica de Serviço

Nesta tarefa são decompostas as principais partes da lógica de serviços, definindo sua arquitetura geral de implementação e implantação na infraestrutura SOA (*middleware*). São definidos os seguintes tipos de lógica, nos quais a lógica de serviço é dividida (tipos adicionais especializados podem ser igualmente requeridos):

- Lógica de serviço a ser implementada diretamente: define a lógica (e sua arquitetura) que será implementada diretamente como um programa de software, muitas vezes publicável como um Serviço Web (SOAP) ou componente, conforme arquitetura de referência. Esse tipo de lógica é a principal parte da lógica de serviços de entidade e serviços utilitários.
- Lógica de composição e orquestração de serviços: define a lógica que será implementada como composição de serviços. Indica a existência de controladores

de composição, gerenciamento de estado e mediação de transações. Esse tipo de lógica é muito comum em serviços de tarefa e serviços de tarefa orquestrada e pode utilizar recursos de composição por orquestração (codificação em BPEL e BPMN 2.0 com uso do orquestrador).

- Lógica de barramento: define lógica que será implementada como lógica de processamento intermediário de mensageria, executada no *Service Bus*. Inclui lógica de transformação (dados, formatos, estrutura, protocolos), enfileiramento, roteamento de mensagens, balanceamento de carga, entre outros. Pode incluir a definição de serviços *proxy*.
- Lógica de pré e pós processamento: lógica adicional que deve ser incluída na execução do serviço. Inclui lógica adicional de barramento, tal como autenticação, autorização e auditoria, uso de políticas, uso de nível de serviço, governança e outros, que não foram tratados na parte funcional. Aqui se projeta ainda as fachadas e cápsulas de legados usadas na arquitetura do serviço. Frequentemente, pode-se empregar arquitetura de agentes de serviços para definição de elementos de lógica de pré e pós processamento de uso comum (ex. agente de autenticação).
- Lógica de monitoração de processos (BAM): lógica de monitoração de processos relacionada ao *Business Activity Monitoring*.

6.5.1.2 Tarefa 2 – Projetar Lógica Principal de Serviço

Nesta tarefa são projetados os principais elementos de lógica que compõem a implementação do serviço. Esta lógica está relacionada com a lógica que implementa o contrato do serviço e que deve, portanto, estar diretamente acoplada a este e desacoplada do resto. Aplica-se padrões de projeto para que essa lógica possa ser adequadamente isolada e desacoplada de outras partes de lógica que compõem a lógica do serviço.

6.5.1.3 Tarefa 3 – Projetar Lógica de Composição e Orquestração

Nesta tarefa são projetados os principais elementos de lógica de composição e orquestração, normalmente com uso de ferramentas para programação do orquestrador. Em especial, são tomadas decisões de centralização de processos (Engines BPEL e BPMN),

centralização de regras (*Business Rules* e *Complex Event Processing* - CEP), gerenciamento e transferência de estado e, mediação de transações.

6.5.1.4 Tarefa 4 – Projetar Lógica de Barramento

Nesta tarefa são projetados os principais elementos de lógica de barramento de serviços, normalmente com uso de ferramentas para programação do *Service Bus*. Em especial, são definidos serviços *proxies* (se houver), regras de transformação de mensagens, enfileiramento, balanceamento de carga, entre outros elementos de lógica que executam no barramento.

6.5.1.5 Tarefa 5 – Projetar Lógica de Pré e Pós Processamento

Nesta tarefa são projetados os elementos adicionais de lógica relacionados ao pré-processamento (antes que a mensagem é entregue ao serviço) e pós-processamento (processamento de mensageria dentro do serviço). Geralmente define lógica a ser implementada no barramento de serviço e através do uso de arquitetura de agentes. Inclui lógica para tarefas comuns, tais como, autenticação, autorização e auditoria, uso de políticas (centralização de políticas), uso de nível de serviço (aplicação de SLA), governança, entre outros.

6.5.1.6 Tarefa 6 – Projetar Lógica de Monitoração de Processos (BAM)

Nesta tarefa são projetados os elementos adicionais de lógica relacionados ao uso de sensores e painéis do *Business Activity Monitoring* (BAM).

6.5.2 Heurísticas para o Projeto de Lógica de Serviço

Esta seção apresenta heurísticas para projeto de serviços que devem ser considerados pelo projetista ao desenvolver o projeto de lógica de serviços para uma iniciativa SOA.

6.5.2.1 Princípio da Padronização de Contrato de Serviço

6.5.2.1.1 Transformações de Dados no ESB

Heurística: *Utilizar transformação de mensagens no barramento para realizar um versionamento implícito dos tipos de dados.*

O versionamento de serviços permite que uma nova versão de um serviço provedor não afete o funcionamento de um consumidor antigo. Contudo, quando o serviço provê algum tipo de dado que será usado por um consumidor, a atualização desse tipo de dado vai influenciar no funcionamento do consumidor e, em alguns casos, exigir que o consumidor recompile seu código em relação ao novo tipo de dado.

Se o provedor introduz uma nova versão de um tipo (adicionando novos atributos, por exemplo), o consumidor terá que se atualizar explicitamente. Caso contrário, o provedor terá que dar suporte às duas versões. Com o versionamento de tipos, o consumidor não terá que fazer nada enquanto as modificações mantenham a compatibilidade.

O versionamento de tipos de dados recai em quatro abordagens:

- Utilizar tipos diferentes para cada versão do tipo de dado: Uma nova versão de um tipo pode ser considerada como um novo tipo de dado. Regras de nomenclatura podem ser utilizadas para nomear versões de tipos. Por consequência, os provedores precisam usar diferentes tipos de dados para fornecer o mesmo tipo de informação e pode ser necessário que mantenham versões de um mesmo consumidor, uma nova versão do tipo e uma versão mais antiga.
- Utilizar o mesmo tipo de dado para diferentes versões do tipo: Neste caso o tipo precisa conter todos os atributos de todas as versões de um mesmo tipo de dado. Todos os serviços irão usar o mesmo tipo, mas eles irão usar apenas os atributos que estão especificados para eles. Isto pode se complicar bastante, pois é preciso documentar quais atributos são válidos para cada versão do serviço e nem sempre diferentes versões serão compatíveis binariamente. Ou seja, pode ser preciso ter certeza que todas as bibliotecas de um processo são compiladas com a mesma versão de um tipo. Outra questão é que se for validar os dados de entrada de acordo considerando determinado tipo, é preciso ter certeza que os atributos adicionais não vão invalidar os dados de entrada.
- Utilizar tipos de dados genéricos: Outra opção de lidar com diferentes versões de tipos de dados é usar apenas código genérico que seja capaz de lidar com qualquer tipo de dado. Neste caso, o formato binário da abordagem anterior não é

impactante. Como consequência, tipos de dados não são importantes em tempo de compilação; todo o processamento acontece em tempo de execução.

- Utilizar listas planas de parâmetros ao invés de dados não estruturados: essa opção não é aplicável quando serviços de alta granularidade enviam dados complexos.

A opção de utilizar apenas tipos de dados genéricos quase não é usada na prática e a opção de compartilhar tipos para diferentes versões é muito perigosa, uma vez que, se o processo não garante que todas as bibliotecas são consistentes (o que é difícil num sistema distribuído), pode resultar em erros e comportamento imprevisto em tempo de execução.

Consumidores de serviço podem fazer sua implementação independente do versionamento dos serviços chamados, usando seus próprios tipos de dados, os quais estão mapeados para o tipo de dado dos serviços quem eles forem usados. Neste caso, os consumidores de serviço devem ter uma camada superior que mapeie tipos de dados externos para tipos internos.

6.5.2.2 Princípio do Baixo Acoplamento de Serviço

6.5.2.2.1 Acoplamento Lógica-para-Contrato

Heurística: Aumentar o acoplamento da lógica do serviço com o contrato de serviço porque isso permite que o contrato de serviço permaneça desacoplado da lógica de serviço.

6.5.2.2.2 Comunicação Assíncrona

Heurística: Para cada serviço, priorizar implementação utilizando comunicação assíncrona. Se não houver restrição de desempenho ou a utilização do padrão assíncrono levar a uma implementação muito complexa, então utilizar padrão síncrono, mas justificando.

Esse tipo de comunicação geralmente compreende um remetente que envia uma mensagem para um destinatário, o qual retornará a resposta para a mensagem. Porém, ao contrário da comunicação síncrona, o remetente, após enviar a mensagem, continuará seu trabalho enquanto a mensagem de resposta não for recebida. Neste caso, o destinatário não precisa estar online no momento do envio da mensagem. Quando o destinatário se torna disponível, a mensagem é entregue a ele para ser processada.

Um problema com comunicação assíncrona ocorre quando o remetente necessita de uma resposta para a mensagem enviada. Como a mensagem de resposta pode demorar a chegar e o remetente continua seu trabalho após o envio da mensagem, é preciso identificar as mensagens para que o remetente identifique a resposta corretamente. O padrão WS-ReliableMessaging pode ser útil em comunicações assíncronas, pois permite diminuir a complexidade do serviço para tratamento do recebimento de mensagens.

6.5.2.2.3 Mediadores

Heurística: *Utilizar mediadores para invocação de serviços.*

Essa forma de baixo acoplamento tem a ver com a forma com que uma chamada de serviço é realizada. Na abordagem ponto-a-ponto o remetente envia a requisição para um sistema específico usando um endereço físico. Essa é a abordagem altamente acoplada. Adicionalmente, se o serviço está fora do ar ou está sobrecarregado de mensagens, mecanismos para gerenciamento de falhas e balanceamento de carga são necessários.

6.5.2.2.4 Verificação Fraca de Tipos

Heurística: *Utilizar verificação de tipos nos mediadores quando contrato dos serviços estiver estável.*

Para que uma infraestrutura SOA (como um ESB) verifique tipos, é necessária alguma informação sobre esses tipos. Se os tipos de dados estão descritos em XML, então o ESB necessita do XMLSchema correspondente. Como consequência, qualquer modificação no tipo não vai afetar somente o consumidor e o provedor, mas também o ESB. Por essa razão, deve-se manter o ESB genérico. Em geral, a mudança nos tipos deveria apenas afetar quem usa o tipo como contrato, mas não quem transfere os dados.

É preciso perceber, contudo, que os tipos de dados são processados pelos sistemas. Então, qualquer inconformidade no tipo de dados recebido por um sistema pode resultar em um processamento errôneo. Assim, é razoável realizar uma verificação de tipos nestes sistemas. O problema principal é quando realizar essa verificação. A grande questão aqui é quão estável o dado está. Digamos que o sistema está sendo utilizado por muito pouco tempo em um ambiente distribuído, então é possível que muitas alterações no formato do tipo de dado ainda poderão ser necessárias por causa de requisitos novos que foram

surgindo. Se, por outro lado, o tipo de dado é considerado estável, a verificação de tipos poderia ser aconselhável.

6.5.2.2.5 Controle de Transações

Heurística: *Definir operações de compensação para as operações dos serviços.*

Controle de transações pressupõe um maior acoplamento entre o consumidor e o provedor do serviço. Em alguns casos, o controle de transação pode ser flexibilizado e ser adotado um controle de compensações. A vantagem da compensação é que a atualização de dados dos sistemas não precisa ser feita de forma síncrona.

Compensações é uma forma interessante de baixo acoplamento. A abordagem comum para resolver problemas de transação é criar um contexto de transação comum usando técnicas como *two-phase commit* (2PC). Com essa abordagem, primeiro realiza-se todas as modificações em todos os serviços que lidam com o dado, exceto o aceite de atualização final do dado. Então, se nenhum serviço sinalizar um problema, então o serviço controlador realiza a atualização em todos os sistemas. 2PC é um dos atributos mais exigidos em *middlewares* distribuídos, mas, na prática, é raramente utilizado em grandes sistemas pois nem todos os sistemas dão suporte a 2PC. O principal problema é que todos os sistemas precisam estar funcionando e têm que fornecer os recursos até que as modificações estejam completas no último sistema. Especialmente quando há acesso concorrente ao dado, isto pode levar a atrasos e *deadlocks*.

A vantagem da compensação é que as atualizações dos sistemas não precisam ser feitas de forma síncrona (alguns sistemas podem até estar fora do ar). O ponto negativo é que se precisa fornecer e chamar explicitamente serviços que revertam a operação realizada por um serviço anterior ou programas para manipular manualmente o erro. Os padrões WS-BPEL e WS-BusinessActivity suportam mecanismos de compensações e podem ser utilizados para compensar operações de serviços.

6.5.2.2.6 Controle da Lógica do Processo

Heurística: *Em ambientes controlados, onde se ter um controlador central não causa problemas de gargalo, utilizar orquestração de serviços para composição de serviços. Caso o controlador central esteja impactando o desempenho da composição, utilize coreografia de serviços.*

Decisões de controle do processo podem também levar a diferentes formas de acoplamento. Ter um componente central controlando toda a lógica do processo cria um gargalo, pois cada sistema envolvido precisa se conectar com esse componente. Falhas neste componente central poderão parar todo o processo.

Por outro lado, ao se ter um controle descentralizado ou distribuído (onde cada componente realiza seu trabalho e conhece quais componentes continuarão depois) evita-se gargalos e, se algum sistema falhar, outros podem continuar trabalhando.

NOTA: A coreografia de serviços lida com uma visão mais global do sistema, aceitando que o controle possa ser distribuído entre diferentes parceiros. A utilização de coreografia faz com que o projeto dos serviços considere mensagens unidirecionais (*one-way messages*) ou eventos, o que, por sua vez, nos leva a considerar o conceito de Arquitetura Orientada a Eventos (EDA – *Event-Driven Architecture*), a qual não será discutida nesse documento.

6.5.2.3 Princípio da Autonomia de Serviço

Heurística: *O projeto de serviços deve considerar o nível de controle sobre seu ambiente de execução subjacente.*

A autonomia de serviço refere-se ao:

- Nível de controle que se tem sobre um serviço em tempo de projeto (autonomia em tempo de projeto);
- Nível de controle que um serviço tem sobre seu ambiente de execução (autonomia em tempo de execução).

Existem diferentes níveis atingíveis de autonomia em tempo de execução, dependendo da extensão do isolamento de uma dada implementação de um serviço.

A aplicação desse princípio depende da quantidade de recursos compartilhados dos quais um serviço pode precisar ou depender. Qualquer parte do ambiente de um serviço sobre o qual este não tenha controle exclusivo pode comprometer a previsibilidade de seu desempenho e comportamento.

Atingir um nível alto de autonomia é especialmente importante no caso de serviços agnósticos, tais como aqueles baseados nos modelos de serviços de entidade e utilitários.

Ao compor serviços, a autonomia é automaticamente reduzida, na medida em que se aproximada dos níveis superiores em uma hierarquia de composição.

6.5.2.3.1 Monitoramento de Serviço

Heurística: *Definir métricas para monitoramento de serviços e utilizar ferramenta de monitoramento de serviços.*

Outra questão importante é a dificuldade em monitorar a quantidade de tráfego que um serviço está recebendo e de onde se não existe um software e métricas de monitoramento de SLA (*Service-Level Agreement*). Se o serviço é utilizado por diferentes clientes de diferentes domínios, é importante particularmente considerar que ele poderá ocorrer uma explosão de tráfego de forma repentina e inesperada devido a composição realizada por clientes fora do seu projeto. Outro problema com invocação direta é que ela adiciona complexidade na sua topografia sem dar visibilidade através de ferramentas.

6.5.2.4 Princípio da Ausência de Estado de Serviço

Heurística: *Evitar criação de serviços que guardam estados.*

Heurística: *Em serviços onde é necessário guardar estados, pode-se utilizar a linguagem WS-BPEL para gerenciar o estado.*

Heurística: *Utilizar a transferência de informações de estado (bases de dados, serviços utilitários, mensagens).*

O gerenciamento de informações de estado em excesso pode comprometer a disponibilidade e a escalabilidade de um serviço. Serviços são, desse modo, idealmente projetados para manter uma condição de independência de estado sempre que apropriado.

Ao automatizar uma tarefa em particular, um serviço deve processar dados específicos desta tarefa. Estes são referidos como dados de estado. Um serviço pode ser ativo, mas pode não estar comprometido com o processamento de dados de estado. Nesta condição

desocupada, o serviço é considerado independente de estado. Um serviço que processa ativamente ou retém dados de estado é classificado como dependente de estado.

Apesar de serviços que gerenciam estados poderem ser disponibilizados, a criação dos mesmos deve ser evitada a fim de se garantir escalabilidade, independência entre instâncias do serviço, balanceamento de carga. A linguagem WS-BPEL pode ser utilizada implementar a lógica que necessidade guardar estados de serviços.. Alternativamente, pode-se recorrer a um mecanismo de transferência de estado.

6.5.3 Padrões de Projeto para Projeto de Lógica de Serviço

Os seguintes padrões de projeto devem ser observados como parte do processo de projeto de lógica de serviço:

- Fachada de Serviço;
- Implementação Redundante;
- Replicação de Dados de Serviço;
- Cápsula (*wrapper*) de Legado;
- Metadados de Mensageria;
- Agente de Serviço;
- Barramento de Serviços Corporativo (ESB):
 - *Broker* de Serviço;
 - Roteamento Intermediário;
 - Enfileiramento Assíncrono;
 - Transformação de Modelos de Dados;
 - Transformação de Formatos de Dados;
 - Conversão de Protocolo;
 - Mensageria Confiável;
 - Centralização de Políticas;
 - Mensageria Direcionada por Eventos.

- Orquestração:
 - Abstração de Processo;
 - Centralização de Processo;
 - Repositório de Estado;
 - Compensação de Transação com Serviços;
 - Transação Atômica com Serviços;
 - Centralização de Regras.

6.5.4 Visibilidade do Processo de Desenvolvimento

A Figura 6-18 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-18 – Entradas e Saídas da Etapa de Projeto de Lógica de Serviço

6.6 DESENVOLVIMENTO DE SERVIÇO

Entradas de Processo	<ul style="list-style-type: none">• Contrato de Serviço• Perfil de Serviço• Documento de Arquitetura de Serviço• Documento de Arquitetura da Solução• Plano de Testes de Serviço• Plano de Testes da Solução• Manual de Uso de Serviço• Manual de Uso da Solução• Ordem de Serviço da Fábrica de Software• Arquitetura de Referência SOA• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Documento de Projeto de Serviço• Documento de Projeto de Solução• Código Fonte de Serviço• Código Fonte de Solução• Documento de Configuração de Serviço• Documento de Configuração de Solução• Perfil de Serviço (atualizado)• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Implementar Serviço
Perfis Executores	<ul style="list-style-type: none">• Desenvolvedor
Perfis Envolvidos	<ul style="list-style-type: none">• Arquiteto SOA• Especialista de Governança SOA• Especialista de Segurança SOA

6.6.1 Processos e Tarefas do Desenvolvimento de Serviço

Devido à ênfase em análise antecipada, modelagem, projeto e no uso de padronizações de projeto, o estágio de implementação de um projeto de SOA é muito mais controlado que em projetos tradicionais. Desse modo, a implementação de serviço é largamente restringida e formatada pelas especificações de serviço e de seu projeto de lógica.

6.6.1.1 Processo 1 – Implementar Serviço

O processo de implementação de serviço depende da tecnologia de implementação usada que, normalmente, vem acompanhada de suas próprias melhores práticas. A discussão do processo de implementação está fora do escopo deste documento devido a esta dependência.

6.6.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-19 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-19 – Entradas e Saídas da Etapa de Desenvolvimento de Serviço

6.7 TESTE DE SERVIÇO

Entradas de Processo	<ul style="list-style-type: none">• Contrato de Serviço• Documento de Arquitetura de Serviço• Plano de Testes de Serviço• Perfil de Serviço• Manual de Uso de Serviço• Documento de Configuração de Serviço• Ordem de Serviço de Fábrica de Testes• Arquitetura de Referência SOA• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Relatório de Testes de Serviço• Perfil de Serviço (atualizado)• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Testes do Serviço• Homologação do Serviço
Perfis Executoros	<ul style="list-style-type: none">• Analista de Testes• Especialista em Governança SOA• Testador
Perfis Envolvidos	<ul style="list-style-type: none">• Arquiteto SOA

6.7.1 Processos e Tarefas do Teste de Serviço

Processos de teste deverão ser definidos e formalizados pela metodologia de testes adotada pela empresa. Esses processos são usados sem maiores variações nesta metodologia SOA. Para uma relação de equivalência, os processos da etapa de Teste de Serviços são associados aos processos de Teste Unitário (ou de componente) em uma metodologia padrão de testes. Entretanto, algumas questões e práticas adicionais específicas de teste de serviços são apresentadas nesta seção.

Esta etapa é composta por dois processos: Testar Serviço e Homologar Serviço.

6.7.1.1 Processo 1 – Testar Serviço

Testar soluções orientadas a serviço requer mais tempo e esforço que em aplicações tradicionais. Os serviços recentemente entregues precisam ser testados individualmente e como parte de suas composições imediatas. Já, os serviços agnósticos, especialmente, requerem testes rigorosos para assegurar que eles estejam prontos para o reuso repetido quando eles são liberados para produção.

A seguir é apresentado um *checklist* de considerações de teste comuns, as melhores práticas gerais para testes de serviços e as melhores práticas para medir a conformidade de um serviço com os níveis de qualidade de serviço esperados durante as fases de teste.

Checklist:

- Que tipos de consumidores de serviço podem potencialmente acessar o serviço?
- Todas as assertivas de políticas de serviço podem ser atendidas com sucesso?
- A que tipo de condições de exceção o serviço pode estar sujeito?
- Quão bem o contrato de serviço comunica as semânticas de serviço?
- Um contrato de serviço revisado altera ou estende versões prévias?
- Quão facilmente o serviço pode ser composto?
- Quão facilmente o contrato de serviço pode ser descoberto?
- A conformidade com perfis WS-I é requerida?

Teste de Serviços:

- Todas as atividades de serviço e composições foram mapeadas?
- Todos os processos de compensação foram testados completamente?
- O que acontece se ocorrerem exceções dentro de processos de compensação?
- Serviços agregados conformam coletivamente com padronizações de projeto?
- Extensões de arquitetura (tais como transações, mensageria confiável, comunicação assíncrona, etc.) são requeridas para habilitar a composição?
- Em qual extensão a composição dependerá de agentes ou de outras extensões proprietárias de tempo de execução?

Melhores práticas gerais:

- Serviços devem ser testados em condições de acesso concorrente por múltiplos consumidores de serviços. Isso permite identificar e ajustar limites antes da implantação;

- Serviços agnósticos devem ser testados em composições de serviços diferentes. Isso é um requisito crítico de modo a assegurar que serviços agnósticos sejam realmente reusáveis e possuam alta capacidade de recomposição;
- Serviços com recursos compartilhados devem ser testados quando esses recursos estão sendo acessados por outros programas. Esse é um passo importante para medir a habilidade do serviço para manter um nível consistente de previsibilidade comportamental;
- Melhores práticas para medir a conformidade com os níveis de qualidade de serviço esperados:
 - A infraestrutura de suporte deve ser testada para determinar limites de uso e outras limitações;
 - Serviços devem ser testados individualmente e como parte de composições de serviços;
 - Serviços testados devem ser chamados em uma taxa que é maior ou igual à sua taxa esperada de chamada;
 - Estatísticas coletadas devem ser documentadas no perfil do serviço de modo que essa informação possa ser referenciada no futuro no caso dos níveis de qualidade de serviço serem reavaliados devido a mudanças na arquitetura ou na infraestrutura.

6.7.1.2 Processo 2 – Homologar Serviço

No processo de homologação, os atores responsáveis pela especificação do serviço deverão conduzir a aceitação. Além do Analista SOA e o Arquiteto SOA, também poderá ser necessária a presença do Analista de Processo, mas este não aparecerá no quadro resumo devido sua atuação ser esporádica.

6.7.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-20 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-20 – Entradas e Saídas da Etapa de Teste de Serviço

6.8 DISPONIBILIZAÇÃO E PUBLICAÇÃO DE SERVIÇO

Entradas de Processo	<ul style="list-style-type: none"> • Ordem de Serviço da Fábrica de Serviço • Contrato de Serviço • Documento de Arquitetura de Serviço • Perfil de Serviço • Documento de Projeto de Serviço • Código Fonte de Serviço • Manual de Uso de Serviço • Documento de Configuração de Serviço • Arquitetura Empresarial (Arquitetura Tecnológica e Arquitetura de Aplicações) • Artefatos de Governança de Serviços Corporativos • Configuração da Infraestrutura Tecnológica • Arquitetura de Referência SOA • Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none"> • Documento de Configuração de Serviço (final) • Perfil de Serviço (final) • Artefatos de Arquitetura Empresarial (Arquitetura Tecnológica e Arquitetura de Aplicações) atualizados • Artefatos de Governança de Serviços Corporativos atualizados • Artefatos de Gerenciamento de Configuração atualizados • Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none"> • Catalogar e Publicar (implantar) Serviço
Perfis Executoros	<ul style="list-style-type: none"> • Administrador de Infraestrutura SOA • Especialista de Governança SOA
Perfis Envolvidos	<ul style="list-style-type: none"> • Arquiteto SOA

6.8.1 Processos e Tarefas da Disponibilização e Publicação de Serviço

Após todo o ciclo de desenvolvimento do serviço, independente de ser parte de uma solução ou simplesmente um novo serviço que será disponibilizado na infraestrutura do órgão, as atividades de publicação e catalogação deverão ser executadas.

6.8.1.1 Processo 1 – Publicar de Serviço

Esta atividade se refere ao ato de disponibilizar o serviço na infraestrutura SOA.

A publicação de serviços no barramento compreende duas etapas. Em primeiro lugar, é necessário realizar o *deploy* das aplicações no servidor de aplicação e, em seguida, publicar

as interfaces de serviços (WSDL) no ESB. O primeiro passo é necessário, uma vez que o serviço será executado no mesmo Servidor de aplicação que está a sua interface. Para realizar esta etapa, o arquivo “.Jar” do serviço é gerado e instalado no servidor de aplicação.

Em seguida, no barramento, deve-se criar o projeto e o arquivo WSDL do serviço publicado como um recurso deste projeto.

Dependendo do barramento adotado, o mesmo pode permitir definir configurações específicas que não estão no WSDL, tais como: configurações de transporte de mensagens do serviço (método de requisição, autenticação etc.); algoritmo de balanceamento de carga que será usado para esse serviço (*none*, *round-robin*, *random* ou *random-weighted*); número de tentativas a ser executado para reexecutar o serviço caso seja necessário, além do intervalo de tempo para as tentativas e se é para tentar novamente se ocorrer erros da aplicação.

6.8.1.2 Processo 2 – Catalogar de Serviço

É o processo de registro de um serviço no inventário de serviços (p.e. padrão UDDI. Embora dependa do serviço de inventário definido e utilizado, o registro (*register*) é responsável pela gerência de metadados úteis durante o *runtime* de serviços para comunicação com um dado serviço. Descrições de serviços, localização e *endpoint*, operações e parâmetros de serviços e, políticas de serviços são exemplos de informação de *runtime* que normalmente o registro gerencia.

O registro pode armazenar descrições e metadados relacionados aos serviços, tais como, políticas, classificações e dependências. O registro não armazena os ativos correspondentes, tais como, WSDLs, esquemas e outros, mas ao invés disso ele aponta para estes ativos. O principal propósito do registro é tornar disponível estes metadados dos serviços para serem consumidos por aplicações para descoberta e uso – este é o motivo para o qual serviços são “publicados” no registro, para serem descobertos e utilizados por toda a empresa.

Os principais registros do mercado tratam funcionalidades para aplicações da empresa, tais como:

- **Segurança:** permite definir controle de acesso granular para cada componente registrado. O responsável pela publicação do componente pode definir permissões de acesso para operações de localização (*find*), recuperação (*get*), modificação e remoção para cada componente publicado;
- **Mecanismos para controle de qualidade e precisão de dados:** para garantir que componentes registrados estão precisos e atualizados em seus respectivos ambientes. Define responsabilidades para cada componente registrado. Ele oferece mecanismos para promoção e aprovação de componentes entre ambientes de desenvolvimento, controle de qualidade e produção;
- **Subscrição e notificação:** para automaticamente notificar usuários a respeito das mudanças nos componentes que eles dependem;
- **Replicação seletiva entre múltiplos registros:** permite propagação automatizada entre diferentes registros (por exemplo, entre registros internos e externos);
- **Gestão avançada de taxonomia** para garantir taxonomias bem definidas;
- **Gestão de desempenho:** para controle, *logging* e auditoria dos processos de publicação e descoberta;
- **Desempenho e escalabilidade:** pela implementação de algoritmos para pilhas de serviços web e bancos de dados e, suportando balanceamento de carga e clusterização.

A definição dos ativos que serão gerenciados pelo registro, seus parâmetros de controle, as políticas, segurança, enfim, todos os aspectos incorporados deverão ser definidos no Modelo de Governança SOA da empresa.

6.8.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-21 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-21 – Entradas e Saídas da Etapa de Disponibilização e Publicação de Serviço

6.9 IMPLEMENTAÇÃO DE APLICAÇÃO COMPOSTA

Entradas de Processo	<ul style="list-style-type: none"> • Ordem de Serviço da Fábrica de Serviço • Documento de Modelagem de Processos de Negócio • Modelo de Informação (Estruturada) • Contrato de Serviço(s) • Perfil de Serviço(s) • Documento de Arquitetura de Serviço(s) • Documento de Arquitetura da Solução • Documento de Projeto de Solução • Código Fonte da Solução • Plano de Testes da Solução • Manual de Uso de Serviço(s) • Manual de Uso da Solução • Documento de Configuração de Serviço(s) • Documento de Configuração de Solução • Arquitetura de Referência SOA • Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none"> • Documento de Arquitetura da Solução (final) • Documento de Projeto de Solução (final) • Código Fonte da Solução (final) • Plano de Testes da Solução (final) • Manual de Uso da Solução (final) • Documento de Configuração da Solução (atualizado) • Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none"> • Integrar elementos da solução
Perfis Executores	<ul style="list-style-type: none"> • Arquiteto SOA • Desenvolvedor
Perfis Envolvidos	<ul style="list-style-type: none"> • Especialista de Governança SOA

6.9.1 Processos e Tarefas da Implementação de Aplicação Composta

Essa etapa é a responsável pela composição final da solução SOA. O Arquiteto SOA é seu principal executor e deverá, dependendo de cada caso, alocar outros perfis profissionais para sua implementação.

Essa “integração” reúne todos os componentes necessários à solução, processo, serviços e interfaces, e compô-los de maneira que operem conjuntamente de acordo com as especificações levantadas na fase de Modelagem do Negócio. É nessa fase que se deve observar todos os quesitos de acessibilidade e segurança definidos.

Toda a solução deve ser operacionalizada, não ficando nada por fazer a não ser sua homologação junto ao requisitante, para assim, entrar em produção.

6.9.1.1 Processo 1 – Implementar Aplicação Composta

Como no caso da implementação do serviço, o processo de implementação de integrações e composição final de soluções orientadas a serviços também depende da tecnologia usada e, dessa forma, sua discussão está fora do escopo deste documento.

6.9.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-22 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-22 – Entradas e Saídas da Etapa de Implementação de Aplicação Composta

6.10 TESTE DA APLICAÇÃO COMPOSTA

Entradas de Processo	<ul style="list-style-type: none">• Ordem de Serviço da Fábrica de Testes• Documento de Modelagem de Processos de Negócio• Modelo de Informação (Estruturada)• Documento de Arquitetura da Solução• Plano de Testes da Solução• Manual de Uso de Solução• Documento de Configuração da Solução• Arquitetura de Referência SOA• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Relatório de Testes da Solução• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Testes da solução
Perfis Executores	<ul style="list-style-type: none">• Analista de Negócio• Analista de Processos• Analista de Testes• Especialista de Governança SOA• Testador
Perfis Envolvidos	<ul style="list-style-type: none">• Arquiteto SOA

6.10.1 Processos e Tarefas do Teste da Aplicação Composta

Agora, com todos os serviços da solução e todas as interfaces de usuário desenvolvidas, a solução deve passar por nova bateria de testes, agora não mais testando individualmente os elementos que a compõe, mais sim, a solução completa e finalizada.

Os processos de teste e homologação de solução são executados conforme metodologia de testes definidas na empresa. Esses processos são usados sem maiores variações nesta metodologia SOA. Para uma relação de equivalência, os processos da etapa de Teste de Solução são associados aos processos de Teste de Solução (ou de aplicação) em uma metodologia de testes padrão.

Esta etapa é composta por dois processos: Testar Solução e Homologar Solução.

6.10.1.1 Processo 1 – Testar Solução

Esse processo é executado pela Fábrica de Testes e visa assegurar que a solução implementada está em acordo com os requisitos (funcionais e não funcionais) definidos.

6.10.1.2 Processo 2 – Homologar Solução

No processo de homologação, os atores responsáveis pela especificação da solução deverão conduzir a aceitação. Nesta etapa é fundamental a participação da área requisitante, que deverá avaliar a solução implementada à luz da demanda realizada, mas trazendo a visão do valor que a solução vai agregar ao seu negócio.

6.10.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-23 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Figura 6-23 – Entradas e Saídas da Etapa de Teste da Aplicação Composta

6.11 DISPONIBILIZAÇÃO DE APLICAÇÃO COMPOSTA

Entradas de Processo	<ul style="list-style-type: none">• Ordem de Serviço da Fábrica de Serviço• Documento de Arquitetura da Solução• Documento de Projeto de Solução• Código Fonte da Solução• Manual de Uso da Solução• Documento de Configuração da Solução• Arquitetura Empresarial (Arquitetura Tecnológica e Arquitetura de Aplicações)• Artefatos de Governança de Serviços Corporativos• Configuração da Infraestrutura Tecnológica• Arquitetura de Referência SOA• Artefatos de Gerenciamento de Projetos
Saídas de Processo	<ul style="list-style-type: none">• Documento de Configuração da Solução (final)• Solução Implantada• Artefatos de Arquitetura Empresarial (Arquitetura Tecnológica e Arquitetura de Aplicações) atualizados• Artefatos de Governança de Serviços Corporativos atualizados• Artefatos de Gerenciamento de Configuração atualizados• Artefatos de Gerenciamento de Projetos
Atividades	<ul style="list-style-type: none">• Implantar da solução
Perfis Executores	<ul style="list-style-type: none">• Administrador de Infraestrutura SOA• Especialista de Governança SOA
Perfis Envolvidos	<ul style="list-style-type: none">• Arquiteto SOA

6.11.1 Processos e Tarefas da Disponibilização de Aplicação Composta

6.11.1.1 Processo 1 – Implantar Solução

O processo de implantação de solução depende da tecnologia de *middleware* usada e, da mesma forma que os demais processos que possuem essa dependência, sua discussão está fora do escopo deste documento.

6.11.2 Visibilidade do Processo de Desenvolvimento

A Figura 6-24 apresenta a visão dos artefatos de entrada e saída para esta etapa do desenvolvimento SOA.



Entrada

- Arquitetura Empresarial (Arquitetura Tecnológica e Arquitetura de Aplicações)
- Artefatos de Governança de Serviços Corporativos
- Configuração da Infraestrutura Tecnológica
- Arquitetura de Referência SOA
- Artefatos de Gerenciamento de Projetos

Artefato	
Documento de Modelagem de Processo de Negócio	✓
Modelo de Informação (Estruturada)	✓
Documento de Definição de Candidatos a Serviços	✓
Perfil de Serviço	✓
Contrato de Serviço	✓
Documento de Arquitetura de Serviço	✓
Documento de Arquitetura da Solução	✓
Plano de Testes de Serviço	✓
Plano de Testes da Solução	✓
Ordem de Serviço de Fabrica de Serviço	✓
Ordem de Serviço de Fabrica de Software	✓
Ordem de Serviço de Fabrica de Testes	✓
Documento de Projeto de Serviço	✓
Documento de Projeto de Solução	✓
Código Fonte de Serviço	✓
Código Fonte da Solução	✓
Relatório de Testes de Serviço	✓
Relatório de Testes da Solução	✓
Manual de Uso de Serviço	✓
Manual de Uso da Solução	✓
Documento de Configuração de Serviço	✓
Documento de Configuração da Solução	🔄

Saída

- Artefatos de Arquitetura Empresarial (Arquitetura Tecnológica e Arquitetura de Aplicações) **atualizados**
- Artefatos de Governança de Serviços Corporativos **atualizados**
- Artefatos de Gerenciamento de Configuração **atualizados**
- Artefatos de Gerenciamento de Projetos

Artefato	
Documento de Modelagem de Processo de Negócio	✓
Modelo de Informação (Estruturada)	✓
Documento de Definição de Candidatos a Serviços	✓
Perfil de Serviço	✓
Contrato de Serviço	✓
Documento de Arquitetura de Serviço	✓
Documento de Arquitetura da Solução	✓
Plano de Testes de Serviço	✓
Plano de Testes da Solução	✓
Ordem de Serviço de Fabrica de Serviço	✓
Ordem de Serviço de Fabrica de Software	✓
Ordem de Serviço de Fabrica de Testes	✓
Documento de Projeto de Serviço	✓
Documento de Projeto de Solução	✓
Código Fonte de Serviço	✓
Código Fonte da Solução	✓
Relatório de Testes de Serviço	✓
Relatório de Testes da Solução	✓
Manual de Uso de Serviço	✓
Manual de Uso da Solução	✓
Documento de Configuração de Serviço	✓
Documento de Configuração da Solução	✓

Figura 6-24 – Entradas e Saídas da Etapa de Disponibilização de Aplicação Composta

7 - IMPLEMENTAÇÃO E APLICAÇÃO DA METODOLOGIA

Conforme já mencionado na seção 1.1, a necessidade de uma metodologia de desenvolvimento SOA foi identificada na experiência de adoção de SOA na Agência Nacional de Vigilância Sanitária – ANVISA (ANVISA, 2010(1)), porém sua concepção inicial foi realizada durante trabalho de consultoria realizado na Agência Nacional de Águas – ANA (contrato realizado a partir de adesão a registro de preços do Ministério da Integração (MI, 2010)). A sistematização completa e aplicação da metodologia apresentada neste trabalho se devem primordialmente ao trabalho realizado no Ministério da Saúde, aonde esta vem sendo utilizada e aprimorada constantemente desde meados de 2012. Estas três experiências são relatadas neste capítulo de modo a demonstrar os resultados de sua aplicação em projetos reais e iniciativas de adoção de SOA.

7.1 AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA – ANVISA

O escopo da experiência de adoção de SOA na ANVISA está associado a projetos de automatização de processos utilizando a plataforma Oracle para BPM e SOA, iniciado em 2010 (ANVISA, 2010(1)). Nesse contexto, foram elaborados uma Arquitetura de Referência SOA preliminar (ANVISA, 2010(2)) e modelos de processos que foram usados para encomendar a produção de software com SOA junto a uma empresa que atuava como Fábrica de Software terceirizada. Tais produtos serviram de especificações funcionais e não funcionais para o desenvolvimento de soluções no modelo SOA, realizado pela Fábrica.

No decorrer dos trabalhos, constatou-se que a falta de maturidade e experiência da Fábrica de Software em técnicas de SOA ocasionou sérios problemas na qualidade dos serviços e soluções que estavam em desenvolvimento na ANVISA, culminando na desistência da empresa terceirizada na continuidade do projeto. Percebe-se que a existência de uma metodologia de desenvolvimento SOA seria uma forma pragmática de reduzir tais dificuldades. Tal metodologia deveria estabelecer processos, papéis e responsabilidades bem definidos, além de descrever como princípios e melhores práticas de SOA podem ser aplicadas durante o desenvolvimento SOA.

Com a saída da Fábrica de Software, a consultoria que acompanhava os trabalhos assumiu os papéis necessários para a condução do projeto e iniciou a criação de artefatos

específicos para agilizar o processo de desenvolvimento e melhorar a qualidade dos resultados. Começavam então as pesquisas sobre metodologias SOA, procurando qual seria a mais adequada para as necessidades observadas. Este estudo preliminar revelou que as metodologias de desenvolvimento SOA disponíveis à época pareciam ocultar informações de toda ordem – fases, disciplinas, papéis e artefatos – além de não demonstrarem com resultados práticos, sua aplicação. Outra característica marcante é que nenhuma metodologia analisada considerava explicitamente aspectos de terceirização dos serviços de desenvolvimento.

A metodologia apresentada neste trabalho é posterior a esta experiência e, portanto, não foi aplicada na ANVISA. Entretanto, os estudos realizados durante os trabalhos foram determinantes para as iniciativas que se sucederam.

7.2 AGÊNCIA NACIONAL DE ÁGUAS – ANA

Os trabalhos na ANA iniciaram-se no início de 2012, já incorporando uma perspectiva de sistematização de melhores práticas de desenvolvimento SOA. Inicialmente, foi realizada avaliação de maturidade SOA da corporação, a partir do modelo *The Open Group Service Integration Maturity Model* – OSIMM (OSIMM, 2009). Esta avaliação ajudou a definir lacunas importantes na prática corporativa que serviram para direcionar os rumos do Plano de Adoção SOA (ANA, 2012(2)) com maior propriedade e adequado às necessidades da Agência.

Foram elaborados em seguida: a Arquitetura de Referência SOA (ANA, 2012(1)), completa e detalhada e o Plano de Governança (ANA, 2012(3)). Iniciou-se, então, a elaboração da metodologia de desenvolvimento, que foi precursora daquela apresentada neste trabalho. Esta metodologia define de forma preliminar o ciclo de desenvolvimento, apoiado em uma estrutura funcional com papéis e responsabilidades relacionados a perfis profissionais, e artefatos específicos para o processo metodológico. Como exemplo, cita-se a previsão dos perfis Auditor SOA, que não está presente em outras metodologias conhecidas, e Especialista em Governança SOA. Esses profissionais atuam conjuntamente para garantir que todos os processos sejam respeitados e os artefatos corretamente elaborados, minimizando a ocorrência de inconformidades.

Apesar da existência de todos esses instrumentos de apoio ao desenvolvimento com SOA, bem como da escolha de projeto inicial para aplicação do paradigma, o processo não pode iniciar. Como grande parte dos órgãos públicos brasileiros, a Agência se utiliza de mão de obra terceirizada por regime de Fábrica de Software. Ao ser convocada para apresentação dos novos paradigmas e padrões de desenvolvimento desejados pela organização, a empresa terceirizada se recusou a executar os serviços nos moldes pretendidos, alegando questões contratuais como a não previsão de desenvolvimento em SOA. Evidentemente, essa forma de desenvolvimento não estava devidamente instruída no contrato de terceirização, que era anterior à iniciativa de adoção de SOA.

Observa-se, então, que contratos vigentes podem ser utilizados, porém, na prática isso dependerá de suas condições contratuais e, por vezes, da boa vontade da empresa, visto que normalmente estes contratos de Fábrica de Software são regidos por pontos de função, o que, ao menos por enquanto, existem apenas ensaios de sua utilização em SOA. Com isso, a situação ideal é, já no início do processo, identificar esta situação e, se for o caso, elaborar um termo de referência específico para contratação das atividades técnicas que deverão ser terceirizadas no Plano de Adoção SOA.

7.3 MINISTÉRIO DA SAÚDE – MS

O projeto de estruturação do Programa de Adoção de SOA do Ministério da Saúde iniciou-se em Março de 2011, através de um contrato com a Universidade de Brasília – UnB.

Instrumentos elaborados no contexto do convênio: Plano do Programa de Adoção 2012-2013 (MS, 2012(3)); Arquitetura de Referência SOA (MS, 2012(1)) e Metodologia de Desenvolvimento SOA (MS, 2012(2)).

A metodologia de desenvolvimento SOA foi aplicada em dois projetos do Programa: CADSUS e o CNES.

7.3.1 CADSUS

O Cadastro Nacional de Usuários do SUS (DATASUS, 2013(2)) é um repositório que contém todas as informações necessárias sobre os indivíduos que usuários do sistema SUS. O cadastramento consiste no processo por meio do qual são identificados os usuários do Sistema Único de Saúde e seus domicílios de residência.

Por meio do cadastro é possível a emissão do Cartão Nacional de Saúde para os usuários e a vinculação de cada usuário ao domicílio de residência, permitindo uma maior eficiência na realização das ações de natureza individual e coletiva desenvolvidas nas áreas de abrangência dos serviços de saúde.

O Cadastramento permite ainda a construção de um banco de dados para diagnóstico, avaliação, planejamento e programação das ações de saúde. A realização de um cadastramento domiciliar de base nacional, aliado à possibilidade de manutenção dessa base cadastral atualizada, pode permitir aos gestores do SUS a construção de políticas sociais integradas e intersetoriais (educação, trabalho, assistência social, tributos etc.) nos diversos níveis do governo.

Este projeto resultou no desenvolvimento e implantação dos serviços e aplicações do Cadastro, em produção desde maio de 2012. Tais desenvolvimentos ocorreram em conformidade com a metodologia exposta neste trabalho. Os artefatos foram produzidos, processos instanciados, perfis profissionais definidos alocados nas etapas e atividades discriminadas na metodologia, resultando em otimização no processo de desenvolvimento e qualidade nos produtos gerados, obedecendo aos princípios e melhores prática do paradigma de orientação a serviços.

7.3.2 CNES

O Cadastro Nacional de Estabelecimentos de Saúde (DATASUS, 2013(1)) é base para operacionalizar os Sistemas de Informações em Saúde, sendo estes imprescindíveis a um gerenciamento eficaz e eficiente. Propicia ao gestor o conhecimento da realidade da rede assistencial existente e suas potencialidades, visando auxiliar no planejamento em saúde em todos os níveis de governo, bem como dar maior visibilidade ao controle social a ser exercido pela população.

O CNES visa disponibilizar informações das atuais condições de infraestrutura de funcionamento dos Estabelecimentos de Saúde em todas as esferas, ou seja, Federal, Estadual e Municipal.

Este projeto, ainda em execução, também utiliza a metodologia em sua plenitude. Os resultados alcançados até o momento demonstram que o mesmo sucesso alcançado no CADSUS, será verificado.

7.3.3 Contratação de Fábrica de Serviço Terceirizada e Consultoria para Centro de Excelência SOA

Com relação à estrutura funcional proposta nesta metodologia, as unidades funcionais corporativas ainda estão em processo de estruturação. As unidades funcionais dedicadas serão alvo de novos processos de terceirização.

O modelo de terceirização a ser adotado resulta das experiências do CADSUS e CNES, onde a presente metodologia pôde ser ajustada e integralmente utilizada na definição do novo modelo de contratação (em andamento).

No caso específico do Ministério da Saúde, observa-se a dificuldade de realização de criação de capacidade interna, nas quantidades e perfis profissionais requeridos, devido aos altos níveis de comprometimento do quadro atual com suas atividades cotidianas e sua atualização tecnológica. Com isso, os perfis profissionais capacitados são importantes partes da especificação dos contratos, pois estes terão que ser providos pelas equipes terceirizadas.

Toda a concepção da metodologia foi utilizada para a geração de produtos específicos para a finalidade de resolver esta questão. São eles:

- TR para contratação dos serviços especializados (MS, 2012(4));
- Plano de Projeto para contratação do Centro de Excelência SOA (convênio).

A metodologia, portanto, vem sendo aplicada como instrumento estruturante do programa de adoção de SOA, servindo de base inclusive para a instrução dos contratos de terceirização.

7.4 OUTROS TRABALHOS JÁ INICIADOS

No início de 2013, o Exército Brasileiro iniciou projeto de desenvolvimento com Arquitetura Orientada a Serviço, em parceria com a Universidade de Brasília. O projeto trata de desenvolvimento de aplicações de Comando e Controle (C2). A metodologia de desenvolvimento está sendo aplicada igualmente neste contexto, onde serviços que serão entregues brevemente já foram especificados em conformidade aos artefatos definidos, a partir de um processo de desenvolvimento que segue os modelos de processo apresentados.

Atualmente, estou contratado como Consultor pela RNP – Rede Nacional de Ensino e Pesquisa, para atuar no MCTI – Ministério da Ciência, Tecnologia e Inovação em um projeto de codinome Aquarius. Basicamente este projeto disponibiliza painéis de informações gerenciais dos programas voltados à ciência, tecnologia e inovação no Brasil. Estes dados são coletados e trabalhados de diversas bases heterogêneas, internas e externas, e também alimentadas por processos de trabalho, cuja automatização também está inserida em seu escopo.

O trabalho que estou coordenando visa disponibilizar uma infraestrutura SOA para o projeto, além de adequá-lo às melhores práticas e princípios da orientação a serviço. Atualmente estamos elaborando o Plano de Adoção SOA, para depois criarmos uma Arquitetura de Referência SOA, uma Metodologia SOA, um Plano de Governança SOA, montarmos um Centro de Excelência SOA e um Escritório de Processos. O cronograma atual prevê um prazo de sete meses, incluindo a composição e capacitação da equipe, definição das tecnologias e softwares e implantação da Arquitetura e Metodologia SOA em uma prova de conceito. Após este trabalho, o Ministério estará apto a iniciar o estudo e adaptação do atual Projeto Aquarius, para uma nova versão mais flexível e em consonância com o paradigma da orientação a serviço. Isto é claro, utilizando esta metodologia em todo o processo de desenvolvimento.

8 - CONCLUSÕES

A metodologia proposta foi construída a partir de pesquisas de campo e experiências relacionadas a iniciativas de adoção de SOA no Brasil.

O principal desafio relacionado com essas iniciativas estava relacionado com a falta de profissionais qualificados disponíveis. Como resultado, recursos humanos existentes estão se tornando bastante dispendiosos e difíceis de encontrar. Isso é especialmente desafiador em organizações com práticas inflexíveis de contratação e aquisição, o que, nos casos estudados, resultam do conjunto legal e regulamentar para aquisições de TI no setor público brasileiro.

Pela experiência adquirida, percebe-se que atividades de desenvolvimento de software com SOA, seja em uma empresa ou órgão público, independente de seu grau de maturidade, frequentemente utilizam forças de trabalho terceirizadas, seja pela cultura ou pela ausência de profissionais nos quadros internos.

Na metodologia proposta, são definidos escopos claros de contratação de serviços, representados pelas unidades funcionais estruturadas pelo modelo de “Fábrica”, as Fábricas de Software, Serviços e Testes.

Um importante aspecto aprendido diz respeito à separação de papéis nas unidades Fábrica de Serviços e de Fábrica de Software. Existe um significativo ônus de governança associado a esta configuração. Portanto, em organizações de baixa maturidade, pode ser melhor mesclar essas unidades funcionais e terceirizá-las com o mesmo fornecedor de serviços. Note, entretanto, que se houver clara governança, as unidades Fábrica de Software e Fábrica de Teste de Software podem ser terceirizadas através de contratos existentes, que são, possivelmente, com base em modelos tradicionais de terceirização de Fábrica de Software.

Adicionalmente, se discute o uso de consultorias experientes que poderão compor o CoE SOA, garantindo conhecimento especializado na iniciativa SOA. Dentre as atribuições dessa equipe inclui-se a personalização da metodologia de desenvolvimento orientada a serviço, a definição da arquitetura de referência (MACKENZIE et al., 2008; ESTEFAN et al. 2008; OPEN GROUP, 2011) e o estabelecimento de práticas de governança (ERL,

2011). Metodologias e arquiteturas de referência SOA devem ser claramente posicionadas como padrões de projeto corporativo (NORTON, 2010). Definições formais para estes elementos estruturantes e práticas de conformidade afins em uma iniciativa SOA, são fatores-chave de sucesso para a adoção de SOA, especialmente no caso da força de trabalho terceirizada (NORTON, 2010).

Na metodologia proposta neste trabalho, estes consultores especialistas estão diretamente relacionados com Centro de Excelência SOA, que tem um modelo de terceirização distinto das unidades de "fábrica". Nossa experiência mostra que quando o *framework* SOA é definido e a maturidade da organização é melhorada (pelo menos para projetos gerenciáveis SOA), a unidade CoE SOA pode ser terceirizada como um conjunto de serviços previamente definidos.

A terceirização da Fábrica de Serviços e do Centro de Excelência SOA requer uma clara definição de processos (unidades de trabalho), perfis profissionais e entregáveis. Estes devem ser claramente especificados em documentos de aquisição (por exemplo, RFPs) e contratos. Esta metodologia tem por objetivo proporcionar uma definição geral para estes, em relação ao papel da Fábrica de Serviços.

Nesta dissertação apresentamos uma metodologia de desenvolvimento de software orientado a serviço utilizada como modelo geral para a terceirização da força técnica de trabalho SOA. A metodologia surge como uma síntese das melhores práticas aprendidas na pesquisa e experiência de campo, estabelecendo e experimentando metodologias de desenvolvimento em grandes organizações do setor público no Brasil. As principais conclusões são:

- Um novo modelo de terceirização é usado para a maior parte do trabalho técnico em função do desenvolvimento em SOA. Isto é principalmente representado por um modelo de "Fábrica de Serviço", descrito na dissertação;
- Os contratos de terceirização existentes para Fábricas de Software e Fábricas de Testes podem ser usados, desde que previstos em contrato ou que o fornecedor concorde com sua adoção. No entanto, há uma sobrecarga de governança que requer maturidade em práticas SOA. Se a maturidade não está presente, a Fábrica de Serviço deve abranger também o papel de Fábrica de Software;

- Processos claramente definidos (unidades de trabalho), perfis profissionais e produtos, que devem estar presentes nos contratos e documentos de aquisição, são essenciais para estabelecer a governança necessária de serviços terceirizados;
- Posicionar as metodologias de desenvolvimento e governança, bem como a arquitetura de referência, como padrões de projeto corporativos e estabelecer práticas para assegurar a conformidade também são fatores-chave de sucesso para projetos em SOA com força de trabalho terceirizada.

Dentre as dificuldades enfrentadas na realização do trabalho, merece citação a exiguidade, na literatura disponível, de métodos de mensuração e respectivos resultados obtidos pelas metodologias analisadas, seja pela sua não divulgação proposital, principalmente por aquelas criadas e mantidas por fabricantes de software, com intuito de não revelar informações que seus concorrentes possam usar para aprimorar seus processos, seja pela simples ausência de casos reais de sua utilização na prática. Assim, uma comparação quantitativa da metodologia proposta com outras metodologias estudadas não foi possível, ficando este estudo indicado como trabalhos futuros. Nesse sentido, aponta-se a necessidade de criação de métricas (efetividade, desempenho, qualidade) para avaliação da metodologia.

Em trabalhos futuros, pretende-se ainda detalhar com maior precisão os processos de Governança SOA, estruturados em um Plano de Governança abrangendo não apenas o ciclo de vida de serviços e processos, mas também todos os artefatos e ativos pertinentes à metodologia, com uso de indicadores de não conformidade para cada etapa do processo e, também, sob a ótica da qualidade dos serviços e soluções SOA produzidas na observância dos princípios da orientação a serviço.

Com isso podemos analisar os objetivos propostos no trabalho:

- OE1: Definir a estrutura funcional necessária à metodologia, de forma a agrupar tarefas e atividades tecnicamente relacionadas;
 - Objetivo atingido através da estruturação das Unidades Funcionais, corporativas e dedicadas, que tem participação efetiva no processo de desenvolvimento em SOA, declarando suas responsabilidades e área de atuação dentro do ciclo de desenvolvimento.

- OE2: Definir os perfis profissionais para a metodologia, tornando clara a divisão de responsabilidades de tarefas e capacidades técnicas necessárias para sua execução;
 - Objetivo atingido através da definição dos Perfis Profissionais, as habilidades e capacidades necessárias, identificando as tarefas de sua responsabilidade e até uma sugestão de treinamentos necessários à sua capacitação.

- OE3: Definir processos de modo a especificar as atividades requeridas no desenvolvimento de software orientado a serviço;
 - Objetivo atingido através da definição do Ciclo de Vida de Desenvolvimento, apresentando suas fases, processos e tarefas, identificando para cada etapa os artefatos de entrada e saída, as heurísticas aplicadas e os responsáveis por sua execução.

- OE4: Validar e testar a metodologia em organizações que estejam iniciando a adoção em SOA.
 - Objetivo atingido através dos estudos de casos do Ministério da Saúde e, mais recentemente, do Exército Brasileiro.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ALLEN P. SOA best practice report: the service oriented process. Technical report, CBDi Journal. 2007.
- [2] ANA – Agência Nacional de Águas. Arquitetura de Referência Versão 1.0. 2012.
- [3] ANA – Agência Nacional de Águas. Plano de Adoção SOA Versão 1.0. 2012.
- [4] ANA – Agência Nacional de Águas. Plano de Governança SOA Versão 1.0. 2012.
- [5] ANVISA – Agência Nacional de Vigilância Sanitária. Processo nº. 25351.667204/2009-97. Pregão Eletrônico/SRP nº 08/2010. 2010.
- [6] ANVISA – Agência Nacional de Vigilância Sanitária. Arquitetura de Referência SOA – ANVISA. 2010.
- [7] ARSANJANI, A. Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA. 2004. Disponível em: <http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/IBM/I041109A.pdf>, acessado em 7/01/2013.
- [8] ARSANJANI, A.; GHOSH S.; ALLAM A.; ABDOLLAH T.; GARIAPATHY S.; HOLLEY K. SOMA: a method for developing service-oriented solutions. 2008. IBM Syst J 47(3):377–396. Disponível em: http://www.cs.jyu.fi/el/tjtse54_09/Artikkelit/ArsanjaniEtAlIBMSsJ.pdf, acessado em 15/01/2013.
- [9] BARBIER, R. La recherche action. Ed. Anthropos/Economica - Paris, 1996.
- [10] BRABEC, T.; BUCHALCEVOVÁ, A. Suitability of XP for service-based application development. Systems Integration. 2007. Disponível em: <http://nb.vse.cz/~buchalc/clanky/xpsi2007.pdf>, acessado em 12/12/2012.
- [11] BREIVOLD, H.; LARSSON, M. Component-based and service-oriented software engineering: Key concepts and principles, in: Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications,

- pp. 13–20. 2007. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4301060&tag=1, acessado em 12/12/2012.
- [12] BIEBERSTEIN, N. et al. Service Oriented Architecture (SOA) Compass. NJ, EUA: IBM, 2006.
- [13] BOUKADI, K.; VINCENT L.; GHEDIRA C.; MAAMAR Z. CSMA: Context-Based, Service-Oriented Modeling and Analysis Method for Modern Enterprise Applications. 2010. Disponível em: <http://www.igi-global.com/article/csma-context-based-service-oriented/43008>, acessado em 22/01/2013.
- [14] BROWN A.; LOPEZ A.; REYES L. Practical Experiences with Software Factory Approaches in Enterprise Software Delivery, ICSEA 2011, The Sixth International Conference on Software Engineering Advances, Barcelona, Spain, October 23, 2011.
- [15] CHANG SH. A systematic analysis and design approach to develop adaptable services in service oriented computing. IEEE congress on services, pp 375–378. 2007. Disponível em: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4278821&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4278821, acessado em 16/01/2013.
- [16] CHANG, SH.; KIM S. A Systematic Approach to Service-Oriented Analysis and Design. 2007. Disponível em: http://link.springer.com/chapter/10.1007/978-3-540-73460-4_32, acessado em 19/01/2013.
- [17] CHERBAKOV, L.; GALAMBOS, G.; HARISHANKAR, R.; KALYANA, S.; RACKHAM, G. Impact of service orientation at the business level. IBM Systems Journal, 44 (4), 653–668. 2005. Disponível em: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5386696&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5386696, acessado em 20/01/2013.

- [18] CLEMENTS P.; NORTHROP L. Software Product Lines: Patterns and Practices, 3rd Edition, Addison Wesley, 2001.
- [19] DATASUS. Cadastro Nacional de Estabelecimentos de Saúde. 2013. Disponível em: <http://cnes.datasus.gov.br/>, acessado em 10/07/2013.
- [20] DATASUS. Cadastro de Usuários do SUS. 2013. Disponível em: <http://cartaonet.datasus.gov.br/>, acessado em 10/07/2013.
- [21] ERRADI A.; ANAND, S.; KULKARNI, N. SOAF: An architectural framework for service definition and realization, in Proceedings of the IEEE International Conference on Services Computing, pp 151-158, Chicago, USA, September 2006. Disponível em: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4026916&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4026916, acessado em 22/01/2013.
- [22] ERL, T. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, 2008.
- [23] ERL, T. SOA Principles of Service Design. Prentice Hall, 2009.
- [24] ESTEFAN, J. A.; LASKEY K.; MCCABE F.; THORNTON D. (editors) - "OASIS Reference Architecture for Service Oriented Architecture Version 1.0", OASIS Draft 1, 2008.
- [25] GU, Q.; LAGO P. Guiding the selection of service-oriented software engineering methodologies. 2011. Disponível em: <http://link.springer.com/article/10.1007%2Fs11761-011-0080-0#page-1>, acessado em 18/01/2013.
- [26] HOLLEY K.; PALISTRANT J.; GRAHAM S. Effective SOA Governance. 2006. Disponível em: http://www-07.ibm.com/my/events/soa/download/soagov_mgmt.pdf, acessado em 15/4/2013.

- [27] HOTLE M.; LANDRY S. Application Delivery and Support Organizational Archetypes: The Software Factory, Gartner Research Report G00167531, May 2009.
- [28] JAYARATNA N. Understanding and evaluating Methodologies: NIMSAD, a systematic framework. McGraw-Hill Inc, New York. 1994.
- [29] JESTON J.; NELIS J. Business Process Management, Second Edition: Practical Guidelines to Successful Implementations, Butterworth-Heinemann, 2008.
- [30] JOACHIM, N.; BEIMBORN, D.; WEITZEL, T. What Are Important Governance and Management Mechanisms to Achieve IT Flexibility in Service-Oriented Architectures (SOA)?: An Empirical Exploration. System Sciences (HICSS), 2011 44th Hawaii International Conference on. Disponível em: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5718959&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5718959, acessado em 14/4/2013.
- [31] JONES S.; MORRIS M. A methodology for service architectures. 2006. Disponível em: <http://www.oasis-open.org/committees/download.php/15071>, acessado em 16/01/2013.
- [32] KIM, Y.; YUN X. An Approach to Modeling Service-Oriented Development Process. 2006. Disponível em: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4026934&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4026934, acessado em 22/01/2013.
- [33] KITCHENHAM, BA. Evaluating software engineering methods and tool part 1: The evaluation context and evaluation methods. SIGSOFT Softw Eng Notes 21(1):11–14. 1996. Disponível em: <http://dl.acm.org/citation.cfm?id=381795>, acessado em 18/01/2013.
- [34] KONTOGOGOS A.; AVGERIOU P. An overview of software engineering approaches to service oriented architectures in various fields. In: Proceedings of the

18th IEEE international workshops on enabling technologies: infrastructures for collaborative enterprises, IEEE Computer Society, pp 254–259. 2009.

- [35] LARRUCEA X. Method Engineering Approach for Interoperable Systems Development. 2008. Disponível em: <http://onlinelibrary.wiley.com/doi/10.1002/spip.371/abstract>, acessado em 20/01/2013.
- [36] MACKENZIE, C.; K. LASKEY; F. MCCABE; P. BROWN; R. METZ (editors) - “OASIS Reference Model for Service Oriented Architecture Version 1.0”, OASIS Standard, 12 October 2006.
- [37] MI – Ministério da Integração. Edital de Pregão Eletrônico No 32/2010. Processo Nº 59000.000886/2010-14. 2010.
- [38] MILANOVIC, N.; MALEK, M. Current solutions for web service composition, IEEE Internet Computing, vol. 8, pp. 51–59. 2004. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1355922, acessado em 13/12/2013.
- [39] MITTAL K. Build your SOA, part 3: the service-oriented unified process. Technical report, IBM developerWorks. 2006. Disponível em: <http://www.ibm.com/developerworks/library/ws-soa-method3/index.html>, acessado em 15/01/2013.
- [40] MS – Ministério da Saúde. Arquitetura de Referência SOA do Ministério da Saúde 2012 – 2013. 2012.
- [41] MS – Ministério da Saúde. Metodologia de Desenvolvimento SOA do Ministério da Saúde 2012 – 2013. 2012.
- [42] MS – Ministério da Saúde. Plano do Programa de Adoção de SOA do Ministério da Saúde 2012 – 2013. 2012.
- [43] MS – Ministério da Saúde. Termo de Referência para Contratação de Serviços Especializados. TR_ServiçosSOA_v4. 2012.

- [44] NESSI - Networked European Software & Services Initiative. NESSI Strategic Research Agenda: Strategy to build NESSI, vol. 3 (2008).
- [45] NORTON David, Case Study: O2 Uses Offshore Software Factory to Drive SOA Initiative. 2010. Gartner (ID Number: G00170930).
- [46] OASIS. Reference Model for Service Oriented Architecture 1.0. 2006. Disponível em: <http://docs.oasisopen.org/soa-rm/v1.0/soa-rm.pdf>, acessado em 7/01/2013.
- [47] OPEN GROUP. The Open Group Architecture Framework – TOGAF, version 9.1, The Open Group Standard, 2011.
- [48] ORACLE. Oracle® Unified Method (OUM) Oracle’s Full Lifecycle Method for Deploying Oracle-Based Business Solutions. 2012. Disponível em: <http://www.oracle.com/us/products/consulting/resource-library/oracle-unified-method-069204.pdf>, acessado em 20/01/2013.
- [49] OSIMM. The Open Group Service Integration Maturity Model (OSIMM).ISBN: 1-931624-81-X. Document Number: C092. 2009.
- [50] PAPAZOGLU, M.P., YANG, J. Design Methodology for Web Services and Business Processes. In: Proceedings of the Third International Workshop on Technologies for E-Services (TES 2002). Volume LNCS; Vol. 2444., Springer-Verlag (2002).
- [51] PAPAZOGLU, M.P. Service oriented architectures: approaches, technologies and research issues. 2007. Disponível em: <http://link.springer.com/article/10.1007%2Fs00778-007-0044-3?LI=true#page-1>, acessado em 14/01/2013.
- [52] PAPAZOGLU, M.P., TRAVERSO, P., DUSTDAR, S., LEYMAN, F. Service-oriented Computing: A Research Roadmap, 2008. Disponível em: <http://www.worldscientific.com/doi/abs/10.1142/S0218843008001816>, acessado em 23/04/2013.

- [53] PAPAZOGLU, MP.; HEUVEL, WJvd. Service-oriented design and development methodology. *Int J Web Eng Technol* 2(4):412–442. 2006. Disponível em: <http://arno.uvt.nl/show.cgi?fid=106517>, acessado em 16/01/2013.
- [54] PMI - Project Management Institute. A Guide to the Project Management Body of Knowledge: PMBOK Guide, 4th edition, Project Management Institute, 2008.
- [55] RAMOLLARI, E.; DRANIDIS, D.; SIMONS A. A Survey of Service Oriented Development Methodologies. 2007. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.99.7494>, acessado em 18/01/2013.
- [56] RICHARDSON, R. J. et al. Pesquisa social: métodos e técnicas. São Paulo: Atlas, 1999.
- [57] SELLTIZ, C. Métodos de pesquisa nas relações sociais. São Paulo: Herder, 1967.
- [58] SIAU K.; ROSSI M. Evaluation of information modeling methods—a review. In: Proceedings of the thirty-first annual Hawaii international conference on system sciences, vol 314. IEEE Computer Society, Washington. 1998. Disponível em: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=648327&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D648327, acessado em 18/01/2013.
- [59] TANG L.; DONG, J.; PENG T. A Generic Model of Enterprise Service-Oriented Architecture. 2008. Disponível em: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4730454&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4730454, acessado em 20/01/2013.
- [60] TSAI, W.T. Service-oriented system engineering: A new paradigm. In: Service-Oriented System Engineering, 2005. SOSE 2005. IEEE International Workshop, Beijing, China (2005).
- [61] TSAI, W.T. et al. Service-oriented system engineering (SOSE) and its applications to embedded system development, 2007. Disponível em:

- <http://link.springer.com/content/pdf/10.1007%2Fs11761-007-0003-2.pdf>, acessado em 18/06/2013.
- [62] WALKER, L. IBM business transformation enabled by service-oriented architecture. IBM Systems Journal (Volume:46 , Issue: 4), 2007. Disponível em: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5386589&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5386589, acessado em 14/04/2013.
- [63] WANG, Lizhe et al. Cloud Computing: a Perspective Study, 2010. Disponível em: <http://link.springer.com/article/10.1007/s00354-008-0081-5>, acessado em 18/06/2013.
- [64] WINTERGREEN RESEARCH. SOA Market Research Report, 2012. Disponível em: <http://www.reportlinker.com/p0153909-summary/Services-Oriented-Architecture-SOA-Infrastructure-Market.html>, acessado em 18/06/2013.
- [65] WINTERGREEN RESEARCH. SOA Market Research Report, 2013. Disponível em: <http://www.reportlinker.com/p0787955-summary/Global-Service-oriented-Architecture-Market.html>, acessado em 18/06/2013.
- [66] ZIMMERMANN O.; KROGDAHL P.; GEE C. Elements of service oriented analysis and design, 2004. Disponível em: <http://www.ibm.com/developerworks/webservices/library/ws-soad1/> , acessado em 15/01/2013.
- [67] ZIRPINS, C.; BAIER, T.; LAMERSDORF, W. A Blueprint of Service Engineering. In: First European Workshop on Object Orientation and Web Service (EOOWS), Darmstadt, Germany (2003).

APÊNDICES

TREINAMENTOS SUGERIDOS

Este apêndice apresenta um conjunto de treinamentos sugeridos para desenvolvimento das capacidades profissionais descritas nos perfis. Esses estão relacionados a conhecimentos específicos de SOA e da plataforma de *middleware* definida.

Administrador de Dados:

- IASA: *Information Architect*;
- Treinamento na ferramenta de AD definida.

Administrador de Infraestrutura SOA:

- IASA: *Infrastructure Architect*.
- Treinamento nas ferramentas de infraestrutura definidas.

Analista de Negócio:

- IASA: *Business Architect*.

Analista de Processo:

- BPMP: Treinamento em CBOK® BPM;
- Treinamento na ferramenta BPMS definida.

Analista SOA:

- SOASchool: Fundamentos de SOA & Computação Orientada a Serviço;
- SOASchool: Arquitetura e Projeto SOA;
- SOASchool: Projetos e Metodologia SOA;
- SOASchool: Tópicos Avançados em Análise e Modelagem de SOA;
- SOASchool: Laboratório de Análise e Modelagem de SOA.

Analista de Testes:

- ISTQB: *Foundation Level*;
- ISTQB: *Test Analyst*;
- ISTQB: *Test Manager*;
- ISTQB: *Test Automation*.

Arquiteto Corporativo:

- IASA: *Foundation Architect*;
- IASA: *Software Architect*;
- IASA: *Infrastructure Architect*;
- IASA: *Information Architect*;
- IASA: *Business Architect*;
- IASA: *Enterprise and Solution Architect*;
- Treinamentos no ambiente de infraestrutura corporativa definido.

Arquiteto SOA:

- SOASchool: Fundamentos de SOA & Computação Orientada a Serviço;
- SOASchool: Conceitos de Tecnologia de SOA;
- SOASchool: Arquitetura e Projeto SOA;
- SOASchool: Tópicos Avançados em Projeto e Arquitetura de SOA;
- SOASchool: Laboratório de Projeto e Arquitetura de SOA;
- Treinamento nas ferramentas de infraestrutura SOA definidas.

Auditor SOA:

- SOASchool: Fundamentos de SOA & Computação Orientada a Serviço;
- SOASchool: Conceitos de Tecnologia de SOA;
- SOASchool: Arquitetura e Projeto SOA;
- SOASchool: Projetos e Metodologia SOA;
- Treinamento nas ferramentas de Governança SOA definidas.

Desenvolvedor:

- Treinamentos específicos para essa formação na plataforma definida.

Especialista em Governança SOA:

- SOASchool: Fundamentos de SOA & Computação Orientada a Serviço;
- SOASchool: Conceitos de Tecnologia de SOA;
- SOASchool: Projetos e Metodologia SOA;
- SOASchool: Fundamentos de Governança de SOA;
- SOASchool: Tópicos Avançados em Governança de SOA;

- SOASchool: Laboratório de Governança de SOA;
- Treinamento nas ferramentas de Governança SOA definidas.

Gerente de Projeto:

- PMI: Formação PMP.

Testador:

- ISTQB: *Foundation Level*.