

ALIPIO FROTA FERRO

**APLICAÇÃO DE UM MÉTODO DE INSTANCIAÇÃO DE PROCESSOS
NO MODELO SOMA**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo, como parte dos requisitos para conclusão do curso de MBA em Tecnologia de *Software*.

SÃO PAULO
2012

ALIPIO FROTA FERRO

**APLICAÇÃO DE UM MÉTODO DE INSTANCIAÇÃO DE PROCESSOS
NO MODELO SOMA**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo, como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de *Software*.

Área de Concentração: Tecnologia de *Software*.

Orientador:
Prof. Dr. Jorge Luís Risco Becerra.

SÃO PAULO
2012

MBA / TI
2012
F418 OP

DEDALUS - Acervo - EPEL



31500022089

FICHA CATALOGRÁFICA

M2012 AQ X

Ferro, Alípio Frota

Aplicação de um método de instanciação de processos no modelo SOMA / A.F. Ferro. -- São Paulo, 2012.

95 p.

Monografia (MBA em Tecnologia da Informação) - Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

1. Programação orientada a objetos 2. Adaptação de processos I. Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II. t.

DEDICATÓRIA

Dedico este trabalho à minha mãe,
Maria do Socorro Frota Ferro, e a
todos que me ajudaram a realizá-lo.

AGRADECIMENTOS

Ao meu orientador Dr. Jorge Luis Risco Becerra, que com toda sua calma, paciência, experiência e sua genialidade me orientou para a realização deste trabalho.

À Universidade de São Paulo – USP, em especial à Escola Politécnica da Universidade de São Paulo – EPUSP, a oportunidade que me foi concedida para realizar o curso de especialização.

Ao Programa de Educação Continuada em Engenharia – PECE, que me ofereceu toda infraestrutura necessária para concluir este trabalho.

À equipe do laboratório de fábrica de *software da* EPUSP, que me ajudou com “dicas” sem as quais este trabalho perderia em qualidade.

À empresa que gentilmente forneceu-me todas as informações necessárias para a concretização deste experimento.

RESUMO

Com a crescente adesão da arquitetura orientada a serviço (SOA), pelas empresas, torna-se essencialmente necessário a adesão de um processo de desenvolvimento orientado a serviços que atenda s necessidades das empresas de forma eficiente e eficaz.

Este estudo propõe realizar a instanciação de um processo de desenvolvimento, para fábrica de *software*, a partir de uma arquitetura de referência SOMA, com a utilização do método proposto neste estudo.

Para verificar a aplicabilidade do método, ele foi aplicado em uma fábrica de *software* atuante no mercado.

ABSTRACT

During the increasing of adhesion service oriented architecture, it is essentially necessary to add a process of development oriented service which satisfies accurately the company's demands.

This article suggests a tailoring process of development (for companies which create software) which uses a reference architecture SOMA. In order to verify the applicability of this method, it has been used in active company which creates software.

LISTA DE ILUSTRAÇÕES

| | PÁG. |
|---------------------------------------------------------------------------------------------------|------|
| Figura 1 – Objetivos estratégicos..... | 19 |
| Figura 2 – Visão processo da fábrica de <i>software</i> que compõe a arquitetura de processo..... | 25 |
| Figura 3 – Objeto processo..... | 26 |
| Figura 4 – Relacionamento dos elementos de um processo..... | 27 |
| Figura 5 – Solução orientada a serviço organizada em camadas..... | 29 |
| Figura 6 – Fase do ciclo de vida do SOMA..... | 30 |
| Figura 7 – Instanciação da arquitetura de processo..... | 37 |
| Figura 8 – Processo de instanciação do SOMA..... | 38 |
| Figura 9 – Atividade de criar caso de uso..... | 45 |
| Figura 10 – Exemplo diagrama estrutural do SOMA e da atividade..... | 47 |
| Figura 1 - Guia de processo da empresa..... | 55 |

LISTA DE TABELAS

| | PÁG. |
|---------------------------------------------------------------------------------------------------|------|
| Tabela 1 – Descrição de requisito..... | 41 |
| Tabela 2 – Categorização de requisitos..... | 41 |
| Tabela 3 – Mapeamento do requisito com o processo da arquitetura de referência | 42 |
| Tabela 4 – Matriz de decisão..... | 43 |
| Tabela 5 – Caso de uso..... | 45 |
| Tabela 6 – Descrição da atividade..... | 46 |
| Tabela 7 – Descrição de papel..... | 46 |
| Tabela 8 – Descrição do artefato..... | 46 |
| Tabela 9 - Variáveis do experimento..... | 50 |
| Tabela 10 - Critérios de avaliação das variáveis..... | 50 |
| Tabela 11 - Relacionamento da referência com processos existente na empresa.. | 52 |
| Tabela 12 – Descrição de requisitos da empresa..... | 52 |
| Tabela 13 - Categorização de requisitos da empresa..... | 53 |
| Tabela 14 - Mapeamento do requisito com o processo da arquitetura de referência com ações..... | 54 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------|------------------------------------------------|
| ABES | Associação Brasileira das Empresas de Software |
| APIs | Application Programming Interface |
| BPMN | Business Process Modeling Notation |
| BPM | Business Process Management |
| CRM | Customer Relationship Management |
| CMM | Capability Maturity Model |
| CMMI | Capability Maturity Model Integration |
| ERP | Enterprise Resource Planning |
| ESB | Enterprise Service Bus |
| GSM | Goal-Service Modeling |
| IBM | International Business Machines |
| KPIs | Key Performance Indicators |
| RM-ODP | Reference Model Open Distributed Processing |
| QoS | Quality Of Service |
| ROI | Return On Investment |
| SOA | Service Oriented Architecture |
| SOMA | Service Oriented Modeling Architecture |
| TI | Tecnologia da Informação |
| UML | Unified Modeling Language |
| WSDL | Web Service Description Language |

SUMÁRIO

| | PÁG. |
|-------------------------------------------------------------------------|-----------|
| 1. Introdução..... | 13 |
| 1.1. Objetivo..... | 14 |
| 1.2. Justificativas | 15 |
| 1.3. Metodologia da pesquisa..... | 16 |
| 1.4. Estrutura do trabalho | 17 |
| 2. Princípios da orientação a serviço e fábrica de software..... | 18 |
| 2.1 Conceitos da computação orientada a serviço | 18 |
| 2.1.1 Princípios de projeto | 21 |
| 2.1.2 Arquitetura orientada a serviço SOA..... | 23 |
| 2.2 Fábrica de software e arquitetura do processo | 24 |
| 2.2.1 Conceitos de objeto processo | 26 |
| 2.3 Arquitetura e modelagem orientada a service SOMA..... | 27 |
| 3. Método de instanciação do SOMA..... | 34 |
| 3.1. Instanciação | 34 |
| 3.2. Estratégia de instanciação | 35 |
| 3.3. Processo de Instanciação | 37 |
| 3.3.1. Atividade 1: identificar requisitos da fábrica de software..... | 38 |
| 3.3.2. Atividade 2: especificar processo | 42 |
| 3.3.3. Atividade 3: Análise do modelo estrutural..... | 46 |
| 3.3.4. Atividade 4: Criar Guia de Processos | 47 |
| 4. Aplicação experimental | 48 |
| 4.1. Planejamento do experimento | 49 |
| 4.2. Aplicação do experimento..... | 51 |
| 4.3. Análise dos resultados | 55 |
| 5. Conclusão..... | 56 |
| 5.1. Trabalhos futuros | 56 |

| | |
|----------------------------------------------------------------|-----------|
| Referências..... | 57 |
| Apêndice A – Especificação de processo da empresa | 61 |
| Apêndice B – Artefatos do gerado pelo processo | 78 |

1. Introdução

Este trabalho consiste na proposta de um processo para realizar uma customização de um processo de desenvolvimento de soluções orientadas a serviço chamado SOMA (Service Oriented Modeling Architecture), que é uma metodologia industrial desenvolvida e utilizada pelos profissionais de serviços da IBM para apoiar a modelagem e arquitetura orientada a serviço SOA (Service Oriented Architecture).

SOA é um conceito usado para se projetar sistemas de *software* e o ambiente que o circunda para conceber serviços a aplicações do usuário final, processos de negócio executáveis ou outros serviços por meio de interfaces de serviços publicadas e que podem ser descobertas.

Serviço é um componente de *software* reusável não firmemente acoplado que engloba a funcionalidade discreta, podendo ser distribuída e acessada por meio de um programa Sommerville (2009).

Hoje em dia, os aplicativos, que são desenvolvidos para a automação de processos de negócio específicos e compartilhamento de dados, normalmente não atendem aos novos requisitos de interoperabilidade, pois não são devidamente projetados para atender a esta necessidade, gerando complicadas arquiteturas de integração ou grandes camadas de *middlewares* Marzullo (2009).

Para atender às necessidades dos requisitos de interoperabilidade é necessário construir um processo de desenvolvimento de *software* orientado a serviços que consiga produzir serviços de forma eficiente às necessidade dos aplicativos de automação.

A construção de serviços, utilizando os métodos tradicionais da orientação a objetos, é insatisfatória para a construção de serviços e para a construção de um ambiente orientado a serviço, porque os processos convencionais de desenvolvimento de *software* não foram construídos baseados nos conceitos da computação orientada a serviço, por isso não atendem plenamente às mudanças e aos obstáculos encontrados na sua concepção. Portanto, para se construir um processo de

desenvolvimento orientado a serviço é necessário basear-se nas técnicas, conceitos e estratégias orientadas a serviços.

1.1. Objetivo

O objetivo deste trabalho é instanciar o processo de desenvolvimento de *software* SOMA, aplicado em uma fábrica de *software*, o qual apoia a modelagem de serviços denominada SOA.

A instanciação é uma adaptação específica e única para um objeto processo de uma determinada classe processo, portanto realizar uma instanciação do SOMA consiste em realizar uma adaptação dos processos do SOMA para serem utilizados por uma fábrica de *software* de forma mais simples e aplicável.

Para este estudo será utilizado o método de instanciação fundamentado na tese de mestrado de Dias,L (2010) que utiliza de técnicas de *tayloring*, adequação dos processos, arquitetura de processos e método de criação de processos. Nesta pesquisa, o método de instanciação utilizará o modelo SOMA como modelo de referência, substituindo o modelo de referência CMMI utilizado por Dias,L (2010).

O processo de instanciação de Dias,L (2010) é baseado em uma arquitetura de processo, em que o objeto processo é seu elemento básico e é definido pela visão processos da RM-ODP e refinado pelas diferentes visões da arquitetura de processo. Assim, o processo de criação de objetos processo será aplicado em diferentes níveis de abstração (referencial e operacional) e será utilizado no ambiente de uma fábrica de *software*.

Segundo Borsoi (2008), ambiente de fabrica é criado com o objetivo de gerar produtos de *software*, com base nos requisitos de negócio da empresa que faz parte de uma organização denominada fábrica.

Diante dessa definição, as atividades da fábrica podem ser entendidas como:

- Atividades operacionais, utilizadas para produção do *software*.

- Atividades de gestão e controle de produção de software.
- Ações estratégicas de negócio, (missão, metas, objetivos) que se relacionam com os sistemas de informação.

A contribuição deste trabalho é realizar a aplicação prática do método de instanciação de processos com o modelo SOMA, em uma fábrica de *software* real, com o objetivo de obter como resultado um processo de desenvolvimento SOMA customizado.

1.2. Justificativas

Atualmente, o número de empresas que utiliza um modelo de qualidade é bem pequeno, devido à falta de método para instanciar os diferentes modelos existentes Dias,L (2010).

Este estudo aplicará um método de como instanciar um processo de desenvolvimento de *software*, fundamentado no modelo SOMA, oferecendo uma visão de como implementar um processo de desenvolvimento de *software* em uma empresa.

Haverá também a possibilidade de interpretação dos problemas que a empresa enfrenta com o processo de desenvolvimento de *software* orientado a serviço.

Ginsberg (1995) afirma que uma quantidade significativa de organizações, que produzem e adquirem *software*, necessitam que suas práticas sejam mantidas, portanto, é necessário entender como tais práticas são realizadas e como os *softwares*, adquiridos, irão atendê-las.

Segundo Zhao (2010), um aplicativo de negócios mostra a distribuição de requisitos de negócios entre *softwares* componentes. No entanto, as abordagens tradicionais de projeto dependem da arquitetura de *software* criada com base nos requisitos de negócios.

A utilização do método de instanciação do modelo SOMA, para criar processos para uma fábrica de *software*, facilita a construção de arquitetura de *software* orientado a serviço.

Após a realização da instanciação do modelo SOMA, obtém-se um processo de desenvolvimento que incorpora os principais processos para construir uma solução orientada a serviço, que, segundo Arsanjani (2011), o modelo SOMA possui aspectos globais e atividade específica para realizar a concepção de uma solução orientada a serviço.

Para fornecer um ambiente de desenvolvimento e recursos de automação ao projetar soluções de SOA de uma forma sistemática, é necessário possuir ferramentas e processo que atendam às necessidades do ambiente, Zhang (2008).

1.3. Metodologia da pesquisa

A pesquisa foi realizada a partir do estudo de métodos de instanciação de processos, arquitetura orientada a serviços, modelo SOMA de desenvolvimento de software. A seguir mostro as etapas da pesquisa:

Fase 1 – Esta fase consiste de realizar um levantamento de artigos, buscando artigos relacionados com o tema da pesquisa.

Fase 2 – Esta fase consiste na escolha de uma arquitetura de processos de desenvolvimento de software orientada a serviço.

Fase 3 – Esta fase consiste na escolha de um método de instanciação de processos para realiza sua aplicação prática.

Fase 4 – Esta fase consiste de um estudo detalhado do método de instanciação de processos escolhido.

Esta fase foi realizada uma adaptação no método de instanciação de processos escolhido, para utilizar a arquitetura de processos orientada a serviços escolhida na fase 2.

Fase 5 – Esta fase consiste em escolher uma empresa para realizar a aplicação do método de instanciação de processos adaptado.

Fase 6 – Esta fase consiste na aplicação do método de instanciação de processos, na empresa escolhida, seguindo as etapas metodológicas do experimento.

Fase 7 – Esta fase consiste em realizar uma conclusão sobre o experimento.

1.4. Estrutura do trabalho

A estrutura da dissertação está dividida em cinco capítulos, sendo eles:

Capítulo 1: Introdução: apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

Capítulo 2: Princípios da orientação a serviço e fábrica de software: apresenta o conceito sobre o tema do trabalho.

Capítulo 3: Método de Instanciação do SOMA: apresenta descrição o método para instanciar o SOMA.

Capítulo 4: Aplicação Experimental: o método proposto é aplicado em um ambiente de uma fábrica de *software* real.

Conclusão: descreve-se a conclusão do estudo realizado.

2. Princípios da orientação a serviço e fábrica de *software*

Este capítulo apresenta os estudos relevantes para o tema deste trabalho: os princípios da orientação a serviço, o processo de desenvolvimento SOMA e o conceito de fábrica de *software*.

2.1 Conceitos da computação orientada a serviço

O conceito da computação orientada a serviço é baseado, de forma mais aperfeiçoada, na computação distribuída, portanto possui o seu próprio princípio de projeto, assim como seu próprio paradigma de orientação a serviço, modelos, conceitos, linguagens, arquitetura, tecnologia de implementação e forma de governança.

Segundo Erl (2008), o paradigma da orientação a serviço é um conjunto abrangente de princípios de projeto específicos, que, quando aplicados, o resultado obtido é uma lógica orientada a serviços.

Serviço pode ser definido como um componente de *software*, que representa uma atividade ou conjunto de atividades de natureza intangível, efetuada por um provedor de serviço para que se alcance um objetivo desejado pelo consumidor do serviço.

As empresas possuem seus próprios objetivos estratégicos e ambiente de atuação, este por sua vez possui seus próprios requisitos e restrições que precisam ser acomodados dentro de serviços que serão fornecidos ou disponibilizados por meio de um *webservice*, pois, segundo Sommerville, é uma representação de alguns recursos computacionais e de informação que podem ser outros programas.

O *webservice*, por sua vez, irá dar suporte à obtenção dos objetivos estratégicos associados à computação orientada a serviços. Assim, os prestadores de serviços terão que desenvolver e publicar tais serviços que podem ser aplicáveis a vários

clientes em potencial, e estes terão que localizar os serviços que necessitam no repositório dos fornecedores.

Quando o cliente, no caso, consumidor, utiliza o serviço de um fornecedor, acaba gerando um potencial acoplamento entre cliente e fornecedor. Tal acoplamento acaba gerando problemas, pois um dos objetivos da computação orientada a serviço é a aplicação dos princípios de projeto. Os serviços em desenvolvimento, para os diversos clientes, em diferentes contextos, devem ser cuidadosamente analisados e projetados.

A computação orientada a serviços é muito atraente a qualquer organização interessada em aprimorar a eficácia da área de Tecnologia da Informação - TI. Segundo Erl (2009), um conjunto de objetivos e benefícios comuns forma uma visão atraente para a utilização da computação orientada a serviço e também estabelece um objetivo bem definido para que uma empresa adote com sucesso a orientação a serviço.

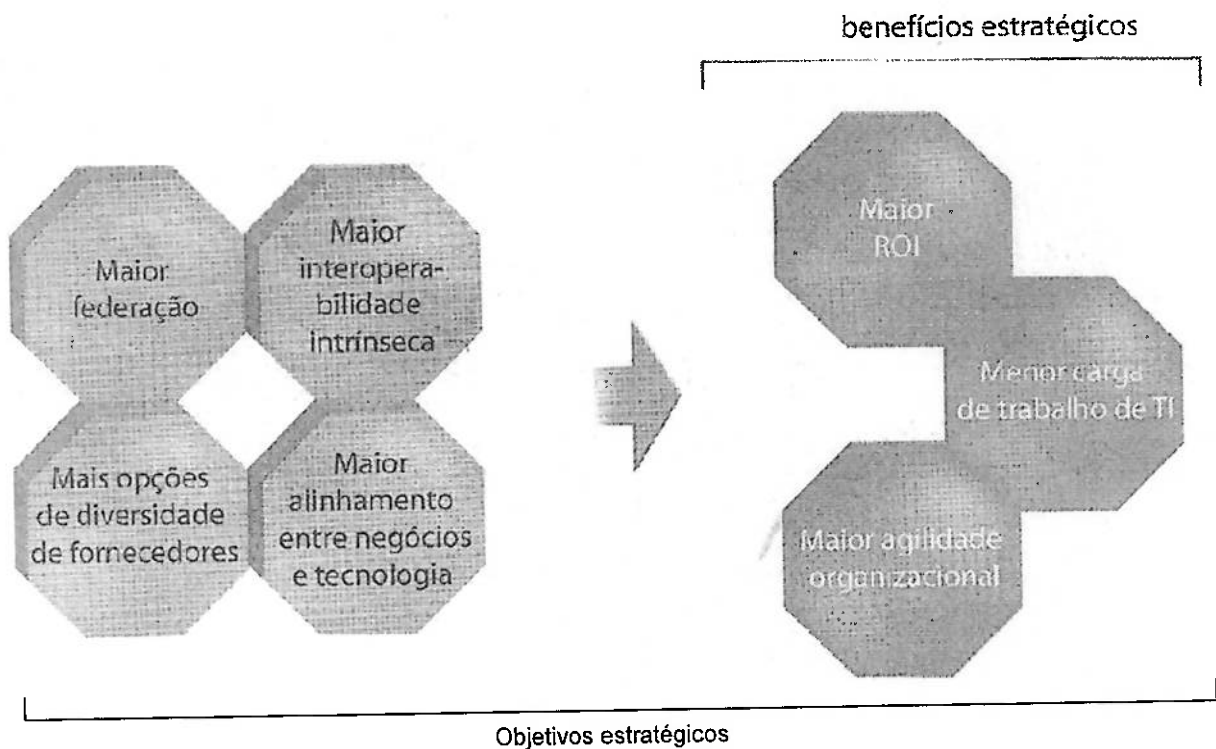


Figura 2 – Objetivos estratégicos, (Fonte: Erl,2009).

Os objetivos estratégicos contidos na Figura 1 são:

- A maior federação está relacionada com o ambiente de TI, em que os recursos e os aplicativos são unidos, mas cada um mantém sua autonomia. Uma solução SOA tem como objetivo aumentar a federação de uma corporação, e para alcançar este objetivo, é necessário aplicar padrões de projeto no desenvolvimento de serviço.
- Maior interoperabilidade está relacionada ao compartilhamento de dados; assim, quanto mais interoperável for o *software*, mais fácil será a troca de informações.
- Mais opções de diversidade de fornecedores, é a capacidade de uma organização de escolher os melhores serviços dos melhores fornecedores e utilizá-los juntos para um determinado fim.
- Maior alinhamento entre o negócio e a tecnologia significa que é necessário estabelecer camadas que agrupem e representem os modelos de negócio, assim os processos de negócio podem ser implementados como um serviço.

Portando, ao atingir tais objetivos, resulta em um aumento dos benefícios estratégicos que são:

1. Maior ROI que significa um retorno sobre o investimento. Assim, quando se fala em maior retorno sobre o investimento, observa-se que o potencial de reuso de serviços aumentou, analisando em quantos cenários diferentes ou em quantas composições o serviço foi utilizado. Somente assim será dada ênfase ao retorno sobre o investimento; portanto, pode-se dizer que o ROI está diretamente relacionado com o reuso de serviços.

2. Menor carga de trabalho de TI consiste em obter o menor desperdício de tempo, menor custo operacional, como despesas associadas à governança e evolução.
3. Maior agilidade organizacional tem um significado estratégico, como a capacidade de uma corporação responder a mudanças do mercado ou dos requisitos de negócio com uma eficiente agilidade organizacional, utilizando menor tempo e poucos recursos.

O entendimento desses objetivos, mostrado na Figura 1, é extremamente importante para que os princípios de projeto sejam vistos de uma forma mais consistente, dentro de um contexto estratégico corporativo. Segundo Erl (2009), existe uma ligação entre a aplicação dos princípios de projeto e os benefícios estratégicos específicos.

Assim, o resultado esperado para obter maior agilidade organizacional é possuir uma capacidade de resposta rápida em um pequeno intervalo de tempo para um potencial mercado.

2.1.1 Princípios de projeto

Atualmente, os trabalhos sobre a análise orientada a serviços e projeto estão em grande parte se concentrando na definição de processos para o desenvolvimento de processos de negócios e serviços, sem considerar a variabilidade nos detalhes nos princípios de projeto, que são aplicação prática de uma diretriz recomendada para a criação de serviço, no momento da concepção dos serviços.

Segundo Erl (2009), um princípio de projeto é uma prática que é utilizada por profissionais para solucionar problemas. Os princípios, aplicados na fase de análise orientada, a serviço do processo de desenvolvimento de uma solução SOA, são:

- Reuso de serviço: Consiste em criar serviços reusáveis que estabeleçam para as corporações recursos efetivos e compartilhados.

Segundo Erl (2009), o princípio de reuso possui os seguintes objetivos: aumentar a agilidade do negócio com a automação dos negócios, por meio da composição de serviço; permitir que a lógica do serviço possa ser reusada de forma que aumente o retorno sobre o seu investimento e que conceba a realização de modelos de serviços.

- **Autonomia de serviço:** É quando um serviço possui a capacidade de autogovernar-se, ou seja, o serviço que possuir maior confiabilidade e desempenho, no momento em que é invocado, e também possuir um alto controle na execução de sua lógica, tem, assim, a capacidade de tomar suas próprias decisões.
- **Visibilidade do serviço:** É quando os serviços são projetados para ser encontrados e interpretados de forma que seu propósito seja facilmente entendível pelos consumidores do serviço.
- **Contrato de serviço padronizado:** O termo contrato refere-se ao documento de descrição de serviços, em que são estabelecidos os termos de compromisso, fornecendo restrições e requisitos técnicos, como informações sobre a semântica que o serviço deve publicar. Segundo Erl (2009), este princípio estabelece que o contrato de serviço deve estar de acordo com os padrões de *projeto* aplicados a outros contratos de serviços.
- **A composição de serviço:** Consiste em projetar serviços que participem e estejam contidos em uma composição de serviços. Tal princípio tem como objetivo a reutilização de diferentes serviços para a concepção de uma composição de serviços, portanto, a composição de serviços concebida almeja assegurar que as características de *projeto*, juntas dão suporte a composição de forma eficaz.

- Baixo acoplamento de serviço: Este princípio estabelece que os serviços devem ser fracamente acoplados, ou seja, reduzir a dependência dos serviços que pode ser com o consumido ou com outros serviços.
- Abstração de serviço: Consiste em projetar um serviço, limitando as informações do contrato de serviço, somente para atender às necessidades específicas para as quais ele foi concebido, ou seja, ocultar as informações que não são necessárias e tornar o serviço funcionalmente útil ao consumidor.

Após aplicação dos princípios, durante a fase de análise orientada a serviço, os serviços devem ser projetados e depois construídos.

2.1.2 Arquitetura orientada a serviço SOA

Uma arquitetura de *software*, segundo a norma IEEE 1471, é uma organização fundamental de um *software*, envolvendo seus componentes, a relação entre eles e o ambiente e os princípios que guiam o seu projeto e sua evolução.

Segundo Dias, J. (2010), SOA é uma abordagem arquitetônica para a construção de aplicativos, usando um conjunto flexível de serviços reutilizáveis e baseados em padrões.

SOA é uma forma de projetar sistemas de *software* para o ambiente que o circula onde se concebe serviços para: Aplicações do usuário final, para processos de negócio executáveis ou para outros serviços por meio de interfaces de serviços publicadas e que podem ser descobertas.

Portanto, pode-se definir SOA como sendo uma forma de organização fundamental de um *software*, envolvendo seus serviços, a relação entre eles e o ambiente e os princípios que guiam o seu projeto.

O SOA também abrange as visões tradicionais de uma arquitetura formal, tais como; visão empresa, visão engenharia, visão informação, visão tecnologia e visão computação, enfim todas as visões estarão associadas ao repositório de serviços da corporação (Marzullo, 2009).

O paradigma da orientação a serviço, mesmo sendo um paradigma abstrato, possui muitas influências de outros paradigmas, isto é, os paradigmas da orientação a objeto, orientação a aspectos e do *Business Process Management* (BPM). Portanto o SOA encontra - se em um estado de desenvolvimento evolutivo sujeito à influência de outros paradigmas (Marzullo, 2009).

2.2 Fábrica de *software* e arquitetura do processo

Borsoi (2008) define a expressão fábrica de *software* como um ambiente de desenvolvimento de *software* constituído por processos que envolvem o ciclo de vida de *software* e são organizados em níveis hierárquicos corporativos: estratégico, gerencial e operacional, cujos processos são baseados de acordo com normas e modelos de qualidade.

Segundo Dias,L (2010), não se pode utilizar o mesmo processo de *software* para produtos diferentes. Partindo desse pressuposto, o entendimento do produto a ser criado torna-se indispensável.

Contudo uma fábrica de *software*, muitas vezes pode ser um departamento da empresa que, por sua vez, deve apresentar resultados financeiros, ter controles de custos, prazos, qualidade e ter respostas em curto, médio e longo prazo, para mudanças de requisitos de *software* (Dias, L, 2010).

Uma fábrica de *software* possui uma arquitetura de processo que é constituída de modelos de processos em que representam conceitos de processo em diferentes pontos de vista. Segundo Dias, L (2010), uma arquitetura de processos possui duas estruturas que podem ser observadas e representadas do ponto de vista estático e do ponto de vista dinâmico, conforme a Figura 2.

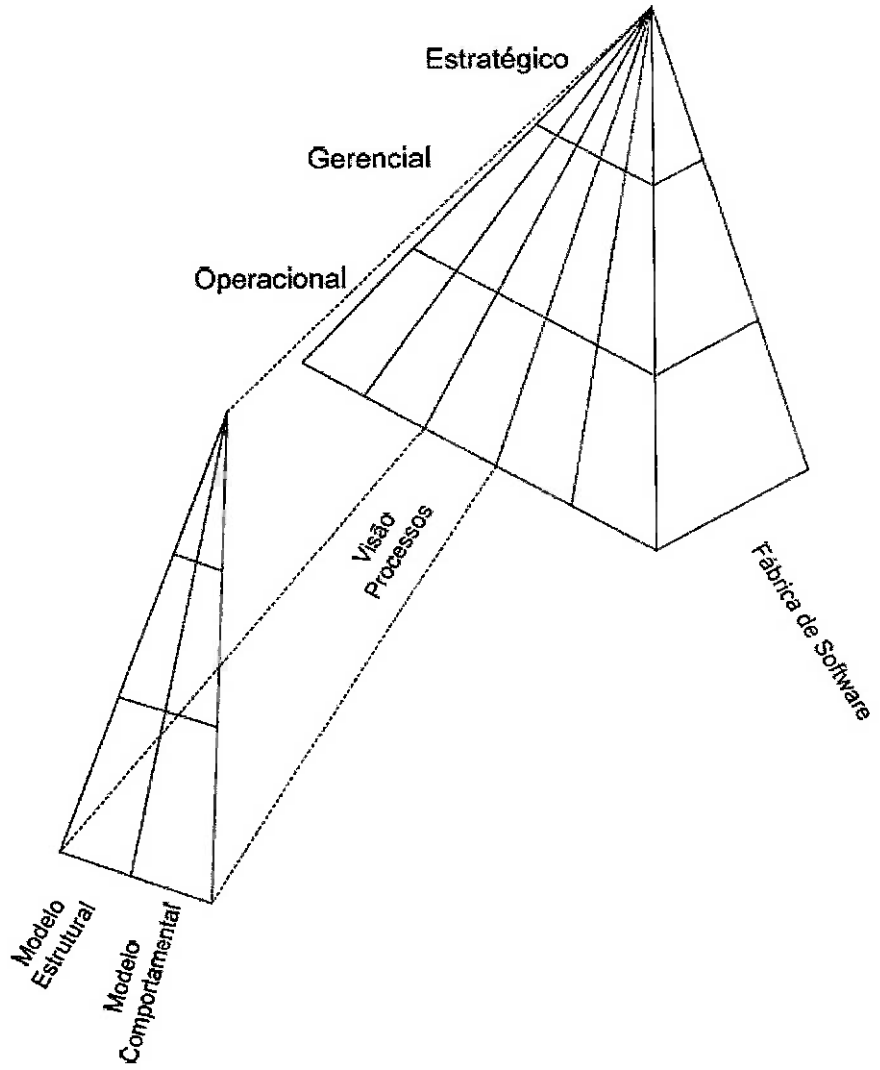


Figura 3 – Visão processo da fábrica de *software* que compõe a arquitetura de processo (Fonte: Dias,L, 2010).

2.2.1 Conceitos de objeto processo

O conceito de objeto processo vem do paradigma de programação e desenvolvimento chamado de orientação a objetos (OO), em que as coisas podem ser representadas por objeto e utiliza a linguagem UML para sua representação, conforme a Figura 3.

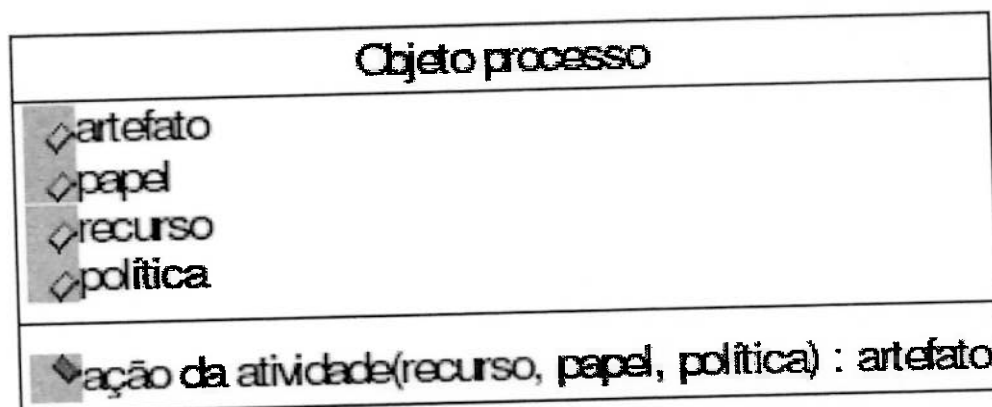


Figura 4 – Objeto processo (Fonte: Borsoi 2008).

Segundo Borsoi (2008), o objeto processo é definido por identidade (nome), comportamento (ações) e atributos (características), em que o comportamento representa as operações do objeto processo e ações dos relacionamentos entre outros objetos processos.

Observa-se que na Figura 3, para realizar a ação, os parâmetros de entrada, consistem em: recurso, papel e política, como resultado é obtido um artefato. Os atributos definem as características do processo e representam a conceitualização do processo: atividades realizadas por atores, que instanciam papéis que, com o auxílio de recursos, produzem artefatos de *software* e que o processo e os seus componentes estão sujeitos a políticas.

A Figura abaixo mostra o relacionamento dos elementos do objeto processo.

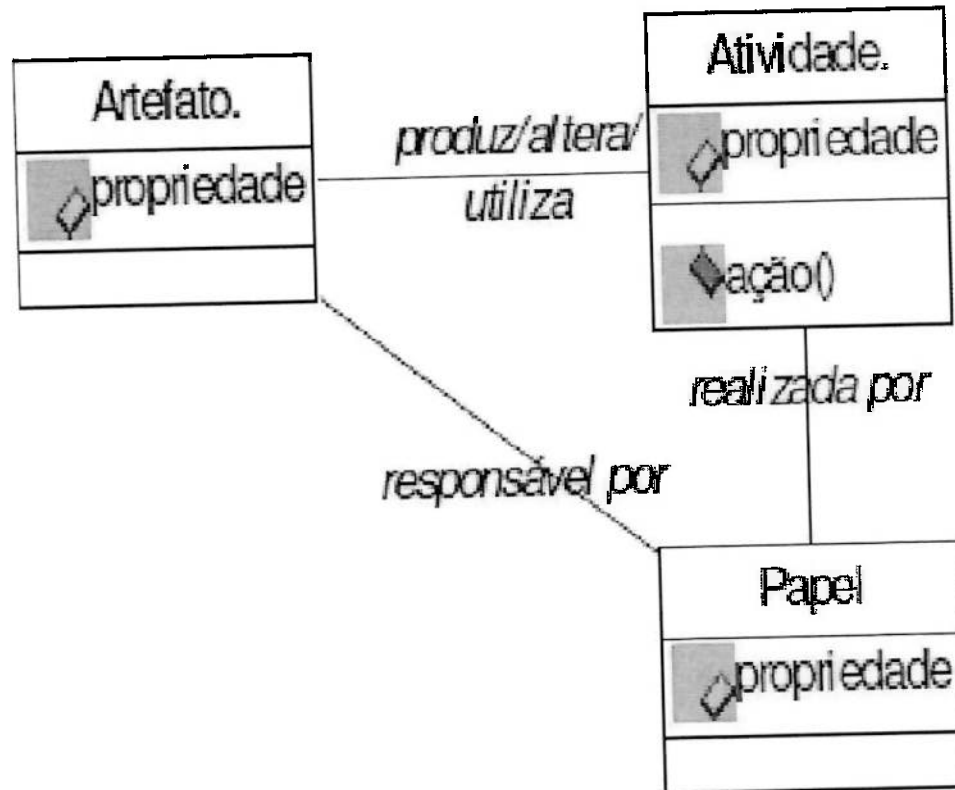


Figura 5 – Relacionamento dos elementos de um processo (Fonte: Borsoi 2008).

2.3 Arquitetura e modelagem orientada a serviço SOMA

A IBM desenvolveu uma metodologia voltada para ajudar a modelagem de uma solução orientada a serviço centrada sobre as técnicas para a identificação, especificação, realização de serviços, fluxos de serviços, composição e componentes em escala corpora identificar e organizar os serviços.

O SOMA (modelagem e arquitetura orientada a serviço) é um método criado pela IBM que incorpora os principais aspectos de uma solução orientada a serviço, abordando as atividades de comunicação, modelagem, construção e entrega, assim podendo ser integrado com os métodos de desenvolvimento de *software* já existentes.

Utilizando o ciclo de vida de desenvolvimento voltado para soluções baseadas em SOA, ou para qualquer outra solução, usando princípios orientados ao serviço, o SOMA define as principais técnicas e descreve os papéis em um projeto de arquitetura orientada a serviço e uma estrutura de divisão de trabalho.

Para organizar uma solução orientada a serviço, Arsanjani (2011) propõe a utilização de uma arquitetura de sete camadas, conforme Figura 5, são elas:

- Camada 1: É a camada que contém o sistema operacional, é formada de aplicações existentes, também chamado de sistemas legados, incluindo *Customer Relationship Management* (CRM), *Enterprise Resource Planning* (ERP) e pacotes de aplicativos existentes.
- Camada 2: É constituída de componentes que a empresa possui, tais componentes são responsáveis por implementar a funcionalidade, realizar manutenção e garantir a qualidade *Quality Of Service* (QoS) dos serviços expostos.
- Camada 3: É a camada de serviços em que se encontram os serviços que a empresa opta por financiar.
- Camada 4: Esta é a camada de orquestração do processos de negócios ou camada de coreografia; onde os serviços são agrupados em um fluxo por meio da orquestração ou coreografia, agindo em conjunto com uma única aplicação.
- Camada 5: É a camada de apresentação, que contém as normas, tais como *webservices*, *portlets* e outras tecnologias, que procuram utilizar os *webservices* nas interfaces dos aplicativos.
- Camada 6: É a camada de integração *Enterprise Service Bus* (ESB), que permite a integração de serviços.

- Camada 7: É a camada de qualidade de serviço (QoS), que fornece os recursos necessários para monitorar, gerenciar e manter a QoS.

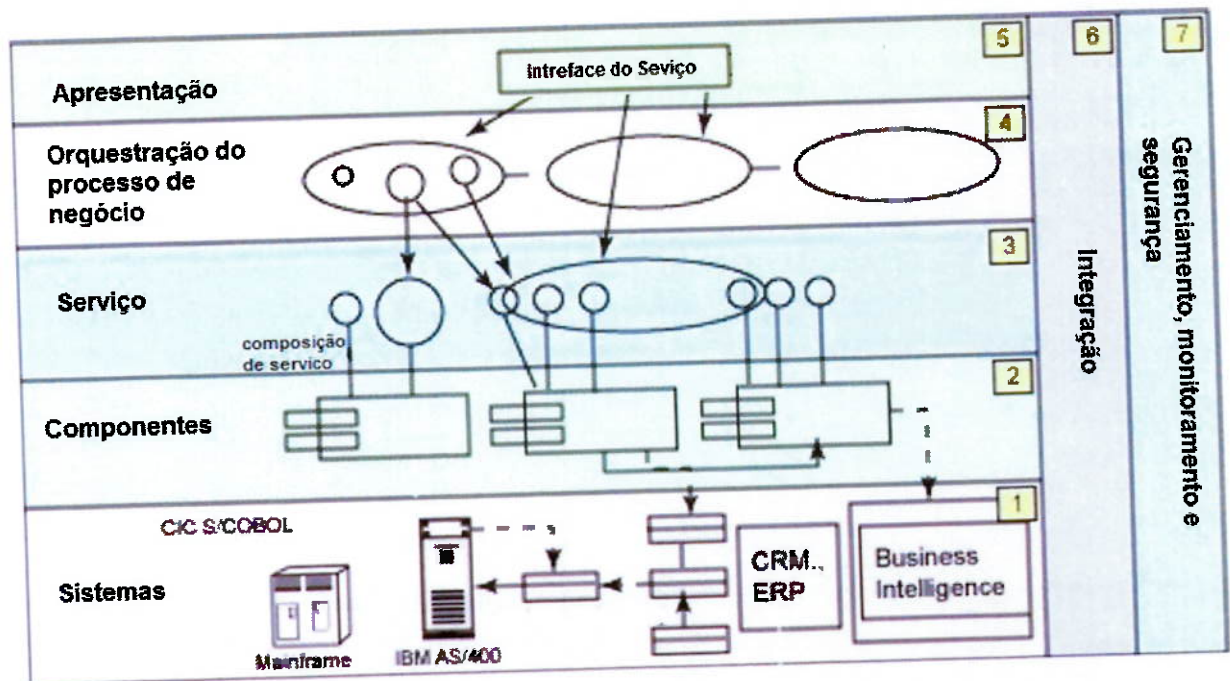


Figura 6 – Solução orientada a serviço organizada em camadas (Fonte: Arsanjani).

Observa-se na Figura 5 que a camada de sistemas é uma camada de middlewares constituída de componentes que realizam a comunicação com os sistemas existentes.

As camadas: orquestração do processo de negócio, serviço e componentes, podem ser entendidas como uma camada de processo de negócio.

As camadas: integração e gerenciamento, monitoramento e segurança podem ser entendidas com um middlewares entre as outras camadas. Os middlewares estabelecem uma comunicação entres as camadas e facilitam a integração.

Para construir um robusto ambiente SOA, é necessário adotar as fases do método do SOMA que contém elementos e recursos que podem ser utilizados conforme a necessidade para a construção de serviço como *software* ou *software como serviço*. A Figura 6 demonstra o ciclo de vida SOMA, em que cada etapa será detalhada a seguir.

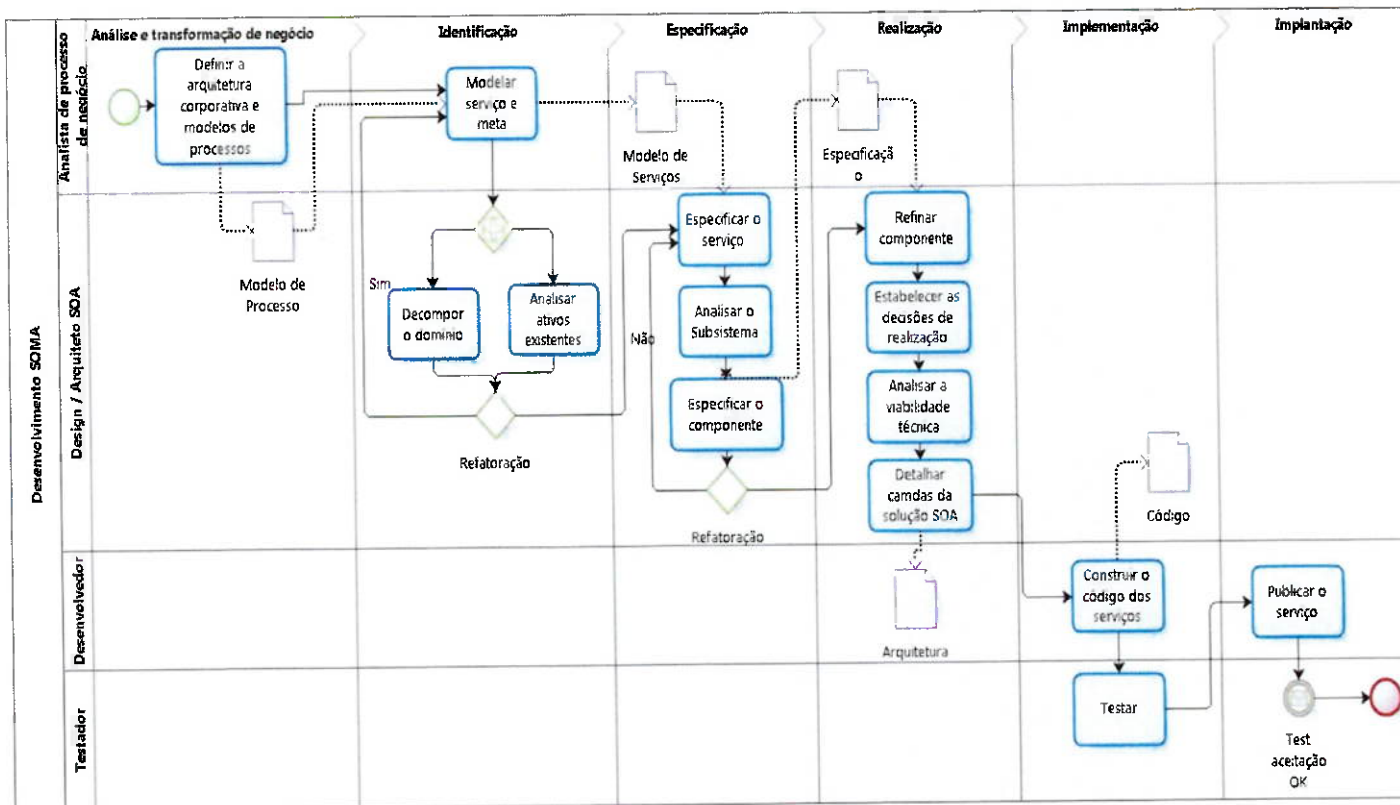


Figura 7 – Fase do ciclo de vida do SOMA (Fonte: adaptada de Arsanjani).

A fase de análise e transformação de negócio consiste na transformação dos processos de uma área da corporação, utilizados para conduzir as fases mostradas na Figura 6.

A análise e transformação do negócio tem início com a definição da arquitetura corporativa em que é apresentada toda a estrutura da empresa. Após a definição da arquitetura corporativa, definem-se os modelos de processos, com um *Business Process Modeling Notation* (BPMN), no qual são representados os serviços de negócios necessários de um domínio específico ou em toda a corporação.

A fase de identificação consiste na identificação dos três pontos fundamentais para a construção de uma solução SOA que são: serviços, componentes e fluxos. Para que os serviços possam ser identificados, utiliza-se um conjunto complementar de técnicas de identificação de serviços.

O processo de identificação de candidatos a serviços do SOMA tem o objetivo de criar um portfolio de serviços alinhado negócios e serviços de TI, que coletivamente, apoiam os processos de negócio e as metas da organização. É feito por meio do processo de avaliação de funcionalidade existente para ver se ele pode ser colocado no modelo de serviço, e pela determinação da falta de recursos de TI que serão necessários para apoiar o alinhamento da estratégia de negócios, metas e processos.

Para realizar a descoberta de serviços, é necessário utilizar três técnicas: modelagem de serviço e meta, decomposição de domínio e análise de recursos existentes.

A modelagem de serviço e meta, também conhecida como *Goal-Service Modeling* (GSM), consiste em manter o alinhamento dos serviços com os objetivos do negócio, refinando o escopo dos processos de negócio, assim, ajudando a identificar as partes do negócio que precisam ser transformadas e mudadas para alcançar as metas estabelecidas, usando os objetivos de negócio que desempenham um papel muito importante na identificação de processos de negócios. Também são identificados *Key Performance Indicators* (KPIs), métricas e eventos associados que possibilitam medir o desempenho dos serviços identificados e avaliá-los com base nas metas. O resultado da identificação são os serviços que têm o maior impacto sobre o negócio, isso também permite a rastreabilidade entre os objetivos de negócio e serviços.

A decomposição de domínio é uma técnica que consiste na identificação de: candidatos a serviços, áreas funcionais, limites para subsistemas, componentes de serviços, processos de negócios e fluxos de serviços. Eles são identificados por meio de atributos comuns encontrados nas funcionalidades de negócios. Nessa técnica, utiliza-se a análise *top-down* de domínios de negócios e modelagem de processos de negócios para identificar os serviços, componentes e fluxos. São analisadas ambas as visões estática e dinâmica do negócio, incluindo as informações, regras e variações.

A análise de recursos existentes consiste em analisar: os sistemas existentes, as aplicações e outros ativos disponíveis necessários para o projeto. O objetivo é identificar os ativos, tais como sistemas já utilizados, funcionalidades do legado que podem ser realizadas por serviços que estejam em conformidade com as necessidades do negócio. O foco está sobre os ativos existentes, em que se realiza

um mapeamento dos processos de negócio e das atividades do processo, sobre as aplicações existentes e suas interfaces.

Ao olhar para as funções disponíveis no sistema, fornecidas pelos ativos existentes, segundo Arsanjani (2009), deve-se realizar uma análise de ativos existentes, aplicando duas técnicas com o objetivo de identificar os candidatos a serviços e construir um inventário de candidatos a serviço que poderão ser aproveitados para construção de sistemas.

As técnicas concentram-se principalmente em sistemas existentes, que irão ajudar a atingir as metas e submetas identificadas no modelo de serviço ou fornecer funcionalidade para as atividades de processos de negócios que estão no escopo (Arsanjani, 2009).

Os serviços são inicialmente projetados como candidato durante a identificação, pois todos os recursos e funcionalidades do negócio, identificados, não devem ser expostos como serviços, pois o projeto provavelmente não terá o financiamento necessário disponível para cobrir todos os serviços identificados. Além disso, apenas o subconjunto relevante de candidato a serviços tornará os serviços expostos.

Portanto, os serviços, que ainda não foram expostos, serão implementados, mas não como webservices e sim como funcionalidades que serão implementadas no curso normal do desenvolvimento do *software*. As funcionalidades serão implementadas como métodos de um componente ou mapeadas para operações de *back-end* sistema legado.

A fase de especificação consiste em três atividades, a primeira é especificação do serviço que ocorre o detalhamento dos serviços, dos fluxos, das dependências entre os serviços, da composição, regras, políticas, operações, mensagens, requisitos não funcionais e as decisões de gestão do estado.

A segunda é análise de subsistema, que consiste em identificar por meio da decomposição de áreas funcionais os subsistemas existentes nas áreas funcionais, e refinados em serviço componentes de alta granularidade que são responsáveis por um aspecto funcional do subsistema.

A terceira é a especificação de componente, que consiste na especificação da interface do componente de serviço, atributos e sua estrutura com a utilização de padrões que ajudam a estruturar os componentes de serviço em um conjunto de componentes funcionais que dão suporte aos negócios.

Nessa etapa também define-se o que será feito na fase de realização, como construir um novo serviço interno ou expor um serviço que utilize uma aplicação legada ou utilizar serviços de terceiros; também se faz a coreografia dos serviços para ativar as funções e os processos de negócios e se realiza o projeto das mensagens dos serviços que incluem entrada, saída e mensagens de erro com o objetivo de eliminar as várias transformações de dados na camada de serviço.

A fase de realização consiste nas atividades de refinamento dos componentes, estabelecer as decisões de realização e análise de viabilidade técnica. As decisões, que foram definidas na etapa anterior, devem ser validadas e refinadas analisando sua viabilidade técnica para atender às decisões de arquitetura.

O refinamento dos componentes consiste em um detalhamento maior das mensagens e das interfaces dos serviços que contêm a assinatura do serviço. As funcionalidades das operações podem ser representadas por um *Web Service Description Language* (WSDL).

Para que se estabeleçam as decisões de realização, segundo Arsanjani (2009), é necessário projetar os riscos por meio de protótipos extensíveis, concebidos desde o início do projeto e a escolha de padrão, assim, pode-se lidar com vários domínios de problemas.

A análise de viabilidade técnica consiste na utilização de padrões de informação de serviço para a concepção da informação, por meio do Enterprise Service Bus (ESB), padrões para cenários de integração e padrões de regra para a criação de regra, que são utilizados com base no nível de maturidade da corporação, ao adotar o SOMA e sua infraestrutura tecnológica. Com o preenchimento gradual das camadas da arquitetura, tem-se uma visão atual que facilita a comunicação e demonstra a evolução da solução orientada a serviço e também tais camadas facilitam as decisões arquiteturais, devido ao melhor entendimento dos requisitos do domínio.

O ciclo de vida do método SOMA utiliza um fluxo iterativo incremental para conceber serviços, com foco nos riscos técnicos da entrega de *software* que são valorizados pela empresa. A fase da realização, por exemplo, é utilizada em todas as interações; assim, uma fase pode ser constituída de recursos que são usados por outras fases.

Na fase de implementação codifica-se os serviços construídos de forma personalizada e também os testes dos serviços são realizados para verificar se foram desenvolvidos, conforme o que foi especificado.

Os serviços são publicados na fase de implantação, para ser consumidos pelos clientes e também se aplicam os testes de aceitação dos serviços.

3. Método de instanciação do SOMA

Este capítulo apresenta uma forma para instanciar o SOMA, empregando um método de instanciação baseado em Dias, L (2010). A contribuição deste trabalho é uma instanciação de uma arquitetura de referência SOMA com a utilização de um método de instanciação para o nível operacional de uma empresa.

3.1. Instanciação

Segundo Borsoi (2008), instanciação é a especialização da arquitetura de processos para um ambiente mais específico, contudo, é necessário entender as características do ambiente específico, ou seja, os requisitos de negócio de uma empresa ou as características do negócio de um projeto específico.

Instanciação de processo pode ser definida com uma customização de processo, em que um processo, com um nível de abstração mais alto, é adaptado para atender aos requisitos do solicitante, portanto, tornando-se uma instância de processo atendendo às necessidades que são os requisitos específicos do cliente.

O método de Dias, L (2010) permite a criação do processo por meio de três atividades:

1. Especificação dos elementos que constituem um processo.
2. Modelagem dos processos que consiste em analisar e modelar os elementos especificados, observando sua dependência, limitações e relacionamentos.
3. Projetar os processos, detalhando o conteúdo e o formato dos *templates*.

3.2. Estratégia de instanciação

A estratégia para realizar uma instanciação do SOMA surgiu do estudo do método de instanciação de uma arquitetura de processo aplicado em fábrica de *software* de DIAS,L (2010), com base em uma arquitetura de referência, que foi criada utilizando o *Capability Maturity Model Integration (CMMI)*, que contém todos os modelos de processo necessários para uma fábrica *software*.

Para introduzir o SOMA, em um ambiente de uma fábrica de *software*, foi utilizado o conceito instanciação, utilizando como modelo de referência o SOMA.

A Figura 7 representa, de uma forma geral, a instanciação de uma arquitetura de referência SOMA para uma arquitetura operacional SOMA de um determinado contexto corporativo. Foi utilizado o diagrama de pacotes da UML para facilitar o entendimento da instanciação, cujo pacote superior representa a arquitetura de referência SOMA, e contém os elementos do processo de desenvolvimento SOMA, de uma forma abrangente, que são: as fases, atividades, tarefa, artefatos e papéis do SOMA.

A seta contínua, denominada instanciação, que sai do pacote arquitetura de referência SOMA e aponta para o pacote arquitetura operacional SOMA, representa a adaptação dos elementos do pacote de referência.

Quando se realiza a instanciação, tendo como base uma arquitetura de referência SOMA, tem-se como resultado da instanciação uma arquitetura operacional SOMA, em que o processo de desenvolvimento de *software* da empresa está em conformidade com os processos existentes na referência.

O "Contexto 1" representa o nível de abstração, em que está inserido o pacote arquitetura de referência SOMA.

O "Contexto 2" representa o nível de abstração operacional, específico de uma fábrica de *software*. O que determina o contexto dois é o grau de instanciação, ou seja, o quanto específico e determinado ele é. O nível de abstração é o grau dado pela instanciação de um contexto mais abrangente para um contexto mais específico.

A seta tracejada, que sai do pacote Arquitetura operacional SOMA e aponta para o pacote Arquitetura de referência SOMA, representa a dependência que existe entre os pacotes; portanto, o pacote Arquitetura operacional SOMA depende do pacote Arquitetura de referência SOMA.

A linha tracejada, que está entre o pacote Arquitetura de referência SOMA e o pacote Arquitetura operacional SOMA, representa a separação do nível de abstração entre o Contexto 1 e o Contexto 2, ou seja, o Contexto 1 mais abstrato e o Contexto 2 é mais específico em relação ao Contexto 1.

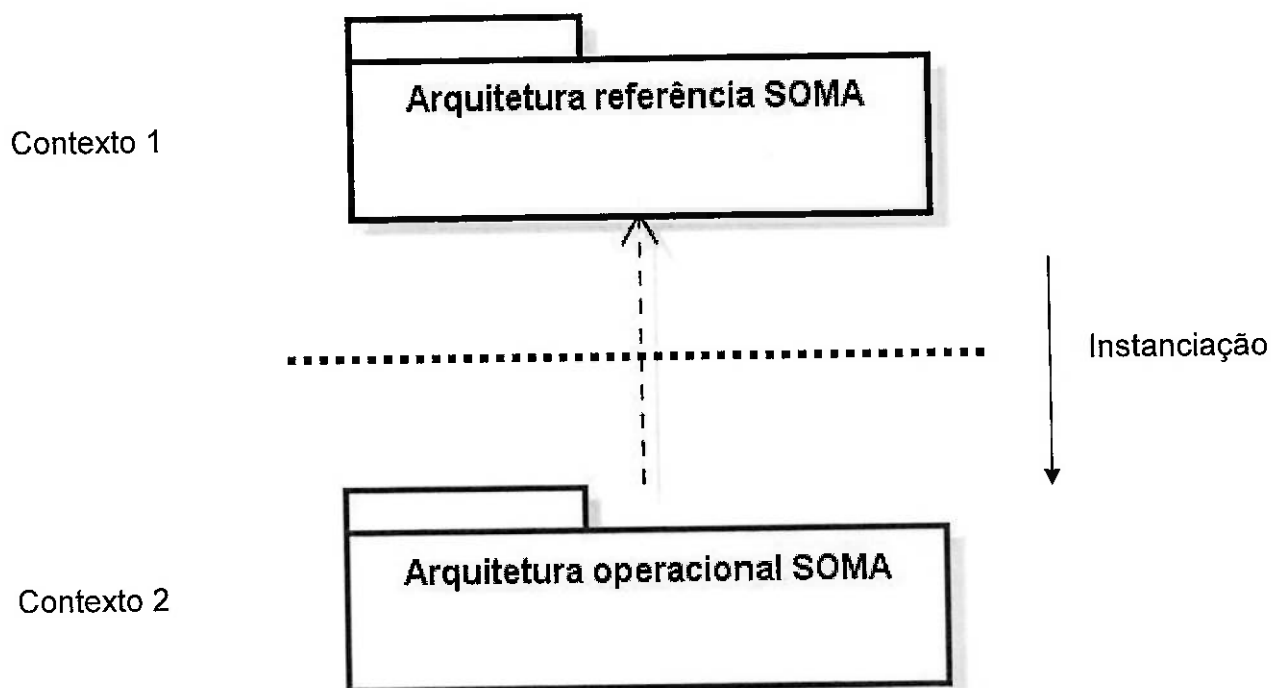


FIGURA 8 – Instanciação da arquitetura de processo.

Para ajudar a tomada de decisão, relacionada com a instanciação, é necessário conhecer o ambiente operacional interno da empresa, pois, só assim será possível definir uma estratégia de instanciação de processo, específica para o Contexto 2.

3.3. Processo de Instanciação

Ao realizar a instanciação, será utilizado o processo representado na Figura 8, que contém os passos necessários para obter uma arquitetura operacional SOMA, tomando-se por base uma arquitetura de referência SOMA.

O processo de Figura 8 foi construído a partir do processo de Dias, L (2010), onde foi realizado um estudo para identificar as principais atividades para realizar a criação de um processo de desenvolvimento de *software*, onde a principal diferença entre o processo representado na Figura 8 e o processo desenvolvido por Dias, L (2010) é

que o processo abaixo utiliza uma arquitetura de referência baseada no modelo SOMA.

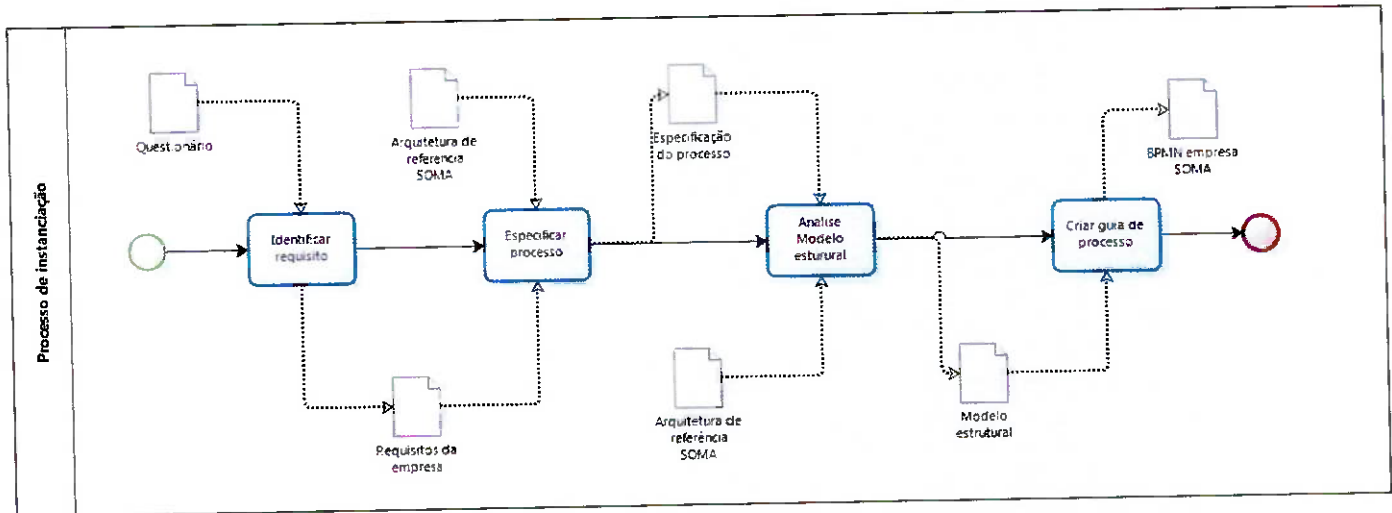


Figura 9 – Processo de instanciação do SOMA.

A Figura 8 utiliza a notação BPMN, pois facilita o entendimento e a organização do processo de instanciação. O círculo de cor verde representa o evento início do processo as setas contínuas, que saem do início e apontam para os retângulos azuis de bordas arredondadas, representa uma sequência, ou seja, a ordem das atividades; já os retângulos azuis de bordas arredondadas representam uma tarefa ou uma atividade básica.

Os artefatos gerados proveem informação do que o processo faz, onde são representados pelo retângulo branco de ponta dobrada, a seta tracejada, que sai de um artefato e entra em uma atividade, representa uma ligação entre artefato e atividade, ou seja, uma associação, o círculo vermelho representa o fim do processo.

3.4. Atividade 1: identificar requisitos da fábrica de software

O processo inicia-se com a atividade de identificar requisitos. Primeiramente há uma reunião com os membros da equipe de desenvolvimento, ocasião em que respondem a um questionário, com o objetivo de identificar os principais processos da empresa que também podem ser chamados de processos chaves.

O questionário é baseado nos processos específicos de implementação de *software* da norma 12207, sendo identificado o ciclo de vida, fase, atividade (início e fim), artefatos gerados e os papéis dos envolvidos no projeto. A estrutura básica do questionário foi dividida em três partes:

1. Introdução ao documento, onde se faz uma breve descrição do objetivo do questionário.
2. Instruções de como o questionário deve ser respondido.
3. Questões são aplicadas para todos os itens listados abaixo:
 - a) Especificação de requisitos.
 - b) Análise e Modelagem.
 - c) Implementação.
 - d) Teste.
 - e) Implantação.
 - f) Validação com o cliente.

As perguntas que são aplicadas aos itens listados acima são:

- Qual é o produto? (documentos)
- Como é realizado? (técnica utilizada)
- Por que é feito?
- Por quem é feito? (ator)
- Qual ferramenta é utilizada?

Também são aplicadas as questões a seguir:

- Quais são os sistemas utilizados para realizar o desenvolvimento do *software*?
- Qual a quantidade de pessoas que trabalham com o desenvolvimento do *software*?
- Qual é o tempo de duração dos projetos que já foram realizados?
- Como é feita a manutenção do sistema?
- Quais dos documentos utilizados pela empresa?

O objetivo desse questionário é entender como é a empresa, como são realizadas as atividades de cada uma das pessoas que participam do processo de desenvolvimento, quais são os artefatos gerados por elas.

Uma vez, com o questionário respondido, pode-se escrever e classificar os requisitos identificados que devem ser escritos e numerados.

Usando as diretrizes de Dias, L (2010), os candidatos a requisitos devem ser descritos da seguinte forma:

- 1) Corretos: Os requisitos de negócio devem ser corretos e representar o que a fábrica terá que executar.
- 2) Entendimento único: Os requisitos de negócio devem ter uma única interpretação.
- 3) Completos: Os requisitos de negócio devem conter todos os elementos necessários para caracterizá-los.
- 4) Consistentes: Devem estar de acordo com os outros documentos a partir dos quais foram entendidos.
- 5) Ordenados por importância: Os requisitos de negócio devem ser ordenados por importância para a instanciação.
- 6) Verificáveis: Os requisitos de negócio precisam ser verificáveis na fábrica instanciada.
- 7) Modificáveis: Os requisitos de negócio precisam ser estruturados de tal maneira que as mudanças ocorridas neles sejam realizadas de uma forma simplificada.
- 8) Rastreáveis: Os requisitos de negócio precisam ser rastreáveis, durante o processo de instanciação.

A descrição dos requisitos de um negócio é uma tarefa importante realizada durante a instanciação, portanto, usando as diretrizes acima, ajuda a descrever os requisitos de forma mais completa.

| Número Requisito | Descrição dos Requisitos |
|------------------|--------------------------|
| | |
| | |

Tabela 1 – Descrição de requisito (Fonte: Dias, L, 2010).

Primeiramente se realiza a descrição dos requisitos, conforme Tabela 1, a seguir são numerados.

Após a numeração os requisitos são classificados, conforme a Tabela 2, que foi criada utilizando as visões RM – ODP e as categorias sugeridas pelo framework CMM.

| Visão | Categoria | Número do requisito |
|------------|--------------------------------------|---------------------|
| Empresa | Políticas | |
| | Procedimentos | |
| | Unidades organizacionais | |
| | Característica da empresa | |
| | Papéis corporativos | |
| | Agências regulamentadoras | |
| | <i>Stakeholders</i> | |
| | Fornecedores | |
| | Mercado | |
| Engenharia | Tipo de <i>Software</i> desenvolvido | |
| | Recursos utilizados | |
| | Processos | |
| Informação | Padrões | |
| | Produtos de <i>software</i> | |
| Tecnologia | Tecnologias utilizadas | |
| Computação | Ferramentas | |

Tabela 2 – Categorização de requisitos (Fonte: Dias, L 2010).

Depois de numerados e devidamente classificados e terminada a atividade de identificação dos requisitos da fábrica.

3.4.1. Atividade 2: especificar processo

A atividade de especificação do processo consiste em identificar em que pontos dos processos contidos na arquitetura de referência SOMA, os requisitos da fábrica de software irão impactar.

Para realizar o mapeamento dos requisitos com os processos da arquitetura de referência, utilizou-se a Tabela 3.

- Na primeira coluna, coloca-se o número do requisito da fábrica de *software*.
- Na segunda coluna, o processo da arquitetura de referência SOMA, onde o requisito da fábrica de *software* irá impactar em alguma mudança.
- Na terceira e última coluna, coloca-se o nome do elemento do processo da arquitetura de referência em que o requisito da fábrica impactará especificamente.

Os elementos do processo de uma arquitetura de referência são: as atividades, os artefatos produzidos pelas atividades e suas respectivas responsabilidades.

O mapeamento, entre o requisito e o elemento do processo da arquitetura de referência, aponta onde cada requisito impacta no elemento do processo da referência, assim tem-se o local onde se deve aplicar ações para adaptar o requisito; pode-se afirmar que, após a adaptação, a empresa está em conformidade com a referência.

| Número Requisito | Processo da Referência Impactado | Elemento do Processo da Referência |
|------------------|----------------------------------|------------------------------------|
| | | |
| | | |
| | | |

Tabela 3 – Mapeamento do requisito com o processo da arquitetura de referência (Fonte: Dias, L 2010).

Após realizar o mapeamento dos requisitos, com o processo de arquitetura de referência, aplica-se a matriz de decisão para realizar as adaptações dos processos.

A construção de matriz de decisão surgiu com base na definição de fatores que ajudam a realizar a adaptação de processos de desenvolvimento para fábricas de software. Ao instanciar os processos de software, os gerentes de processo devem identificar os pontos principais de um processo.

A matriz decisão é utilizada como guia para ações de adaptação necessárias no momento da instanciação. A coluna fator, na matriz decisão da Tabela 4, representa um fator crítico de sucesso para a corporação, que considera relevante para a fábrica de software. Alguns fatores comuns foram identificados:

1. Importância: É quanto o requisito é importante para o negócio.
2. Complexidade: É quanto o requisito é simples ou complicado de ser realizado, ou seja, a complexidade da realização de alguma coisa.
3. Interdependência: É quanto o requisito é dependente de outro requisito da fábrica.

A segunda coluna da matriz decisão é a coluna nível, que significa uma medida atribuída a cada fator descrito acima com o objetivo de quantificar um fator. Os níveis são: alto, médio baixo.

| Fator | Nível | Ação |
|------------------|-------|-----------|
| Importância | Alto | Adicionar |
| | Médio | Redefinir |
| | Baixo | Remover |
| Complexidade | Alto | Redefinir |
| | Médio | Redefinir |
| | Baixo | Redefinir |
| Interdependência | Alto | Adicionar |
| | Médio | Redefinir |
| | Baixo | Remover |

Tabela 4 – Matriz de decisão.

A coluna ação, da Tabela 4, representa as ações de adaptação:

1. Adicionar: quando é necessário criar um elemento do objeto processo na arquitetura de processo operacional, que não existe na arquitetura de processos referência.
2. Redefinir: quando é necessário redefinir os elementos de um objeto processo.
3. Remover: quando é necessário retirar um elemento do objeto processo que existia na arquitetura de processos de referência.

As ações de adaptação servem para realizar a instanciação do processo da arquitetura de referência SOMA, para o caso de uso associado à ação.

Após a adaptação de todos os elementos do processo, aplica-se a técnica de descrição de cenário que consiste em descrever os processos com a utilização de caso de uso.

O caso de uso é uma narrativa de texto que descreve os processos da empresa de forma detalhada; assim, facilitando o entendimento dos processos, ou seja, a forma de como a empresa deve trabalhar.

As atividades necessárias para a criação dos cenários estão representadas na Figura 9. A bola de cor preta representa o início do processo, possui uma seta que aponta para a atividade "Definir os atores" que, por sua vez, é representada pelo retângulo de borda arredonda, onde são identificados os atores que atuam na fábrica de software.

Após a identificação dos atores, é realizada a atividade "Escrever o cenário", onde é relatado como são realizadas as tarefas da empresa.

A bola de cor branca, com uma bola de cor preta em seu interior, representa o fim da atividade de criação dos casos de uso.

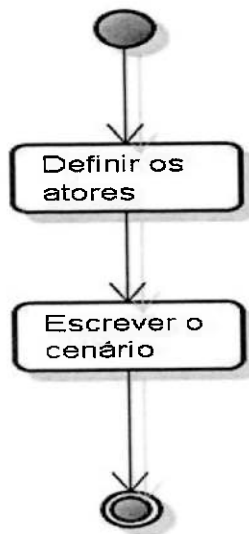


Figura 10 – Atividade de criar caso de uso.

Cada cenário deve ser descrito em forma de caso de uso, conforme a Tabela 5 que apresenta as seções contidas dentro de um documento de caso de uso, com seus respectivos comentários.

| Seção do caso de Uso | Comentário |
|-----------------------------------------|--------------------------------------------------------------------|
| Nome do caso de uso | Começa com um verbo. |
| Escopo | Escopo do processo. |
| O que quer atingir | Objetivo do processo ou de uma subfunção. |
| Ato Principal | Quem executa o processo. |
| Interessado ou interesses | Quem se importa com caso de uso e o que deseja? |
| Pré-condições | Quais artefatos são necessários para iniciar o processo? |
| Garantia de sucesso | O que precisa ser verdade quando a finalização é bem sucedida? |
| Cenário de sucesso principal | Um caminho típico, incondicional e otimista do cenário de sucesso. |
| Extensões | Cenários alternativos de sucesso. |
| Requisitos especiais | Requisito não relacionados. |
| Lista de variantes tecnológicas e dados | Método de entrada e saída e formatos de dados variáveis. |
| Frequência de ocorrência | Quantas vezes é realizada a tarefa? |
| Diversos | Ponto em aberto. |

Tabela 5 – Caso de uso.

Após a realizar a descrição dos cenários, é feita uma descrição detalhada dos papéis, atividades e artefatos produzidos no processo de desenvolvimento de software da empresa em foco, conforme as tabelas 6, 7 e 8, assim terminando esta atividade com todos os processos especificados.

| Descrição da atividade | |
|-------------------------------|-----------------------------------------------------------------------------------------|
| Nome da atividade: | Descrição do nome da atividade |
| Ator: | Papel responsável pela execução da atividade |
| Tipo: | Serve para organizar as atividades em grupos |
| Descrição: | Detalhamento sobre a definição atividade |
| Meta: | Meta a ser atingida durante a execução da atividade |
| Pré-condições: | Condições necessárias para início da atividade |
| Artefato de entrada: | Artefato utilizado durante a execução da atividade |
| Tarefa: | Detalhamento passo a passo da atividade |
| Recursos: | Recursos necessários para execução da atividade |
| Habilidades: | Habilidades necessárias ao ator para executar a atividade |
| Artefato de saída: | Artefato resultante após a execução das tarefas |
| Controle: | Restrição à realização da atividade, por exemplo, referentes a políticas ou legislação. |
| Métricas: | Medições que serão realizadas para avaliar o desempenho da atividade |

Tabela 6 – Descrição da atividade (Fonte: Dias, L 2010).

| Descrição de papel | |
|---------------------------|----------------------------|
| Nome do papel: | Descrição do nome do papel |
| Descrição do papel: | Detalhes sobre o papel |

Tabela 7 – Descrição de papel (Fonte: Dias, L 2010).

| Descrição do artefato | |
|------------------------------|----------------------------------------------------------------------|
| Nome: | Descrição do nome do artefato |
| Descrição: | Detalhamento sobre o artefato |
| Tipo: | Tipo que classifica os artefatos em grupos |
| Versão: | Versão atual do artefato |
| Proprietário: | Pessoa responsável pela criação do artefato |
| Política: | As políticas que o artefato está sujeito |
| Modelo: | <i>Template</i> , orientações ou critérios para produzir o artefato. |
| Composto: | Indicações dos artefatos que compõem o artefato descrito |
| Derivado: | Indicação do artefato que origina o artefato descritivo |

Tabela 8 – Descrição do artefato (Fonte: Dias, L 2010).

3.4.2. Atividade 3: Análise do modelo estrutural

Com a finalidade de realizar uma análise do modelo estrutural de processo, é construído o diagrama estrutural do processo, que é uma representação de várias

classes conceituais, concebidas com base na decomposição de uma fábrica de software em objetos, que representam o domínio conforme, Figura 10.

Nesta atividade, pegam-se os processos de uma fábrica de software, juntos com os elementos que constituem os processos para realizar a concepção do diagrama estrutural de processos. Este diagrama será constituído com os elementos do processo da referência do SOMA, como demonstrado na Figura 10.

O modelo estrutural contém a estrutura do objeto processo e dos seus elementos que são representados pelo diagrama estrutural do processo, em que são definidos todos os elementos que compõem um processo. Após a construção do diagrama estrutural de processos, esta atividade foi concluída.

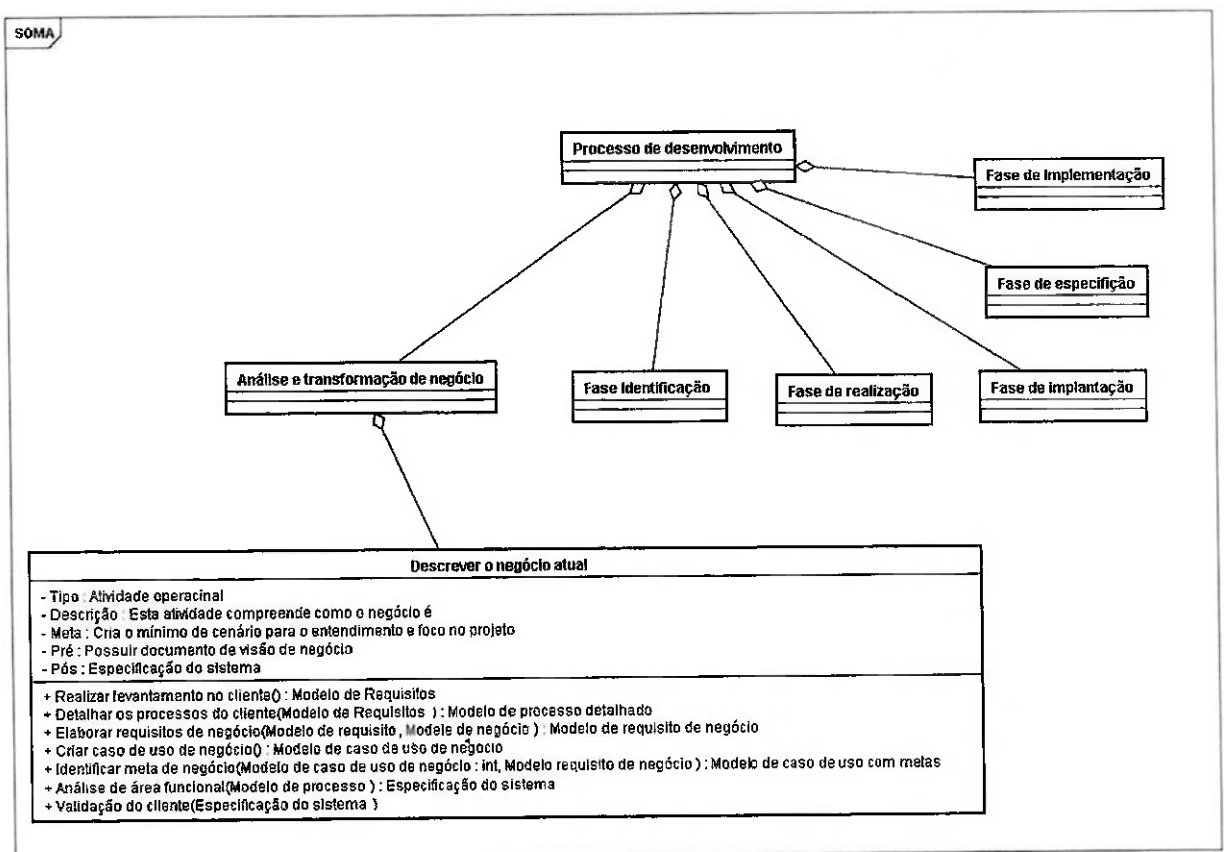


Figura 11 – Exemplo diagrama estrutural de processos.

3.4.3. Atividade 4: Criar o Guia de Processos

Nesta atividade concebe-se uma guia de processos para a corporação, que é representada por um Business Process Modeling Notation (BPMN). O BPMN é uma linguagem de representação de processos, assim, facilitando o entendimento e estudo do processo da empresa.

O guia de processos é um diagrama que contém todos os processos da empresa, e é apresentado ao fim da instanciação para a equipe de desenvolvimento com o objetivo de facilitar o entendimento dos processos da fábrica de *software*.

Ao criar o guia de processos SOMA, para corporação, utilizou-se o diagrama estrutural de processos, gerado na atividade anterior, neste guia são feitos alguns ajustes de nomenclatura e organização dos processos.

O guia de processos, produto concebido nesta etapa, é um modelo de processos operacional SOMA de uma fábrica de *software*, onde são apresentados os processos, que podem ser visualizados por todos os membros da equipe de desenvolvimento por meio de um BPMN, no qual se visualiza as atividades dos papéis da fábrica de *software*.

4. Aplicação experimental

O planejamento do experimento consiste em detalhar as fases do experimento, demonstrando as variáveis que serão analisadas e medidas no domínio de uma fábrica de *software*.

O experimento é constituído de variáveis que serão experimentadas e de variáveis fixas, cujo objetivo é obter o resultado para realizar comparações com outros experimentos que são utilizados para se comprovar teorias e validar medidas.

1. Planejamento do experimento

O experimento foi dividido em sete partes, que serão apresentadas a seguir, com base na tese de Dias, L (2010).

2. Motivação

Este experimento tem principal motivação o a aperfeiçoar entendimento da criação dos processos, para uma fábrica de software por meio da instanciação do modelo SOMA em uma empresa atuante no mercado.

3. Objeto

O objeto deste experimento é a aplicação do método proposto no capítulo três, utilizando o SOMA como referência para uma empresa e analisar sua instanciação para aperfeiçoar o conhecimento sobre instanciação.

4. Ponto de vista

O experimento será executado sob o ponto de vista do pesquisado.

5. Domínio

O domínio deste experimento é uma empresa de desenvolvimento de software, de pequeno porte, com sede em São Paulo, atuando no mercado desde 1996, tem como objetivo principal a prestação de serviços de desenvolvimento de *software* para o mercado corporativo. Seu público-alvo é composto por pequenas e médias empresas, dos mais diversos segmentos: comércio, indústria e serviços. Ao longo da sua existência conseguiu desenvolver módulos comerciais de software que aderem cerca de 70% dos processos de negócios atuais dos clientes, e os 30% que não aderem serão customizados.

6. Escopo

O escopo deste experimento é a aplicação do método proposto no capítulo três, para realizar a instanciação do SOMA para o domínio apresentado.

4.1. Planejamento do experimento

O planejamento da aplicação experimental consiste:

- definir as variáveis que serão observadas,
- definir critérios para avaliá-las,
- definir forma de medição.

- **Variáveis definidas**

As variáveis observadas neste experimento foram definidas como as principais variáveis para realizar a utilização do método proposto, portanto, as variáveis deste experimento são os artefatos do método. Ao observar estes artefatos, entende-se o seu relacionamento na prática, o que caracteriza a instanciação do SOMA.

As variáveis são apresentadas na Tabela 9.

| Variáveis |
|---------------------------------------------------------------------|
| Descrição dos papéis |
| Tabela de categorização de requisitos |
| Descrição das atividades |
| Descrição dos artefatos |
| Mapeamento do requisito com o processo da arquitetura de referência |

Tabela 9 - Variáveis do experimento.

- **Crítérios de avaliação**

Para cada variável foi definido um critério baseado no que será avaliado da variável escolhida.

| Variáveis | Crítério |
|---------------------------------------|---------------------------------------------------------------------------------------------|
| Descrição dos papéis | Verifica-se a alteração dos nomes dos papéis em relação à referência SOMA. |
| Tabela de categorização de requisitos | Verificam-se se todos os requisitos foram classificados. |
| Descrição das atividades | Verificam-se as alterações realizadas nas atividades dos processos em relação à referência. |

| | |
|---------------------------------------------------------------------|------------------------------------------------------------------------------|
| Descrição dos artefatos | Verificam-se as alterações realizadas nos artefatos em relação à referência. |
| Mapeamento do requisito com o processo da arquitetura de referência | Será analisado o relacionamento do requisito com os processos da referência. |

Tabela 10 – Critérios de avaliação das variáveis.

- **Forma de medição**

Para cada variável do experimento são realizadas medições, qualitativas levando em consideração os critérios definido para cada variável.

4.2. Aplicação do experimento

A execução do experimento teve início com a solicitação das informações a seguir:

- A apresentação da empresa.
- Como é realizado o levantamento dos requisitos na empresa.
- Como é realizada a especificação dos processos,
- Como é realizado o treinamento,
- Como é realizada a implementação de seus produtos,
- Quais documentos a empresa utiliza no processo de desenvolvimento.
- Informações sobre a equipe de trabalho e guia atual de processo.

Assim foi possível apurar todos os artefatos existentes na empresa com as informações fornecidas, sendo desnecessário realizar a aplicação de questionário e a descrição dos processos por meio de narrativa de caso de uso.

Como a empresa já produzia *software*, o primeiro passo foi realiza uma comparação do processo de desenvolvimento da empresa com os processos da referência, conforme demonstrado na Tabela 11.

| Atividades | Existe | Não Existe | Parcial | Observações |
|---------------------------------------|--------|------------|---------|---------------------------------------------------------------------------|
| Identificar serviço | | x | | |
| Especificar serviço | | x | | |
| Realizar serviço | | x | | |
| Teste | x | | | |
| Implementação | x | | | |
| Modelagem e transformação de negócio. | | | x | Possui processo para identificar requisitos de software e não de negócio. |
| Implantação | | | x | Possui processo de implantação, mas não para implantar serviços. |

Tabela 11 - Relacionamento da referência com processos existentes na empresa.

Após realizar a comparação dos processos, foi notada a necessidade de manter 12 processos, já existentes, com o objetivo de aumentar a aderência da fábrica de software ao SOMA.

| Número do requisito | Descrição do requisito | Decisão |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| 1 | Projetos de implantação de ERP próprio, adaptando e desenvolvendo novas funcionalidades. | Complexidade média |
| 2 | Melhorar a estimativa de prazos e esforço para o projeto. | Importância alta |
| 3 | Identificar serviços que tragam valor para o negócio. | Complexidade média |
| 4 | Analisar o negócio para identificar serviços. | Complexidade média |
| 5 | Transformação da funcionalidade existente no cliente para um modelo de serviços. | Complexidade média |
| 6 | O sócio da empresa exerce o papel de responsável de infraestrutura, esta atividade deve ser mantida, porém separa das demais. | Importância alta |
| 7 | Manter um controle sobre as versões de serviços implantados aos clientes para facilitar a manutenção dos serviços. | Interdependência alta |
| 8 | Utilizar o SOMA. | Complexidade alta |

Tabela 12 – Descrição de requisitos da empresa.

Após realizar a descrição dos requisitos, foi atribuído um fator de decisão para cada requisito, uma vez que não foi necessário realizar uma narrativa de caso de uso, porque a empresa já possuía os processos documentados. Ao fim da descrição, foi realizada a classificação, conforme Tabela 13.

As categorias com número de requisito em branco significa que não foi identificado nenhum requisito que se enquadrasse nessas categorias.

| Visão | Categoria | Número do requisito |
|------------|--------------------------------------|---------------------|
| Empresa | Políticas | |
| | Procedimentos | 4 e 5 |
| | Unidades organizacionais | |
| | Característica da empresa | |
| | Papéis corporativos | |
| | Agências regulamentadoras | |
| | <i>Stakeholders</i> | |
| | Fornecedores | |
| | Mercado | 3 |
| Engenharia | Tipo de <i>Software</i> desenvolvido | 1 |
| | Recursos utilizados | |
| | Processos | 2 e 6 |
| Informação | Padrões | 8 |
| | Produtos de <i>software</i> | |
| Tecnologia | Tecnologias utilizadas | |
| Computação | Ferramentas | 7 |

Tabela 13 - Categorização de requisitos da empresa.

Após realizar a categorização dos requisitos, foi criada a Tabela 14 para observar e aplicar as ações da matriz de decisão nos elementos da arquitetura de referência.

| Número requisito | Processo da Referência Impactado | Elemento do Processo da Referência | Ação |
|------------------|----------------------------------|------------------------------------------------------------------------------|-----------|
| 1 | Desenvolvimento | Todos os artefatos e atividades | Redefinir |
| 2 | Desenvolvimento | Atividade de estimativas | Adicionar |
| 3 | Desenvolvimento | Atividade de descrever processo de negócio | Redefinir |
| 4 | Desenvolvimento | Atividade de identificação de serviço | Redefinir |
| 5 | Desenvolvimento | Atividade de especificar e realizar serviço Artefato de modelo de serviço | Redefinir |
| 6 | Desenvolvimento | Todos os artefatos e atividades | Adicionar |
| 7 | | Atividade gestão de mudança e requisitos mais artefatos | Adicionar |
| 8 | Desenvolvimento | Todos os artefatos e atividade | Redefinir |

Tabela 14 - Mapeamento do requisito com o processo da arquitetura de referência com ações.

Após realizar a instanciação, foi especificado o novo processo de desenvolvimento SOMA para a empresa, observado no Apêndice A, e os artefatos gerados pelo processo no Apêndice B.

Com a especificação concluída, a guia de processo gerada para empresa pode ser observada na Figura 11. Como a empresa já possuía uma guia de processo, a Figura

11 foi sinalizada com um retângulo tracejado onde o processo onde foi realizada a instanciação.

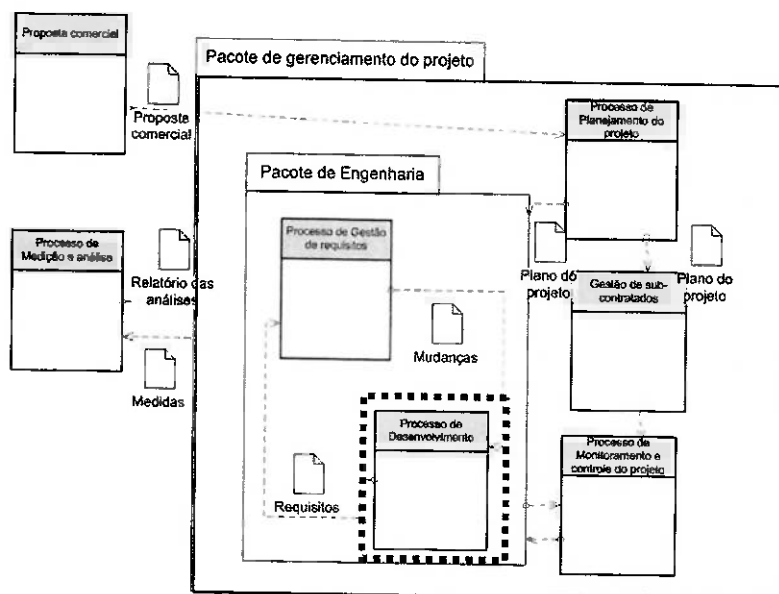


Figura 12 - Guia de processo da empresa.

4.3. Análise dos resultados

Os nomes dos papéis da referência sofrem alterações, porque, para manter a integridade da empresa, foram mantidos os nomes dos cargos da empresa.

Para a tabela de categorização de requisitos, todos os requisitos foram classificados, mas a maioria das categorias não foram preenchidas.

Para a descrição das atividades, muitas foram alteradas no processo de desenvolvimento, pois não eram voltadas para identificação de serviços.

Para os artefatos gerados pelas atividades do processo de desenvolvimento foram adaptados de forma que satisfaça o requisito associado.

O Mapeamento do requisito com o processo da arquitetura de referência mostrou-se preciso para realizar a adaptação do processo de desenvolvimento.

5. Conclusão

Este estudo foi baseado em conceito do processo de desenvolvimento do SOMA e técnicas de customização de processos para criação de processo de desenvolvimento *software* orientado a serviço para fábrica de *software*.

A teoria da instanciação aplicada em objetos processos ajuda o entendimento prático sobre a realização de uma instanciação de processos de desenvolvimento orientada a serviço para fábrica de *software*.

A aplicação prática do método mostra a importância de entender as necessidades de empresas produtoras de *software* com o seu processo de desenvolvimento.

A criação do processo de desenvolvimento de *software* foi aderente ao um modelo SOMA, portanto, a aplicação do método proposto contribui para a criação de um processo orientado a serviços.

5.1. Trabalhos futuros

Com este trabalho, outros temas poderão ser desenvolvidos, dentre eles:

- Melhorias no método de instanciação, adicionado conteúdos ou outras referências.
- Realizar análise mais profunda das verdadeiras vantagens, que SOMA instanciado traz para empresa tanto do ponto de vista estratégico quando do organizacional.
- Aplicação prática do método em diversas empresas para avaliar verdadeiras dificuldades encontradas na utilização das instanciações dos processos.

Referências

ARSANJANI, A.; ABDOLLAH, T.; ALLAM, A.; GANAPATHY, S.; GHOSH, S.; HOLLEY, K. SOMA: A method for developing service-oriented solutions. *Jornal IBM Systems*, Volume 47, Edição 3, p 377 – 396, Publicação 2008.

ARSANJANI, IBM, A. Service-oriented modeling and architecture, How to identify, specify, and realize services for your SOA. Disponível em: <<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-projeto1/>>. Acessado em 13 julho 2011.

Associação Brasileira das Empresas de Software. Mercado Brasileiro de Software: panorama e tendências, 2011. Disponível em: <http://www.abes.org.br/UserFiles/Image/PDFs/Mercado_BR2011.pdf>. Acessado em 10 julho 2011.

BORSOI, B. Arquitetura de processo aplicada na integração de fábricas de software. Tese (doutorado em engenharia elétrica) – Universidade de São Paulo. Orientador: Jorge Luis Risco Becerra, 2008, 177 p.

DAN, A.; ARSANJANI, A.; JOHNSON, R. Information as a Service: Modeling and Realization. *Systems Development in SOA Environments*, 2007. SDSOA '07: ICSE Workshops 2007. International Workshop on.

KELECIC, G.; CAR, Z.; Teaching Software Process: An Experience in Implementing RUP in a Student Project. *Telecommunications*, 2005. ConTEL 2005. Proceedings of the 8th international conference on 2.

DIAS, L.D. Método De Instanciação De Uma Arquitetura De Processos Aplicado Em Fábrica De Software. Tese (mestrado em engenharia elétrica) – Universidade de São Paulo. Orientador: Jorge Luis Risco Becerra, 2010, 114 p.

DIAS, J.J.L.; ALMEIDA, E.S.; LEMOS M.S.R. The Analysis Activity in a Systematic SOA-based Architecture Process. Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14 th IEEE International 2010, p. 147 – 156.

Erl, Thomas. SOA Princípios de desing de serviço, 2009.

GINSBERG, M. QUINN, L. Process tailoring and the software Capability Maturity Model. Technical report, Software Engineering Institute (SEI), 1995.

HAKI, M.K.; FORTE, M.W. Proposal of a Service Oriented Architecture Governance Model to serve as a practical framework for Business-IT Alignment. New Trends in Information Science and Service Science (NISS), 2010 4 th International Conference on 2010, p. 410 – 417.

IEEE. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE Std 1471-2000: Recommended Practice For Architecture Description Of Software-Intensive System, IEEE Computer Society, 2000.

KOKKO, T.; ANTIKAINEN, J.; SYSTA, T. Adopting SOA – Experiences from nine Finnish organizations. Software Maintenance and Reengineering, 2009. CSMR '09. 13 th European Conference on 2009, p.129 – 138.

MARZULLO, F. P. SOA na prática: Inovando seu negócio por meio de soluções orientadas a serviço, 2009, p. 123-148, p.311-329.

PRESSMAN, R. S. Engenharia de software: Uma abordagem profissional. 7ª ed., 2010, p. 52-80.

RM-ODP. Reference Model of Open Distributed Processing. ISO/IEC 10746-1 | ITUTRec. X.901. 1996.

RM-ODP. Reference Model of Open Distributed Processing. ISO/IEC 10746-2 | ITUTRec. X.902. 1996.

RM-ODP. Reference Model of Open Distributed Processing. ISO/IEC 10746-3 | ITU114 T Rec. X.903. 1996.

RM-ODP. Reference Model of Open Distributed Processing. ISO/IEC 10746-4 | ITUTRec. X.904. 1996.

RUI, H.; HAO, W.; ZHIQING, L. A Software Process Tailoring Approach Using a Unified Lifecycle Template. Computational Intelligence an Software Engineering, 2009, CiSE 2009, International Conference on 2009.

Sommerville, Ian. Engenharia de software 8ª edição, 2007. p. 510-523.

ZHANG, L.-J.; ARSANJANI, A.; BERNARDINI, F.; CHEE, Y.-M.; JALALDEEN, A.; PONNALAGU, K.; SINDHGATTA, R.; ZHOU, N. SOMA-ME: A platform for the model-driven *projeto* of SOA solutions. *Jornal IBM Systems*, Volume 47, Edição 3, p 397 – 313, Publicação 2008.

ZHAO, X., ZOU, Y. A Business Process Driven Approach for Generating Software Architecture. *Quality software (QSIC)*, 2010, 10th International Conference on 2010.

XU P.; Ramesh, B. Using Process Tailoring to Manage Software Development Challenges. *IT Professional* v.10, n.4, p.39-45, 2008.

Apêndice A – Especificação de processo da empresa**Especificação dos processos da
Fábrica de Software da empresa**

| | |
|---------------------------|-------------------------------|
| Autor: Alipio Frota Ferro | Data de emissão: 5/01/2011 |
| Revisor: | Data de revisão: |

Índice

| | |
|------------------------------------------------------|-----------|
| Objetivo | 63 |
| Processo..... | 63 |
| Processo de Desenvolvimento PD | 63 |
| PD 1. Descrever Processo de negócio atual | 65 |
| PD 2. Identificação de serviço | 66 |
| PD 5. Executar a especificação do serviço | 67 |
| PD 6. Executar a análise do subsistema | 68 |
| PD 7. Executar especificação de componente | 69 |
| PD 8. Decisão de realização | 70 |
| PD 9. Elaboração do Plano de Testes | 71 |
| PD 10. Analisa Plano de Testes | 71 |
| PD 11. Implementação..... | 72 |
| PD 12. Testes..... | 72 |
| PD 13. Correção de Problemas | 73 |
| PD 14. Implantação | 73 |
| PD 15. Especificação da infraestrutura | 74 |
| PD 16. Modelagem da infraestrutura | 74 |
| PD 17. Implementação da infraestrutura..... | 75 |
| PD 18. Testes da infraestrutura montada..... | 76 |
| PD 19. Correção de Problemas na infraestrutura | 76 |
| PD 20. Implantação da infraestrutura..... | 76 |

Objetivo

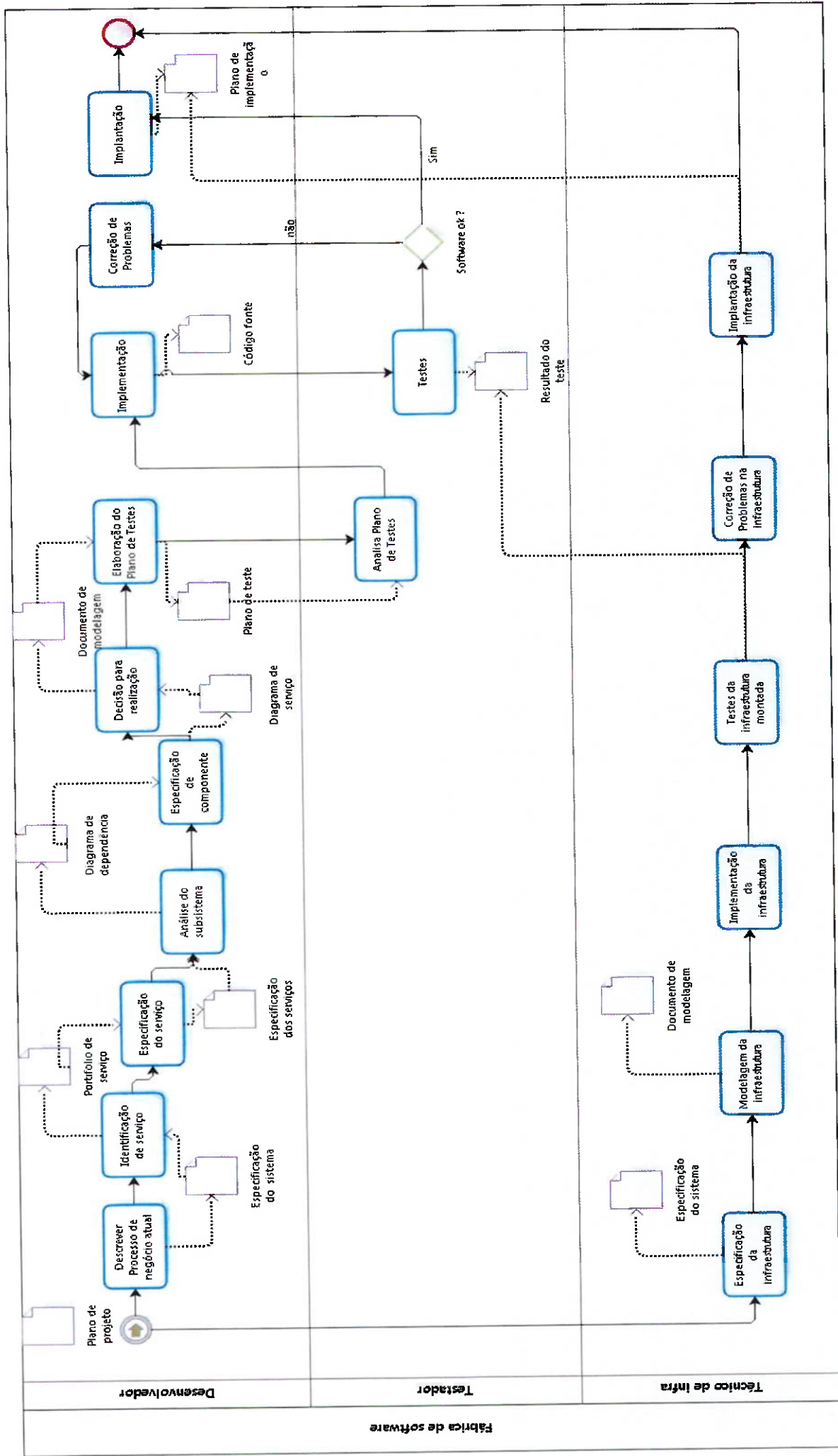
Este documento tem como objetivo apresentar a modelagem e descrição dos processos da Fábrica de Software da Empresa. Para isso será usada como base a modelagem BPMN de cada processo, descrevendo o seu significado e detalhando as atividades definidas no modelo.

Processo

Neste item será detalhado o processo de desenvolvimento que compõem a Fábrica de Software.

Processo de Desenvolvimento PD

Neste item são apresentados os elementos processuais que compõem o processo de desenvolvimento da Fábrica de Software. O processo é apresentado na Figura 2.



[1] Figura 13 – Processo de Desenvolvimento

PD 1. Descrever processo de negócio atual

Descrição: Geração do documento de especificação do sistema.

Papel: Desenvolvedor.

Pré-condições: Plano do projeto realizado.

Artefato (entrada): Proposta aprovada pelo cliente, plano do projeto, template do documento de especificação do sistema.

Tarefas:

- 1) Recuperação da base de templates o template de especificação do sistema.
- 2) Realização de um levantamento detalhado no cliente.
- 3) Detalhamento dos processos do cliente.
- 4) Elaboração dos requisitos do negócio.
- 5) Criação dos casos de uso de negócio (Igual, mas o foco é processo do negócio e não do sistema como é feito).
- 6) Identificar meta de negócio para cada caso de uso.
- 7) Análise de área funcional.
- 8) Validação da especificação com o cliente.

Recursos:

Base de templates

Habilidades: Conhecimento em engenharia de software, análise de área funcional e modelagem de área funcional.

Artefato (saída):

Especificação do sistema.
Timesheet preenchido.

PD 2. Identificação de serviço

Descrição: Concentra-se em análise top-down de domínios de negócios e modelagem de processos de negócios para identificar os serviços, componentes e fluxos. Serão analisada as visões estática e dinâmica do negócio, incluindo as informações, regras e variações.

Ator: Desenvolvedor

Pré-condições: Versão da especificação validada pelo cliente

Artefato (entrada): Especificação do sistema.

Tarefas:

- 9) Recuperação da base de templates o documento de modelagem.
- 10) Recuperação da especificação realizada.
- 11) Realiza a decomposição do processo
- 12) Realiza a análise de processo de negócio.
- 13) Realiza a análise de caso de uso de negócio (SOA).
- 14) Executar análise orientada a variação.
- 15) Associar serviço à meta.
- 16) Analisar recursos existentes.

Recursos:

Base de Templates
Ferramenta para modelagem de Classes

Habilidades:

Conhecimento em engenharia de software, decomposição de processo de negócio, modelagem de organizações, portfolio de serviços, análise de área funcional.

Artefato (saída):

Documento de modelagem
Timesheet preenchido

PD 5. Executar a especificação do serviço

Descrição: A especificação do serviço consiste na elaboração do projeto, modelagem e detalhamento dos serviços. As operações de serviços são designadas com base na granularidade de serviço, afinidade funcional, e coesão dos serviços na hierarquia de serviços. Elas são invocadas para executar uma função de negócios em uma implementação de TI, portanto, são um complemento fundamental no modelo de serviço.

Ator: Desenvolvedor

Pré-condições: Possuir documento de Modelagem

Artefato (entrada): Modelo de serviço

Tarefas:

- 17) Aplicar teste de exposição de serviços.
- 18) Dependências do Serviço.
- 19) Composição e Fluxos do Serviço.
- 20) Identificar Padrões de Segurança para os serviços.
- 21) Especificar Mensagens dos serviços.

Recursos:

Base de Templates

Habilidades: Conhecimento em engenharia de software, projeto de serviço,

Artefato (saída):

Documento de modelagem
Timesheet preenchido

PD 6. Executar a análise do subsistema

Descrição: O subsistema mostra os limites da lógica de TI para a funcionalidade do negócio. Eles são identificados por decomposição de áreas funcionais. Os subsistemas que existem nessas áreas funcionais são refinados em serviço componentes de alta granularidade que são responsáveis por um aspecto funcional do subsistema.

Ator: Desenvolvedor

Pré-condições: Ter realizado a especificação do serviço

Artefato (entrada): Modelo de serviço, modelo de análise de negócio.

Tarefas:

- 22) Dependências do Subsistema.
- 23) Identificar Componentes do Serviço.
- 24) Identificar Componentes Funcionais.
- 25) Identificar Componentes Técnicos.

Recursos:

Base de Templates

Habilidades: Conhecimento em engenharia de software, projeto de serviço.

Artefato (saída):

Documento de modelagem
Timesheet preenchido

PD 7. Executar especificação de componente

Descrição: Durante a especificação do componente, estruturam-se os componentes de serviço em um conjunto de componentes funcionais, que suportam as capacidades de negócios e também são observados os componentes técnicos responsáveis pelo de apoio da tecnologia e infraestrutura.

Ator: Desenvolvedor

Pré-condições: Ter realizado a análise do subsistema

Artefato (entrada): Diagrama de dependência de subsistema

Tarefas:

- 26) Especificar atributos dos componentes.
- 27) Identificar eventos e mensagens.
- 28) Criar diagrama de sequência do componente.
- 29) Criar diagrama de componente.
- 30) Executar Projeto Orientado à Variação.

Recursos:

Base de Templates

Habilidades: Conhecimento em engenharia de software e projeto de serviço.

Artefato (saída):

Documento de modelagem
Timesheet preenchido

PD 8. Decisão de realização

Descrição: Procura-se exercer as decisões de arquitetura, e projetar os fatores de risco através de protótipos extensíveis, concebido e desenvolvidos desde o início.

Ator: Desenvolvedor

Pré-condições: Possuir documento de modelagem

Artefato (entrada): Modelo de serviço

Tarefas:

- 31) Decisões da Realização de Serviço.
- 32) Alocar Componentes em Camadas.
- 33) Executar Exploração de Viabilidade Técnica Detalhada.

Recursos:

Base de Templates

Habilidades: Conhecimento em engenharia de software e arquitetura orientada a serviço.

Artefato (saída):

Documento de modelagem
Timesheet preenchido

PD 9. Elaboração do Plano de Testes

Descrição: Elaboração do plano de testes, segundo a especificação do sistema e as funcionalidades criadas.

Papel: Desenvolvedor

Pré-condições: Especificação e modelagem do sistema concluído.

Artefato (entrada): Especificação e modelagem do sistema concluído.

Tarefas:

- 1) Especificação do sistema e documento de modelagem.
- 2) Preparação dos planos de testes de caixa branca e caixa preta segundo especificação e modelagem do sistema

Recursos: Planilhas de teste.

Habilidades: Conhecimento em engenharia de software

Artefato (saída):

Plano de teste
Timesheet preenchido

PD 10. Analisa Plano de Testes

Descrição:

Verificar se todas as funções do software foram considerados no plano de teste.

Papel:

Testador

Pré-condições:

Plano de teste criado

Artefato (entrada):

Plano de teste

Tarefas:

- 1) Verificar todas as funções e os limites de restrição de entrada de dados, identificar problemas na base de dados, integrações, arquivo texto, configurações, mensagens de erros, dentre outros
- 2) Preparação do ambiente de testes

Recursos:

Planilhas de teste, bug tracker, ferramenta para execução de testes

Habilidades:

Conhecimento em engenharia de software

Artefato (saída):

Plano de teste revisado
Timesheet preenchido

PD 11. Implementação

| | |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descrição: | Geração dos códigos |
| Papel: | Desenvolvedor |
| Pré-condições: | Documento de Modelagem do sistema finalizado e revisado Ambiente de desenvolvimento instalado e testado |
| Artefato (entrada): | Modelos do sistema e especificação do sistema |
| Tarefas: | <ol style="list-style-type: none"> 1) Consulta as regras de codificação da empresa 2) Construir protótipo a fim de elaborar uma estrutura básica para o sistema 3) Verificação da possibilidade de reutilização de componentes, funções e códigos. 4) Realização da programação segundo os documentos de especificação e modelagem do sistema 5) Realização dos testes unitários a fim de remover os erros |
| Recursos: | Base de códigos para reuso Ambiente de desenvolvimento |
| Habilidades: | Conhecimento da linguagem de programação, UML e engenharia de software |
| Artefato (saída): | Código-Fonte Timesheet preenchido |

PD 12. Testes

| | |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descrição: | Realização dos testes, segundo o plano de testes efetuados e apontamento dos erros encontrados. |
| Papel: | Testador |
| Pré-condições: | Código fonte testado |
| Artefato (entrada): | Código fonte Plano de testes |
| Tarefas: | <ol style="list-style-type: none"> 1) Realização dos testes sugeridos no plano de testes utilizando dados reais. 2) Sugerir testes que não tenham sido contemplados. 3) Aponta os erros encontrados no sistema |
| Recursos: | Ambiente de testes, Planilhas de teste, bug tracker, ferramenta para execução de testes |
| Habilidades: | Conhecimento em engenharia de software |

Artefato (saída): Resultado do teste

PD 13. Correção de Problemas

Descrição: Correção dos problemas encontrados no teste

Papel: Desenvolvedor

Pré-condições: Plano de testes executados e erros apontados

Artefato (entrada): Resultado do teste

Tarefas:

- 1) Avaliação dos erros apontados *versus* a especificação
- 2) Correção dos erros
- 3) Testes das correções

Recursos: Ambiente de testes, planilhas de teste, bug tracker, ferramenta para execução de testes

Habilidades: Conhecimento em engenharia de software

Artefato (saída): Sistema corrigido

PD 14. Implantação

Descrição: Instalação do sistema no ambiente do cliente para que seja possível a homologação e então a entrada do sistema em produção.

Papel: Desenvolvedor

Pré-condições: Aceitação do cliente, testes do sistema e levantamento da infraestrutura do cliente realizados.

Artefato (entrada): Especificação do sistema, sistema desenvolvido

Tarefas:

- 1) Confecção dos planos de implantação
- 2) Confecção do manual do usuário
- 3) Documento de aceitação com cenários de teste
- 4) Implantação no cliente
- 5) Agendamento com o cliente para aceitação do sistema
- 6) Execução do roteiro de aceitação com o cliente
- 7) Assinatura da aceitação
- 8) Treinamento

Recursos: Código fonte disponível e acesso ao ambiente do cliente

Habilidades:

Artefato (saída): Conhecimento em engenharia de software
Plano de implantação, Manual do usuário, roteiro de aceitação.

PD 15. Especificação da infraestrutura

Descrição: Geração do documento de especificação dos requisitos para a infraestrutura.

Papel: Técnico de infraestrutura

Pré-condições: Plano do Projeto realizado.

Artefato (entrada): Proposta aprovada pelo cliente, plano do projeto, template do documento de especificação do sistema

Tarefas:

- 1) Recuperação da base de templates o documento de especificação do sistema
- 2) Realização do levantamento detalhado na infraestrutura do cliente
- 3) Detalhamento dos processos do cliente quanto à infraestrutura
- 4) Elaboração dos requisitos da infraestrutura do sistema baseado na situação da infraestrutura atual do cliente
- 5) Especificação do ambiente de desenvolvimento necessário para sistema.
- 6) Especificação dos equipamentos necessários para o projeto segundo a infraestrutura do cliente e dos requisitos do sistema
- 7) Anexa a especificação da infraestrutura do sistema ao documento de especificação do sistema.

Recursos: N/A

Habilidades: Conhecimento em engenharia de software e equipamentos de infraestrutura computacional.

Artefato (saída): Especificação da infraestrutura (contemplada no documento especificação de sistema)

PD 16. Modelagem da infraestrutura

Descrição: Geração dos modelos em UML da infraestrutura necessária para implantação do sistema

Papel: Técnico de infraestrutura

Pré-condições: Especificação do sistema finalizada.

| | |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Artefato (entrada): | Especificação do sistema. |
| Tarefas: | <ol style="list-style-type: none"> 1) Recuperação da base de templates o documento de modelagem da infraestrutura 2) Recuperação da especificação do sistema realizada. 3) Elaboração do diagrama da infraestrutura necessária para o cliente |
| Recursos: | Ferramenta para modelagem |
| Habilidades: | Conhecimento em engenharia de software |
| Artefato (saída): | Documento de Modelagem da infraestrutura do sistema e do ambiente de desenvolvimento Timesheet preenchido |

PD 17. Implementação da infraestrutura

| | |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descrição: | Aquisição, instalação e configuração dos equipamentos para o ambiente de desenvolvimento do sistema |
| Papel: | Técnico de infraestrutura |
| Pré-condições: | Documento de Modelagem do sistema e modelagem da infraestrutura |
| Artefato (entrada): | Modelos do sistema e especificação do sistema |
| Tarefas: | <ol style="list-style-type: none"> 1) Consulta das regras para equipamentos de infraestrutura da empresa 2) Consulta da especificação do sistema 3) Aquisição dos equipamentos especificados 4) Realização da instalação do ambiente de infraestrutura de desenvolvimento como modelado 5) Realização da configuração do ambiente de infraestrutura de desenvolvimento como especificado 6) Realização dos testes básicos unitários 7) Criação do manual de uso do ambiente (senhas, endereços de rede, nome de banco de dados, dentre outros) |
| Recursos: | Equipamentos de rede, computadores, sistemas operacionais, software de banco de dados e manuais |
| Artefato (saída): | Ambiente de desenvolvimento |
| Habilidades: | Conhecimento de redes, banco de dados e sistemas operacionais |
| Artefato (saída): | Ambiente de desenvolvimento de software instalado e configurado Timesheet preenchido |

PD 18. Testes da infraestrutura montada

| | |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descrição: | Realização dos testes, segundo o plano de testes e apontamento dos erros encontrados. |
| Papel: | Técnico de infraestrutura |
| Pré-condições: | Plano de testes elaborado |
| Artefato (entrada): | Plano de testes |
| Ações: | <ol style="list-style-type: none"> 1) Realização dos testes 2) Sugerir testes que não estão no plano 3) Identificação dos erros encontrados no ambiente de testes |
| Recursos: | Planilhas de teste, bug tracker, ferramenta para execução de testes |
| Habilidades: | Conhecimento em engenharia de software |
| Artefato (saída): | Resultado do teste do ambiente de desenvolvimento montado |

PD 19. Correção de Problemas na infraestrutura

| | |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descrição: | Correção dos problemas encontrados nos testes |
| Papel: | Técnico de infraestrutura |
| Pré-condições: | Plano de testes executados e erros apontados |
| Artefato (entrada): | Resultado do teste |
| Tarefas: | <ol style="list-style-type: none"> 1) Avaliação dos erros apontados <i>versus</i> a especificação 2) Correção dos erros 3) Testes das correções |
| Recursos: | Planilhas de teste, bug tracker, ferramenta para execução de testes |
| Habilidades: | Conhecimento em engenharia de software |
| Artefato (saída): | Ambiente de desenvolvimento corrigido |

PD 20. Implantação da infraestrutura

| | |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Descrição: | Instalação e configuração dos equipamentos e softwares para criar a infraestrutura básica para a instalação e configuração do sistema desenvolvido. |
| Papel: | |

| | |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pré-condições: | Técnico de infraestrutura |
| Artefato (entrada): | Testes do sistema realizados e correções efetuadas |
| Tarefas: | Especificação do cliente, sistema desenvolvido <ol style="list-style-type: none">1) Confecção do plano de implantação2) Aquisição dos equipamentos e software necessários3) Confecção do manual do usuário referente à infraestrutura4) Documento de aceitação com cenários de validação5) Agendamento com a equipe de desenvolvimento e o cliente para instalação e aceitação do sistema6) Implantação no cliente7) Testes da infraestrutura implantada8) Execução do roteiro de aceitação com o cliente9) Assinatura da aceitação da infraestrutura instalada. |
| Recursos: | Código fonte disponível, aquisição dos equipamentos e acesso ao ambiente do cliente |
| Habilidades: | Conhecimento em engenharia de software |
| Artefato (saída): | Plano de implantação, Manual do Usuário, roteiro de aceitação Timesheet preenchido |

Apêndice B – Artefatos do gerado pelo processo**Especificação do sistema XXXX**

| | |
|----------|------------------|
| Autores: | Data de emissão: |
| Revisor: | Data de revisão |

FOLHA DE CONTROLE DE REVISÕES

| Número da versão | Data de emissão | Registro de modificações |
|------------------|-----------------|--------------------------|
| 00 | | |
| 01 | | |
| 02 | | |
| 03 | | |
| 04 | | |
| 05 | | |
| 06 | | |
| 07 | | |
| 08 | | |
| 09 | | |
| 10 | | |

Índice

| | | |
|----------|----------------------------------------------------------------------|----|
| <u>1</u> | <u>Objetivo do Documento</u> | 81 |
| <u>2</u> | <u>Objetivo do Sistema</u> | 81 |
| | <u>2.1 Nome do Sistema</u> | 81 |
| | <u>2.2 Escopo</u> | 81 |
| | <u>2.3 Definições, Siglas e Abreviaturas</u> | 81 |
| <u>3</u> | <u>Descrição Geral</u> | 81 |
| | <u>3.1 Processos de Negócio do Cliente</u> | 81 |
| | <u>3.2 Perspectivas do Negócio</u> | 81 |
| | - <u>3.2.1 Entidade do negócio</u> | 81 |
| | - <u>3.2.2 Relacionamentos da entidade de negócio</u> | 81 |
| | - <u>3.2.3 Metas da entidade de negócio</u> | 81 |
| | - <u>3.2.4 Restrições</u> | 81 |
| | <u>3.3 Funções das entidades de negócio</u> | 82 |
| | <u>3.4 Requisitos não funcionais</u> | 82 |
| <u>4</u> | <u>Diagrama de Navegação do processo de negócio (Workflow)</u> | 82 |
| <u>5</u> | <u>Casos de uso Negócio</u> | 82 |
| <u>6</u> | <u>Referências</u> | 83 |

Objetivo do Documento

O objetivo do documento é auxiliar na obtenção de métricas que serão utilizadas para melhorias do processo produtivo da fábrica de componentes educacionais.

Objetivo do Sistema

Nome do Sistema

O sistema será chamado de Extranet Fit 4.

Escopo

Definições, Siglas e Abreviaturas

Propesct – Candidato a uma vaga de trabalho.

Representante Comercial – Vendedor da empresa que possui seu histórico de vendas nos bancos de dados

Avatar – imagem que o usuário utiliza em seu cadastro, o intuito desta imagem é identificá-lo.

Descrição Geral

Processos de Negócio do Cliente

Perspectivas do Negócio

Entidade do negócio

Ex.:

Relacionamentos da entidade de negócio

Metas da entidade de negócio

Restrições

Deve-se descrever restrições que afetarão o projeto dos serviços como, por exemplo:

- linguagem de programação a ser usada;
- *framework* a ser usado;
- ferramenta de desenvolvimento;
- sistema gerenciador de banco de dados a ser usado;
- sistema operacional;
- limites de recursos;

Não deve impor restrições que o cliente não tenha sugerido no documento de especificação.

Funções das entidades de negócio

Requisitos não funcionais

Deve-se descrever os requisitos não funcionais do projeto. Os requisitos não funcionais podem ser:

- Usabilidade
- Confiabilidade
- Disponibilidade
- Segurança de acesso (*security*)
- Manutenibilidade
- Portabilidade
- Segurança (*safety*)
- Desempenho

Atenção: Os requisitos não funcionais devem ser descritos de uma maneira que permita que sejam testados. O ideal é serem quantitativamente testados. Ex.: Disponibilidade d 99,9% em um ano. Chave de segurança de 128bits e aderência a norma ISO 17799.

Diagrama de Navegação do processo de negócio (Workflow)

Casos de uso de negócio

1. Uma breve descrição do caso de uso de negócio
2. Metas de Desempenho: uma especificação das métricas relevantes ao caso de uso de negócios, definida por um ator a área foco do caso de uso de negócio e uma definição de suas metas.
 - a. <nome da meta de desempenho>
 - b. Breve descrição da meta de desempenho.
3. Fluxo de Trabalho: o fluxo de trabalho do caso de uso ou Workflow.
 - a. Fluxo de Trabalho Básico
 - b. <nome da etapa do fluxo de trabalho>
 - c. 3.2. Fluxos de Trabalho Alternativos
 - d. 3.2.1 <nome da etapa do fluxo de trabalho>
4. Requisitos Especiais: os requisitos de casos de uso de negócios não cobertos pelo fluxo de trabalho de acordo com sua descrição.
 - a. <nome do requisito especial>
5. Pontos de Extensão
6. Pontos de extensão do caso de uso de negócios.
 - a. <nome do ponto de extensão>
Definição do local do ponto de extensão no fluxo de eventos.

7. Relacionamentos: os relacionamentos que envolvem o caso de uso são incluídos.
 - Para associações de comunicação, incluem-se uma breve descrição, a multiplicidade e os atores associados, além da navegabilidade da função do caso de uso.
 - Para relacionamentos de inclusão e de extensão, incluem-se os casos de uso associados e uma breve descrição.
8. Diagramas de Atividades
Esta seção inclui os diagramas de atividades que ilustram o fluxo de trabalho.
9. Diagramas de Casos de Uso
Esta seção inclui os diagramas de casos de uso que envolvem este caso de uso de negócios e seus relacionamentos.

Referências

Sistema XXX

Modelo Para Implantação

Autores:

Data de emissão:

FOLHA DE CONTROLE DE REVISÕES

| Número da versão | Data de emissão | Registro de modificações |
|------------------|-----------------|--------------------------|
| 00 | | |
| 01 | | |
| 02 | | |
| 03 | | |
| 04 | | |
| 05 | | |
| 06 | | |
| 07 | | |
| 08 | | |
| 09 | | |
| 10 | | |

Índice

| | | |
|------------|-----------------------------------------------------------------------------|----|
| <u>1</u> | <u>Objetivo do Documento</u> | 87 |
| <u>2</u> | <u>Descrição do ambiente de implantação (segundo a especificação)</u> | 87 |
| <u>2.1</u> | <u>Software do cliente</u> | 87 |
| <u>2.2</u> | <u>Hardware do cliente</u> | 87 |
| <u>2.3</u> | <u>Diagrama de topologia de rede do cliente</u> | 87 |
| <u>3</u> | <u>Atividades para implantação</u> | 87 |
| <u>3.1</u> | <u>Itens para backups</u> | 87 |
| <u>3.2</u> | <u>Versões de software para instalação</u> | 87 |
| <u>3.3</u> | <u>Migração de dados</u> | 87 |
| <u>3.4</u> | <u>Scripts e configurações necessárias</u> | 87 |
| <u>4</u> | <u>Testes após implantação</u> | 87 |
| <u>5</u> | <u>Validação do cliente</u> | 87 |

Objetivo do Documento

Esta seção é obrigatória e apresenta o documento, sua organização e público-alvo.

O objetivo do documento de Descrição de Testes é apresentar a estratégia de teste, contemplando os testes do objeto educacional. Descreve os testes a serem realizados e estabelece o planejamento das atividades de teste do sistema.

Descrição do ambiente de implantação (segundo a especificação)

Software do cliente

Software *versus* Versão que tenham influência na implantação

Hardware do cliente

Diagrama de topologia de rede do cliente

Atividades para implantação

Itens para backups

Versões de software para instalação

Migração de dados

Scripts e configurações necessárias

Testes após implantação

Validação do cliente

Modelo de Serviço XXX

| | |
|----------|------------------|
| Autores: | Data de emissão: |
|----------|------------------|

FOLHA DE CONTROLE DE REVISÕES

| Número da versão | Data de emissão | Registro de modificações |
|------------------|-----------------|--------------------------|
| 00 | | |
| 01 | | |
| 02 | | |
| 03 | | |
| 04 | | |
| 05 | | |
| 06 | | |
| 07 | | |
| 08 | | |
| 09 | | |
| 10 | | |

Índice

| | | |
|-----|-------------------------------------------------|----|
| 1 | <u>Objetivo do Documento</u> | 91 |
| 2 | <u>Portfólio de serviço</u> | 91 |
| 2.1 | <u>Lista de serviço (Identificação)</u> | 91 |
| 3 | <u>Especificação</u> | 91 |
| 3.1 | <u>Teste de exposição:</u> | 93 |
| 3.2 | <u>Dependência de serviços</u> | 93 |
| 3.3 | <u>Composição e fluxo de serviços</u> | 93 |
| 3.4 | <u>Requisitos não funcionais</u> | 94 |
| 3.5 | <u>Mensagem do serviço</u> | 94 |
| 4 | <u>Realização</u> | 94 |
| 4.1 | <u>Decisão de arquitetura</u> | 94 |
| 4.2 | <u>Alocar serviços em componentes</u> | 94 |
| 4.3 | <u>Decisão de gerenciamento de estado</u> | 95 |
| 5 | <u>Diagrama com os serviços</u> | 95 |
| 6 | <u>Diagrama de sequencia dos serviços</u> | 95 |
| 7 | <u>Diagrama de camada SOA</u> | 95 |

Modelagem do Sistema XXX

Objetivo do Documento

O objetivo do documento de Modelagem de Serviço é descrever o diagrama de classes necessárias para a construção do sistema.

Portfolio de serviço

Lista de serviço (Identificação)

| Categoria do serviço | Nome do serviço | Operações | Atributo de entrada | Atributo de saída | Exceção |
|----------------------|-----------------------------------|---------------------------------------|---------------------|-------------------|---------|
| Pagamento | Serviço relacionados ao pagamento | Funções necessárias para um pagamento | | | |

Tabela 1 - Lista categorizada de serviço

Categoria do serviço: É o nome dado a um agrupamento de serviços.

Nome do serviço: O nome do serviço pode ser associado às funções de negócio, objetivos de negócio, os ativos, tais como os sistemas existentes e a indentificação de um serviço já existente no negócio.

Operações: quais funções o serviço possui.

| Assinatura | Descrição |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| EquaçãoRetaSimplificada() | Constrói um objeto equação da reta simplificada com coeficientes zerados. |
| EquaçãoRetaSimplificada(float a, float b) | Constrói um objeto equação da reta simplificada com coeficientes definidos, conforme os parâmetros a e b. |

Atributo de entrada: informações necessárias para as funções.

| Tipo | Nome | Descrição |
|------|-------------|---------------------------------------------------------------------|
| int | _tentativas | Número de vezes que o usuário tentou acertar o resultado da equação |

Atributo de saída: informações geradas pelos pelas funções.

| Retorno | Assinatura | Descrição |
|---------|----------------|--------------------------------------------|
| double | sin (double a) | Retorna o seno trigonométrico do ângulo a. |
| | | |

Exceção: caso ocorra algum erro, como deve proceder.

| Candidato a serviço | Descrição | Status | Associação do serviço | | |
|---------------------|-----------|--------|-----------------------|------|---------|
| | | | Função/ processo | Meta | Sistema |
| | | | | | |
| | | | | | |
| | | | | | |

Tabela 2 - Portfolio de serviço

Candidato a serviço: Nome de um serviço candidato.

Descrição: uma descrição das responsabilidades do serviço.

Implements: Lista de interfaces que o serviço XXX implementa

Extends: Lista de interfaces que a serviço XXX estende

Status: pode ser se o serviço vai ser exposto ou não.

Função / processo: nome do processo de negócio que está associado.

Meta: A meta que o serviço está associado.

Sistema: Nome do sistema. Se o processo de negócio, cujo candidato a serviço está representado, é realizado por um sistema ou utiliza um sistema.

Especificação

Teste de exposição:

| Serviço | Vai ser exposto (s/n)? | Teste de exposição | | | | Obs. |
|---------|------------------------|-------------------------------|-----------------------------------|------------------------------------------|--------------------------|------|
| | | Atende à processo de negócio? | Atende à regra de uma composição? | Atende à regra de algum serviço externo? | É um serviço redundante? | |
| | | | | | | |

Serviço: Nome do serviço.

Vai ser exposto: verifica-se se o serviço exposto.

Atende ao processo de negócio: Verificar - se as metas do processo de negócio serão atendidas, ou seja, o serviço está alinhado ao processo de negócio.

Atende a uma regra de uma composição: Verificar - se vai pertencer a alguma composição de serviço, e se pertence tem que atender às regras da composição.

Atende à regra de algum serviço externo: Verificar - se vai atender à regra de algum serviço externo.

É um serviço redundante: Verifica se a responsabilidade do serviço não esta sendo repetida.

Obs: Observação que mereça ser comentada.

Dependência de serviços

Dependência funcional:

Pode ser representado por um diagrama de componentes da UML.

É quando um serviço é formado por uma composição de outros serviços.

Dependência do processo de negócio:

Pode ser representado por um diagrama de diagrama de sequência da UML.

É quando um serviço é utilizado no representado contexto de um processo de negócio, ou seja, dever possuir:

Pré-condição: A chamada de outro serviço deve ser executada com sucesso para que este serviço seja executado.

Dependência de processamento: O processamento de outro serviço deve ser concluído com sucesso para ser concretizar o processamento do serviço atual.

Pós-condição: Quando após o processamento do serviço é necessário chamar outro serviço.

Composição e fluxo de serviços

É onde é determinado a ordem de execução de uma composição de serviços que não pode ser interrompida, e também classificamos a composição em relação ao seu tempo de duração de sua execução em curta ou longa.

| Composição de serviços | Ordem de execução dos serviços | Tipo (longa/curta) |
|-------------------------------|---------------------------------------|---------------------------|
| | | |

Requisitos não funcionais

| Serviço | Requisito não funcional |
|----------------|--------------------------------|
| | |

Mensagem do serviço

| Serviço | Mensagem de entrada | Mensagem de saída | Obs. |
|----------------|----------------------------|--------------------------|-------------|
| | | | |

Realização

Decisão de arquitetura

| Pontos de decisão | Decisão | Serviço |
|-----------------------------------------------------|----------------|----------------|
| Como o serviço será exposto? | | |
| Como é formada a estrutura da mensagem? | | |
| Como e onde a transformação da mensagem será feita? | | |
| Outras questões. | | |

Alocar serviços em componentes

| Serviço | Componente | Obs. |
|----------------|-------------------|-------------|
| | | |

Decisão de gerenciamento de estado

| Nome da composição de serviço | Estado do exigido pela composição | Local onde o estado da composição está sendo realizado. |
|--------------------------------------|------------------------------------------|----------------------------------------------------------------|
| | | |

Diagrama com os serviços

Esta seção deverá conter o diagrama de serviços

Diagrama de sequência dos serviços

Diagrama de camada SOA