



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

José Paulo Telles Pires de Faria

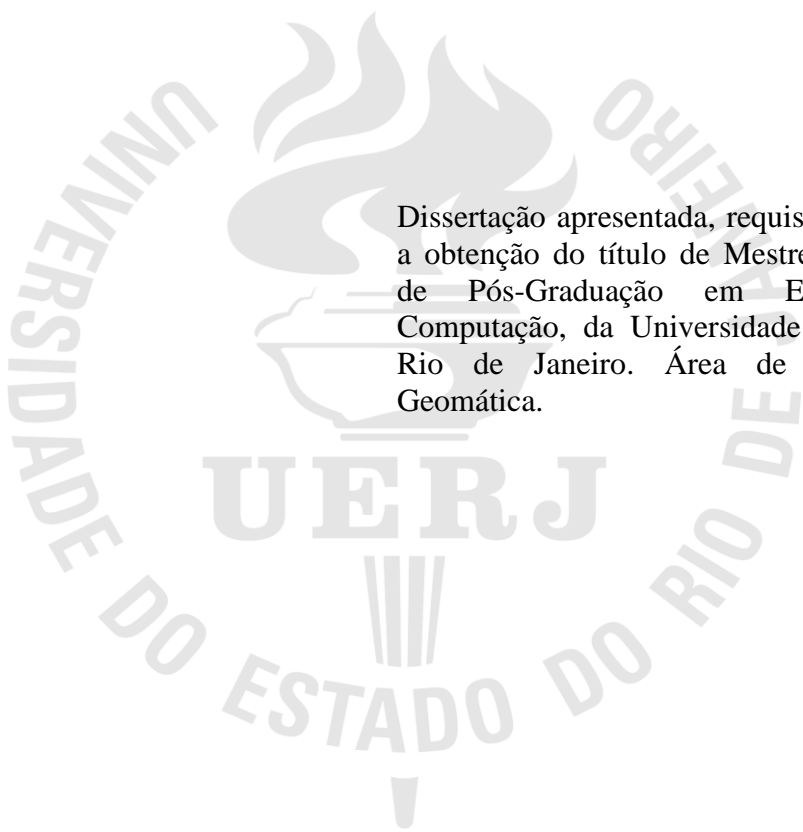
**O ZEP Framework: um *framework* para a criação de métodos para
envolvimento de *stakeholders* no desenvolvimento de soluções de software**

Rio de Janeiro

2011

José Paulo Telles Pires de Faria

O ZEP Framework: um *framework* para a criação de métodos para envolvimento de *stakeholders* no desenvolvimento de soluções de software



Dissertação apresentada, requisito parcial para a obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de Concentração: Geomática.

Orientador: Prof. Dr. Orlando Bernardo Filho

Coorientadora: Prof.^a Dr.^a Vera Maria Benjamin Werneck

Rio de Janeiro

2011

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

F224 Faria, José Paulo Telles Pires de.
O ZEP Framework: um *framework* para a criação de métodos para envolvimento de *stakeholders* no desenvolvimento de soluções de software. / José Paulo Telles Pires de Faria – 2011.
149 f.

Orientador: Orlando Bernardo Filho.
Coorientadora: Vera Maria Benjamin Werneck.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia de Computação. 2. Engenharia de software – Teses. I. Bernardo Filho, Orlando. II. Werneck, Vera Maria Benjamin. III. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. IV. Título.

CDU 004.41

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

Assinatura

Data

José Paulo Telles Pires de Faria

**O ZEP Framework: um *framework* para a criação de métodos para envolvimento
de *stakeholders* no desenvolvimento de soluções de software**

Dissertação apresentada, requisito parcial para a obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de Concentração: Geomática.

Aprovado em: 04 de outubro de 2011.

Banca Examinadora:

Prof. Dr. Orlando Bernardo Filho (Orientador)
Faculdade de Engenharia – UERJ

Prof.^a Dr.^a Vera Maria Benjamin Werneck (Coorientadora)
Faculdade de Engenharia – UERJ

Prof.^a Dr.^a Neide dos Santos
Faculdade de Engenharia – UERJ

Prof.^a Dr.^a Karin Koogan Breitman
Pontifícia Universidade Católica do RJ - PUC-Rio

Rio de Janeiro

2011

DEDICATÓRIA

Para minha família.

RESUMO

FARIA, José Paulo Telles Pires de. **O ZEP Framework: um *framework* para a criação de métodos para envolvimento de *stakeholders* no desenvolvimento de soluções de software.** 2011. 149f. Dissertação (Mestrado em Engenharia de Computação) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro. Rio de Janeiro, 2011.

A especificação dos requisitos de *software* pressupõe que se conheçam os requisitos do sistema do que será parte. Os requisitos do sistema, por sua vez, pressupõem o conhecimento do negócio (*business*) onde o sistema será utilizado. Para que estes conhecimentos sejam obtidos é importante o envolvimento dos *stakeholders* tanto no nível de sistema quanto no nível de negócio. As literaturas sobre Engenharia de Requisitos, Engenharia de Software e Engenharia de Sistemas concordam que o envolvimento dos *stakeholders* é fundamental. O tratamento dispensado ao assunto, no entanto, é pequeno, dada a importância do tema. Esta dissertação, utilizando conceitos da Engenharia de Métodos Situacionais e de *Design Science*, apresenta o ZEP Framework, um artefato, produzido com o software EPF Composer, que permite a criação de métodos para envolver o *stakeholder*. Estes métodos, para serem criados, devem levar em consideração as peculiaridades da organização, dos recursos disponíveis e do projeto em si. São apresentados, ainda, alguns cenários, na área de Turismo, como exemplos da utilização do *framework*.

Palavras-chave: Engenharia de software; Engenharia de métodos; Engenharia de requisitos; Turismo.

ABSTRACT

The specification of software requirements implies that the system requirements are known. The system requirement requires knowledge about the business of which the system will be a part. To obtain this knowledge it is important to engage stakeholders not only at the system level, but at the business level as well. The literature on Requirements Engineering, Software Engineering and Systems Engineering consider the involvement of stakeholders as fundamental. Despite this, the treatment of the subject is not proportional to its importance. This dissertation, using concepts from Situational Method Engineering and Design Science, presents the ZEP Framework, an artifact built with the EPF Composer software, that allows the creation of methods to engage stakeholders. These methods consider the specific situation of the organization, the resources available and the characteristics of the initiative itself. Some scenarios, in the Tourism field, are presented as examples of the utilization of the framework.

Keywords: Software engineering; Method engineering; Method engineering; Requirements engineering; Design Science; Tourism.

LISTA DE FIGURAS

Figura 1 - Relacionamento dos Componentes de uma Teoria de <i>Design</i>	15
Figura 2 - Contexto do artefato de TI	16
Figura 3 - O artefato de TI e sua rede nomológica	17
Figura 4 - O ciclo Gerar/Testar	18
Figura 5 - Abordagens para a Engenharia de Métodos	23
Figura 6 - Métodos e processos.....	25
Figura 7 - O conteúdo principal do repositório do MFESA	27
Figura 8 - Processo de Criação de Métodos.....	28
Figura 9 - O modelo S3.....	32
Figura 10 - Utilização do SPEM	34
Figura 11 - Terminologia mapeada para Conteúdo de Métodos e Processo.....	35
Figura 12 - Camadas do modelo para UML e SPEM 2.0	36
Figura 13 - Exemplo de instanciação das camadas.....	37
Figura 14 - Estrutura do Meta-Modelo SPEM.....	38
Figura 15 - Os estereótipos do Perfil UML 2 do SPEM 2.0 definidos no pacote <i>Core</i>	41
Figura 16 - <i>Work Definition</i> e seus elementos relacionados.....	41
Figura 17 - As partes do <i>Describable Element</i> e suas subclasses.....	42
Figura 18 - Taxonomia dos <i>Describable Elements</i> centrais	43
Figura 19 - Principais <i>Method Content Elements</i> e seus relacionamentos	44
Figura 20 - Estereótipos de <i>Work Product Kind</i>	45
Figura 21 - Taxonomia de <i>Work Product</i>	45
Figura 22 - O <i>Work Product Relationship Kinds</i>	46
Figura 23 - Visão geral das principais classes e associações do pacote <i>Process Structure</i> ..	47
Figura 24 - Uma estrutura analítica é definida por <i>Activities</i> que englobam <i>Breakdown Elements</i>	48
Figura 25 - Estereótipos de tipos de atividades.....	48
Figura 26 - A <i>Work Definition</i> de Atividade e seus relacionamentos com <i>Performers</i> e Entradas/Saídas	50
Figura 27 - <i>Work Definition</i> e seus elementos relacionados.....	51
Figura 28 - Os estereótipos do SPEM 2.0 UML 2 Profile definidos no pacote <i>Process</i>	

	<i>with Methods</i>	52
Figura 29 -	Taxonomia das classes do meta-modelo definidas no pacote <i>Process with Methods</i>	53
Figura 30 -	Taxonomia e principais relacionamentos de <i>Breakdown</i>	54
Figura 31 -	Exemplos de <i>Method Content Use</i> (direita) referenciando <i>Method Content</i> (esquerda).	55
Figura 32 -	Uma <i>Method Library</i> é um <i>container</i> para <i>Plugins</i> e <i>Configurations</i>	56
Figura 33 -	<i>Method Plugins</i> são <i>containers</i> para pacotes de métodos	57
Figura 34 -	Definição de <i>Method Library</i> e <i>Method Configurations</i>	58
Figura 35 -	EPF Composer: Representação de conteúdo do método	59
Figura 36 -	EPF Composer: Representação do processo de autoria	62
Figura 37 -	Visão da firma por <i>stakeholders</i>	63
Figura 38 -	Oito degraus na escada de participação dos cidadãos	65
Figura 39 -	Uma escada de gerenciamento e envolvimento de <i>stakeholders</i>	66
Figura 40 -	Fontes de poder	68
Figura 41 -	Indicadores de poder	68
Figura 42 -	Urgência como função de criticalidade e sensibilidade ao tempo nos negócios em geral	70
Figura 43 -	Diagrama de cebola de <i>stakeholders</i>	71
Figura 44 -	Modelo de 'cebola' dos <i>stakeholders</i> em um sistema típico	72
Figura 45 -	Tipologia de diagnóstico de <i>stakeholders</i> organizacionais	73
Figura 46 -	Matriz de <i>stakeholders</i>	73
Figura 47 -	Tipologia de <i>stakeholders</i> : Um, dois e três atributos presentes	74
Figura 48 -	Categorias de <i>stakeholders</i>	76
Figura 49 -	Ciclo de desenvolvimento da arquitetura	78
Figura 50 -	Níveis de modelagem e o ZEP Framework	81
Figura 51 -	Exemplo de projeto com uma única fase	82
Figura 52 -	Representação de uma Fase no ZEP Framework, baseado no PMBOK	82
Figura 53 -	Exemplo de projeto com três fases	83
Figura 54 -	Exemplo de projeto com fases sobrepostas	83
Figura 55 -	Exemplo de especificação de relacionamento entre Fases no ZEP Framework usando o EPF Composer	84
Figura 56 -	Grupos de Processos no Gerenciamento de Projetos	84

Figura 57 - Representação do Grupo de Processos de Iniciação no EPF Composer, segundo o PMBOK.	85
Figura 58 - Representação do Grupo de Processos de Planejamento no EPF Composer, segundo o PMBOK	85
Figura 59 - Representação do Grupo de Processos de Execução no EPF Composer, segundo o PMBOK	86
Figura 60 - Representação do Grupo de Processos de Monitoração e Controle no EPF Composer, segundo o PMBOK.....	87
Figura 61 - Representação do Grupo de Processos de Finalização no EPF Composer, segundo o PMBOK	87
Figura 62 - A chave para o planejamento de envolvimento	88
Figura 63 - Implementação do Engagement Workbook	89
Figura 64 - Os cinco estágios (<i>stages</i>) do Envolvimento de <i>Stakeholders</i>	90
Figura 65 - Os cinco estágios (<i>stages</i>) do Manual de Envolvimento de <i>Stakeholders</i>	91
Figura 66 - Fases em um processo ideal de análise e decisão sobre uma ação propostas	91
Figura 67 - Representação de método de Babiuch et al. no EPF Composer.....	92
Figura 68 - Principais Elementos de Conteúdo de Método e seus relacionamentos.....	94
Figura 69 - Estrutura Analítica de Processos para o Grupo de Processos de Iniciação	104
Figura 70 - Diagrama de Atividades para Criação de TAP (Termo de Abertura de Projeto)	105
Figura 71 - Padrão de Recurso para o nível corporativo.....	106
Figura 72 - Padrão de Recurso para atividades de planejamento para o nível corporativo	107
Figura 73 - Atividade de Planejamento para o nível de negócio	108
Figura 74 - Atividade de Planejamento para o nível de projeto.....	109
Figura 75 - Método para o Projeto de Regionalização do Turismo (resumido)	109
Figura 76 - Método para o Projeto de Regionalização do Turismo (expandido).....	110
Figura 77 - Método para o Projeto de Regionalização do Turismo (município)	111
Figura 78 - Método para o Projeto de Qualificação Profissional para Turismo	111

LISTA DE TABELAS

Tabela 1 - Componentes de uma Teoria de <i>Design</i>	15
Tabela 2 - Métodos de avaliação	18
Tabela 3 - Orientações da pesquisa em <i>Design Science</i>	20
Tabela 4 - Framework de March e Smith.....	21
Tabela 5 - Critérios propostos por Firesmith (2009), Jiang (2005) e Harmsen (2001)	30
Tabela 6 - Indicadores de Harmsen	32
Tabela 7 - Exemplos de Níveis de Conformidade no SPEM	39
Tabela 8 - Comparação estrutural dos metamodelos.....	61
Tabela 9 - Componentes do EPF Composer, PMBok e BABoK	62
Tabela 10- Níveis de Impacto Público do IAP2.....	67
Tabela 11- Exemplo de classificação preliminar de <i>stakeholders</i>	76
Tabela 12- Matriz de poder dos <i>stakeholders</i>	77
Tabela 13- Matriz de Papéis e Tarefas no BABOK	79
Tabela 14- Planejamento de envolvimento	93
Tabela 15- Papéis em Análise de Negócios	96
Tabela 16- Atributos de situação	100
Tabela 17- Componentes do EPF Composer, PMBok e BABoK	101

SUMÁRIO

	INTRODUÇÃO	12
1	FUNDAMENTAÇÃO TEÓRICA	14
1.1	Design Science	14
1.1.1	<u>Design Science para Sistemas de Informação</u>	14
1.1.2	<u>Princípios de Design Science</u>	18
1.1.3	<u>Framework para Design Science</u>	20
1.1.4	<u>Considerações</u>	21
1.2	Engenharia de Métodos	20
1.2.1	<u>Objetivos da Engenharia de Métodos</u>	22
1.2.2	<u>Conceitos Básicos</u>	24
1.2.3	<u>Criação do método</u>	27
1.2.4	<u>SPEM (Software & Systems Process Engineering Meta-Model Specification)</u>	34
1.2.4.1	Arquitetura geral do Meta-modelo SPEM.....	37
1.2.4.2	O pacote Core	40
1.2.4.3	O pacote Managed Content	42
1.2.4.4	O pacote Method Content.....	42
1.2.4.5	O pacote Process Structure.....	46
1.2.4.6	O pacote Process Behavior.....	50
1.2.4.7	O pacote Process with Methods	51
1.2.4.8	O pacote Method Plug-in	55
1.2.4.9	Considerações finais.....	58
1.2.5	<u>EPF (Eclipse Process Framework) Composer</u>	58
1.2.6	<u>Atributos de um componente de método</u>	60
1.3	Stakeholders	62
1.3.1	<u>Definição</u>	63
1.3.2	<u>Níveis de envolvimento</u>	64
1.3.3	<u>Atributos dos stakeholders</u>	67
1.3.4	<u>Classificação dos Stakeholders</u>	71
1.3.5	<u>Arquitetura corporativa e TOGAF</u>	75

1.3.6	<u>Gerência de Projetos e Análise de Negócios</u>	79
2	O ZEP FRAMEWORK	81
2.1	Estrutura de Processos no ZEP Framework	81
2.2	Conteúdo de Métodos no ZEP Framework	93
2.2.1	<u>Work Products (Produtos de Trabalho) no ZEP Framework</u>	94
2.2.2	<u>Role (Papel) no ZEP Framework</u>	95
2.2.3	<u>Tasks (Tarefas) no ZEP Framework</u>	97
2.3	Processo de seleção de componentes	100
2.4	Considerações	100
3	CENÁRIOS DE APLICAÇÃO DO FRAMEWORK	102
3.1	Evento inicial do processo (comum a todos os cenários)	103
3.2	Cenário 1: Sistema para órgão estadual de Turismo	104
3.2.1	<u>Iniciação</u>	104
3.2.2	<u>Planejamento</u>	105
3.2.2.1	Nível corporativo.....	106
3.2.2.2	Nível de negócio (Turismo)	107
3.2.2.3	Nível de projeto (Regionalização do Turismo)	108
3.2.2.4	Criação do método.....	109
3.3	Cenário 2: Sistema para um órgão municipal de Turismo	110
3.4	Cenário 3: Segundo sistema para o órgão estadual de Turismo	111
3.5	Considerações	112
4	CONCLUSÃO	113
	REFERÊNCIAS	115
	GLOSSÁRIO	121
	APÊNDICE A – Exemplo de descrição dos componentes do framework.....	132
	APÊNDICE B – Técnicas de Envolvimento.....	146

INTRODUÇÃO

Esta dissertação se propõe a tratar de um problema que persiste na prática, apesar de muitos esforços para resolvê-lo: o problema dos requisitos de software incorretos. O argumento pode ser resumido da forma seguinte.

- Para determinar os requisitos do software a ser construído é necessário determinar como ele irá ajudar o sistema a atingir os seus objetivos.
- Para determinar os requisitos do sistema é necessário determinar o problema a ser resolvido ou a oportunidade a ser aproveitada.
- Para determinar o problema a ser resolvido, ou a oportunidade a ser aproveitada, é necessário conhecer os objetivos do negócio. Um problema é algo que atrapalha a realização de um objetivo do negócio. Uma oportunidade é uma situação que pode permitir atingir algum objetivo do negócio.
- Para conhecer os objetivos do negócio é preciso se informar junto aos envolvidos no contexto do negócio, ou seja, junto aos seus *stakeholders*.
- Portanto para determinar os requisitos do futuro software é preciso, inicialmente, identificar os *stakeholders* que estão envolvidos no problema para, então, descobrir os objetivos do negócio. Só neste momento será possível avaliar se os custos para desenvolver ou adquirir um sistema de informações podem ser justificados.

Este raciocínio, embora aparentemente simples, não é acompanhado, na prática, por ações que realmente criem um envolvimento com os *stakeholders*. Isto se deve a fatores como a falta de tempo dos desenvolvedores e dos outros envolvidos, o custo do levantamento de informações (por exemplo, viagens ou visitas) e a falta de preparo para um tipo de trabalho “social” (ao invés de “técnico”), Muitas vezes, embora a necessidade de realizar este trabalho seja reconhecida, falta um método claro sobre como realizá-lo (processo) e sobre o resultado a ser obtido (produto).

A solução é criar métodos que especifiquem, em detalhes, as atividades, produtos e papéis que as pessoas devem exercer para que o envolvimento se torne um fim que possa ser atingido com mais facilidade.

O objetivo da dissertação é contribuir para a solução deste problema através do ZEP Framework. Este *framework* é composto por componentes de métodos com a descrição de

processos, tarefas, papéis e produtos a serem criados. Estes componentes deverão ser combinados, levando em conta a situação presente, de forma a estabelecer como o envolvimento com os *stakeholders* deverá se realizar. Depois de criados, os métodos podem ser armazenados para uma eventual reutilização ou adaptação.

O primeiro passo é determinar quais os componentes a serem incluídos no *framework*. Para isso foi realizada uma pesquisa sobre *stakeholders*: definição, atributos, classificações e técnicas para envolvimento.

Em seguida foi preciso determinar como criar métodos para obter o envolvimento adequado a cada situação. A opção pela Engenharia de Métodos Situacionais foi considerada a mais conveniente por permitir a flexibilidade desejada e por contar com uma ferramenta *freeware*, o Eclipse Process Framework Composer (EPF Composer) para a implementação do ZEP Framework.

O capítulo 1 contém a fundamentação teórica para a dissertação. Na seção 1.1 é feita uma introdução aos princípios de *Design Science*, os quais servem para fundamentar a opção pela criação de artefatos como objetivo da pesquisa. Em seguida, na seção 1.2, é apresentada a fundamentação teórica sobre a Engenharia de Métodos Situacionais e, na seção 1.3, sobre *stakeholders*. O ZEP Framework é apresentado no capítulo 2. No capítulo 3 são desenvolvidos cenários para ilustrar a utilização do *framework* em situações típicas. O capítulo de conclusão destaca as contribuições e limitações da pesquisa e indica tópicos para pesquisas futuras.

1 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são examinadas: a literatura sobre *Design Science*, Engenharia de Métodos e *stakeholders*. Estes três assuntos são necessários para a criação do ZEP Framework.

1.1 *Design Science*

Nesta dissertação é adotado o modelo de pesquisa conhecido como *Design Science*. Na primeira parte deste capítulo é feita uma breve retrospectiva sobre *Design Science* e sua utilização para pesquisas na área de sistemas de informação. Em seguida são vistas as principais abordagens que ela utiliza e são mostrados os produtos e o processo de uma Teoria de *Design*.

1.1.1 *Design Science* para Sistemas de Informação

Walls, Widmeyer e El Sawy (Walls et al.,1992, p.41) levam os conceitos de *Design Science* para as pesquisas em sistemas de informação, argumentando sobre a necessidade da definição de teorias da própria área para que ela possa se estabelecer como ciência. Uma teoria de *design* deve ser uma teoria prescritiva, isto é, deve dizer como o processo de *design* deve ser realizado para que seja eficaz e viável, ao contrário das teorias das ciências naturais e físicas, que são explicativas e preditivas.

A característica principal de uma teoria de *design* é que ela trata de comportamentos voltados para atingir um objetivo. Nas ciências naturais não se pode, realmente, falar do objetivo de um elétron. Já nas ciências sociais isto é possível, porém elas apenas procuram explicar porque os objetivos existem e tentam prever os resultados destes objetivos.

A teoria de *design* procura ajudar a atingir objetivos e tem as seguintes características (Walls et al, 1992, p. 41): (i) as teorias de *design* devem tratar os objetivos como contingências. (ii) uma teoria de *design* nunca deve tratar puramente de explicações ou previsões, (iii) as teorias de *design* são prescritivas. (iv) as teorias de *design* são teorias compostas que englobam *kernel theories* das ciências naturais, sociais e matemáticas, (v) enquanto as teorias explicativas dizem “o que é”, as teorias preditivas dizem “o que será” e as teorias normativas dizem “o que deve ser”, as teorias de *design* dizem “como fazer e porque”,

(vi) as teorias de *design* mostram como as teorias explicativas, preditivas e normativas podem ser usadas na prática e, (vii) as teorias de *design* são teorias de racionalidade procedural.

A Tabela 1 mostra um resumo dos componentes de uma Teoria de *Design* para Sistemas de Informação em relação ao produto e ao processo e a Figura 1 os relacionamentos desses elementos.

Tabela 1 - Componentes de uma Teoria de *Design*

1 - Produto

1.1 – Meta-requisitos	Descreve as classes de objetivos aos quais a teoria se aplica
1.2 – Meta- <i>design</i>	Descreve a classe de artefatos que, por hipótese, atendem os meta-requisitos
1.3 – <i>Kernel theories</i>	Teorias das ciências naturais e sociais que regem os requisitos de <i>design</i>
1.4 – Hipóteses testáveis sobre os produtos do <i>design</i>	Usadas para testar se o meta- <i>design</i> satisfaz os meta-requisitos

2 - Processo

2.1 – Método de <i>design</i>	Descrição dos procedimentos para a construção do <i>design</i>
2.2 – <i>Kernel theories</i>	Teorias das ciências naturais e sociais que regem o próprio processo de <i>design</i>
2.3 – Hipóteses testáveis sobre o processo de <i>design</i>	Usadas para testar se o método de <i>design</i> resulta em um artefato que é consistente com o meta- <i>design</i> .

Fonte: WALLS et al., 1992, p. 43

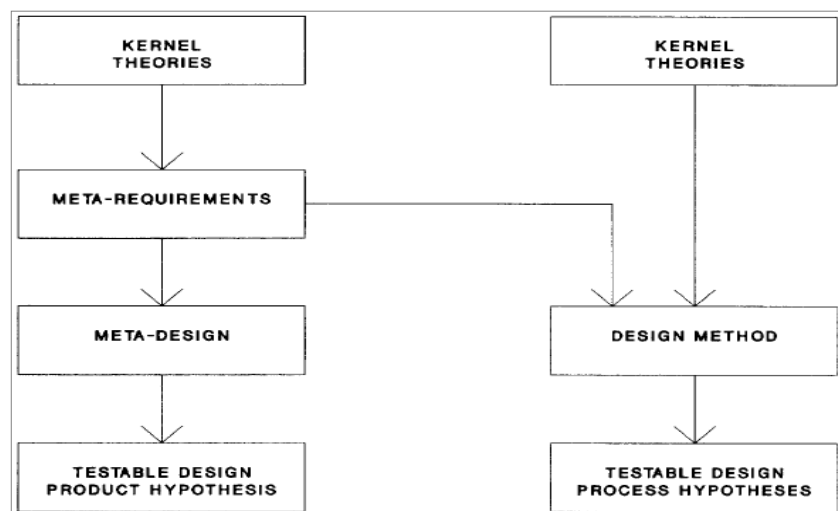


Figura 1 - Relacionamento dos Componentes de uma Teoria de *Design*.

Fonte WALLS et al., 1992

BENBASAT e ZMUD (2003) enfatizam a falta de um consenso sobre o escopo das pesquisas em sistemas de informação e definem uma proposta que se baseia em considerar o artefato de TI (Tecnologia da Informação) como sendo o objeto central das pesquisas, como mostrado na Figura 2.

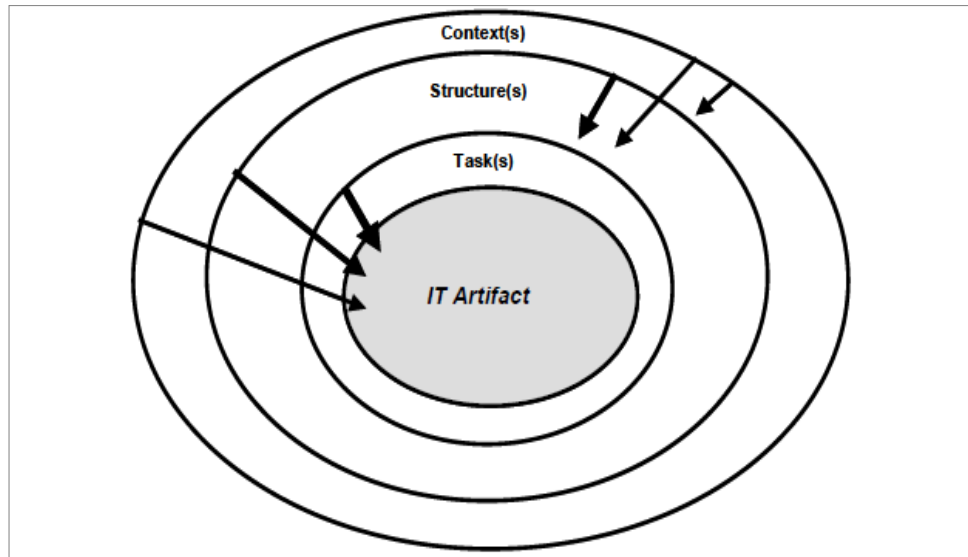


Figura 2 - Contexto do artefato de TI.

Fonte: BENBASAT; ZMUD; 2003, p. 187

O Artefato de TI é a aplicação da Tecnologia da Informação para “permitir ou apoiar alguma tarefa contida em uma estrutura [de tarefas (ou estruturas)] que, por sua vez, está contida em um contexto [(ou contextos)]” (op. Cit. p. 186). No entanto, a disciplina de Sistemas de Informação engloba outros elementos, além do Artefato de TI, em uma rede nomológica,, conforme mostra a Figura 3.

Uma rede nomológica estabelece relacionamentos entre: “(a) observable properties or quantities to each other; or (b) theoretical constructs to observables; or (c) different theoretical constructs to one another. These "laws" may be statistical or deterministic.” (CRONBACH et al., 1955).

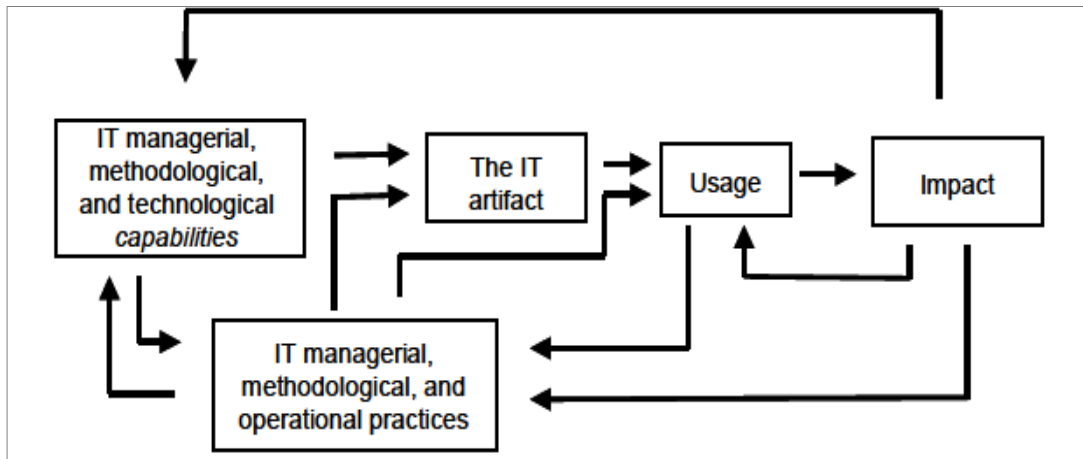


Figura 3 - O artefato de TI e sua rede nomológica

Fonte: BENBASAT; ZMUD; 2003, p. 187

Pesquisadores e profissionais de Sistemas de Informação buscam ampliar seus conhecimentos sobre como os artefatos de TI são concebidos, construídos e implementados, como os artefatos de TI são usados, suportados e melhorados e como os artefatos de TI impactam (e são impactados) pelos contextos nos quais estão contidos. (BENBASAT e ZMUD, 2003. Pg 186).

Ainda em relação à Figura 3 dois aspectos são vitais. “Em primeiro lugar, os construtos estão *intimamente* relacionados com o artefato de TI (...). Segundo, (...) [os elementos] são responsáveis por causalidade ativa e retroativa” (op. cit. p.186).

Benbasat and Zmud (2003) propõem, entre outras, algumas regras práticas para aumentar a diferenciação entre as pesquisas de Sistemas de Informação e as de outras áreas afins. Uma das regras é que o estudo nessa área investiga relações que estão dentro da rede nomológica (Figura 3). Outra regra é existência de *níveis de separação* entre os construtos de Sistemas de Informação e os principais construtos no modelo de pesquisa do estudo. Assim essas sugestões, permitem um progresso considerável em tornar mais clara a natureza da erudição sobre Sistemas de Informação e, portanto, da identidade da disciplina de Sistemas de Informação.

Walls, Widmeyer e El Sawy (Walls et al., 2004) fazem uma avaliação dos resultados da sua proposta de 1992 (Walls et al., 1992), sendo que nesse intervalo foram publicados 26 artigos que abordam a Teoria de *Design* para Sistemas de Informação. Esta análise foi motivada pela proposta de Hevner et al. (2004) que define alguns princípios descritos na seção seguinte.

1.1.2 Princípios de *Design Science*

Hevner et al., (2004) definem princípios básicos (Tabela 2) a serem seguidos nas pesquisas em *Design Science*. Um princípio importante em *Design Science* é que a resolução de problemas e o conhecimento são adquiridos durante o próprio processo da resolução do problema. O resultado da pesquisa em Sistemas de Informação com *Design Science* consiste num artefato criado intencionalmente para resolver um importante problema organizacional (HEVNER et al., 2004, pg 82).

A definição de artefato, por sua vez, inclui construtos, modelos e métodos aplicados ao desenvolvimento e uso de sistemas de informação. Esta é a primeira orientação definida por HEVNER et al. (2004).

A segunda orientação é a de que a pesquisa deve tratar de um problema relevante.

Em terceiro lugar, é preciso fazer uma avaliação do artefato criado, podendo ser realizada de várias formas.

A quarta orientação estabelece que é preciso que a pesquisa faça alguma contribuição.

Tabela 2 – Métodos de avaliação

Métodos de avaliação	
1 – Observacionais	Estudo de caso: Estudar a fundo o artefato em um ambiente real. Estudo de campo: Monitorar o uso do artefato em vários projetos.
2 – Analíticos	Análise estática: Examinar a estrutura do artefato em suas qualidades estáticas (por exemplo, sua complexidade) Análise de arquitetura: Estudar se o artefato se encaixa na arquitetura técnica de Sistemas de Informação Otimização: Demonstrar que o artefato possui qualidades ótimas ou fornece limites para um funcionamento ótimo. Análise dinâmica: Estudar o artefato em uso pelas suas qualidades dinâmicas (por exemplo, desempenho)
3 – Experimentais	Experimento controlado: Estudar o artefato em um ambientes de características de qualidade controladas (por exemplo, usabilidade) Simulação: Executar o artefato com dados artificiais.
4 – Testes	Teste funcional (<i>black-box</i>): Executar as <i>interfaces</i> do artefato para descobrir falhas ou identificar defeitos. Teste estrutural (<i>white-box</i>): Executar testes de cobertura com alguma métrica (por exemplo, caminhos de execução) em uma implementação artificial.

5 – Descritivos

Argumentação informada: Usar informações de uma base de conhecimento (por exemplo, pesquisa relevante) para construir uma argumentação convincente sobre a utilidade do artefato.

Cenários: Construir cenários detalhados em torno do artefato para demonstrar a sua utilidade

Fonte: HEVNER et al., 2004, p. 86

O rigor da pesquisa (a quinta orientação) pode ser comprovado pela utilização de fontes amplamente aceitas e da rigorosa e detalhada atribuição da origem dos elementos utilizados.

A sexta orientação é a de que o *design* é um processo de busca. A procura do melhor (ou ótimo) *design* é frequentemente um problema intratável para problemas de sistemas de informação realistas. As estratégias de busca procuram soluções boas e viáveis que possam ser implementadas no ambiente real. Assim procura soluções satisfatórias (*satisficing* e não *satisfying*), se usarmos o termo de Simon (SIMON, 1996). Este processo gera um ciclo de geração de novos *designs* que são testados e melhorados repetidamente, como pode ser visto na Figura 4.

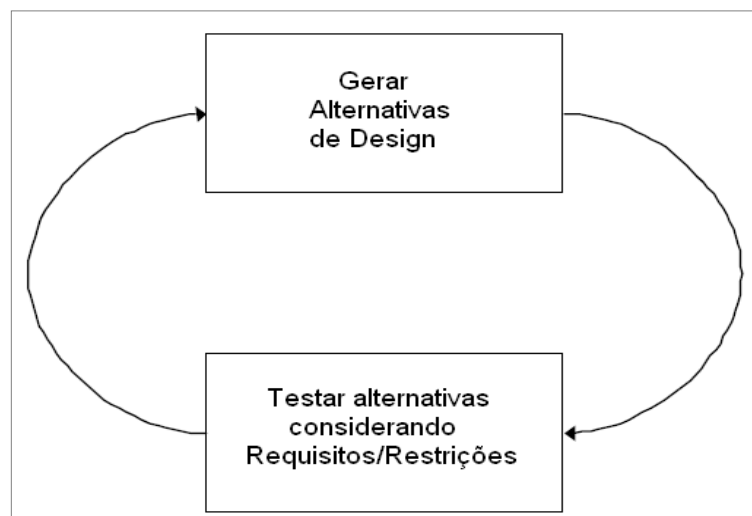


Figura 4 – O ciclo Gerar/Testar.

Fonte: HEVNER et al., 2004, p. 89.

Finalmente, a pesquisa deve ser comunicada de forma que possa ser compreendida por audiências técnicas e administrativas. A plateia técnica deve ter detalhes técnicos suficientes para permitir que o artefato seja compreendido e utilizado. A plateia administrativa deve receber informações suficientes para decidir pela utilização do artefato em sua organização.

Neste ponto a dissertação procurou usar tanto a terminologia técnica quanto a terminologia gerencial para que o artefato possa ser compreendido.

Tabela 3 – Orientações da pesquisa em *Design Science*

Orientações da pesquisa com <i>Design Science</i>	
Orientação	Descrição
1 – <i>Design</i> como Artefato	A pesquisa em <i>Design Science</i> deve produzir um artefato viável na forma de um construto, modelo, método ou instanciação.
2 – Problema relevante	O objetivo da pesquisa em <i>Design Science</i> é desenvolver soluções baseadas em tecnologia para problemas importantes e relevantes para o negócio.
3 – Avaliação do <i>design</i>	A utilidade, qualidade e eficácia de um artefato deve ser rigorosamente demonstrada através de métodos de avaliação bem executados.
4 – Contribuições da pesquisa	Uma pesquisa efetiva em <i>Design Science</i> deve contribuições claras e verificáveis nas áreas de <i>design</i> de artefatos, <i>design</i> de fundamentos e/ou metodologias de <i>design</i>
5 – Rigor na pesquisa	A pesquisa em <i>Design Science</i> depende da aplicação de métodos rigorosos tanto na construção quanto na avaliação do artefato projetado.
6 – <i>Design</i> como processo de procura	A procura por um artefato eficaz exige a utilização dos meios disponíveis para alcançar os resultados desejados e, ao mesmo tempo, a satisfação das leis no ambiente do problema.
7 – Comunicação da pesquisa	A pesquisa em <i>Design Science</i> deve ser apresentada de forma eficaz tanto para plateias orientadas a tecnologia quanto para as orientadas para administração.

Fonte: HEVNER et al., 2004, p. ? 86.

1.1.3 Framework para *Design Science*

Um *framework* para *Design Science* deve definir a priori os produtos e atividades da pesquisa. March e Smith (March et al., 1995) combinam as dimensões de produtos e de atividades (Tabela 4).

Os seguintes produtos de pesquisa para uma Teoria de *Design* foram definidos (March e Smith, 1995): construto, modelo, método e instância. Os **construtos**, ou conceitos, formam o vocabulário de um domínio e podem ter vários níveis de formalismo. Um **modelo** é um conjunto de proposições ou declarações que expressam relacionamentos entre os construtos.

Um **método** é um conjunto de etapas (ou algoritmo ou orientação) usado para realizar uma tarefa. Uma **instância** é a realização de um artefato em seu ambiente

As atividades de pesquisa em *Design Science* são construir e avaliar. A atividade de **Construir** está relacionada a construção de artefatos, demonstrando que um artefato deste tipo *pode* ser construído. **Avaliar** significa desenvolver critérios e avaliar o desempenho do artefato em relação a estes critérios (MARCH E SMITH, 1995. pg. 258)

É possível aplicar os métodos das ciências naturais aos artefatos da *Design Science*. As **teorizações** “explicam as características do artefato e suas interações com o ambiente que levam ao desempenho observado” (MARCH E SMITH, 1995. pg. 259) Na **comprovação** são apresentadas evidências para atestar a teoria.

Tabela 4 – Framework de March e Smith

		Atividades de Pesquisa			
		Construção	Avaliação	Teorização	Comprovação
Produtos da pesquisa	Construtos				
	Modelos				
	Métodos				
	Instâncias				

Esta dissertação está estruturada, seguindo o padrão proposto por March e Smith (1995), pois permite posicionar os artefatos produzidos pela pesquisa em relação ao tipo de produto e à atividade realizada na pesquisa para produzi-lo.

1.1.4 Considerações

Nesta seção vimos como a abordagem de *Design Science* enfatiza a criação de artefatos para Sistemas de Informação e quais são eles.

1.2 **Engenharia de Métodos**

Para que o modelo seja útil é necessário criar um método para a sua aplicação. No entanto, como o método deve ser adequado para cada situação é necessário criar um *framework* para permitir a criação de métodos.

Uma disciplina importante para o desenvolvimento de sistemas de informação é o controle do processo de desenvolvimento. Para isto existe a necessidade de métodos que sejam gerenciáveis pois tempo, orçamento, pessoas e outros recursos precisam ser controlados através de monitoramento e avaliação de resultados. Isto leva à ideia de reutilizar o mesmo método em todos os projetos. Esta reutilização traz várias vantagens como (i) menor custo, pois o método está pronto para ser utilizado, (ii) mais rapidez, pois o método pode ser utilizado imediatamente e (iii) gerenciabilidade, pois a sua estrutura (*WBS: Work Breakdown Structure*) já é conhecida.

Infelizmente, a reutilização do mesmo método tem se mostrado inadequada na maioria das situações. Isto ocorre porque cada projeto tem características próprias. Esta especificidade implica em um método diferente para cada projeto. Porém a flexibilidade obtida com a criação de novos métodos, a partir das características de cada projeto, tem muitas desvantagens como (i) alto custo, pois sempre haverá uma etapa para a criação de um novo método, (ii) pouca rapidez, pois a criação do método consome tempo e (iii) pouca gerenciabilidade, pois a comparação dos projetos será mais difícil.

Para enfrentar esta situação é preciso encontrar uma maneira de se obter, ao mesmo tempo, reutilização e flexibilidade.

1.2.1 Objetivos da Engenharia de Métodos

O principal objetivo da Engenharia de Métodos é encontrar um ponto de equilíbrio entre os objetivos de flexibilidade e reutilização.

De acordo com Harmsen (Harmsen, 1997), a flexibilidade pode se apresentar em vários níveis. Em um extremo, teríamos (i) os métodos rígidos, sem nenhuma flexibilidade. Aumentando gradualmente a flexibilidade teríamos as opções de (ii) selecionar entre vários métodos rígidos, (iii) a seleção de um caminho, entre vários caminhos predefinidos, (iv) a seleção e ajustes a partir de uma estrutura geral (*outline*) de método até chegarmos ao máximo de flexibilidade que é (v) a construção do método a partir de componentes.

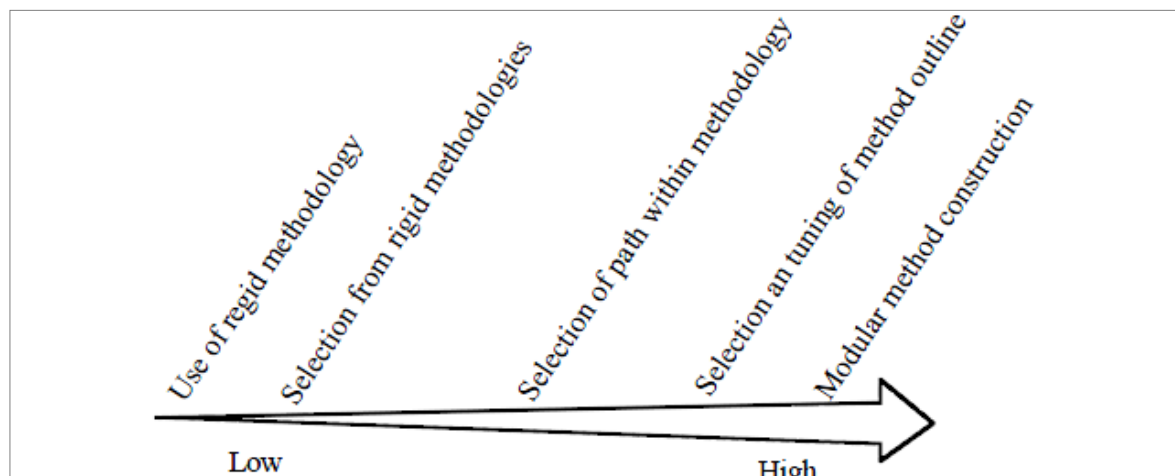


Figura 5 - Abordagens para a Engenharia de Métodos. Fonte: ROLLAND, 1997.

Usar um método rígido significa seguir um método sem que haja nenhum desvio em relação aos produtos e atividades previstas. O conjunto de fases e produtos deve ser executado sempre da mesma forma.

Outra abordagem é a seleção de um caminho dentro de métodos que oferecessem uma flexibilidade maior conforme apresentado na Figura 5. Por exemplo, depois da fase de requisitos poderíamos optar entre métodos orientados a objeto ou não. Poderíamos escolher entre um método para desenvolver aplicações *desktop* ou aplicações Web. Dependendo da aplicação, um protótipo pode, ou não, ser desenvolvido. No entanto, todas as opções estão predefinidas.

Na opção “*selection and tuning of a method outline*” da Figura 5, a estrutura geral está predefinida e o engenheiro de métodos escolhe as técnicas que melhor se ajustam a cada tipo de aplicação.

Finalmente, na construção modular de métodos, um novo método é criado especificamente para a situação atual. O Engenheiro de Métodos cria um novo método, levando em consideração as características da empresa e as necessidades da aplicação que será desenvolvida.

A solução básica para obter reutilização com flexibilidade é criar componentes de métodos seguindo um padrão (*framework*). Como todos os componentes seguem o mesmo padrão, estes componentes poderão ser combinados da forma mais adequada para cada situação. Desta forma teremos a reutilização (dos componentes) e a flexibilidade (pela adequação do método montado a partir dos componentes) nos novos métodos.

1.2.2 Conceitos Básicos

A definição de alguns conceitos gerais como métodos, metodologia e processos é importante para área de Engenharia de Métodos.

A palavra “método” tem origem grega. É composta pela palavra “μετά” (meta), que significa “além” ou “o que estamos procurando” (isto é, um objetivo ou meta) e “ὁ δόξ” (odos) que significa “caminho” ou “estrada”. Portanto “temos um método quando seguimos um certo 'caminho' para atingir um certo fim, proposto previamente” (FERRATER-MORA, 1964, pg. 196)

No contexto do desenvolvimento de sistemas de informação, Brinkkemper (1996, pg. 275-6) define um método como uma abordagem para realizar um projeto de desenvolvimento de sistemas, com base em uma forma específica de pensar.

A palavra “metodologia” é composta pela palavra “método” (“μετό δόξ” em Grego) e “λόγος” (logos) que significa, entre outras coisas, “discurso”, mas é geralmente empregada com o sentido moderno de “estudo de”, como, por exemplo em “geologia”, o estudo da Terra.

Apesar das origens etimológicas, a maioria das pessoas utiliza “método” e “metodologia” como sinônimos. Nesta dissertação, seguiremos a etimologia e empregaremos as palavras com seu sentido próprio.

Outro termo que deve ser definido é “processo”. Esta palavra é empregada com muitos significados diferentes. Para o IEEE (IEEE; 1990, p.57) um processo é: “(i) Uma sequência de etapas realizada com um propósito; por exemplo, o processo de desenvolvimento de software. (ii) Uma unidade de execução gerenciada pelo escalador (*scheduler*) de um sistema operacional. Ver também: tarefa (*task*), trabalho (*job*). (iii) Realização de operações sobre dados.”

Firesmith et al. (FIRESMITH et al, 2009) estabelecem a diferença entre “método” e “processo” da seguinte maneira:

Note that there is no universal consensus on the meaning of the terms *method* and *process*. Some authors consider these terms to be synonyms. Others consider processes to be large-scale activities and methods to be smallscale ways of implementing the processes. However, from a method engineering standpoint, it is very important to clearly differentiate the documented *as-intended* approach (the *method* documented in plans, procedures, and standards) from the *as-performed* approach (the *process* actually used by people on the endeavor). (FIRESMITH et al., 2009, p. 53)

Além desta distinção, eles chamam atenção para os níveis de abstração a serem levados em consideração ao tratar de métodos e processos. A Figura 6 mostra os três níveis considerados por Firesmith et al. e a relação entre métodos e processos.

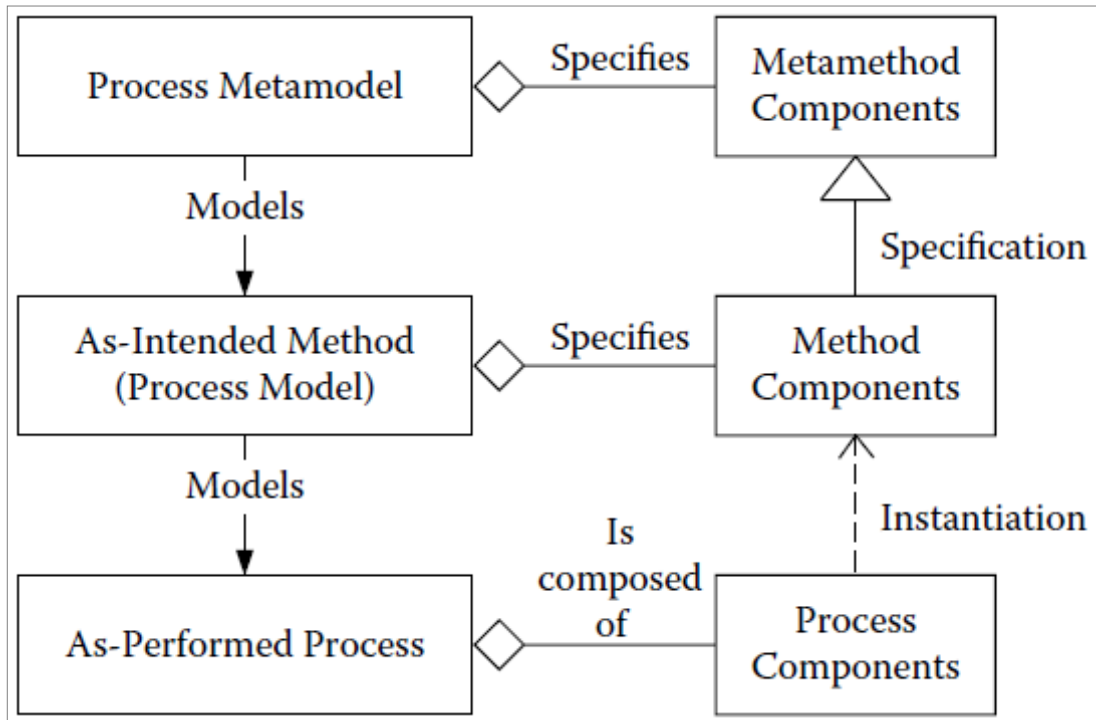


Figura 6 – Métodos e processos.

Fonte: FIRESMITH et al., 2009, p. 54.

A apresentação, a seguir, segue o conteúdo de Firesmith et al nas páginas 53 e seguintes..

No nível inferior (à esquerda) da Figura 6 estão os processos tal como existem no mundo real. Um processo, como existe em um projeto concreto, pode ser considerado como composto de Componentes de Processo (na Figura 6, nível inferior, à direita) como, por exemplo, uma analista de negócios chamada Alice, executando uma tarefa (rever documento) para criar um produto (versão 2.3 do documento).

O nível intermediário é composto de métodos (*As-Intended Method*, do lado esquerdo da Figura 6). Estes métodos podem ser considerados, levando-se em conta a sua relação com o nível inferior, como sendo Modelos de Processos (*Process Model*). Os métodos descrevem a maneira como o método deve ser executado (*As-Intended*) e não a maneira como serão

realmente utilizados (*As-Performed*).. De forma semelhante aos processos, os métodos podem ser considerado como sendo constituídos por Componentes de Método (*Method Component*, no nível intermediário, lado direito, da Figura 6). Os Componentes de Método especificam como os Componentes de Processo (*Process Components*, nível inferior, lado esquerdo, da Figura 6) devem ser executados. Por exemplo, um Componente de Método do tipo Termo de Abertura de Projeto, pode ocorrer em vários projetos, sendo instanciado de forma diferente em cada um.

O nível mais alto contem metamodelos. O Metamodelo de Processo (*Process Metamodel*, no nível superior, lado esquerdo, da Figura 6) é um modelo para especificar como devem ser os *Process Models* (no nível intermediário). O Metamodelo de Processo é especificado por Componentes de Metamodelo (nível superior, lado direito, da Figura 6). No MFESA (*Method Framework for Engineering System Architectures*) de Firesmith et al., os Componentes de Metamodelo são *Architecture Worker* (quem realiza o trabalho), *Architecture Work Unit* (o trabalho a ser realizado) e *Architecture Work Product* (o produto criado pelo trabalho realizado).

Esta distinção entre método e processo permite diferenciar dois tipos (entre outros) de usuários do ZEP Framework: Engenheiros de Métodos e Engenheiros de Processo. Os primeiros seriam responsáveis pela criação de Componentes de Métodos como, por exemplo, como realizar uma entrevista. Os segundos seriam responsáveis pela criação de processos como, por exemplo, um processo para obter informações sobre *stakeholders* que incluisse, nesta ordem, pesquisa documental, entrevista com o solicitante do serviço e uma oficina para validar os resultados obtidos na pesquisa e nas entrevistas.

Os Componentes de Método no MFESA são apresentados na Figura 7. Deve-se notar que o MFESA foi criado para a criação de arquiteturas, porém os nomes na Figura 7 podem ser adaptados a outras situações que não a criação de arquiteturas..

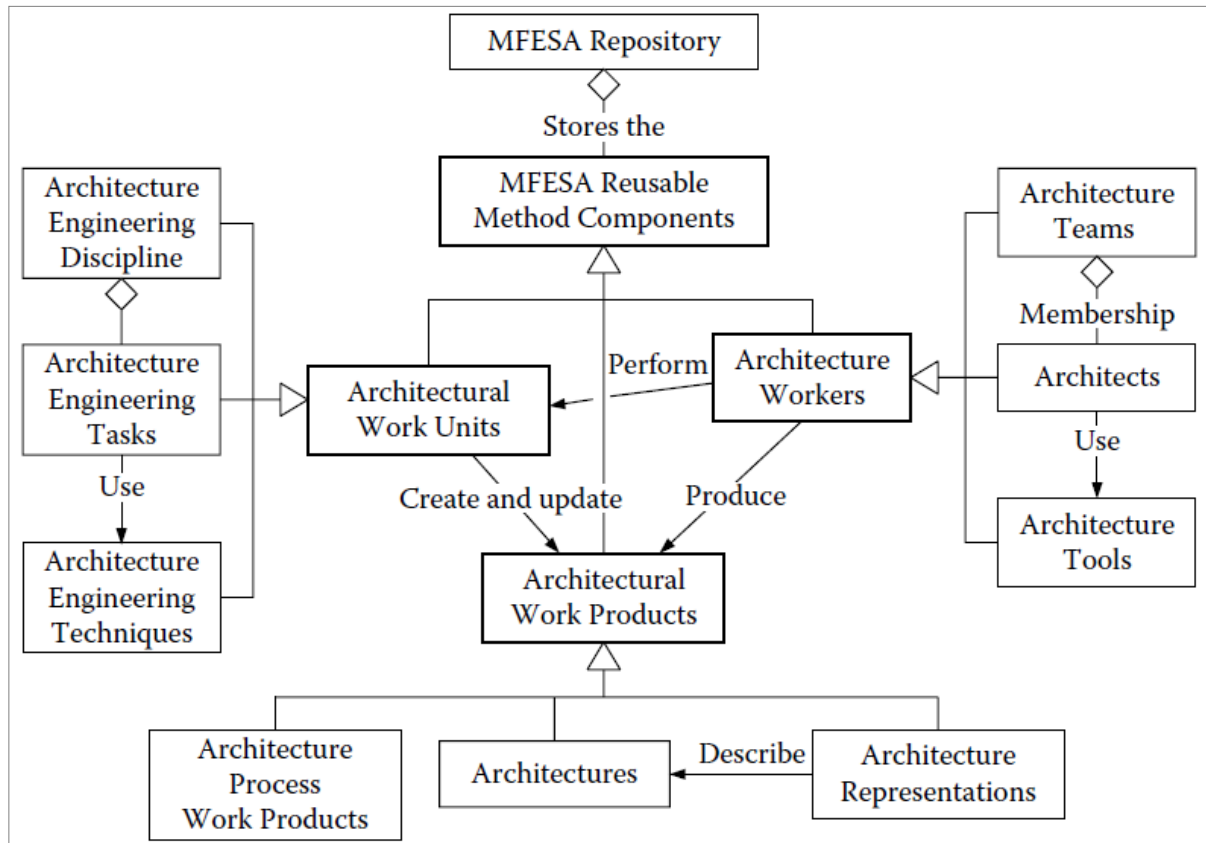


Figura 7 – O conteúdo principal do repositório do MFESA.

Fonte: FIRESMITH et al., 2009, p. 57.

Em um projeto de Análise de Negócio, por exemplo, os *Architecture Workers* seriam os envolvidos em Análise de Negócio podendo ser Analistas que são membros de uma equipe de Análise de Negócio e usam ferramentas de Análise de Negócio.

1.2.3 Criação do método

O MFESA (*Method Framework for Engineering System Architectures*) (Firesmith; 2009) é, como indica o nome, um *framework* para métodos de criação de arquiteturas de sistemas. No entanto, Firesmith (2009) também apresenta um método para a construção de métodos que é baseado nas ideias da Engenharia de Métodos Situacional (*Situational Method Engineering*). A Figura 8 representa graficamente este meta-método

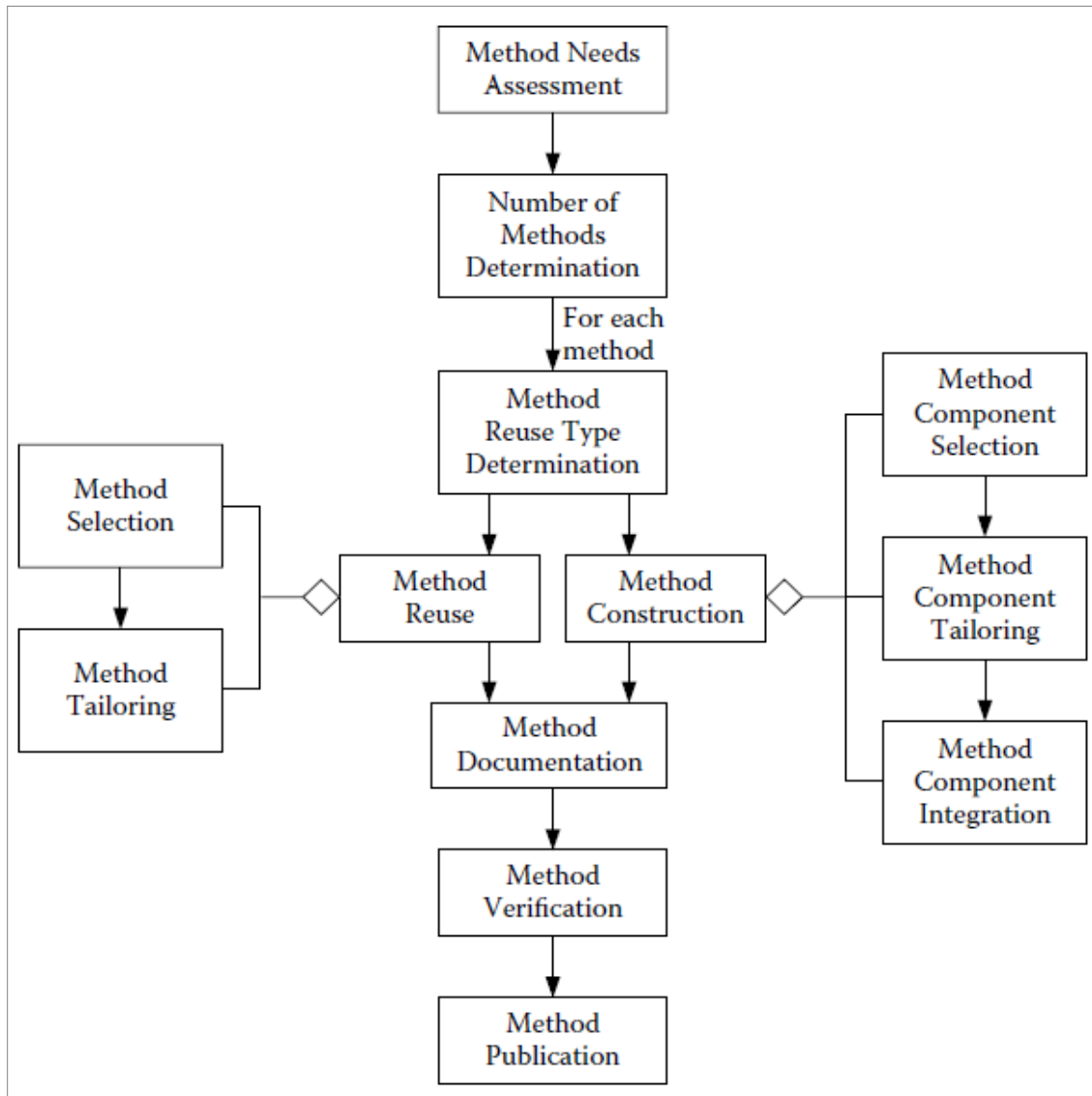


Figura 8 - Processo de Criação de Métodos. Fonte: FIRESMITH; 2009, p. 342.

É importante observar que a figura apresenta as doze etapas em uma ordem específica, porém elas são executadas de uma maneira iterativa, incremental, paralela e limitada no tempo (FIRESMITH, 2009, p. 340).

A primeira etapa desse processo (Figura 8) é a avaliação das necessidades (ou requisitos) do método a ser construído. Para isto é necessário analisar as características do sistema que será construído, as características da iniciativa e da organização e, finalmente, possíveis restrições de recursos.

As características do sistema incluem sua possível reutilização, por exemplo, se será utilizado em uma linha de produtos de software ou se será um esforço único. Deve ser feita uma estimativa quanto ao número provável de requisitos a serem elicitados. O tamanho do

futuro sistema também deve ser considerado, pois um sistema pequeno provavelmente irá precisar de um método mais simples. Embora a complexidade normalmente aumente com o tamanho, um sistema pode ser complexo mesmo que seja pequeno. O mesmo pode ser dito quanto à sua importância. Se um sistema é crítico para a missão do negócio, esforços maiores devem ser feitos para atingir um alto nível de qualidade na determinação dos requisitos.

As características da iniciativa (um projeto isolado, um conjunto (*portfolio*) de projetos ou a empresa como um todo) também devem ser levadas em consideração. O tempo disponível e o orçamento também têm impacto sobre o método a ser montado.

A organização possui várias características que podem afetar a seleção de técnicas tais como cultura organizacional, experiência com Engenharia de Métodos, uso interno do Método ou por uma empresa contratada, distribuição geográfica da empresa.

Finalmente, pode haver restrições ao próprio método como a integração a algum ciclo de vida de desenvolvimento (por exemplo, cascata ou espiral) ou a utilização de algum método padrão (por exemplo, RUP). Essas características propostas em FIRESMITH (2009) estão descritas na Tabela 5.

Jiang (Jiang, 2005) apresenta também vários critérios para seleção de técnicas. O seu trabalho aborda técnicas para desenvolvimento de requisitos. A Tabela 5 apresenta uma visão geral dos critérios propostos.

A segunda etapa na Figura 8 é para a determinação do número de métodos que serão necessários. Este número é obtido levando em consideração as informações obtidas na etapa anterior.

Na etapa de determinação do tipo de reutilização de método é decidido se algum método já existente deve ser reutilizado, ou se será necessário criar um novo método.

No caso da decisão ser a reutilização de um método já existente duas atividades podem ser realizadas: (i) Seleção de Método e (ii) Adequação do Método. Elas podem ser realizadas várias vezes, de forma iterativa e incremental (porém com um limite de tempo) até que o método esteja pronto para ser reutilizado.

Na Seleção de Método, as informações obtidas na Etapa 1, servem de critério para a escolha do método. No entanto, alguns componentes podem não estar ainda adequados à situação e será necessário executar a etapa de adequação.

Na Adequação do Método podem ser feitas alterações como, por exemplo, a retirada de uma etapa e/ou a substituição de um produto a ser gerado por outro.

A Construção de método pode também deve ser realizada de forma iterativa e incremental (porém com um limite de tempo). Dependendo da situação podem ser necessárias até quatro das subetapas seguintes: (i) Seleção do componente de método que pode começar selecionando primeiro a unidade de produto ou primeiro a unidade de trabalho. (ii) Construção do componente de método onde se o método desejado ainda não estiver no repositório, ele deverá ser criado. (iii) Adequação de componente que pode envolver a retirada de elementos que não são desejados, a inclusão de novos componentes ou a alteração de um componente já existente. (iv) Integração de componentes que envolvem a verificação de que não há componentes “soltos” (por exemplo, um produto que não é utilizado posteriormente) nem a falta de componentes (por exemplo, uma atividade que tem como entrada um artefato que não foi produzido anteriormente).

O objetivo da etapa de Documentação do método é a criação de documentos que permitam a utilização do método por pessoas que não participaram da sua criação. A documentação pode ser em papel ou como um *site* da *Web*.

É recomendável realizar uma verificação do método. Duas maneiras de fazê-lo são (1) uma verificação informal por um colega e (2) uma verificação pela equipe de processos da empresa.

Finalmente, o método deve ser tornado público para que possa ser utilizado. Por exemplo, o EPF Composer publica os métodos na forma de *Web site*.

Combinando as tabelas de Firesmith, Jiang e Harmsen temos a Tabela 5 onde o destaque em cinza são os critérios comuns aos três autores.

Tabela 5 - Critérios propostos por Firesmith (2009), Jiang (2005) e Harmsen (2001)

	Firesmith (2009)	Jiang (2005)	Harmsen (2001)
Características do sistema	✓		
- Quantidade de sistemas	✓	Nível de reutilização	✓
- Quantidade de requisitos	✓		
- Tamanho do sistema	✓	Tamanho do sistema	✓
- Complexidade do sistema	✓	Complexidade do sistema	✓
Volatilidade dos requisitos		✓	✓

Nível de segurança (<i>safety</i>) do sistema		✓	
Nível de qualidade do produto		✓	✓
Tipo do produto		✓	✓
Nível de inovação do produto		✓	✓
– Criticidade do sistema	✓		
Características da iniciativa	✓		
– Tipo de iniciativa	✓	Categoria do projeto	
– Tipo de contrato	✓	Nível de <i>outsourcing</i>	
– Restrições de tempo	✓	✓	✓
– Restrições de custo	✓	✓	✓
Heterogeneidade dos <i>stakeholders</i>		✓	✓
Tamanho da equipe		✓	✓
Importância de elicitar conhecimento implícito		✓	
Características da organização	✓		
– Cultura organizacional	✓		
– Experiência e <i>expertise</i>	✓	Conhecimento do domínio	✓
		Conhecimento de Engenharia de Requisitos	
		Conhecimento dos requisitos	
		Disponibilidade de facilitador	
– Separação organizacional	✓	Relacionamento organização-cliente	
– Separação geográfica	✓		
Disponibilidade do cliente		✓	
Restrições ao método	✓		
– Conformidade obrigatória	✓		
– Ciclo de vida e/ou desenvolvimento	✓		
– Ágil vs. documentado	✓		

Um outro modelo é o Modelo S3 de Harmsen et al. (Harmsen et al., 2003). Neste modelo os critérios para selecionar componentes de métodos estão divididos em (1) indicadores de *performance* (S*uccess*), (2) fatores sobre a situação (S*ituation*) e (3) aspectos do cenário (S*cenario*). Isto ocorre pois, segundo os autores, “the required aspects in a situational method, indicated by the project scenarios, depend on both the project situation and the required project success” (HARMSSEN, 2003, p. 2). Isto foi representado pelos autores na Figura 9.

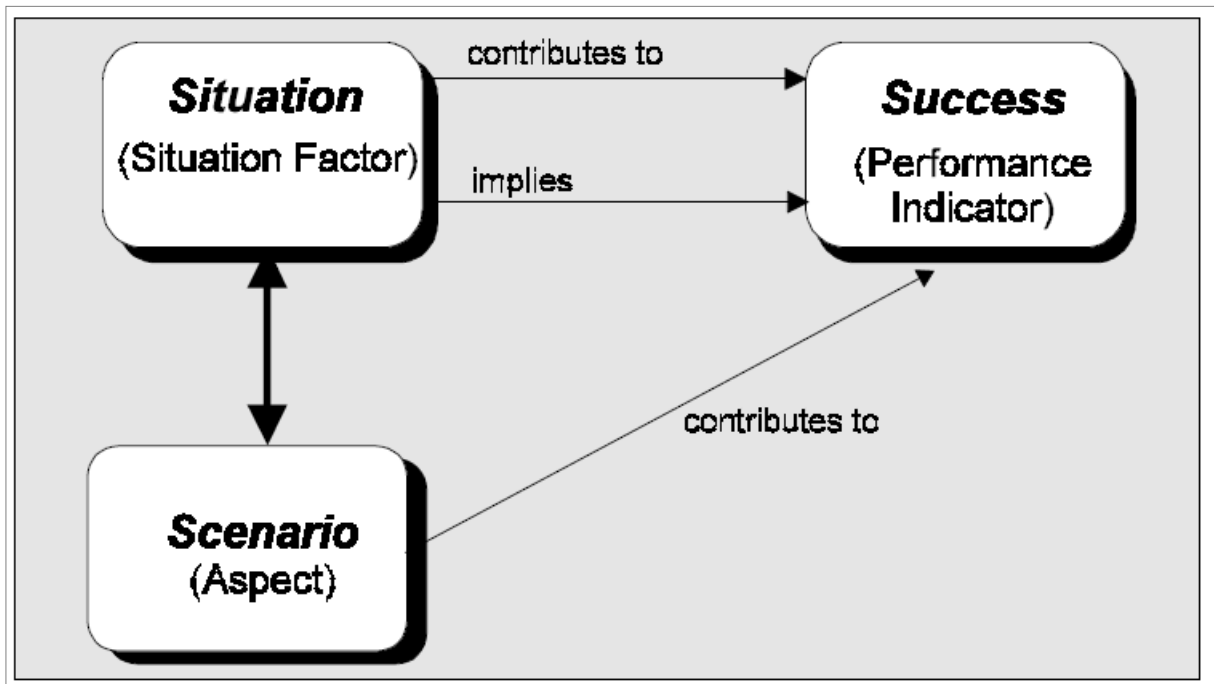


Figura 9 – O modelo S3. Fonte: HARMSEN et al, 2003, p. 4.

Os indicadores de *performance* podem ser de três tipos: (i) indicadores de *performance* do processo, (ii) indicadores de *performance* do produto e (iii) indicadores de *performance* do resultado. Além disso, fatores relacionados com a situação devem ser considerados. Estes fatores estão relacionados ao ambiente e ao projeto. Finalmente, um cenário possui vários aspectos que devem ser considerados no processo de seleção. A Tabela 6 mostra estes critérios.

Tabela 6 – Indicadores de Harmsen

3.1 – Indicadores relacionados com <i>performance</i> (<u>S</u> uccess)	3.1.1 – Relacionados ao processo	3.1.1.1 – Tempo
		3.1.1.2 – Dinheiro
		3.1.1.3 – Gerenciamento organizacional
		3.1.1.4 – Gerenciamento de informações
		3.1.1.5 – Controle de qualidade
		3.1.1.6 – Relacionamento com cliente
		3.1.1.7 – Gerenciamento de incertezas [riscos]
		3.1.1.8 – Controle de complexidade
		3.1.1.9 – Instalação conforme especificações
	3.1.2 – Relacionados ao produto	3.1.2.1 – Funcionalidade
	3.1.2.2 – Confiabilidade	
	3.1.2.3 – Usabilidade	
	3.1.2.4 – Eficiência	

		3.1.2.5 – Manutenibilidade
		3.1.2.6 – Portabilidade
	3.1.3 – Relacionados ao resultado	3.1.3.1 – Satisfação de necessidade de informação
		3.1.3.2 – Melhoria de processo de negócio
		3.1.3.3 – Aceitação do sistema
		3.1.3.4 – Adequação à organização
3.2 – Fatores da situação (<i>Situation</i>)	3.2.1 - Ambiente	3.2.1.1 - Comprometimento gerencial
		3.2.1.2 – Importância estratégica
		3.2.1.3 – Impacto organizacional
		3.2.1.4 – Nível de resistência
		3.2.1.5 – Potencial de conflito
		3.2.1.6 – Escassez de pessoas e recursos
		3.2.1.7 – Estabilidade
		3.2.1.8 – Formalidade
		3.2.1.9 – Conhecimento e experiência do usuário
	3.2.2 – Organização do projeto	3.2.2.1 – Capacidade da equipe
		3.2.2.2 – Tamanho da equipe
		3.2.2.3 – Dependência de outros projetos
	3.2.3 - Projeto	3.2.3.1 – Clareza do objetivo do projeto
		3.2.3.2 – Estabilidade do objetivo do projeto, qualidade das especificações, tamanho do projeto
		3.2.3.3 – Complexidade do sistema
		3.2.3.4 – Nível de inovação
3.3 – Aspectos do cenário (<i>Scenario</i>)	3.3.1 – Objetivo do projeto	
	3.3.2 - Abordagem	
	3.3.3 – Nível de participação do usuário	
	3.3.4 – Estratégia de instalação	
	3.3.5 – Gerenciamento de tempo	
	3.3.6 – Gerenciamento do orçamento	
	3.3.7 – Gerenciamento de informações	
	3.3.8 – Controle de qualidade	
	3.3.9 – Gerenciamento da organização do projeto	
	3.3.10 – Gerenciamento de contratos e aquisições	
	3.3.11 – Suporte metodológico e treinamento	
	3.3.12 – Divisão em fases	
	3.3.13 - Responsabilidades	
	3.3.14 – Nível de reutilização	
	3.3.15 – Ponto de partida para análise	

- 3.3.16 – Estratégia de validação
- 3.3.17 - Ferramentas
- 3.3.18 – Aspectos de modelagem
- 3.3.19 – Perspectivas de modelagem

Fonte: criada a partir de HARMSSEN et al., 1995, p. 5-8

1.2.4 SPEM (Software & Systems Process Engineering Meta-Model Specification)

Dada a necessidade de reutilizar o conhecimento sobre os componentes de métodos, é necessário encontrar uma maneira de os definir, armazenar e recuperar. Para definir os componentes de um método é preciso utilizar um meta-modelo de método, como vimos na seção 1.2. Um meta-modelo deste tipo foi criado pelo OMG (*Object Management Group*). O OMG é conhecido por ser o responsável por vários padrões, sendo que o mais famoso é o UML (*Unified Modeling Language*). O meta-modelo criado pela OMG é o SPEM (*Software & Systems Process Engineering Meta-Model Specification (Version 2.0)*).

A Figura 10 mostra como o *framework* é usado.

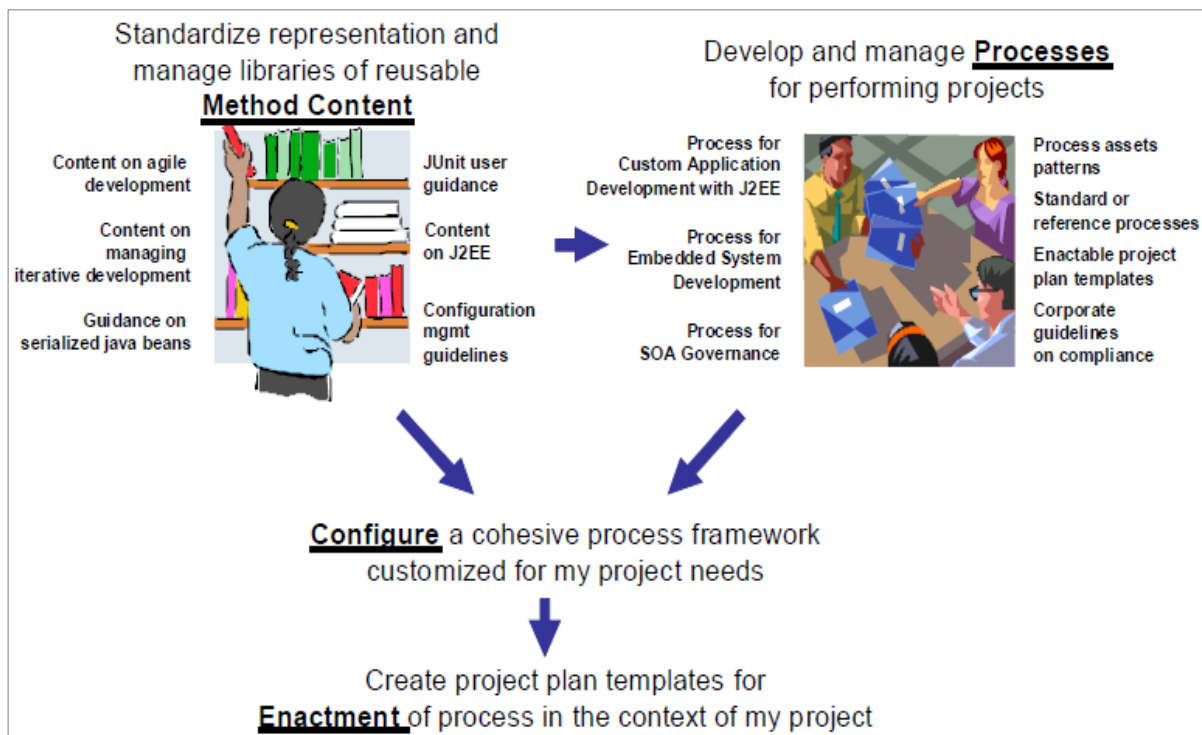


Figura 10 – Utilização do SPEM. Fonte: OMG, 2008, p.10.

O Conteúdo de Método (Method Content, na Figura 10) é criado usando o meta-modelo. Desta forma, é possível criar conteúdo a partir de várias fontes, ao mesmo tempo, com a garantia de compatibilidade formal. O objetivo é criar componentes de método (embora o padrão não use este termo). Os componentes de método permitirão atingir o objetivo de reutilização, pois poderão ser usados em várias combinações.

Um Processo (na Figura 10, o Processes) irá especificar combinações dos componentes de métodos (*Method Content*). Estas combinações são conhecidas na ferramenta EPF (*Eclipse Process Framework*), vista na seção 1.2.5, como “*capability patterns*”. Estes processos também são Componentes de Método (embora o SPEM não os considere como *Method Content*).

Em uma certa situação, os componentes a serem utilizados, para um tipo específico de projeto serão selecionados em uma Configuração (Configure na Figura 10).

Depois de criada a Configuração, na etapa Realização (Enactment na Figura 10), ela será concretizada com, por exemplo, a especificação dos nomes dos programadores e as datas das entregas. Para isto pode ser usado um software de gerenciamento de projetos como o Microsoft Project ou o dotProject.

Esta separação entre o conteúdo do método e o processo de desenvolvimento não é nova, pois já foi usada no RUP, IBM Rational Unified Process e está refletida também na estrutura do SPEM, como pode ser visto na Figura 11.

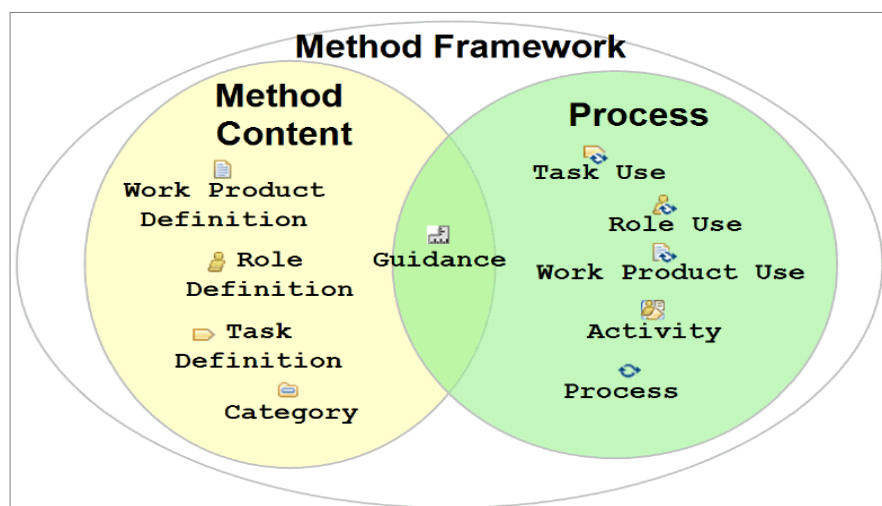


Figura 11 – Terminologia mapeada para Conteúdo de Métodos e Processo.

Fonte: OMG, 2008, p. 14.

Existem vários níveis de abstração envolvidos na utilização do SPEM. A Figura 12 mostra os relacionamentos entre os vários níveis.

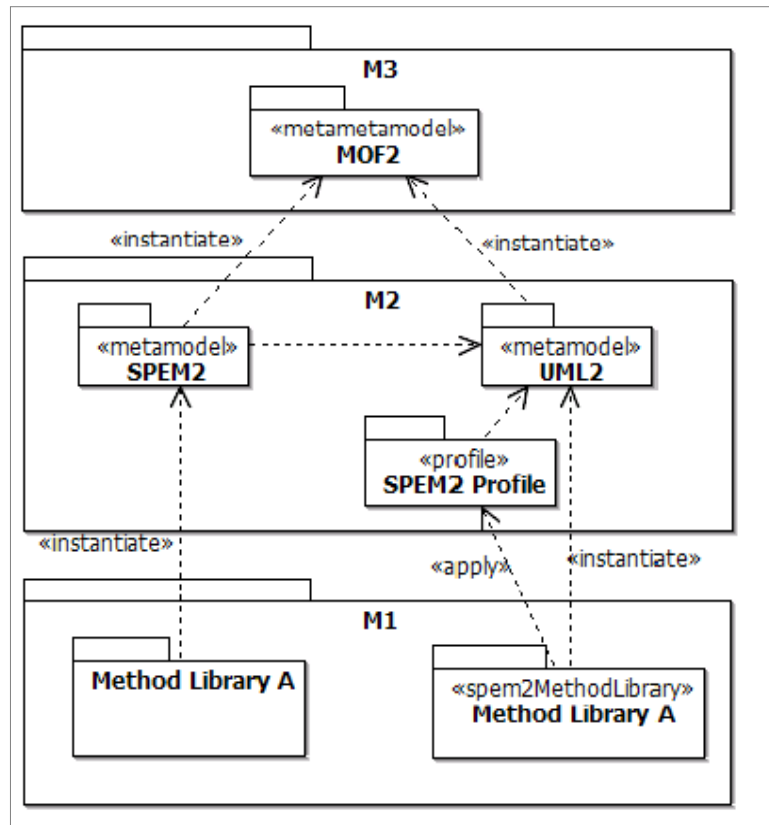


Figura 12 – Camadas do modelo para UML e SPEM 2.0.

Fonte: OMG, 2008, p. 20.

Como pode ser visto na Figura 12, o SPEM (SPEM2 na Figura) é um meta-modelo (no nível M2) usado pelo Engenheiro de Métodos para criar a sua Biblioteca de Métodos (*Method Library*) no nível M1. Esta irá conter os componentes de método a serem usados para criar modelos para os métodos específicos para uma determinada situação (no nível M0 da Figura 13).

A Figura 13 fornece um exemplo de como os níveis de abstração dos modelos OMG se relacionam com o padrão SPEM.

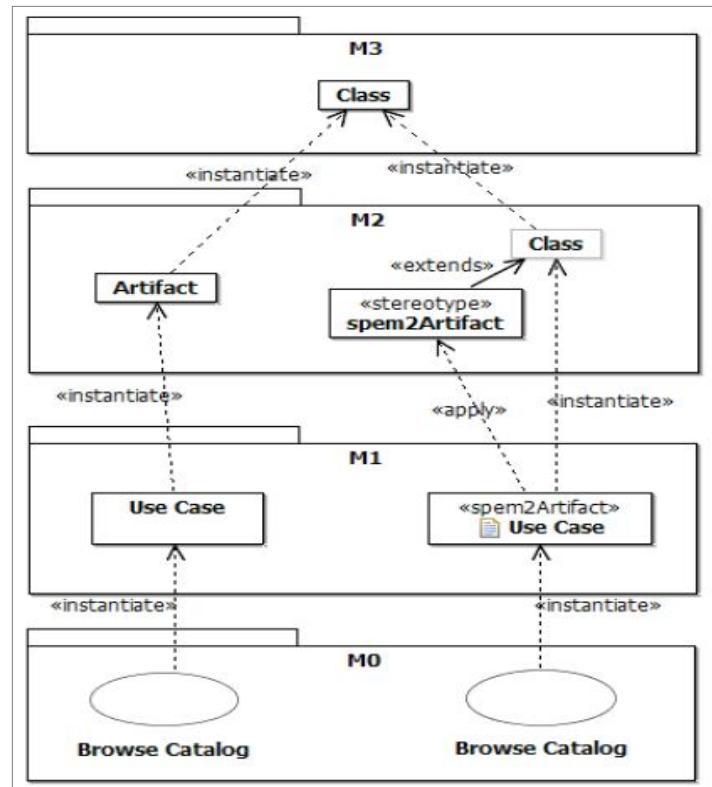


Figura 13 – Exemplo de instânciação das camadas. Fonte: OMG, 2008, p. 21.

No nível M0 (ver Figura 13) temos o uso real (por exemplo, no software de gerenciamento de projetos dotProject) do método definido em M1. O método em M1 usa elementos do meta-modelo SPEM (*Artifact*, neste exemplo) para criar o artefato Caso de Uso (“Use Case”, na Figura 13). O Artefato (“*Artifact*”) do nível M2 é, por sua vez, uma instância de “*Class*”, definida no nível M3. O padrão para o nível M3 é o MOF (*Meta Object Facility*), também do OMG.

Comparando a Figura 13 (relativa ao SPEM) com a Figura 6 (relativa ao MFESA de Firesmith et al.) é possível dizer que o nível inferior do MFESA corresponde ao nível M0 do SPEM, o nível intermediário do MFESA ao nível M1 e o nível superior aos níveis M2 e M3 do SPEM.

Existem ferramentas para utilizar a especificação SPEM. Nesta dissertação usaremos o EPF (*Eclipse Process Framework*) Composer, visto na seção 1.2.5.

1.2.4.1 Arquitetura geral do Meta-modelo SPEM

O meta-modelo do SPEM possui a estrutura mostrada na Figura 14. Esta estrutura

permite que o implementador de um modelo baseado em SPEM selecione apenas os pacotes (*packages*) necessários para as suas necessidades. Em todas as implementações, o pacote *Core* deve estar sempre implementado.

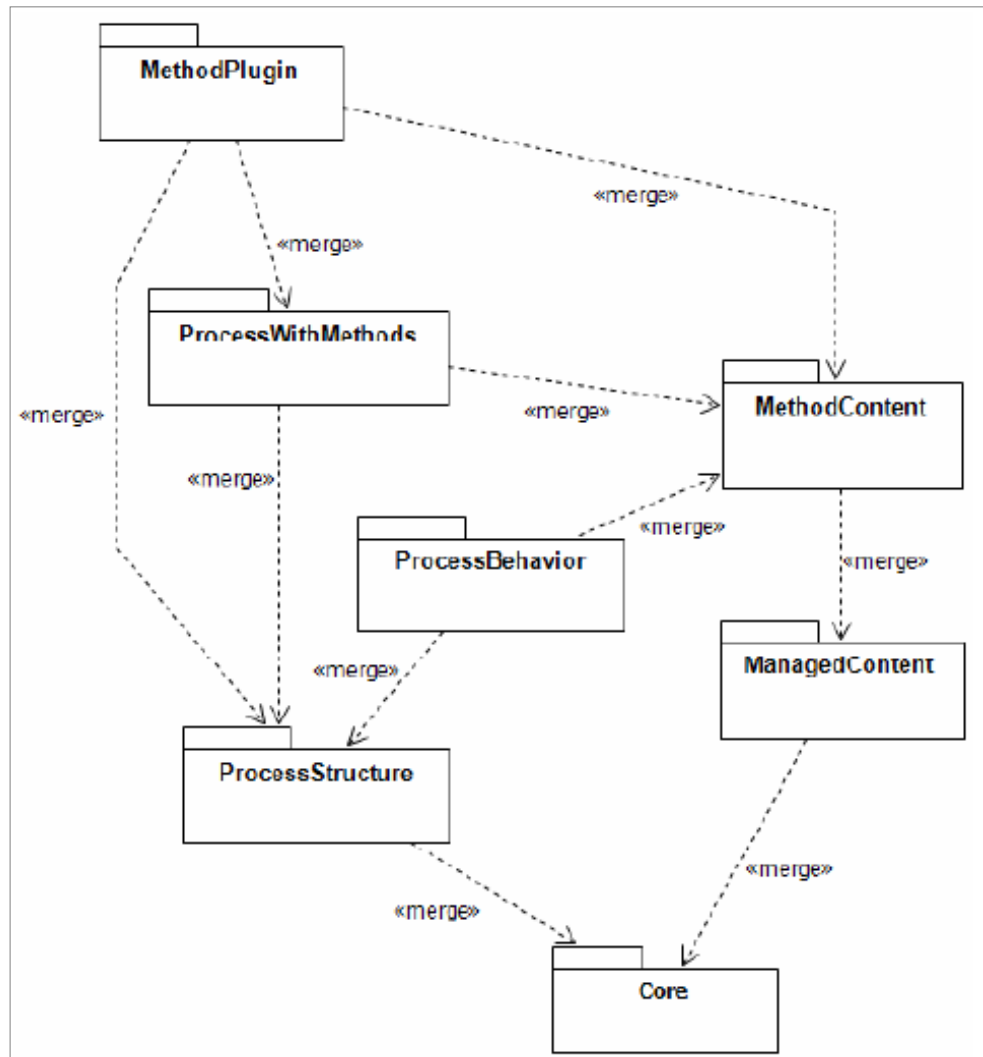


Figura 14 – Estrutura do Meta-Modelo SPEM. Fonte: OMG, 2008, p. 2.

O pacote *Core* (OMG, 2008, p. 3) possui as classes que servem de base para as classes de todos os outros pacotes do meta-modelo.

O pacote *Process Structure* define a base para todos os modelos de processo. A estrutura básica de um processo é uma decomposição de atividades conhecida como *Work Breakdown Structure (WBS)*. O *WBS* é formado por *Activities*, onde cada *Activity* pode ser decomposta em uma ou mais *Activities*. É possível definir tipos (*Kinds*) de atividades. Por

exemplo, uma Fase é uma atividade que não se repete, composta por uma ou mais atividades. Uma Iteração é uma atividade que pode ser executada repetidas vezes. O próprio conceito de Processo, no SPEM, é definido como uma *Activity*.

O pacote de *Process Behavior* permite representar o comportamento dos processos, embora não especifique um padrão de comportamento específico. É possível estabelecer ligações entre os elementos estáticos do pacote *Process Structure* e modelos como o UML 2 *Activity Diagram* ou o *Business Process Modeling Notation (BPMN)*.

O pacote *Managed Content* permite associar descrições textuais aos elementos do modelo. É possível criar um método menos formal que utilize apenas descrições textuais.

O pacote *Method Content* permite criar uma base de conhecimento, independente de situações específicas, com conteúdo que pode ser usado na criação de métodos. Seu objetivo é permitir a reutilização da experiência adquirida em outros projetos.

O pacote *Process with Methods* define estruturas que permitem integrar o pacote *Process Structure* com o pacote *Method Content*, isto é, os processos que foram especificados usando o pacote *Process Structure* podem utilizar o conhecimento detalhado especificado no pacote *Method Content*.

Finalmente, o pacote *Method Plugin* permite a criação de métodos complexos que podem ser configurados através de mecanismos de variabilidade como Contribuição (*Contributes*), Substituição (*Replaces*), Estende (*Extends*) e Estende-e-Substitui (*Extends-Replaces*). É possível então obter reutilização com adequação a cada situação.

Não é necessário implementar todos os pacotes. O SPEM define vários modelos de conformidade (*Compliance Points*) que podem ser utilizados, dependendo das condições encontradas no momento. A Tabela 7 mostra as opções mencionadas no SPEM (OMG, 2008, p. 4-7).

Tabela 7 – Exemplos de Níveis de Conformidade no SPEM

	<i>Core</i>	<i>Process Structure</i>	<i>Process Behavior</i>	<i>Managed Content</i>	<i>Method Content</i>	<i>Process with Methods</i>	<i>Method Plugin</i>
1. <i>SPEM Complete</i>	✓	✓	✓	✓	✓	✓	✓
2. <i>SPEM Process with Behavior and Content</i>	✓	✓	✓	✓			
3. <i>SPEM Method Content</i>	✓			✓	✓		
4. <i>Process Structure and Behavior</i>	✓	✓	✓				

	<i>Core</i>	<i>Process Structure</i>	<i>Process Behavior</i>	<i>Managed Content</i>	<i>Method Content</i>	<i>Process with Methods</i>	<i>Method Plugin</i>
5. <i>Process Structure and Managed Content</i>	✓	✓		✓			
6. <i>Process Structure, Behavior and Method Plugin</i>	✓	✓	✓				✓
7. <i>Process with Methods, Structure, Method Content e Managed Content</i>	✓	✓		✓	✓	✓	

Fonte: OMG, 2008, p. 4-7.

A segunda opção da Tabela 7 é indicada para os que querem se concentrar na modelagem do processo (isto é, na estrutura de atividades) considerando apenas o *WBS (Work Breakdown Structure)* e os diagramas de *workflow*.

A terceira opção pode ser usada quando o objetivo é apenas documentar técnicas, produtos e orientações. Esta documentação pode ser consultada durante a execução do projeto.

Na quarta opção, o conteúdo textual do *Managed Content* não é necessário e, portanto, não é incluído.

Na quinta opção, o *Process Behavior* não é incluído, o que significa que apenas a estrutura estática do *WBS* e a sua especificação textual devem ser incluídos.

Na sexta opção, apenas a estrutura e o comportamento são necessários. Para obter a reutilização com flexibilidade, os recursos de variabilidade do pacote *Method Plugin* também são incluídos.

Finalmente, na sétima opção os recursos de variabilidade do pacote *Method Plugin* não são necessários.

1.2.4.2 O pacote *Core*

O pacote *Core* deve sempre fazer parte de uma implementação SPEM pois define as classes que serão a fundação sobre a qual serão definidas as outras. As classes definidas neste pacote têm duas finalidades principais: permitir que as outras classes possam ser qualificadas com um Tipo (*Kind*) e definir as classes que permitirão expressar os processos.

A Figura 15 mostra os principais estereótipos do pacote.

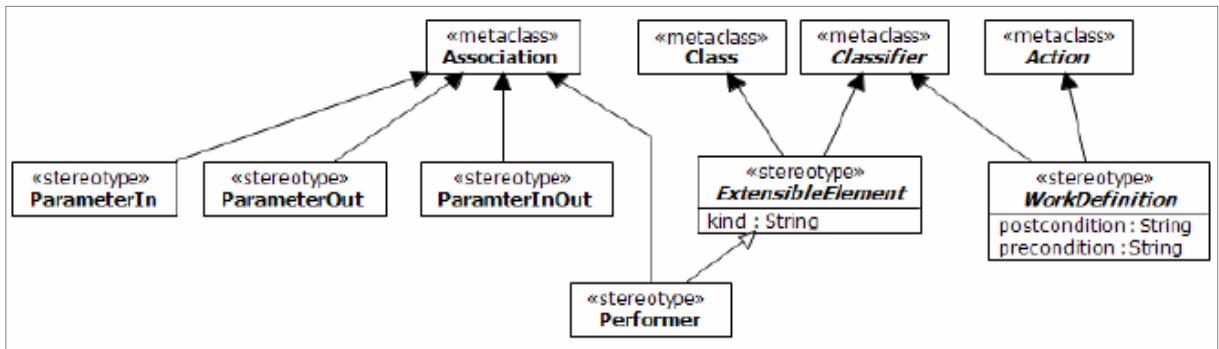


Figura 15 – Os estereótipos do Perfil UML 2 do SPEM 2.0 definidos no pacote *Core*.

Fonte: OMG, 2008, p. 35.

Na Figura 15 a classe *ExtensibleElement* vai permitir a criação dos Tipos (*Kinds*) e a classe *WorkDefinition* permitirá a representação dos processos. A Figura 16 mostra a classe *WorkDefinition* com mais detalhes.

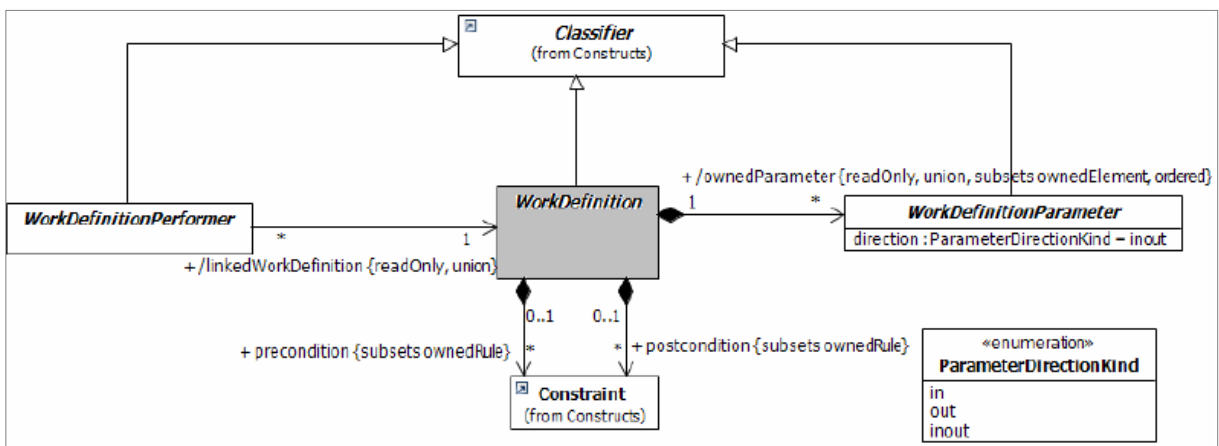


Figura 16 – *Work Definition* e seus elementos relacionados

Fonte: OMG, 2008, p. 38, (meu sombreado).

“A classe *Work Definition* é um Classificador abstrato que generaliza todas as definições de trabalho dentro do SPEM 2.0” (OMG, 2008, p. 38). O trabalho está associado a parâmetros (entrada, saída e entrada/saída), a quem irá executá-lo (*Performer*) e, opcionalmente, a restrições de entrada e saída (pré-condições e pós-condições). Uma *Work Definition* (Figura 16) pode ser representada usando uma *Activity* de UML 2, se for realizado um *merge* com o pacote *Process Behavior*.

1.2.4.3 O pacote *Managed Content*

Este pacote torna possível a descrição textual dos elementos de processo. A classe *Describable Method* servirá como superclasse para as classes *Process Element* (do pacote *Process Structure*) e *Method Content Element* (do pacote *Method Content*) que, portanto, herdarão os seus atributos. A Figura 17 mostra as partes do *Describable Element* e as suas subclasses.

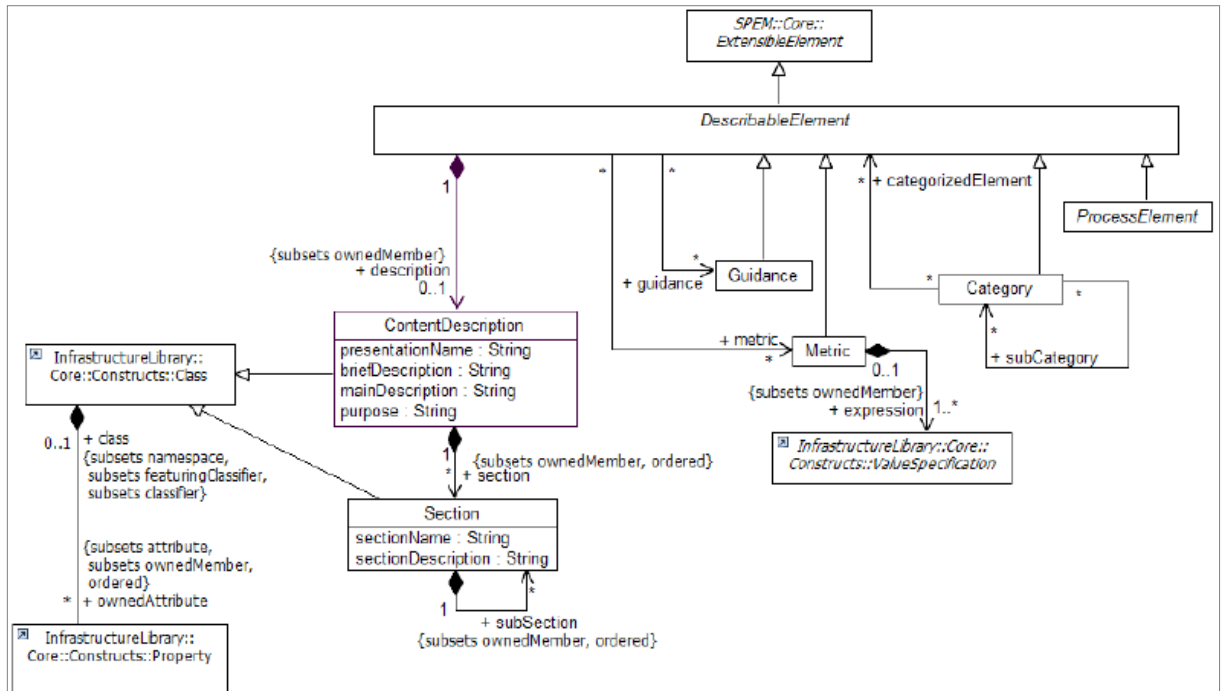


Figura 17 – As partes do *Describable Element* e suas subclasses. Fonte: OMG, 2008, p. 74.

Um *Describable Element* é uma “generalização abstrata de todos os elementos do SPEM 2.0 que podem ser documentados com descrições textuais” (OMG, 2008, p. 77). Como pode ser visto na Figura 17 um *Describable Element* pode conter um elemento *Content Description* que guarda as descrições textuais. Isto é, o *Content Description* guarda, além de seu nome, um nome para apresentação, um descrição breve e uma descrição principal e a sua finalidade. Além disso, pode ser dividido em seções, como na estrutura de um livro.

A descrição de um elemento pode ser complementada por uma ou mais ocorrências da classe *Guidance*. Esta orientação pode ser de vários tipos como, por exemplo, modelos, *checklists*, estimativas e conceitos.

A classe *Category* é um *Describable Element* que pode ser associado a um grupo de *Describable Elements* para criar grupos de *Describable Elements*, baseados em critérios

definidos pelo usuário do SPEM (OMG, 2008, p 74). Uma *Category* pode ter subcategorias levando à criação de hierarquias com um número arbitrário de níveis. Um *Describable Element* pode estar associado a várias categorias. As *Categories*, portanto, não definem taxonomias, apenas facetas.

1.2.4.4 O pacote *Method Content*

Este pacote define os elementos centrais de um método, como *Roles*, *Tasks* e *Work Product Definitions*. Observe que os elementos do pacote podem formar o conteúdo de um método, porém não são o próprio método. O que esta dissertação chama de “método” é equivalente ao que o SPEM chama de *Delivery Process*. A Figura 18 mostra a taxonomia dos *Describable Elements* centrais.

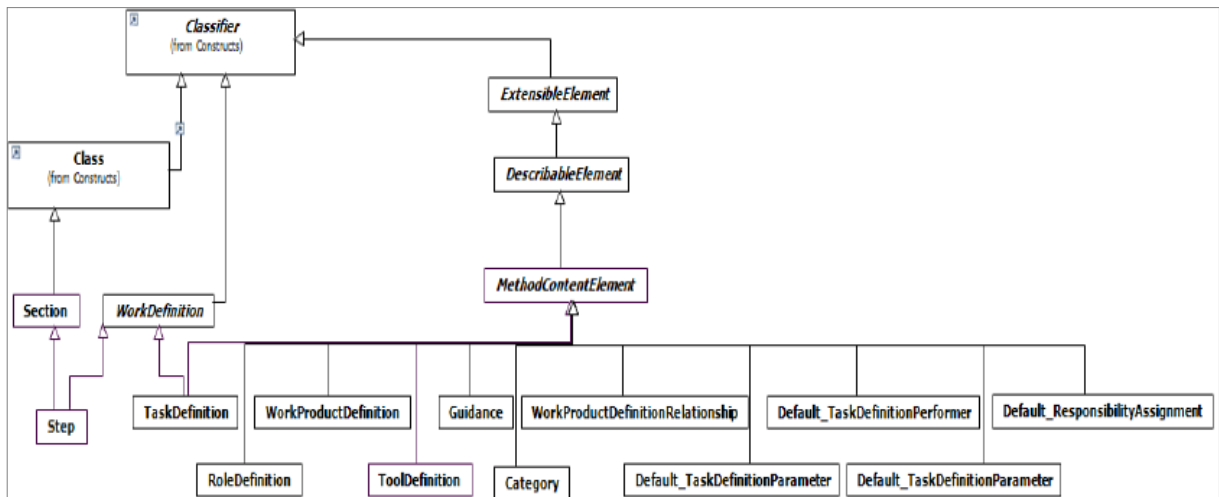


Figura 18 – Taxonomia dos *Describable Elements* centrais. Fonte: OMG, 2008, p. 82.

Pode-se observar na Figura 18 que todos os elementos são derivados de *Method Content Element* e, como este é derivado de *Describable Element*, podem conter uma descrição textual.

Os principais elementos do pacote possuem relacionamentos como os que são mostrados na Figura 19. A descrição do conteúdo de método é composta, fundamentalmente, por *Task Definitions*, com *Steps*, que tem *Work Product Definitions* como entradas e saídas e que são executadas por *Roles Definitions*. Alguns destes *Roles* são os responsáveis e outros são apenas participantes. (OMG, 2008, p.82)

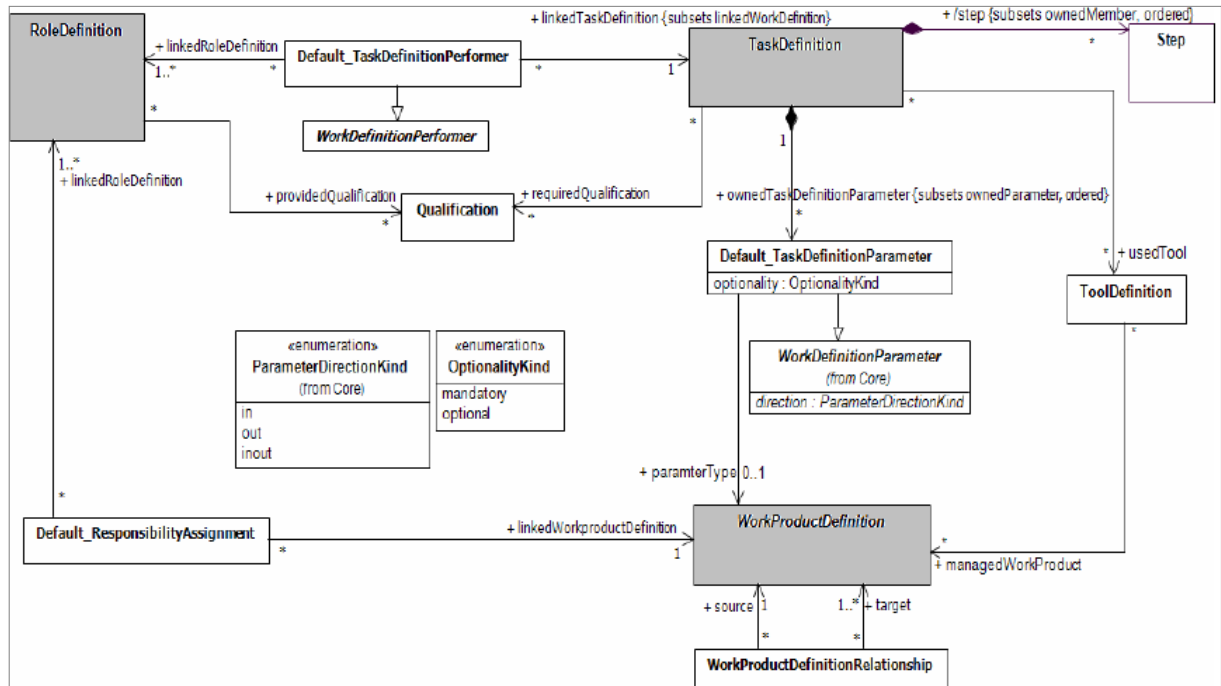


Figura 19 – Principais *Method Content Elements* e seus relacionamentos (meu sombreado)

Fonte: OMG, 2008, p. 83.

Uma *Task Definition* é um *Method Content Element* e um *Work Definition* (ver Figura 18) que define o trabalho a ser feito (*performed*) por instâncias de *Role Definition* (*performers*) Uma *Task* está associada a *Work Products* de entrada e saída e sua execução deve ser responsabilidade de algum *Role Definition* (OMG, 2008, p. 90).

Uma *Task Definition* deve ter um objetivo (*purpose*, derivado de *Content Description*). Deve fornecer instruções detalhadas (*Steps*) sobre como realizar o trabalho. Estas instruções devem ser independentes do momento, dentro de um processo, em que são executadas. Isto permite que a *Task Definition* possa ser reutilizada em vários pontos de processo e/ou por processos diferentes, seguindo o princípio da separação entre *Method Content* e *Process* que foi apresentada anteriormente.

“Um *Role Definition* é um *Method Content Element* que define um conjunto de habilidades, competências e responsabilidades” (OMG, 2008, p.87). Os Papéis (*Roles*) representam quem vai executar uma Tarefa (*Task*).

Um *Work Product Definition* é um *Method Content Element* que pode ser recebido por uma Tarefa como um parâmetro de entrada ou produzido pela Tarefa como um parâmetro de saída.. Um *Work Product* pode ser responsabilidade de um *Role Definition*. O SPEM 2.0

Base Plug-in define alguns tipos de *Work Product* que são mostrados na Figura 20.

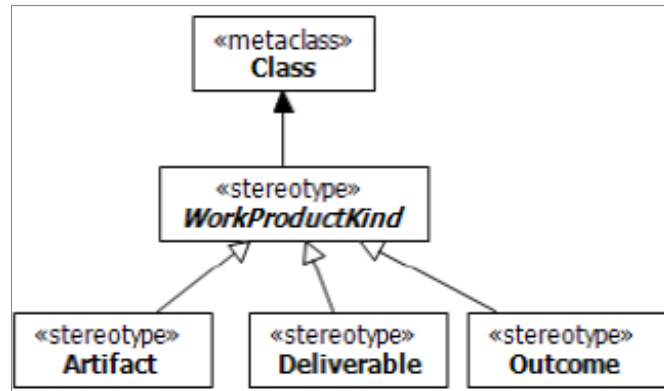


Figura 20 – Estereótipos de *Work Product Kind*. Fonte: OMG, 2008, p. 165.

Um *Artifact* é um “*Work Product Definition* que fornece uma descrição e definição para tipos de produtos tangíveis” (OMG, 2008, p. 166). Já os *Outcomes* “fornecem uma descrição e definição para produtos de trabalho não-tangíveis” (OMG, 2008, p. 165). Finalmente um *Deliverable* “fornece uma descrição e definição para empacotar outros *Work Products* e podem ser entregues para alguém interno ou externo” (OMG, 2008, p. 166). Ou seja, uma Entrega (*Deliverable*) agrega outros *Work Products* (inclusive outras Entregas).. Geralmente é entregue no fim de um Processo (ver definição no pacote *Process Structure*).

Como pode ser visto na Figura 21 um *Work Product Definition* pode estar relacionado a outro através de um *Work Product Definition Relationship*.

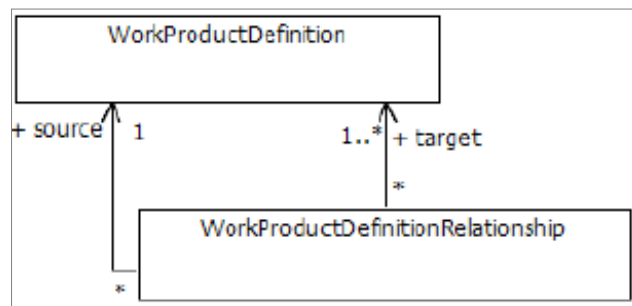


Figura 21 – Taxonomia de *Work Product*. Fonte: OMG, 2008, p. 92.

O SPEM 2.0 *Base Plug-in* define alguns tipos de *Work Product Relationship*, que são mostrados na Figura 22.

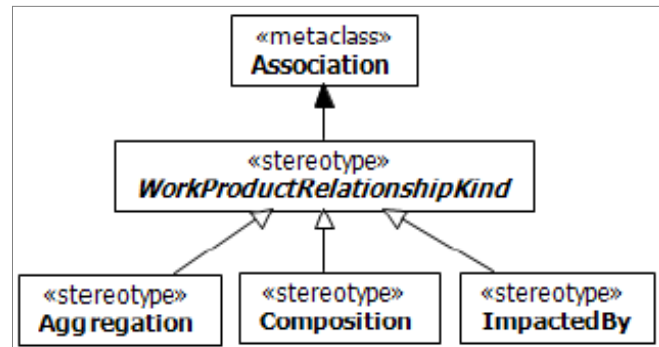


Figura 22 – O *Work Product Relationship Kinds*. Fonte: OMG, 2008, p.167.

Uma *Composition* indica que uma instância de um *Work Product* é parte de outra instância. Uma *Aggregation* indica que um produto de trabalho “is used with another” (OMG, 2008, p. 167). Finalmente, *ImpactedBy* indica que uma instância pode afetar outra. Por exemplo, se uma instância mudar isto irá afetar uma outra.

1.2.4.5 O pacote *Process Structure*

Este pacote contém os elementos básicos da estrutura para definir processos de desenvolvimento. (OMG, 2008, p. 43). No SPEM os processos são representados por meio de *Work Breakdown Structure (WBS)* de Atividades (ver a seguir), que são decompostas em outras atividades, formando uma árvore hierárquica, até chegar aos elementos “folha”.

A Figura 23 mostra uma visão geral das classes no pacote *Process Structure*.

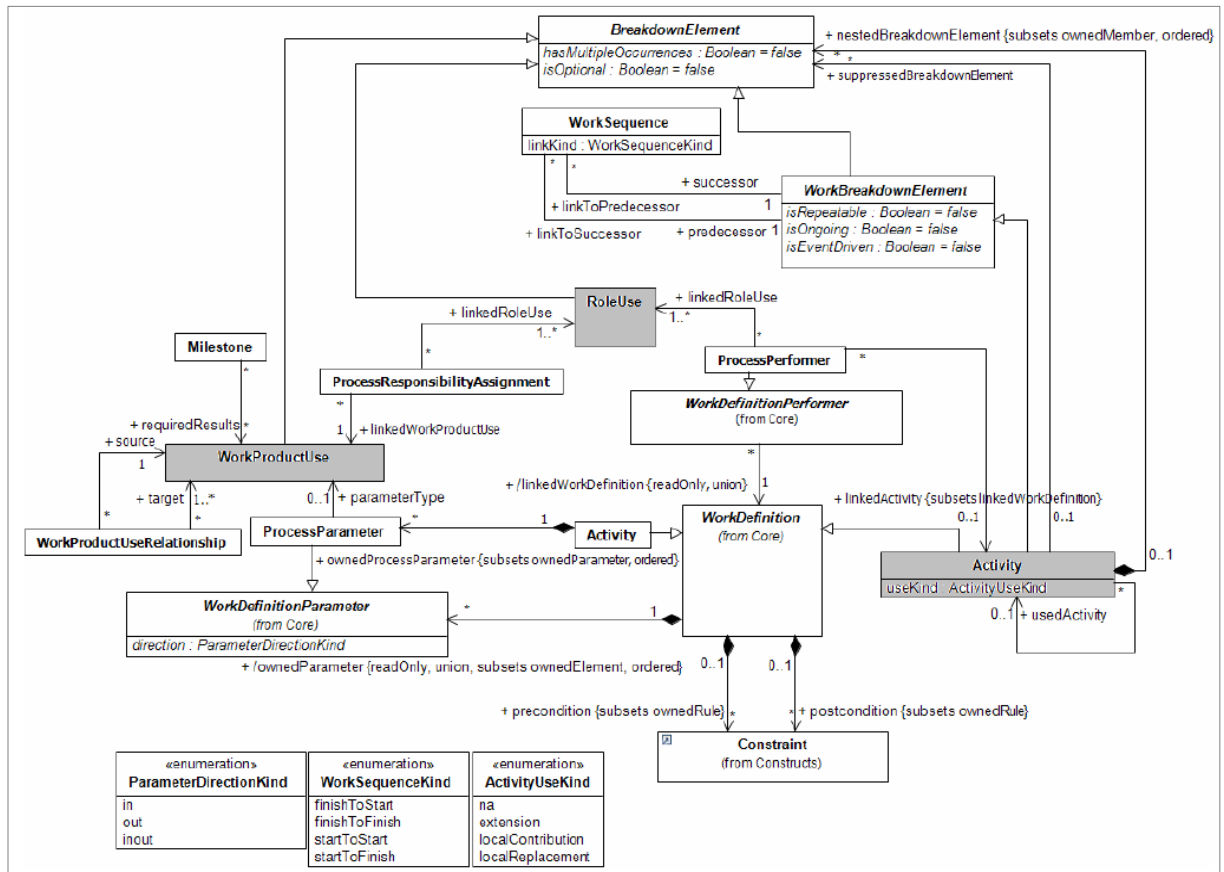


Figura 23 – Visão geral das principais classes e associações do pacote *Process Structure*

Fonte: OMG, 2008, p. 44, meu sombreado.

A classe *Breakdown Element* é uma classe abstrata que pode representar qualquer *Process Element* que faça parte de uma estrutura de *breakdown* (OMG, 2008, p.54) A Figura 24 mostra que a associação *nestedBreakdownElement* permite a criação da hierarquia mencionada anteriormente.

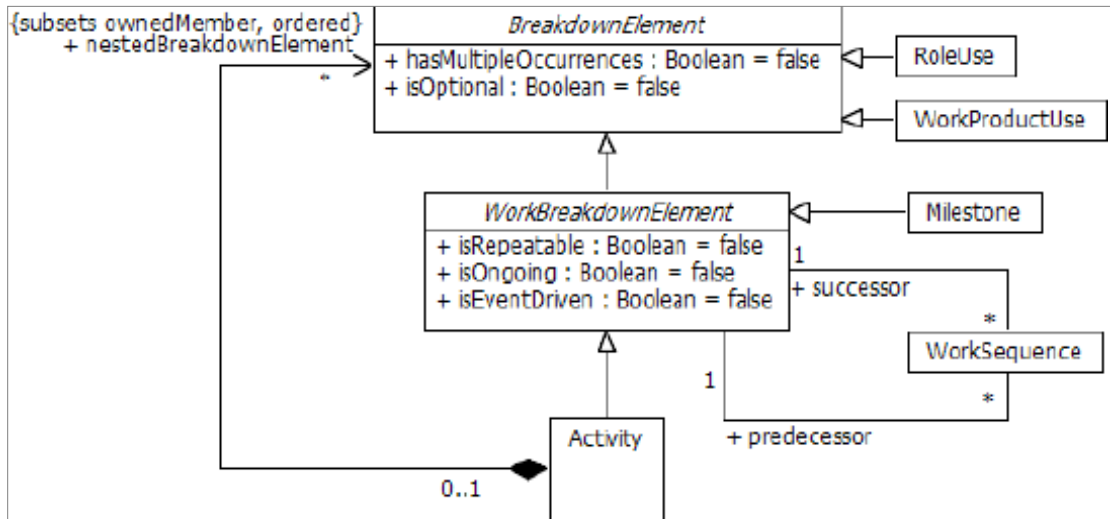


Figura 24 – Uma estrutura analítica é definida por *Activities* que englobam *Breakdown Elements*. Fonte: OMG, 2008, p. 54.

A classe *Activity* “defines basic units of work within a Process as well as a Process itself. In other words, every Activity represents a Process in SPEM 2.0.” (OMG, 2008, p. 46). Isto significa que, no SPEM, os processos são representados internamente como atividades. É possível definir Tipos de Atividade (*Activity Kinds*). A definição de tipos de atividade segue o padrão do *SPEM 2.0 Base Plug-in* e é mostrada na Figura 25.

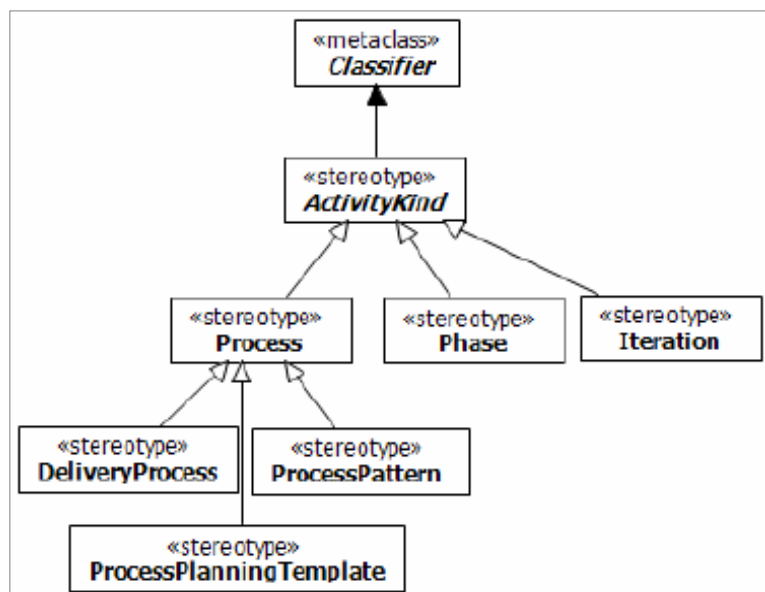


Figura 25 – Estereótipos de tipos de atividades. Fonte: OMG, 2008, p.155.

Um *Process* (na Figura 25) é o nome do Tipo de Atividade que é aplicado a “uma *Activity* especial que descreve a estrutura para tipos específicos de projetos de desenvolvimento ou para partes deles” (OMG, 2008, p. 156). Os processos podem ser de três tipos: *ProcessPattern*, *DeliveryProcess* e *ProcessPlanningTemplate*.

Um *ProcessPattern* é “um processo especial que descreve um conjunto reutilizável de *Activities* em uma área geral de processo que fornece uma abordagem consistente para problemas comuns” (OMG, 2008, 158). É uma boa prática que ele crie uma ou mais Entregas e que seja independente de processos específicos (para que possa ser reutilizado).

Um *DeliveryProcess* “é um processo especial que descreve uma abordagem completa e integrada para realizar um tipo de projeto específico” (OMG, 2008, 157). Nesta dissertação é o que é chamado de Método. Assim, a criação de um método específico para atender a uma solicitação em uma situação dada significa, em termos de SPEM, a criação de um *Delivery Process* que leva a situação em consideração. O *Delivery Process*, tipicamente, é a base para a definição de um projeto. Isto é feito por meio da criação de um *ProcessPlanningTemplate*. Este último é um “special Process that is prepared for instantiation by a project planning tool” (OMG, 2008, p. 160).

Uma Fase (*Phase*) “represents a significant period in a project, ending with major management checkpoint, milestone, or set of Deliverables” (OMG, 2008, p.156). Geralmente é uma *Activity* com o atributo *isRepeatable* especificado como '*False*'. Este atributo é, por *default*, colocado como '*True*' para uma Iteração (*Iteration*), pois um Iteração é um conjunto de *Activities* que podem ser repetidas mais de uma vez (OMG, 2008, p. 156)

A Figura 26 mostra os relacionamentos da classe *Activity*. As classes *RoleUse* e *WorkProductUse* representam utilizações das classes *Role Definition* e *Work Product Definition*, vistas a seguir.

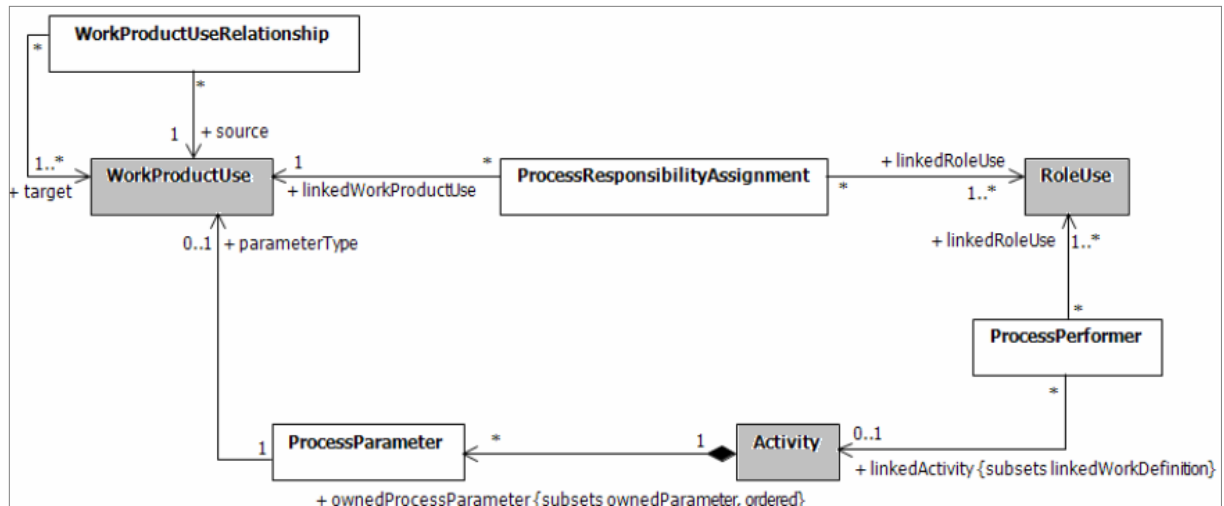


Figura 26 – A *Work Definition* de Atividade e seus relacionamentos com *Performers* e Entradas/Saídas. Fonte: OMG, 2008, p. 46.

Um *Role Use* “is a special Breakdown Element that either represents a performer or a participant of the Activity. If it is a performer, the Role Use and Activity need to be related via a Process Performer ” (OMG, 2008, p. 59). Se for um participante, fica em uma composição *nestedBreakdownElement*. Um *Role Use* representa uma ocorrência específica de um Papel pré-definido (*Role Definition*).

Um *Work Product Use* “is a special Breakdown Element that either represents an input and/or output type for an Activity or represents a general participant of the Activity. If it is an input/output, then the Work Product Use needs to be related to the Activity via the Process Parameter class. (OMG, 2008, p. 62). Se for um participante, fica em uma composição *nestedBreakdownElement*

1.2.4.6 – O pacote *Process Behavior*

O SPEM 2.0 foi projetado para trabalhar em conjunto com outros modelos que incorporam comportamento. A especificação sugere o uso do *UML 2 Activity Diagram*, *UML 2 State Machine Diagram* ou até mesmo *Business Process Definition Metamodel/Business Process Modeling Notation (BPDM/BPMN)*. Para permitir esta integração, o SPEM define classes que servem como intermediárias entre o metamodelo SPEM e os outros. A Figura 27 mostra como estas classes (que são indicadas pelo prefixo “ext_”) são utilizadas.

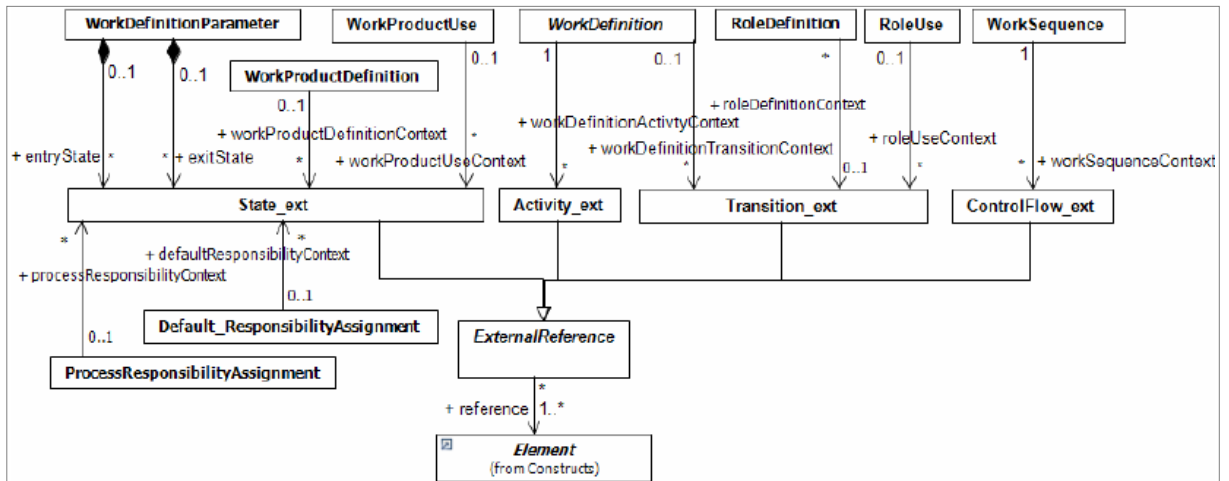


Figura 27 – Ligações de rastreabilidade entre as principais abstrações do modelo de comportamento e os principais *Process Elements*. Fonte: OMG, 2008, p. 69.

As classes *Activity_ext*, *Control Flow_ext*, *State_ext* e *Transition_ext* representam uma referência a uma classe em um modelo externo de comportamento.

1.2.4.7 O pacote *Process with Methods*

O nível de conformidade 2 (*SPEM Process with Behavior and Method*) apresentado na Tabela 7 da seção 1.2.4.1, permite desenvolver um método situacional (chamado de *Process* no SPEM) para cada novo projeto. Já o nível de conformidade 3 (*SPEM Method Content*) permite a documentação detalhada de Componentes de Método (chamado de *Method Content* no SPEM) com o objetivo de documentar e reutilizar conhecimentos sobre como alguém (*Role* no SPEM) pode trabalhar a partir de Produtos de Entrada (*Work Products* de entrada no SPEM), seguindo instruções precisas (*Steps* no SPEM) para obter Resultados (*Work Products* de saída no SPEM). O pacote *Process with Methods* permite combinar (*merge*) as duas abordagens criando um método (*Process* no SPEM) para atender a uma determinada situação e utilizando a base de conhecimentos de técnicas e resultados armazenados no pacote *Method Content*.

Um princípio fundamental do SPEM é a separação entre o Conteúdo de Método e Processo que vimos na Figura 10. Nos níveis de conformidade 2 e 3 é utilizada a classe *Package* para conter todas as outras classes de um pacote. Para utilizar os pacotes destes dois

níveis em conjunto, porém mantendo a sua separação, foram criadas as classes *ProcessPackage* e *MethodContentPackage* (Figura 28). Estas classes são especializações da classe *Package* que podem conter, respectivamente, apenas elementos de processo ou elementos de conteúdo de método.

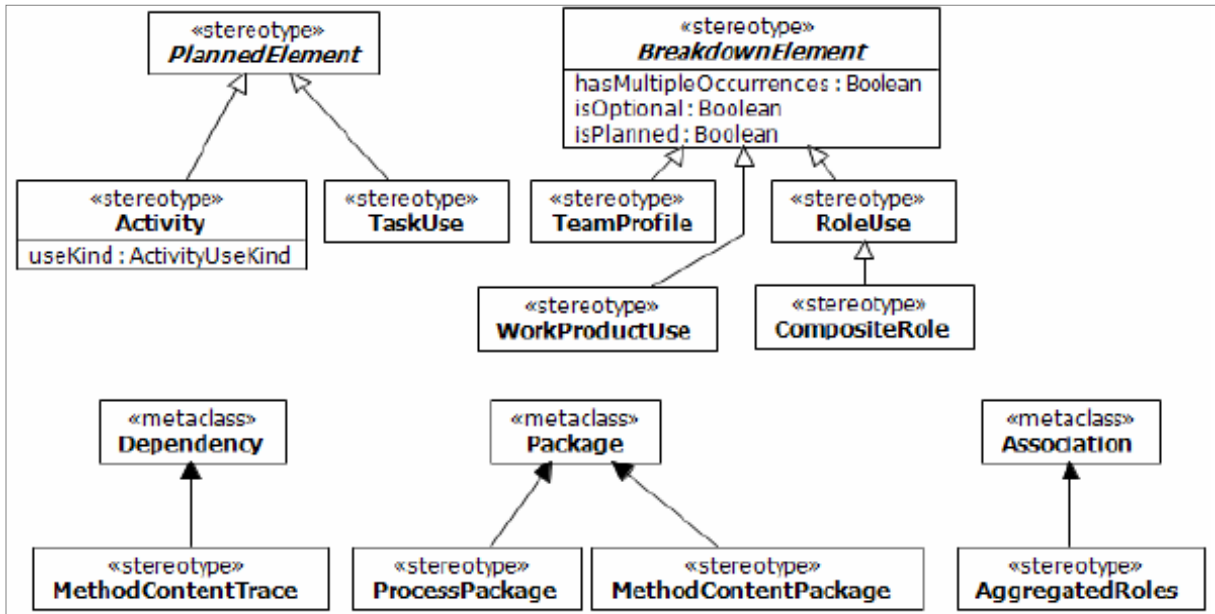


Figura 28 – Os estereótipos do SPEM 2.0 UML 2 Profile definidos no pacote *Process with Methods*. Fonte: OMG, 2008, p. 95.

A Figura 29 mostra as classes apresentadas no pacote *Process with Methods* para tornar possível esta separação. As classes *Method Content Element* (apresentada no pacote *Method Content*) e *Process Element* (apresentada no pacote *Process Structure*) passam a ser categorizadas com as classes *Method Content Kind* e *Process Kind*, respectivamente. Ao serem categorizados, os elementos das classes *Method Content Element* e *Process Element*, se tornam classes dos tipos *MethodContentPackageableElement* e *ProcessPackageableElement*, respectivamente. É possível, então, definir que os elementos de um Pacote de Conteúdo de Método (*MethodContentMethod*) ou um Pacote de Processo (*Process Package*) só podem conter elementos “empacotáveis” com os tipos respectivos.

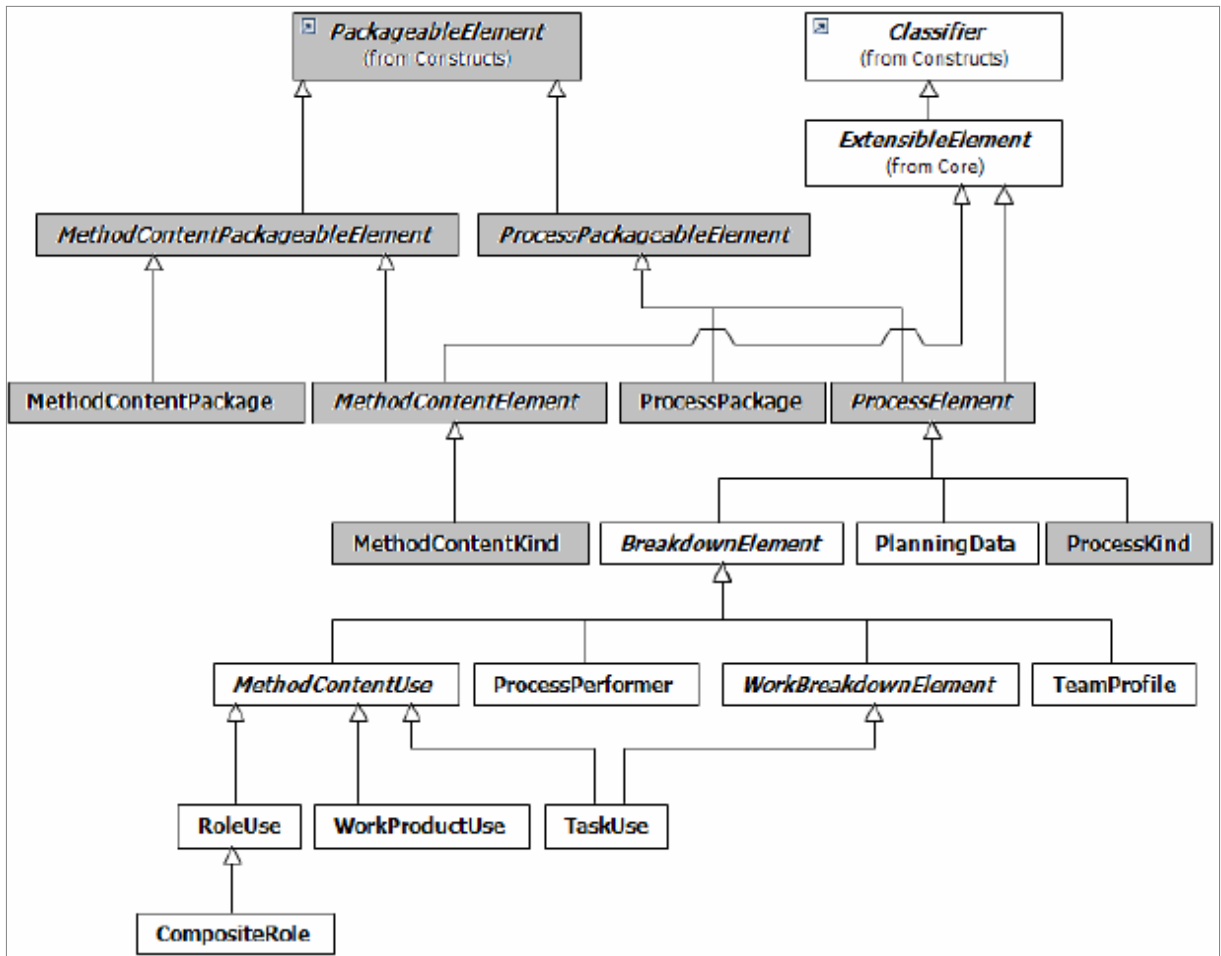


Figura 29 – Taxonomia das classes do meta-modelo definidas no pacote *Process with Methods*. Fonte: OMG, 2008, p. 96, meu sombreado.

A classe *Method Content Use* é uma classe abstrata que representa o principal conceito para permitir o uso conjunto de Conteúdo de Método (*Method Content*) e Processo (*Process*) no SPEM 2. Na Figura 30 vemos as três principais classes derivadas de *Method Content Use*: *TaskUse*, *WorkProductUse* e *RoleUse*.

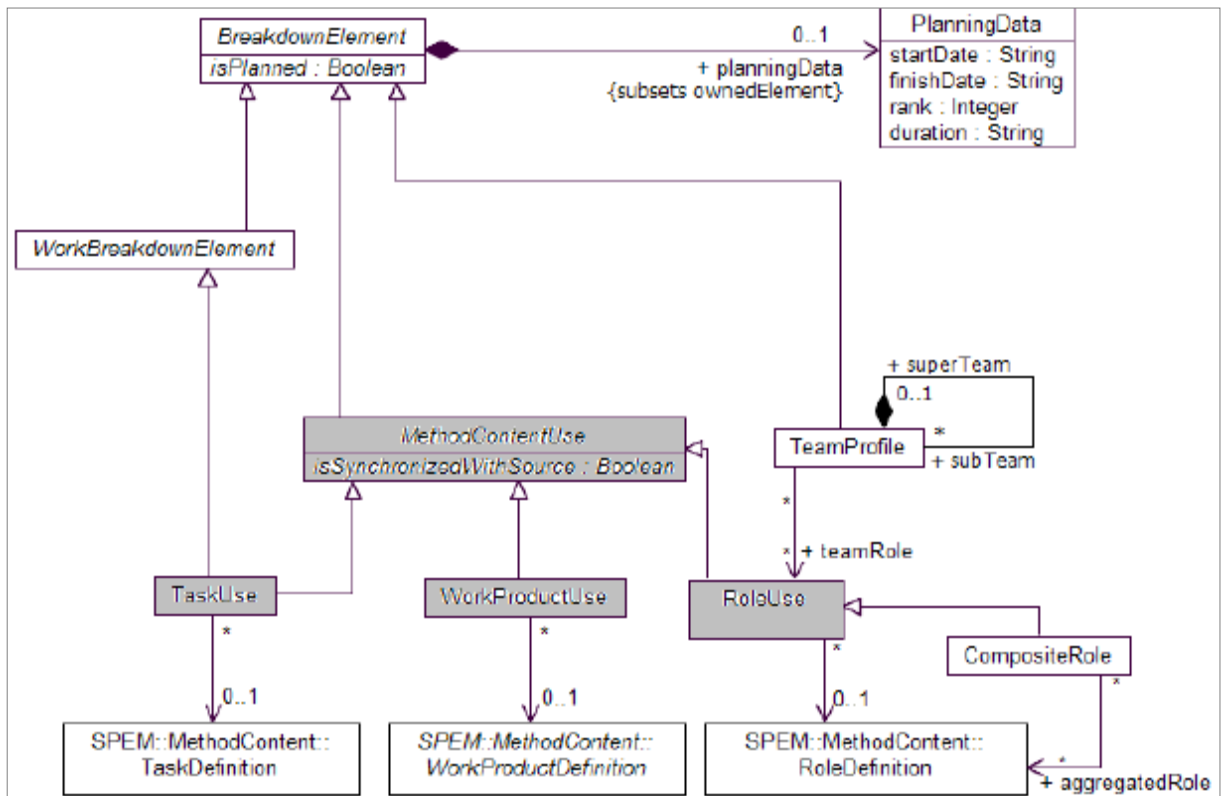


Figura 30 – Taxonomia e principais relacionamentos de *Breakdown*.

Fonte: OMG, 2008, p. 97, meu sombreado.

As classes derivadas de *Method Content Use* servem como intermediárias (*proxy*) entre as definições feitas nos pacotes *Method Content* e *Process Structure*. O exemplo da Figura 31 mostra como o relacionamento *<<content trace>>* é usado para ligar as classes destes pacotes.

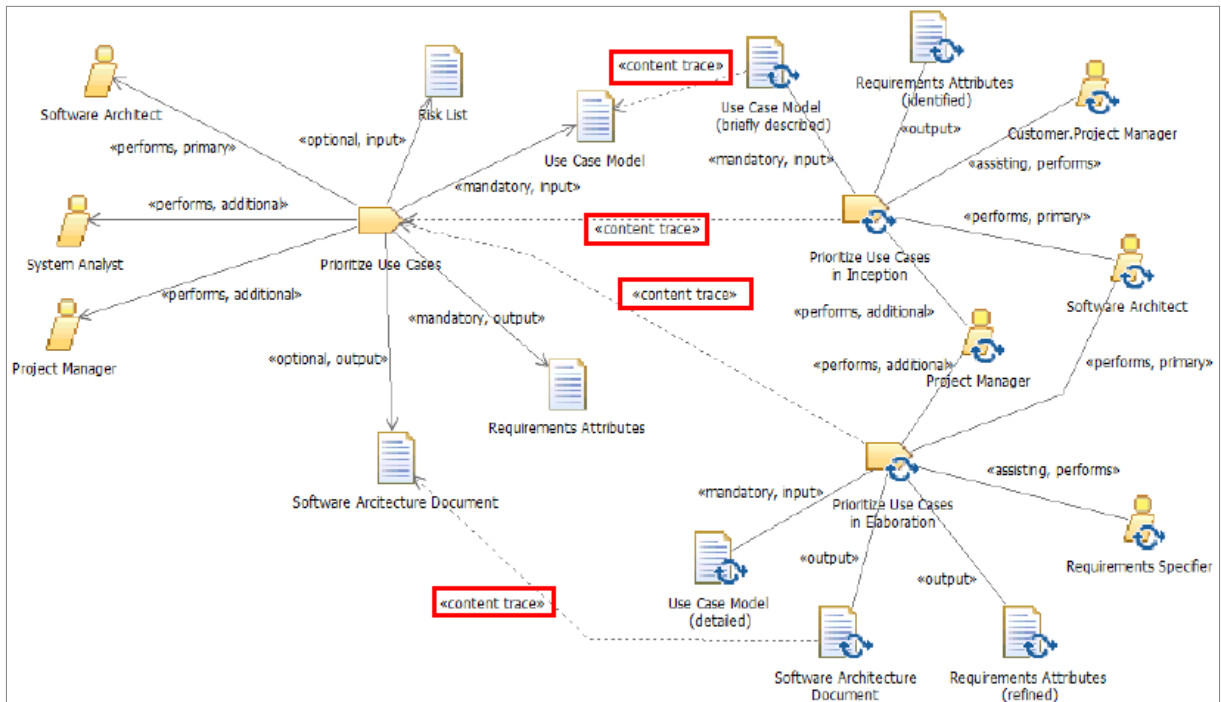


Figura 31 – Exemplos de *Method Content Use* (direita) referenciando *Method Content* (esquerda). Fonte: OMG, 2008, p. 105, meu destaque.

1.2.4.8 O pacote *Method Plug-in*

Para obter mais flexibilidade na criação de métodos, o conteúdo dos pacotes de conteúdo de método (*Method Content Packages*) e de processos (*Process Packages*) podem ser agrupados em uma unidade de reutilização chamada de *Method Plugin*. É possível então criar uma Biblioteca de Métodos (*Method Library*) apenas com os *Plugins* de Métodos (*Method Plugins*) que serão utilizados por uma organização. Além disso, é possível selecionar, dentro de uma Biblioteca de Métodos, quais os *Plugins* de Métodos (e quais de seus elementos), que serão válidos em determinadas situações. Esta seleção é chamada de Configuração de Métodos (*Method Configurations*) e também faz parte de uma Biblioteca de Métodos.

A Figura 32 mostra a estrutura geral de uma Biblioteca de Métodos (*Method Library*) destacando os seus elementos principais: *Method Plugins* e *Method Configurations*. Todos os elementos do SPEM 2.0 são guardados (*stored*) em *Method Libraries*. Uma *Method Library* é um *container* físico para *Method Plugins* e *Method Configurations*

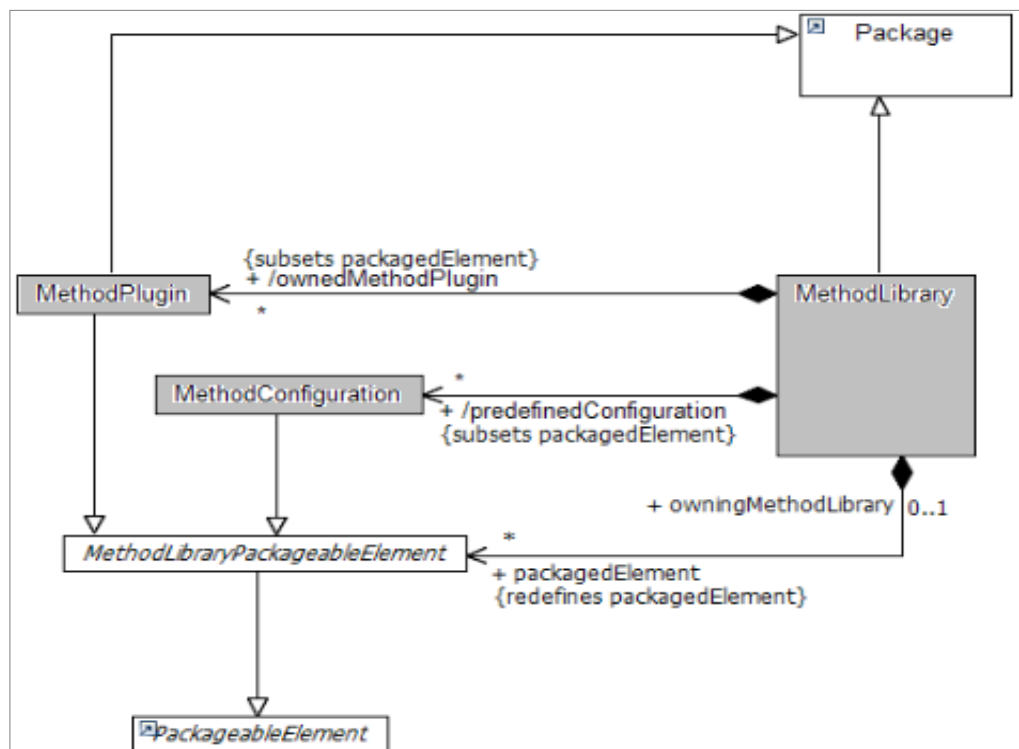


Figura 32 – Uma *Method Library* é um *container* para *Plugins* e *Configurations*.

Fonte: (OMG, 2008, p. 122, meu sombreado).

O pacote *Method Plugin* contém pacotes de conteúdo de métodos (*Method Content Packages*) e pacotes de processos (*Process Packages*). Na Figura 33 vemos a estrutura de um *Method Plugin*.

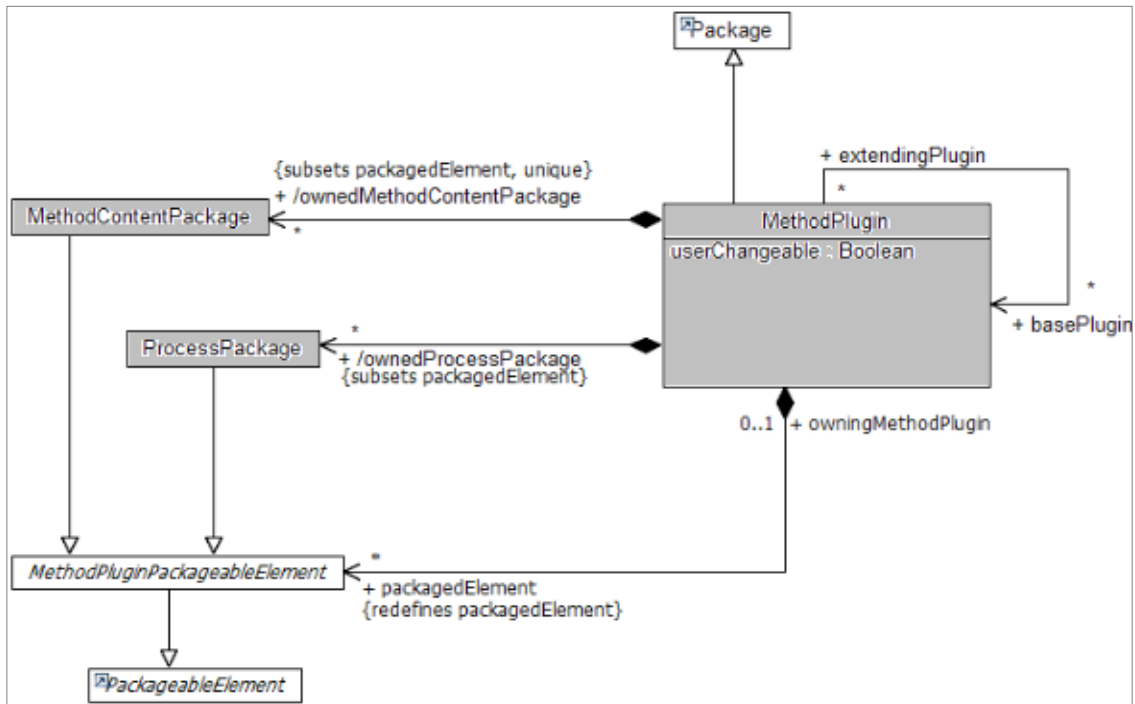


Figura 33 – *Method Plugins* são *containers* para pacotes de métodos

Fonte: OMG, 2008, p. 124, meu sombreado

Uma *Method Configuration* permite selecionar apenas os *Plugins* (com `/methodPluginSelection`) e os elementos específicos dentro de um *Plugin* (com `/contentPackageSelection` e `/processPackageSelection`) que serão levados em consideração durante o processo de criação de um novo método (*Delivery Process*, no SPEM). Além disso, é possível indicar que apenas os elementos de conteúdo de método (*Method Content Elements*) e os elementos de processo (*Process Elements*) que estão associados a certas categorias deverão estar visíveis durante o processo de criação. Isto pode ser visto na Figura 34.

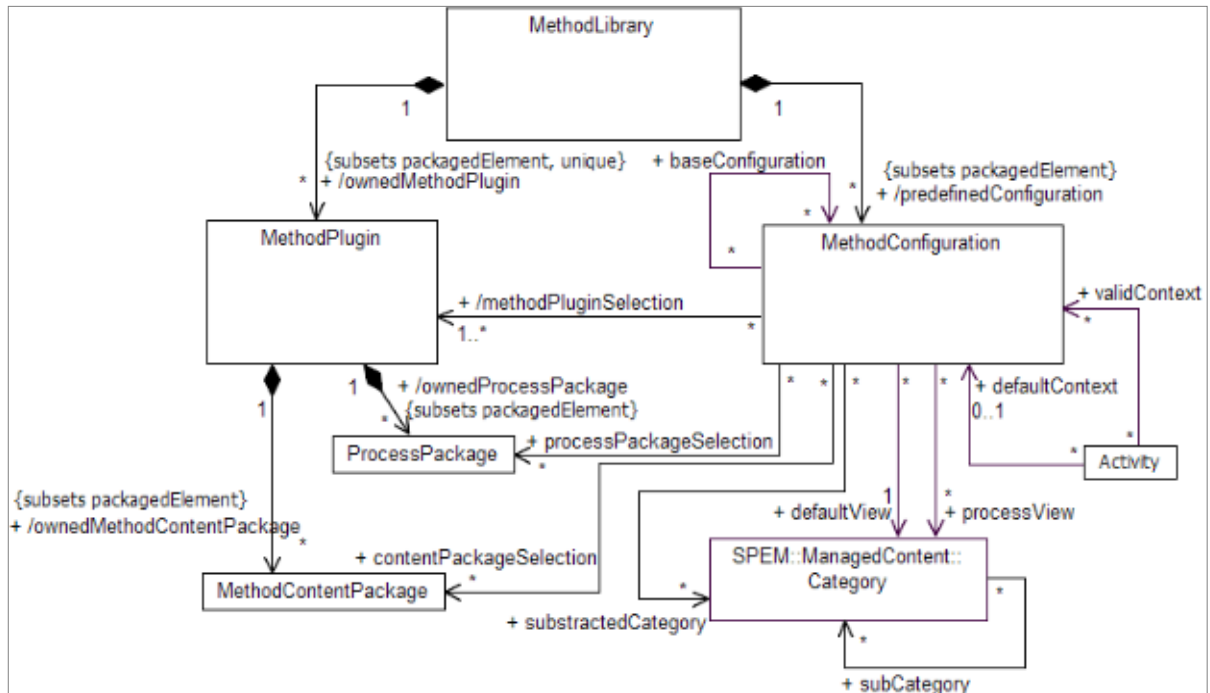


Figura 34 – Definição de *Method Library* e *Method Configurations*.

Fonte: OMG, 2008, p.120.

1.2.4.9 Considerações finais

O SPEM fornece um metamodelo cujo objetivo é alcançar os objetivos de flexibilidade e reutilização propostos pela Engenharia de Métodos Situacionais. Embora utilize uma terminologia um pouco diferente, é fácil reconhecer os conceitos da Engenharia de Métodos Situacionais. Para que seja utilizado é necessária uma ferramenta que faça a sua implementação. A seção seguinte apresenta o *Eclipse Process Framework Composer (EPF Composer)* que implementa o SPEM.

1.2.5 EPF (Eclipse Process Framework) Composer

O *Eclipse Process Framework (EPF) Composer* é uma ferramenta *freeware* desenvolvida no ambiente Eclipse. Pode ser usada para criar, armazenar (no formato XMI), recuperar e modificar componentes de métodos. Permite também a combinação dos componentes de método para a criação de *Delivery Processes*, isto é, de métodos completos.

Os componentes de métodos e os processos são baseados no meta-modelo SPEM. A Figura 35 é um exemplo de utilização do EPF.

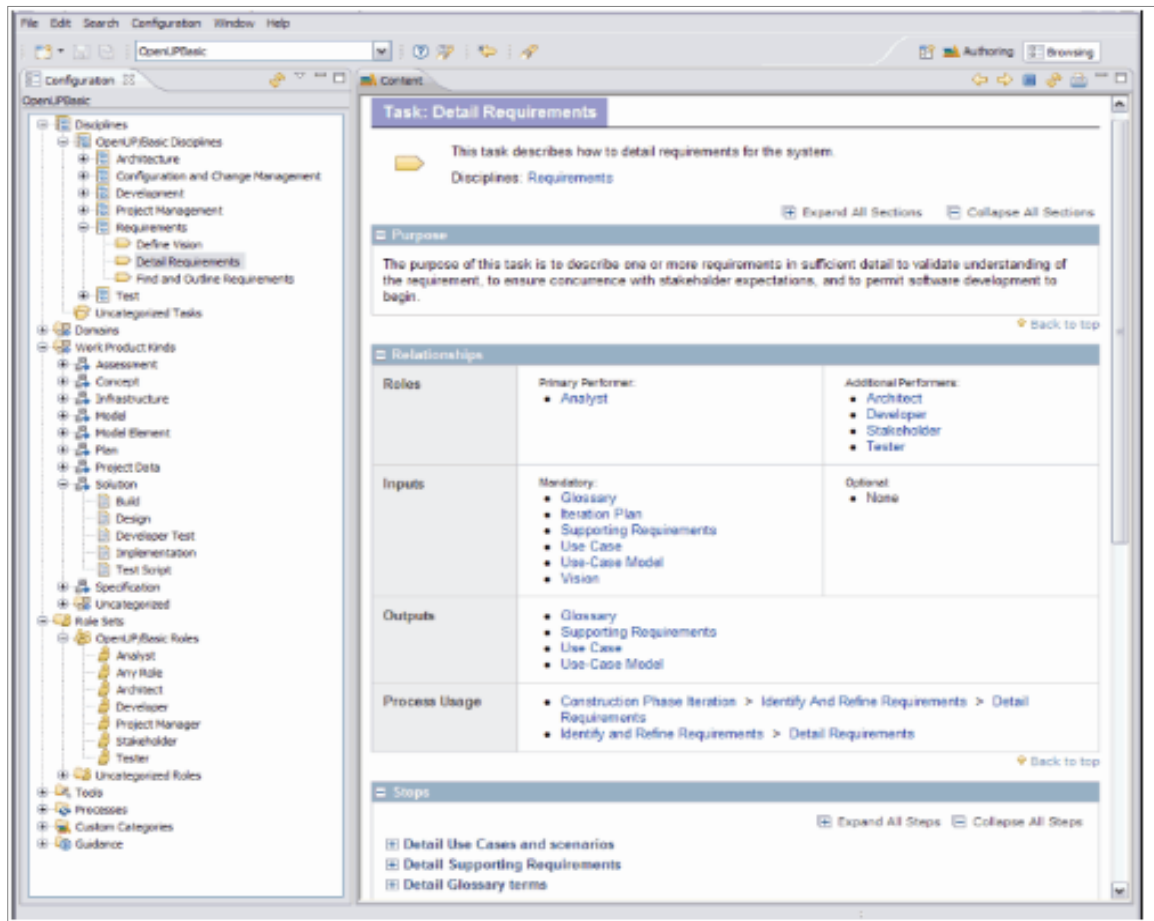


Figura 35 – EPF Composer: Representação de conteúdo do método.

Fonte: TUFT; 2010; pg 30.

Como se pode ver, uma *Task* (um tipo de componente de método que representa uma tarefa a ser realizada) pode ser descrita em detalhe. Uma *Task* tem uma Finalidade (“*Purpose*” na Figura 35), e Relacionamentos (“*Relationships*”, na Figura 35) com outros componentes (*Roles*, *Inputs*, *Outputs* e *Process Usage*).

Um Processo (outro tipo de componente de método) pode ser visto na Figura 36.

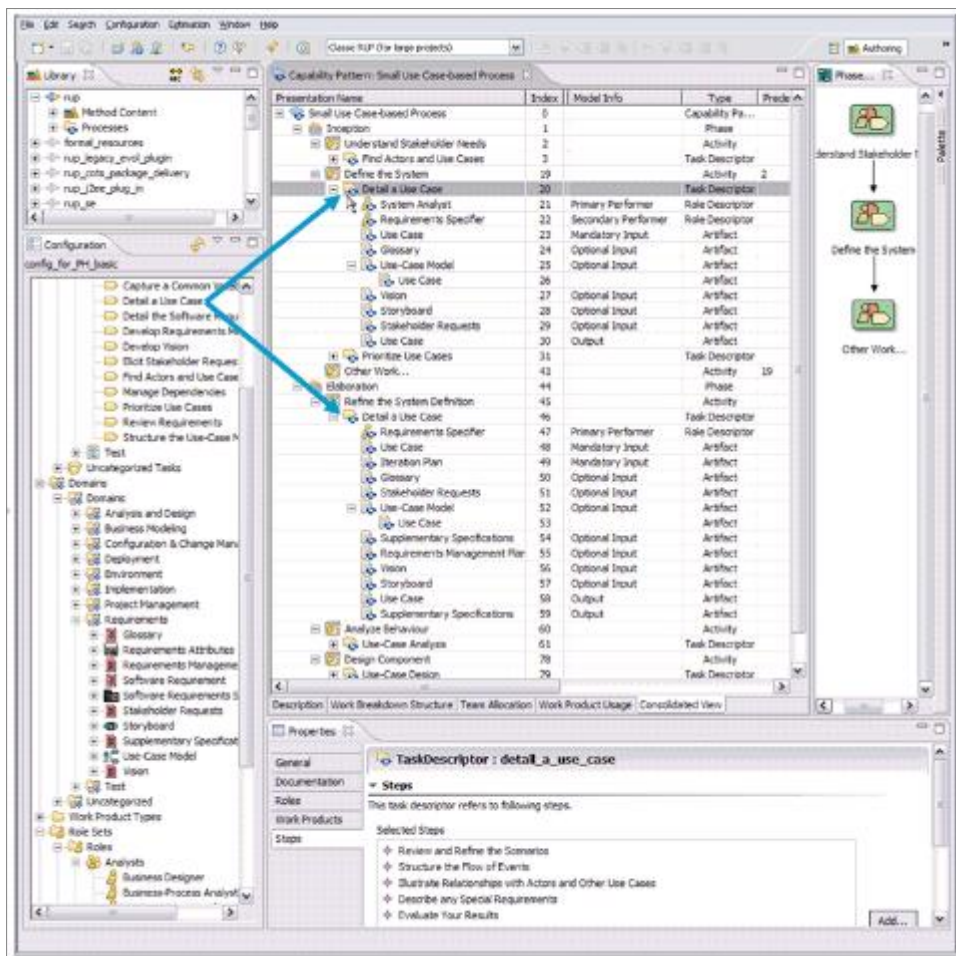


Figura 36 – EPF Composer: Representação do processo de autoria.

Fonte: TUFT; 2010; pg 31.

Além do ambiente para desenvolvimento dos componentes de métodos, o EPF permite a publicação, em formato HTML, de toda a documentação sobre o método, o que permite a sua utilização pelos desenvolvedores.

1.2.6 Atributos de um componente de método

A descrição de um componente de método pode incluir vários atributos. A Tabela 8 (COSSENTINO, 2006) resume algumas abordagens para especificar os atributos.

Tabela 8 - Comparação estrutural dos metamodelos

		Fragmento de método	Pedaço de método	Fragmento OPF	Fragmento FIPA
Processo	O quê (What)	Estágio, atividade, tarefa e passos		Atividade, tarefa	Atividade
	Entrada/Saída (input/output)		Situação (<i>input</i>), intenção (<i>output</i>)		Dados da atividade (<i>input/output</i>)
	Como (How)		Orientação (<i>guideline</i>), interface	Técnica	Orientação (<i>guideline</i>)
	Quando (When)			Estágio, montagem, fase	
	Quem (Who)			Produtor (direto, indireto)	<i>Role, Actor</i> (Realiza, ajuda)
	Reusabilidade e montagem	Relação com outros fragmentos	Descritor (ID, nome, tipo, domínio da aplicação)	Orientação (<i>guideline</i>)	Glossário, aspecto, orientação-de-composição, dependências do fragmento
Produto	Entregas, documentos dos marcos, modelos e diagramas	Orientação Produto-Parte	Planilha Modelo Documento	WorkProduct. MMMElement	
			Linguagem de modelagem		

Fonte: COSENTINO, 2006, p. 11.

É possível elaborar uma tabela semelhante para incluir novos modelos (Tabela 9). A tabela analisa as categorias da tabela anterior aplicadas aos modelos do EPF Composer, PMBoK e BABoK.

Tabela 9 - Componentes do EPF Composer, PMBoK e BABoK

	EPF Composer (*)	PMBoK	BABoK
Processo	O quê (What)	Tarefa, atividade, <i>capability-pattern</i> , processo	Elementos
	Entrada/Saída (input/output)	Entrada (<i>input</i>), saída (<i>output</i>)	
	Como (How)	Orientação (<i>guideline</i>), passos	Ferramentas e técnicas
	Quando (When)	Fase, iteração, atividade	Figura com Fluxo de dados (<i>data flow</i>)
	Quem (Who)	Papel (<i>role</i>)	<i>Stakeholders</i>
	Reusabilidade e montagem	Descritor de componente	Orientações (<i>guideline</i>)
Produto		Artefatos (<i>Artifacts</i>), resultados (<i>outomes</i>), entregas (<i>deliverables</i>)	Planilhas Modelos Documentos

(*) EPF Composer possui outros elementos, além dos que foram incluídos aqui.

1.3 Stakeholders

Embora os primeiros registros do atual conceito de *stakeholder* possam ser encontrados ainda nos anos 30, o termo só começou a ser usado, na área gerencial, na década de 60 (Freeman; 1984). Seu uso, no entanto, ficou restrito a alguns pesquisadores até que o livro de Freeman, “*Strategic Management: A Stakeholder Approach*” fosse lançado em 1984. Esta obra foi um verdadeiro divisor de águas sobre o assunto. Sua principal proposição era que um negócio deve se preocupar não só com os acionistas (*stockholders*) mas também com os *stakeholders*, isto é, “todos os grupos e indivíduos que podem afetar, ou serem afetados, pela realização do propósito da organização” (Freeman, 1984, p. 25). A empresa deveria reconhecer um conjunto de grupos muito maior do que era comum na época. Esta é uma definição muito abrangente e pode tornar difícil operacionalizar o envolvimento de todos os *stakeholders*. A solução é atribuir prioridades a eles e criar níveis de envolvimento de acordo com a importância de cada um. A Figura 37 mostra alguns exemplos de *stakeholders* e as setas indicam que a comunicação deve ocorrer nos dois sentidos.

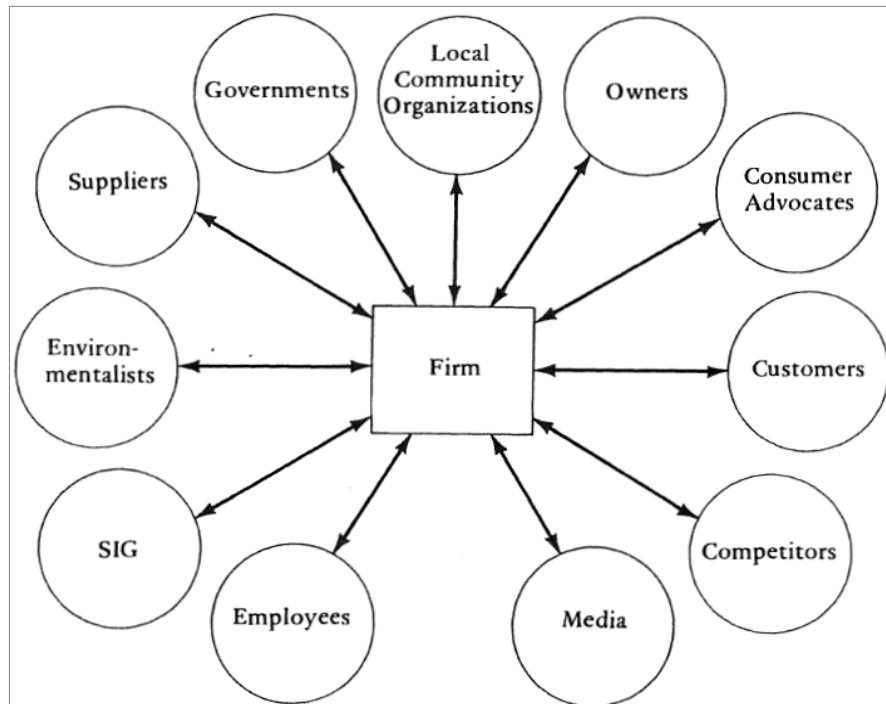


Figura 37 – Visão da firma por *stakeholders*. Fonte: (FREEMAN, 1984, pg 25).

1.3.1 Definição

Friedman e Miles (2006, p. 5-8) apresentam cinquenta e cinco definições, em ordem cronológica, criadas entre 1963 e 2003. Estas definições podem ser mais amplas ou mais restritivas, considerando-se a abrangência dos que aí seriam incluídos. Em um extremo, temos as definições que consideram apenas os acionistas (*stockholders*) como sendo de interesse para a empresa e para os seus gerentes (Friedman, 1970). No outro, podemos considerar como *stakeholders* até mesmo a Natureza (meio ambiente) e seres inexistentes, como as futuras gerações.

A literatura sobre *stakeholders* cresceu muito após o livro de Freeman (1984). As diversas definições e abordagens levaram Donaldson e Preston (1995) a dividir a literatura sobre *stakeholders* em três categorias: descritiva, instrumental e normativa. A abordagem descritiva ocorre quando é feita uma descrição da maneira como os gerentes e empresários se relacionam com os *stakeholders*. Ela evita emitir juízos de valor e pretende ser objetiva.

Na abordagem instrumental o objetivo é propor maneiras para que os objetivos da organização possam ser alcançados da forma mais eficiente possível através do gerenciamento

das relações com os *stakeholders*. Por exemplo, como se antecipar às possíveis queixas e evitar conflitos.

Finalmente, na abordagem normativa, o objetivo é estabelecer como devem ser as relações com os *stakeholders* a partir de posições éticas. O *stakeholder* não é visto como um meio para atingir um fim (abordagem instrumental), nem como um objeto neutro a ser estudado objetivamente mas sim como possuidor de um valor intrínseco. O *stakeholder* deve ser levado em consideração porque isto é o “certo”.

Donaldson e Preston consideram que as três abordagens estão relacionadas. A abordagem fundamental é a normativa, pois serve para justificar as outras duas. Esta dissertação utiliza uma abordagem instrumental, pois o nosso objetivo é criar um *framework* para a criação de métodos de envolvimento de *stakeholders*.

1.3.2 Níveis de envolvimento

O envolvimento dos *stakeholders* pode ocorrer em vários níveis da organização e até fora dela: projeto, unidade de negócio, empresa ou órgão público, ambiente local, regional, nacional e até mesmo internacional. A política de participação deve ser compatível com o nível considerado.

O objetivo final do Modelo de Envolvimento de *Stakeholders* é fornecer orientação sobre o tipo de envolvimento que deverá ser adotado para cada tipo de *stakeholder*. Para isto é necessário definir os tipos de envolvimento possíveis.

Arnstein (Arnstein, 1967) propõe um modelo de participação com oito níveis crescentes de poder de decisão para os *stakeholders*. Este modelo é representado por uma escada, como na Figura 38, onde os degraus indicam que os níveis de participação podem ser mais altos ou mais baixos.

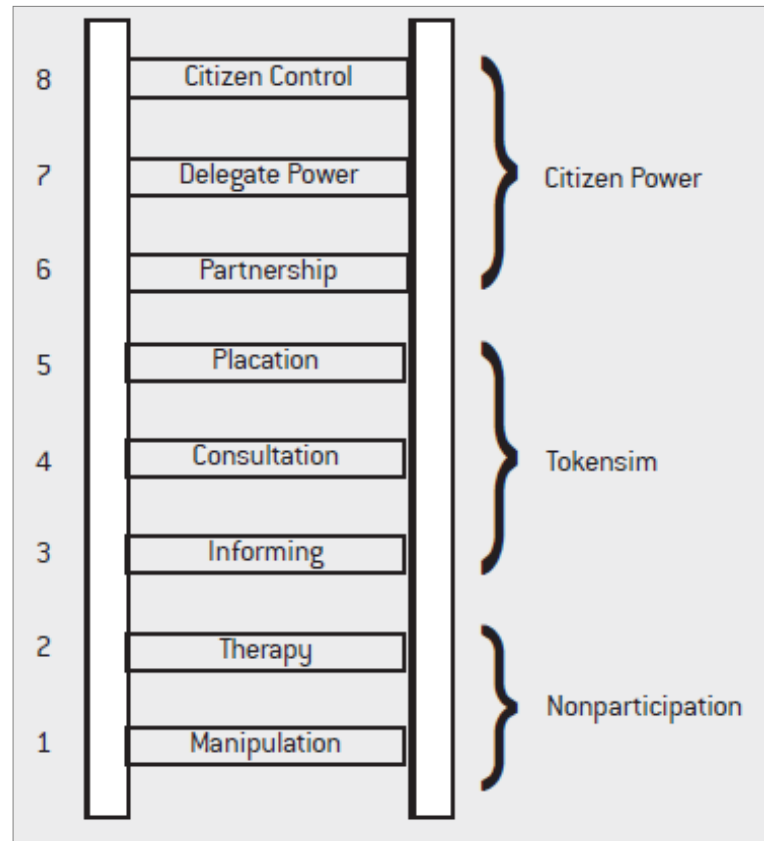


Figura 38 – Oito degraus na escada de participação dos cidadãos.

Fonte: ARNSTEIN, 1967, p.217, apud CANADA; 2006, p. 9.

Nos degraus inferiores temos uma “não-participação”. A Manipulação e a Terapia tem como objetivo “curar” os *stakeholders* das suas pretensões. Nos níveis 3 e 4, temos o que Arnstein chama de “tokenism”, isto é, uma participação apenas simbólica na qual os cidadãos podem falar e ser ouvidos, porém sem poder para que suas opiniões sejam acatadas. No nível 5 (*Placation*) a participação ainda é simbólica, porém o cidadão pode se expressar de modo mais amplo, mas apenas para que se acalme. No nível 6 (parceria), já é possível haver algumas “trocas” com os poderosos. Finalmente, nos níveis mais altos (7 e 8) os cidadãos podem tomar decisões que serão acatadas.

Arnstein (1967) enfatiza que seu modelo é uma simplificação da realidade, pois “no mundo real das pessoas e programas, pode haver 150 degraus com distinções menos nítidas e ‘puras’ entre eles”. Mesmo assim fica claro que o poder de participação e de decisão não é igual para todos.

Friedman e Miles (2006) propõem uma “escada” com mais níveis. A complexidade do seu modelo (Figura 39), no entanto, o torna mais difícil de ser utilizado.

Stakeholder management tool and nature of response		Intention of engagement	Level of influence	Style of dialogue and associated examples
Degrees of Stakeholder power	12. Stakeholder control	Majority representation of stakeholders in decision-making process	Forming or agreeing to decisions	Multi-way dialogue, e.g. community projects
	11. Delegated power	Minority representation of stakeholders in decision-making process		Multi-way dialogue, e.g. board representation
	10. Partnership	Joint decision-making power over specific projects		Multi-way dialogue, e.g. joint ventures
Degrees of involvement	9. Collaboration	Some decision-making power afforded to stakeholders over specific projects	Having an influence on decisions	Multi-way dialogue, e.g. strategic alliances
	8. Involvement	Stakeholders provide conditional support; if conditions are not met support is removed. The organization decides the extent of conformity		Multi-way dialogue, e.g. constructive dialogue
	7. Negotiation			Multi-way dialogue, e.g. reactive: bargaining
Degrees of tokenism	6. Consultation	Organization has the right to decide. Stakeholders can advise.	Being heard before a decision	Two-way dialogue, e.g. questionnaires, interviews, focus groups, task forces, advisory panels
	5. Placation	Appease the stakeholder. Stakeholders can hear and be heard, but have no assurance of being heeded by the organization		
	4. Explaining	Educate stakeholders		
Non-participation	3. Informing	Educate stakeholders	Knowledge about decisions	One-way dialogue, e.g. verified corporate social reports
	2. Therapy	'Cure' stakeholders of their ignorance and preconceived beliefs		One-way dialogue, e.g. briefing sessions, leaflets, magazines, newsletters, green glossy social corporate reports, or other publications
	1. Manipulation	'Misleading' stakeholders, attempting to change stakeholder expectations		

Figura 39 – Uma escada de gerenciamento e envolvimento de *stakeholders*.

Fonte: FRIEDMAN e MILES, 2006, p. 162.

Um modelo muito utilizado é o da International Association for Public Participation (IAP2). Neste modelo, os níveis inferiores do modelo de Arnstein não aparecem, pois não são considerados como sendo um nível de participação. Os níveis superiores foram consolidados em “Autorizar” (*Empower*). A Tabela 10 mostra o espectro de participação que pode ocorrer.

Tabela 10 - Níveis de Impacto Público do IAP2

===== Níveis crescentes de impacto público =====>				
Informar	Consultar	Envolver	Colaborar	Autorizar
Objetivo da Participação Pública	Objetivo da Participação Pública	Objetivo da Participação Pública	Objetivo da Participação Pública	Objetivo da Participação Pública
Fornecer ao público informações objetivas e equilibradas para ajudá-lo a compreender os problemas, alternativas e/ou soluções	Obter <i>feedback</i> público sobre análises, alternativas e decisões	Trabalhar diretamente com o público durante todo o processo para garantir que problemas (<i>issues</i>) e preocupações públicas sejam compreendidos e considerados de forma consistente.	Criar uma parceria com o público a respeito de todos os aspectos da decisão inclusive desenvolvimento de alternativas e identificação da solução preferida.	Colocar a decisão final nas mãos do público
Promessa ao público	Promessa ao público	Promessa ao público	Promessa ao público	Promessa ao público
Iremos mantê-lo informado	Iremos mantê-lo informado, ouvir e confirmar preocupações e fornecer <i>feedback</i> sobre como o público influenciou a decisão	Iremos trabalhar com vocês para garantir que seus problemas (<i>issues</i>) e preocupações estejam refletidos diretamente nas alternativas desenvolvidas e para fornecer <i>feedback</i> sobre como o público influenciou a decisão.	Iremos procurá-los para conselhos e inovações na formulação de soluções e incorporar, no máximo possível, seus conselhos e recomendações nas decisões.	Iremos implementar o que vocês decidirem.
Exemplo de ferramentas	Exemplo de ferramentas	Exemplo de ferramentas	Exemplo de ferramentas	Exemplo de ferramentas
<ul style="list-style-type: none"> · Fact sheets · Web sites · Open houses 	<ul style="list-style-type: none"> · Comentário público · Focus group · Pesquisas · Reuniões públicas 	<ul style="list-style-type: none"> · Oficinas · Pesquisa deliberada 	<ul style="list-style-type: none"> · Comitê Consultivo de Cidadãos · Construir consenso · Tomada de decisão participativa 	<ul style="list-style-type: none"> · Juri de cidadãos · Votos · Decisões delegadas

Fonte: IAP2, 2000.

1.3.3 Atributos dos stakeholders

Para classificar os *stakeholders* é necessário definir quais os atributos (ou combinação de atributos) que devem ser levados em consideração durante o processo de classificação. Estes atributos podem ser usados de forma isolada ou em combinações específicas de dois,

três ou mais atributos. Além disso, os atributos podem ser dicotômicos, podem ter alguns valores discretos (dentro de um conjunto pré-definido) ou podem ter uma variação contínua dentro de uma faixa de valores.

O poder é um atributo presente (isoladamente ou em conjunto com outros) em quase todas as classificações de *stakeholders*. A definição de poder é um assunto muito complexo. São mostradas apenas algumas definições associadas com classificações de *stakeholders*.

Uma definição de poder é “a habilidade de indivíduos ou grupos de persuadir, induzir ou coagir os outros a seguir determinados cursos de ação” (JOHNSON et al., 2008, p. 160). A Figura 40 mostra algumas fontes de poder, dentro e fora da organização.

Fontes de poder

Dentro das organizações

- Hierarquia (poder formal),
por exemplo, tomada de decisão autocrática
- Influência
por exemplo, liderança carismática
- Controle de recursos estratégicos,
por exemplo, produtos estratégicos
- Possuir conhecimentos e habilidades,
por exemplo, especialistas em computadores
- Controle do ambiente humano,
por exemplo, habilidades de negociação
- Envolvimento na implementação de estratégia
por exemplo, exercer poder de escolha

Para *stakeholders* externos

- Controle de recursos estratégicos,
por exemplo, materiais, trabalho, dinheiro
- Envolvimento na estratégia de
implementação,
por exemplo, rede de distribuição, agentes
- Possuir conhecimentos e habilidades,
por exemplo, subcontratados, parceiros
- Através de ligações internas,
por exemplo, influência informal

Figura 40 – Fontes de poder. Fonte: JOHNSON et al., 2008, p. 161.)

Como muitas vezes um conhecimento preciso destas fontes de poder não está disponível. Um recurso, então, é usar indicadores de poder, como os da Figura 41.

Indicadores de poder

Dentro das organizações

- Status
- Reivindicação sobre recursos
- Representação
- Símbolos

Para *stakeholders* externos

- Status
- Dependência de recursos
- Acordos de negociação
- Símbolos

Figura 41 – Indicadores de poder. Fonte: JOHNSON et al., 2008, p. 161.

Mitchell et al. (1997) adotam uma definição de poder em que esse é definido como “A relação entre os atores sociais em que um ator social, A, pode entrar como um outro ator social, B, para fazer algo que B não teria feito de outra forma” (MITCHELL et al., 1997, p. 869). As bases para que este poder exista podem ser de três tipos: coercitivas, utilitárias e

normativas. Uma base coercitiva utiliza a força, ou a ameaça de usar a força. A base utilitária usa recursos materiais como bens ou dinheiro. A base normativa utiliza recursos simbólicos como prestígio e estima.

Um atributo importante dos *stakeholders* é a legitimidade. A palavra legitimidade vem de legítimo que, por sua vez, se refere ao que está conforme a lei (o radical de legitimidade e de legítimo vem da palavra latina *legis*, que é o genitivo singular de *lex*, que significa lei). Esta lei pode ser baseada “no direito, na razão ou na justiça” (Ferreira, 1986, p. 1017). A legitimidade baseada no Direito pode ser comprovada através de contratos. Desta forma um acionista é um *stakeholder* legítimo. Da mesma maneira um fornecedor possui um contrato que obriga a organização a pagar o que foi acordado. A legitimidade baseada no razão e na justiça é mais difícil de estabelecer, porém se baseia no que a sociedade considera como sendo razoável e justo.

Em relação ao poder, pode-se observar desde logo que um *stakeholder* pode ter legitimidade sem ter poder. Inversamente pode ter poder sem ter legitimidade.

Outro atributo que pode ser usado pelos gerentes para classificar os *stakeholders* é a urgência. Urgente pode ser definido como:

urgente. [Do latim *urgente*.] Adj. 2 g. **1.** Que urge; que é necessário ser feito com rapidez. **2.** Indispensável, imprescindível. **3.** Iminente, impendente. (FERREIRA, 1986, p. 1741) .

A primeira explicação para definição de urgente enfatiza o aspecto temporal, enquanto a segunda enfatiza a criticidade. Os dois aspectos são necessários para caracterizar a urgência.

Estes dois aspectos são considerados também por Mitchell et al. (Mitchell et al., 1997) onde a urgência se caracteriza por estar (1) relacionada com o tempo e (2) ser importante, ou crítica, para o *stakeholder*. Não basta apenas estar relacionada com o tempo é necessário também que a criticalidade esteja presente. Estes dois aspectos da urgência estão representados graficamente na Figura 42 (Mitchell et al., 2011, p. 246) onde se pode observar que a urgência só está presente quando está relacionada com o tempo (*Time Sensitivity*) e é crítica (*Criticality*).

		Time	Sensitivity
		YES	NO
Criticality	YES	1. Urgency present	2. Long-term important
	NO	3. Short-term unimportant	4. Urgency absent

Figura 42 – Urgência como função de criticalidade e sensibilidade ao tempo nos negócios em geral. Fonte: MITCHELL et al., 2011, p. 246.

A organização pode favorecer ou prejudicar (de forma real ou apenas aparente) um grupo de *stakeholders*. É importante identificar para cada grupo, se ele está contra, é a favor ou é neutro em relação a uma questão. Isto irá influenciar o modo de envolvimento a ser adotado em relação a ele. Mesmo quando o *stakeholder* não se manifesta abertamente é necessário determinar o potencial que ele tem para ser contra ou a favor e a intensidade de sua posição.

Schmeer

Schmeer (Schmeer, 2000) propõe uma classificação quanto ao posicionamento com cinco níveis: (1) apoio, (2) apoio moderado, (3) neutro, (4) oposição moderada e (5) oposição. Os níveis são obtidos a partir de entrevistas. Cada entrevistado avalia a sua posição e, indiretamente, a posição de outros *stakeholders*. No cálculo final, são levadas em consideração a média das avaliações feitas pelas outros e a autoavaliação. O objetivo deste procedimento é garantir que o posicionamento será o mais objetivo possível.

Savage

Savage e outros (SAVAGE et al., 1991) propõe um modelo baseado no potencial para ameaça e no potencial para cooperação.

1.3.4 Classificação dos Stakeholders

Os *stakeholders* podem ser classificados em função da proximidade que têm em relação a questão considerada no projeto. A representação gráfica é feita com círculos ou elipses concêntricas e pode variar dependendo da área do projeto.

Business Analysis Book of Knowledge (BABOK)

O ponto de vista do BABoK (IIBA, 2008) é a análise de negócio. A Figura 43 mostra os tipos de *stakeholders* em função da proximidade com a solução a ser fornecida.

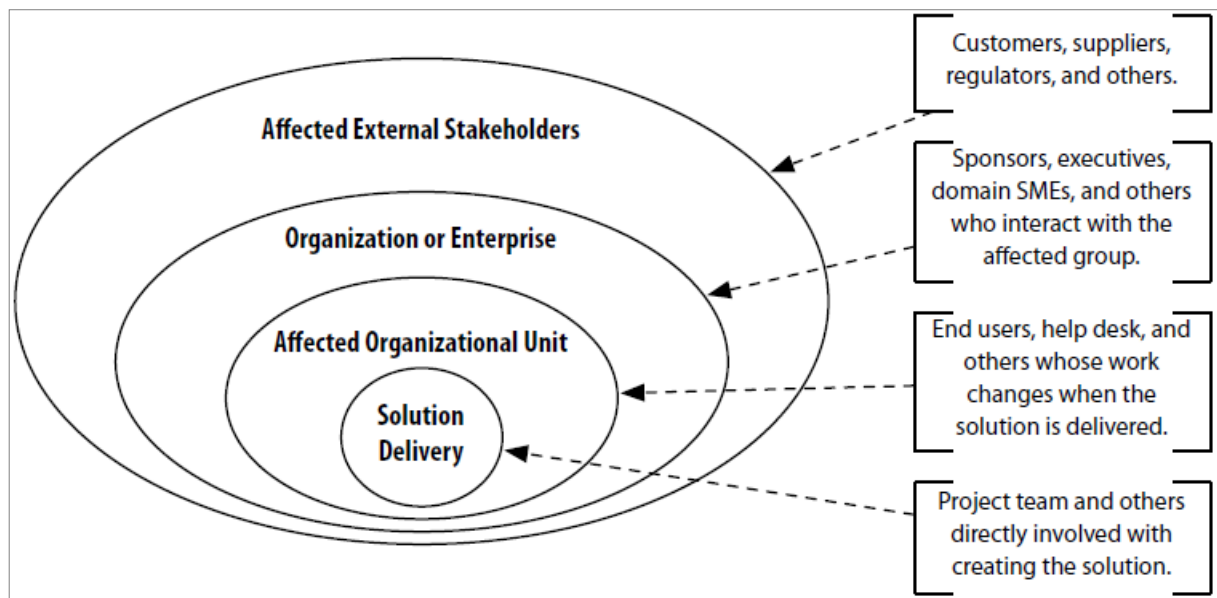


Figura 43 – Diagrama de cebola de *stakeholders*.

Fonte: IIBA, 2008, p. 30.

Alexander et al., (2009) definem que o ponto de vista é um pouco diferente, porém a ideia de distância permanece a mesma. Isto pode ser visto na Figura 44.

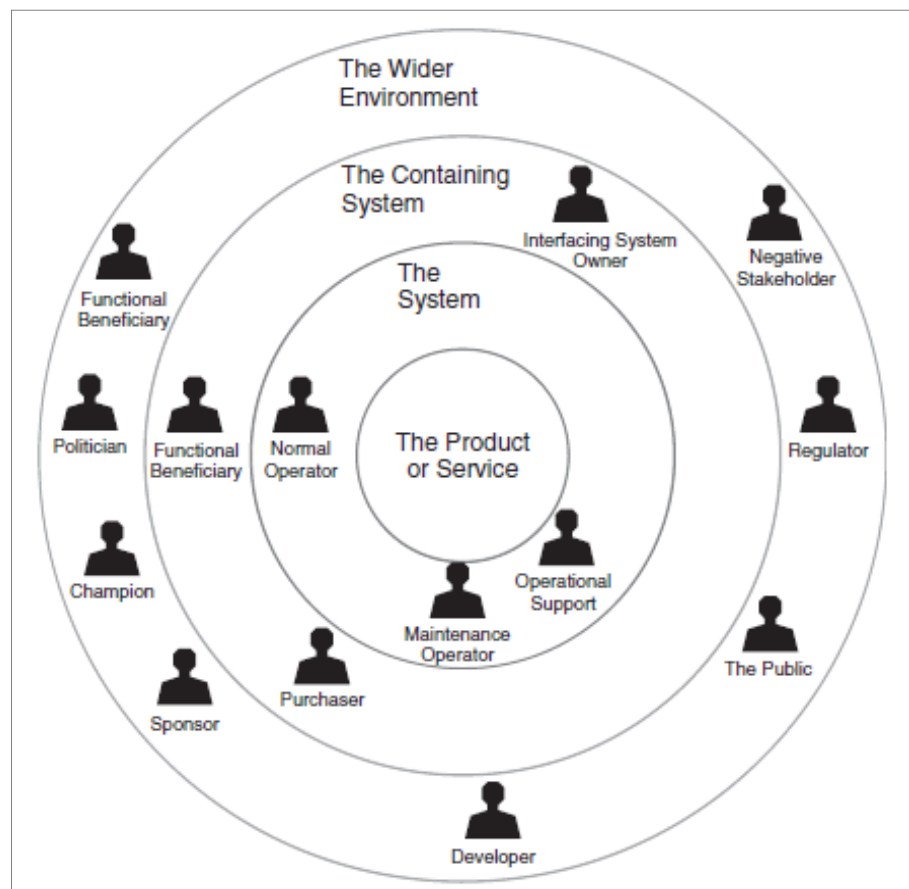


Figura 44 – Modelo de 'cebola' dos *stakeholders* em um sistema típico.

Fonte: ALEXANDER; BEUS-DUKIC, 2009, p. 29.

Os atributos examinados anteriormente podem ser combinados, dois a dois, em uma matriz. Com isto é possível criar classificações os quatro elementos resultantes.

Combinando o potencial para ameaças com o potencial para cooperação Savage (2001) propõe que os *stakeholders* sejam classificados como na Figura 45. Para cada tipo de *stakeholder* é sugerida uma estratégia de envolvimento.

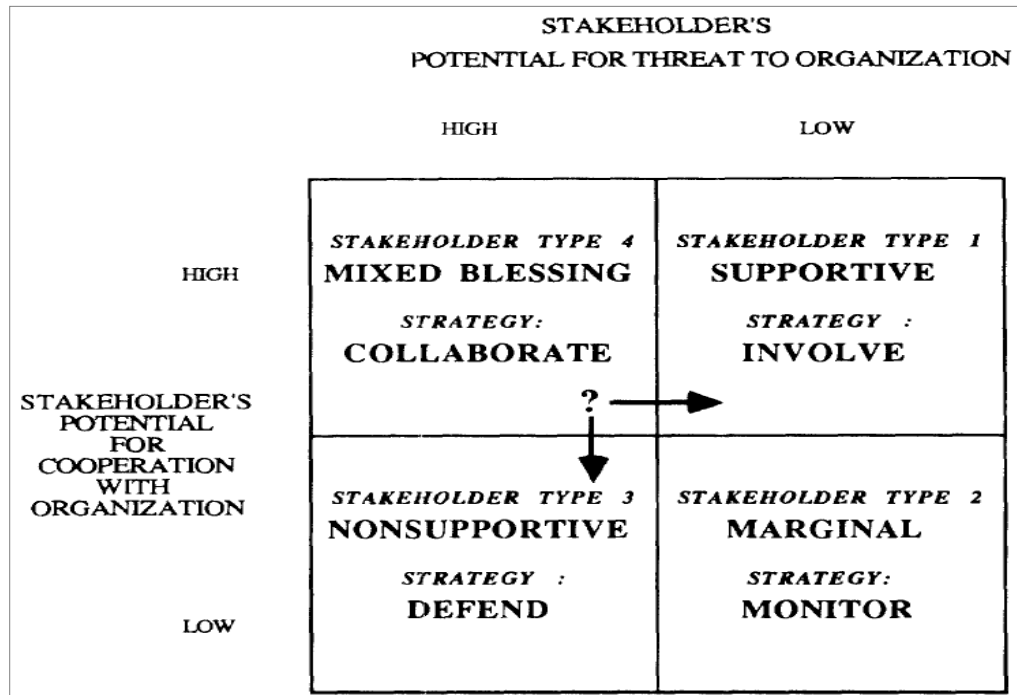


Figura 45 – Tipologia de diagnóstico de *stakeholders* organizacionais.

Fonte: SAVAGE et al., 1991, p. 65.

Outra classificação muito comum na literatura (IIBA, 2008) é a que combina influência (poder) e impacto (interesse). A Figura 46 mostra um exemplo. O texto dentro das células indica o tipo de envolvimento recomendado para cada combinação de atributos.

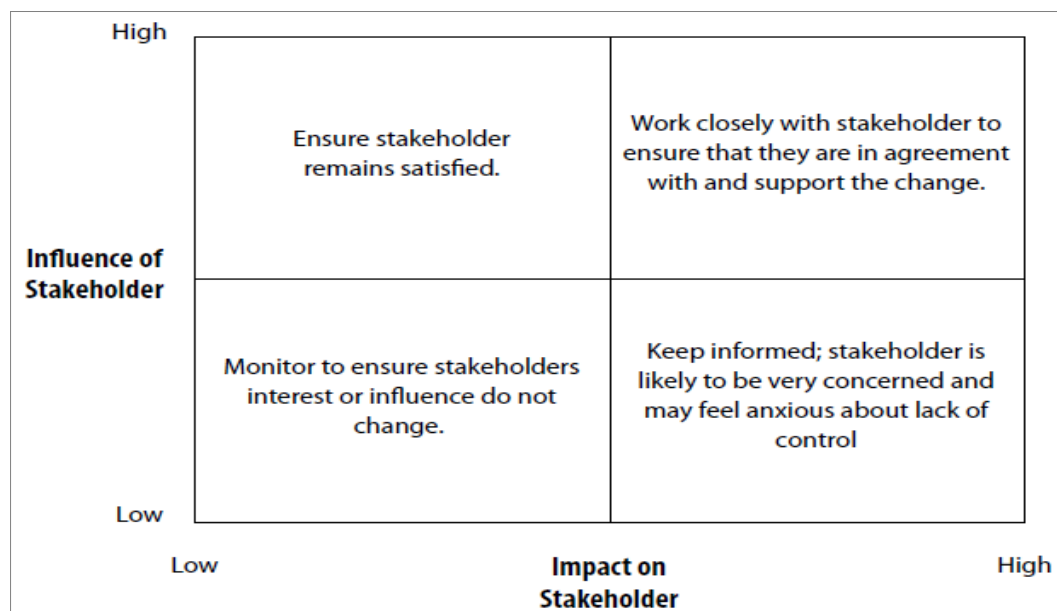


Figura 46 – Matriz de *stakeholders*. Fonte: IIBA, 2008, p. 30.

Outras classificações fazem combinações de atributos não estando limitadas a somente dois atributos. Com o emprego de três atributos é possível criar modelos com 8 tipos de *stakeholders*. Um exemplo disto é o modelo de Mitchell (1997).

A classificação de *stakeholders*, seguindo o modelo de Mitchell (1997) é uma das mais citadas e utilizadas das classificações. Neste modelo são considerado, simultaneamente, os atributos de poder, legitimidade e urgência. Neste modelo (Figura 47) os atributos ou estão presentes ou não estão, ou seja, embora existam vários níveis de poder, urgência e legitimidade, considera-se apenas se eles estão presentes ou não. Quando o *stakeholder* possui apenas um atributo (tipos 1, 2 e 3) ele é considerado como “latente” e a sua importância para os gerentes não é muito grande. Quando passa a ter dois atributos (tipos 4,5 e 6) a sua importância aumenta. Finalmente, quando possui três atributos (tipo 7) trata-se de um *stakeholder* que não pode ser ignorado.

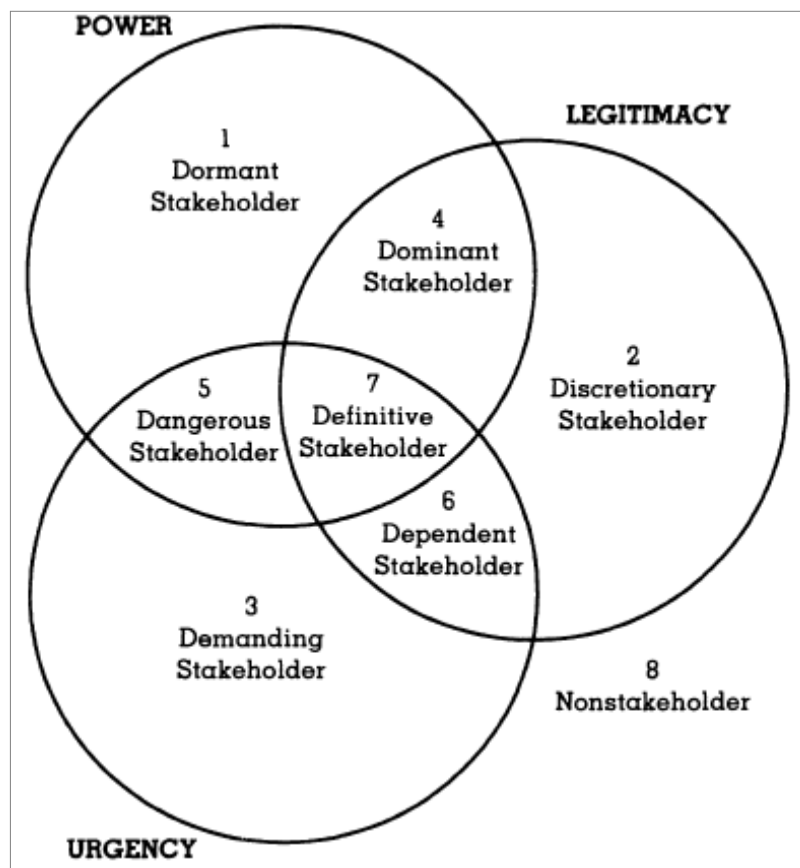


Figura 47 – Tipologia de *stakeholders*: Um, dois e três atributos presentes.

Fonte: MITCHELL et al., 1997, p. 872.

1.3.5 Arquitetura corporativa e TOGAF

A literatura de *Enterprise Architecture (EA)* também trata dos *stakeholders*. Togaf define um *stakeholder* como sendo “Um indivíduo, equipe ou organização (ou classes dele) com interesses em, ou preocupações em relação ao, resultado da arquitetura” (TOGAF, 2009, 38)

Esta definição embora possa ser considerada semelhante à de Freeman (Freeman, 1984), é mais restritiva, pois não inclui os que são afetados pelos efeitos colaterais da arquitetura, apenas os que são afetados pelo resultado da arquitetura. A “arquitetura”, na definição, se refere ao processo geral relativo à arquitetura como se pode depreender da observação: “Gestão de *Stakeholder* é um disciplina importante que profissionais de arquitetura de sucesso podem usar para ganhar o apoio dos outros. Que os ajuda garantir que seus projetos tenham sucesso onde outros falham.” (TOGAF, 2009, 282)

O gerenciamento deve começar na Fase A (Visão da Arquitetura). Durante as outras etapas estão previstas revisões da abordagem definida inicialmente tanto quanto à identificação dos *stakeholders* quanto à maneira como serão envolvidos no projeto.

As atividades relacionadas ao gerenciamento de *stakeholders* são: (i) identificação, (ii) classificação, (iii) abordagem para cada categoria de *stakeholder* e (iv) definição dos artefatos a serem utilizados por cada categoria.

A recomendação é que a identificação inicial dos *stakeholders* seja feita através de um *brainstorm*. Entre os participantes sugeridos estão os executivos de alto nível, o órgão de gerenciamento de projetos, desenvolvedores, parceiros e outros. Como critério para a escolha as seguintes perguntas podem servir de orientação (Togaf, 2009, 283): (i) Quem ganha e quem perde com a mudança?, (ii) Quem controla o gerenciamento de mudança de processos?, (iii) Quem projeta os sistemas novos?, (iv) Quem toma as decisões?, (v) Quem pesquisa sistemas de TI e quem decide o que comprar?, (vi) Quem controla os recursos?, (vii) Quem possui habilidades especializadas necessárias ao projeto?, e (viii) Quem tem influência?

Além de pessoas em cargos formais é preciso identificar os influenciadores. É importante chegar aos nomes das pessoas (e não apenas os nomes de áreas envolvidas) pois haverá a necessidade de contato individualizado.

O Togaf (2009) distingue cinco grandes categorias de *stakeholders* (Figura 48):

Funções corporativas, organização do usuário final, organização do projeto, operações do sistema e *stakeholders* externos.

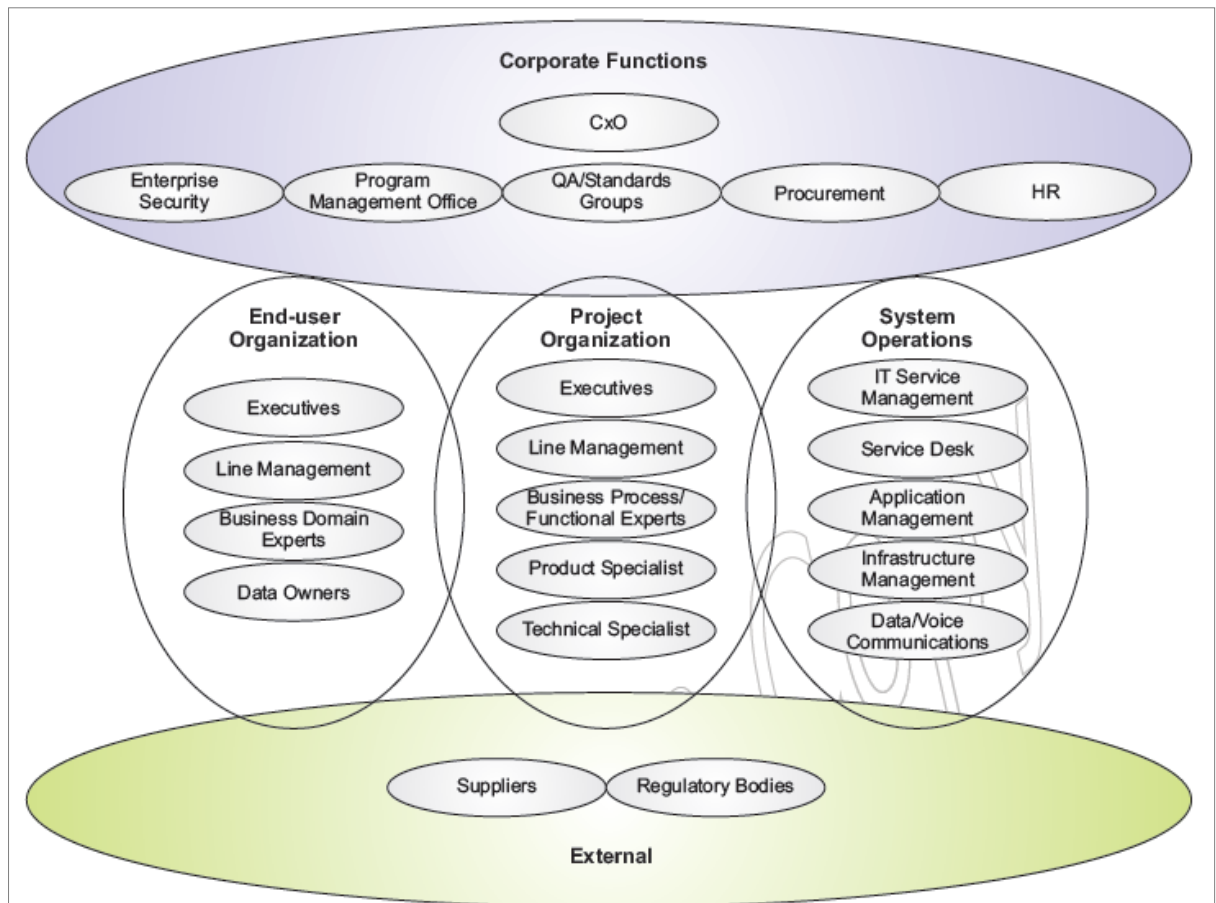


Figura 48 – Categorias de *stakeholders*. Fonte: (TOGAF, 2009, 284).

Utilizando as categorias desta figura é possível fazer uma classificação preliminar dos *stakeholders*. Um exemplo é mostrado na Tabela 11.

Tabela 11 – Exemplo de classificação preliminar de *stakeholders*.

Grupo	Nome	Capacidade de prejudicar a mudança	Compreensão atual	Compreensão necessária	Comprometimento atual	Comprometimento necessário	Apoio necessário
Executivos	Pedro	Alta	Média	Alta	Baixo	Alto	Alto
Analistas	Maria	Média	Média	Média	Médio	Médio	Médio

Fonte: TOGAF, 2009, p. 284.

As dimensões sugeridas na classificação dos *stakeholders*, portanto, são: ameaça aos objetivos do projeto, conhecimento e comprometimento com o projeto e a necessidade de obter o apoio do *stakeholder*.

A classificação dos *stakeholders* é feita com o objetivo de definir o tipo de relacionamento que será mantido com eles. O Togaf leva em consideração (1) o poder e (2) o interesse. Estas duas variáveis do modelo levam a quatro classificações como mostrado na Tabela 12.

Tabela 12 – Matriz de poder dos *stakeholders*

	<i>Alto</i>	C Manter satisfeitos	D Os mais importantes
<u>Poder</u>			
	<i>Baixo</i>	A Esforço mínimo	B Manter informado
		<i>Baixo</i>	<i>Alto</i>
		<u>Nível de interesse</u>	

Fonte: TOGAF, 2009, p. 285.

Finalmente é preciso definir como envolver cada uma das classificações. Por exemplo, a classe D, os mais importantes, devem participar das reuniões onde são tomadas decisões, enquanto o nível B pode receber *e-mails* informando sobre o andamento do projeto.

O Togaf possui um método próprio para desenvolver uma arquitetura, o ADM (*Architecture Development Model*). O tipo de envolvimento dos *stakeholders* depende, também, da etapa específica do processo de desenvolvimento onde o envolvimento ocorre. A Figura 49 mostra as etapas.

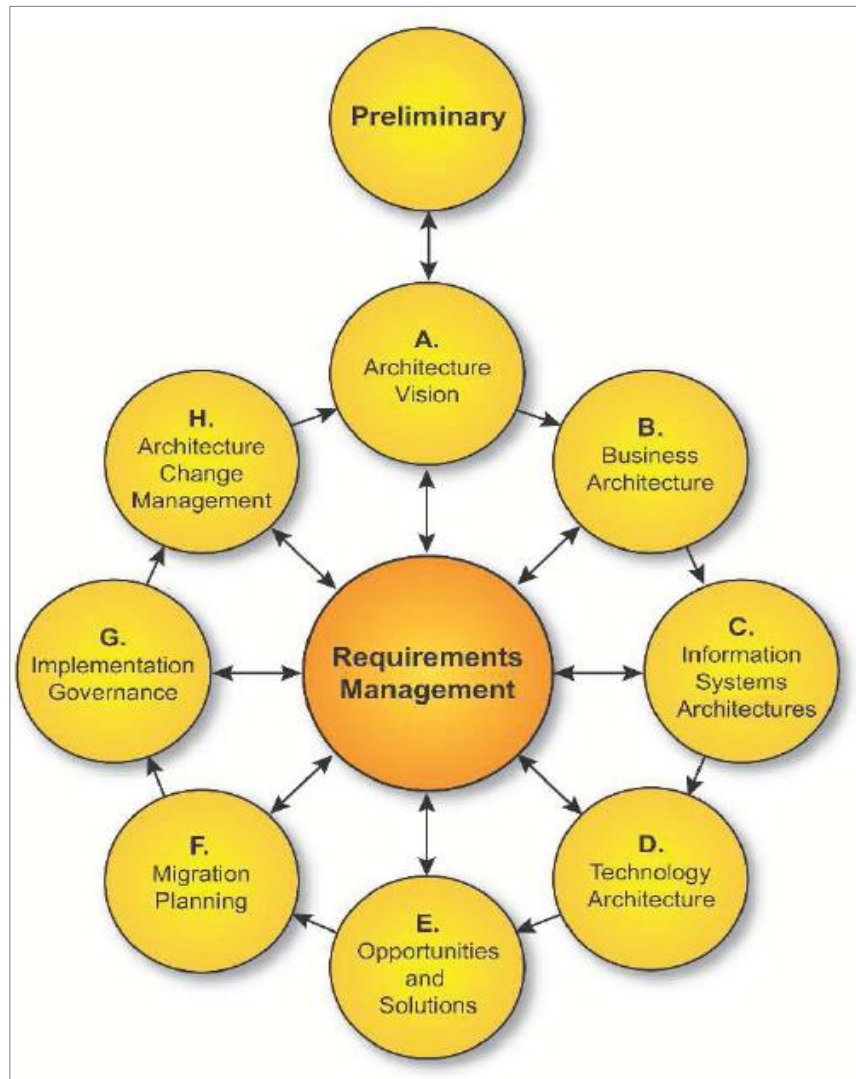


Figura 49 – Ciclo de desenvolvimento da arquitetura. Fonte: TOGAF, 2009, 54.

A fase preliminar serve para definir quem irá participar do projeto da arquitetura, os recursos disponíveis (em especial, o tempo) e o escopo dentro da empresa. No que diz respeito aos *stakeholders*, é necessário criar uma lista contendo, no mínimo, o *sponsor* e o gerente do projeto.

Na etapa Visão da Arquitetura o processo descrito anteriormente serve para definir quem irá participar do projeto da arquitetura, os recursos disponíveis (em especial, o tempo) e o escopo dentro da empresa. No que diz respeito aos *stakeholders* é necessário criar uma lista contendo, no mínimo, o *sponsor* e o gerente do projeto.

1.3.6 Gerência de Projetos e Análise de Negócios

A literatura sobre gerência de projetos e a literatura sobre análise de negócios também consideram o gerenciamento de *stakeholders* como sendo fundamental para o sucesso dos projetos.

O Business Analysis Body of Knowledge (BABoK, a partir daqui) é uma compilação de disciplinas e técnicas, criado pelo International Institute of Business Analysis (IIBA) para servir como referência para Analistas de Negócio. O IIBA também fornece certificações para os analistas que comprovarem experiência prática na área e passarem em um exame criado pelo próprio IIBA. A Tabela 13 mostra o envolvimento de cada tipo de *stakeholder* em cada uma das disciplinas consideradas pelo BABoK. A tabela foi incluída pois os papéis farão parte do ZEP Framework.

Tabela 13 - Matriz de Papéis e Tarefas no BABOK (Parte-1)

Role	Tasks														
	2						3					4			
	2_1	2_2	2_3	2_4	2_5	2_6	3_1	3_2	3_3	3_4	4_1	4_2	4_3	4_4	4_5
<i>Business Analyst</i>									✓						
<i>Customer</i>			✓	✓			✓	✓		✓					✓
<i>Domain SME</i>	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
<i>End User</i>	✓		✓	✓	✓	✓	✓	✓		✓				✓	✓
<i>Implementation SME</i>	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
<i>Project Manager</i>	✓	✓	✓	✓	✓	✓	✓	✓		✓					✓
<i>Supplier</i>	✓		✓	✓			✓	✓		✓					✓
<i>Tester</i>	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓		✓	✓
<i>Regulator</i>	✓	✓		✓			✓	✓		✓					✓
<i>Sponsor</i>	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓			✓	✓
<i>Operational Support</i>			✓	✓	✓	✓	✓	✓		✓					✓
<i>2.1 Plan Business Analysis Approach</i>															
<i>2.2 Conduct Stakeholder Analysis</i>															
<i>2.3 Plan Business Analysis Activities</i>															
<i>2.4 Plan Business Analysis Communication</i>															
<i>2.5 Plan Requirements Management Process</i>															
<i>2.6 Manage Business Analysis Performance</i>															
<i>3.1 Prepare for Elicitation</i>															
<i>3.2 Conduct Elicitation Activity</i>															
<i>3.3 Document Elicitation Results</i>															
<i>3.4 Confirm Elicitation Results</i>															
<i>4.1 Manage Solution Scope & Requirements</i>															
<i>4.2 Manage Requirements Traceability</i>															
<i>4.3 Maintain Requirements for Reuse</i>															
<i>4.4 Prepare Requirements Package</i>															
<i>4.5 Communicate Requirements</i>															

Fonte: IIBA, 2009.

Tabela 13 - Matriz de Papéis e Tarefas no BABOK (Parte 2)

Role	Tasks																	
	5					6						7						
	5_1	5_2	5_3	5_4	5_5	6_1	6_2	6_3	6_4	6_5	6_6	7_1	7_2	7_3	7_4	7_5	7_6	
<i>Business Analyst</i>								✓	✓	✓	✓							
<i>Customer</i>	✓	✓	✓					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Domain SME</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>End User</i>	✓	✓	✓				✓	✓	✓	✓	✓		✓		✓	✓	✓	✓
<i>Implementation SME</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Project Manager</i>				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Supplier</i>	✓	✓	✓					✓	✓	✓	✓	✓	✓					✓
<i>Tester</i>								✓	✓	✓	✓		✓					✓
<i>Regulator</i>	✓							✓	✓	✓	✓						✓	✓
<i>Sponsor</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
<i>Operational Support</i>								✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

5.1 Define Business Need

5.2 Assess Capability Gaps

5.3 Determine Solution Approach

5.4 Define Solution Scope

5.5 Define Business Case

6.1 Prioritize Requirements

6.2 Organize Requirements

6.3 Specify and Model Requirements

6.4 Define Assumptions and Constraints

6.5 Verify Requirements

6.6 Validate Requirements

7.1 Assess Proposed Solution

7.2 Allocate Requirements

7.3 Assess Organizational Readiness

7.4 Define Transition Requirements

7.5 Validate Solution

7.6 Evaluate Solution Performance

Fonte: IIBA, 2009.

2 O ZEP FRAMEWORK

O ZEP Framework, apresentado nesta dissertação, é um modelo criado a partir do metamodelo SPEM. A Figura 50 mostra os níveis de modelagem associados ao ZEP Framework.

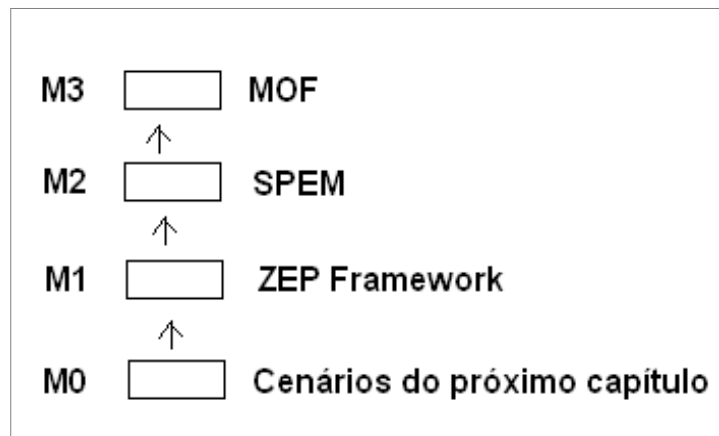


Figura 50 – Níveis de modelagem e o ZEP Framework

O ZEP Framework pode ser usado em vários níveis de conformidade, como definido na seção 1.2.4.1.

2.1 Estrutura de Processos no ZEP Framework

No nível de conformidade 5 (*SPEM Process Structure with Managed Content*) são utilizados apenas os pacotes *Core*, *Process Structure* e *Managed Content*. O pacote *Managed Content* contém a descrição textual dos elementos criados a partir do pacote *Process Structure*. O pacote *Process Structure* permite descrever a estrutura estática dos processos.

Um método completo e pronto para ser implementado (chamado de *Delivery Process* no SPEM) pode ser composto por uma ou mais fases. Dentro de uma fase, alguns processos podem ser iterativos, isto é, podem ocorrer mais de uma vez. As fases “são divisões, dentro de um projeto que exigem um controle extra para gerenciar eficazmente que uma entrega importante seja completada.” (PMIa, 2008, p. 18); A Figura 51 mostra a estrutura de uma fase, indicando os grandes grupos de processos.

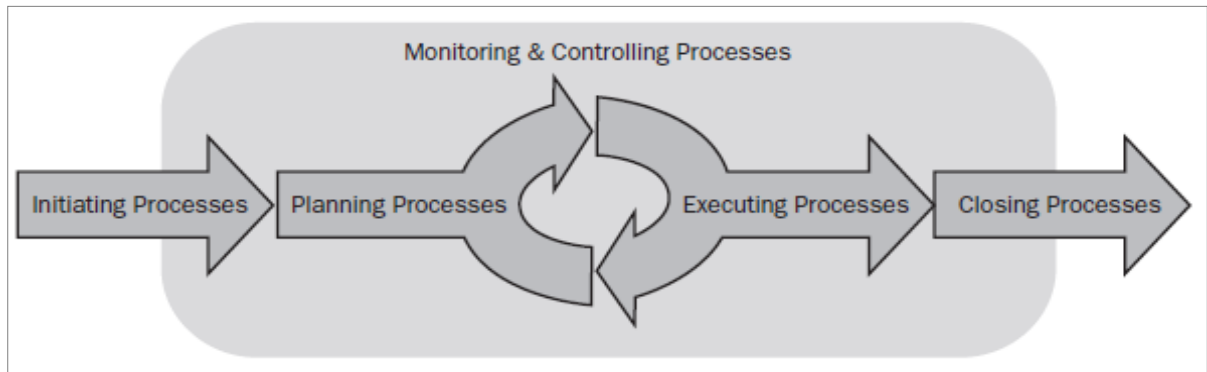


Figura 51- Exemplo de projeto com uma única fase.

Fonte: PMIa, 2008, p. 19.

A implementação do modelo da Figura 51, no ZEP Framework, é mostrada na Figura 52. Esta implementação permite que a estrutura básica de uma Fase, segundo o PMBOK, possa ser reutilizada várias vezes.

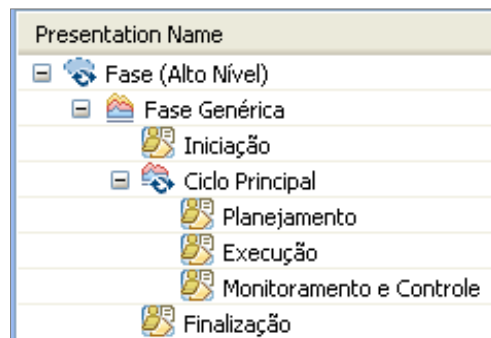


Figura 52 – Representação de uma Fase no ZEP Framework, baseado no PMBOK

Fonte: PMIa, 2008.

As fases de um projeto podem ser organizadas de forma sequencial, sobreposta ou uma combinação de ambas. A organização sequencial é usada quando o início de uma fase depende do término de outra fase que já teve início. A Figura 53 mostra um exemplo para a remoção de resíduos perigosos em uma área.

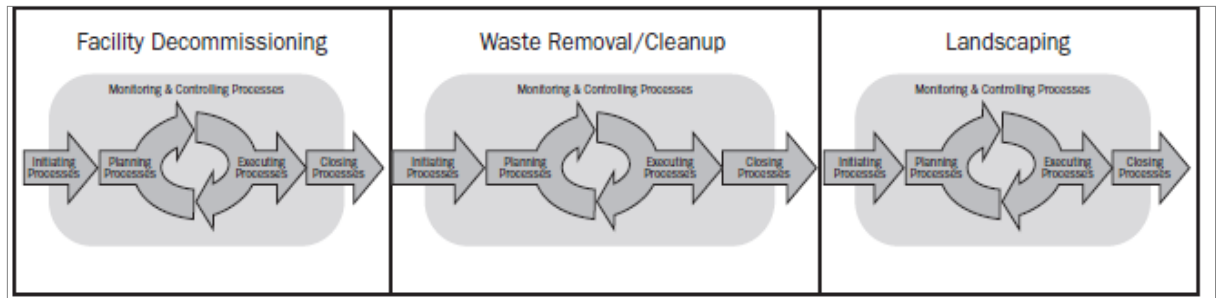


Figura 53 - Exemplo de projeto com três fases.

Fonte: PMIA, 2008, p. 21.

No relacionamento sobreposto (Figura 54), o início de uma fase não depende do término de outra que tenha começado antes. A execução sobreposta aumenta o nível de complexidade do controle de projeto mas pode resultar em economia de tempo.

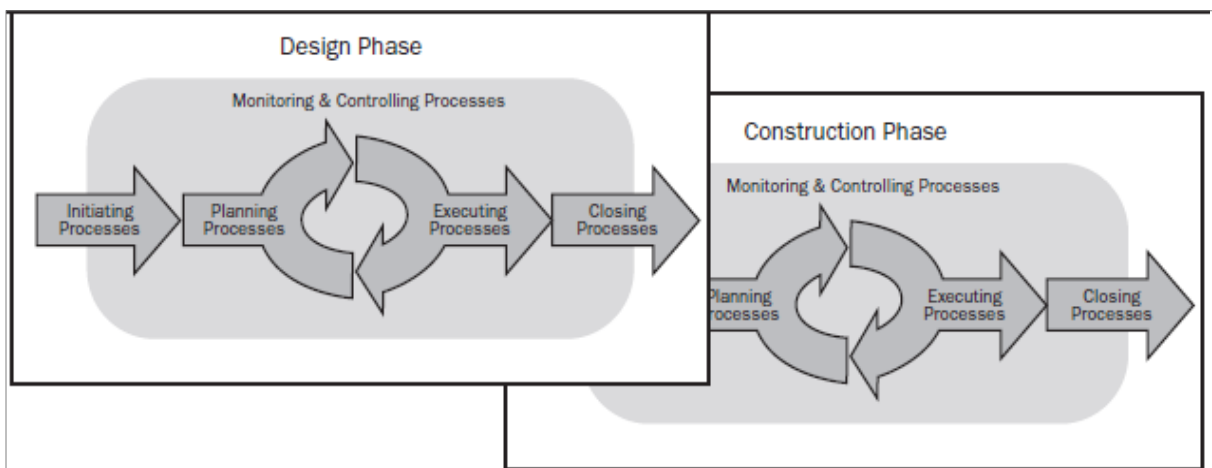


Figura 54 – Exemplo de projeto com fases sobrepostas.

Fonte: PMIA, 2008, p. 21.

No ZEP Framework, os recursos do EPF Composer permitem a especificação da relação entre as Fases. A Figura 55 apresenta um exemplo. Nesta Figura é especificado que a Fase “Scope” deve terminar para que a Fase “Act_(Planning)” possa começar (*Finish-to-Start*), o que corresponde a uma estrutura sequencial entre estas duas Fases.

Phase : Act_[Planning]

General Information
Provide general information about this phase.

Name: Act_[Planning]

Presentation name: Act [Planning]

Optional
 Event Driven
 Multiple Occurrences
 Ongoing
 Planned
 Repeatabe
 Suppressed

Index	Presentation Name	Dependency
1	Scope	Finish To Start

Dependency:

Figura 55 – Exemplo de especificação de relacionamento entre Fases no ZEP Framework, usando o EPF Composer

As atividades dentro de um projeto, segundo o PMBOK, podem ser classificadas em Grupos de Processos. A Figura 56 mostra como estes Grupos de Processo se relacionam.

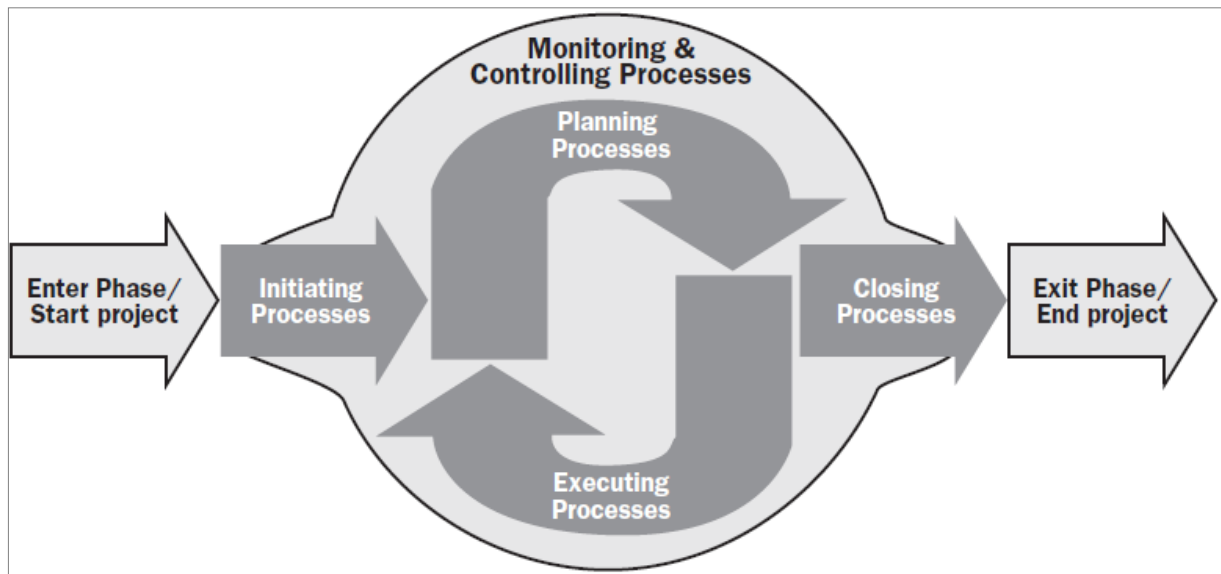


Figura 56 – Grupos de Processos no Gerenciamento de Projetos. Fonte: PMIa, 2008, p. 40.

Os processos de iniciação (*Initiating Processes*) (ver Figura 56) são executados no início do projeto ou no início de uma fase e têm como objetivo identificar as partes envolvidas e obter autorização para prosseguir. A Figura 57 mostra a sua implementação no ZEP Framework.



Figura 57 – Representação do Grupo de Processos de Iniciação no EPF Composer, segundo o PMBOK. Fonte: PMIa, 2008.

Os processos do Grupo de Processos de Planejamento têm como objetivo “estabelecer o escopo do projeto [ou fase], definir e refinar os objetivos e desenvolver a linha de ação necessária para atingir estes objetivos” (PMI, 2008, p.46). A Figura 58 mostra a sua implementação no ZEP Framework.

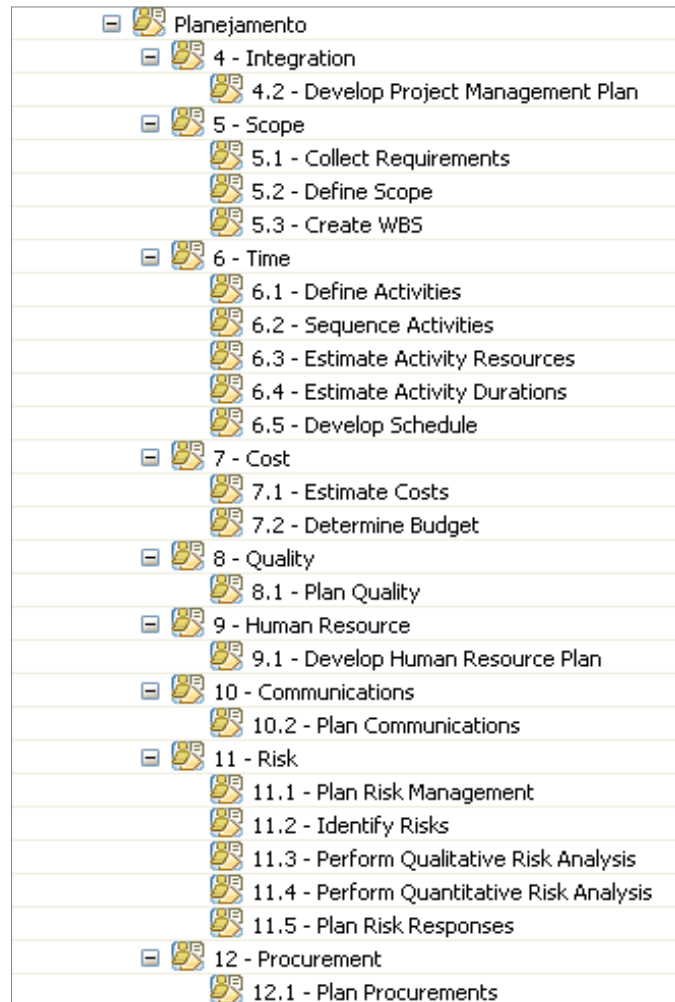


Figura 58 – Representação do Grupo de Processos de Planejamento no EPF Composer, segundo o PMBOK. Fonte: PMIa, 2008.

No Grupo de Processos de Execução estão os processos que têm como objetivo “concluir o trabalho definido no plano de gerenciamento do projeto de forma a cumprir as especificações do projeto” (PMIA, 2008, p. 55). A Figura 59 mostra a sua implementação no ZEP Framework.

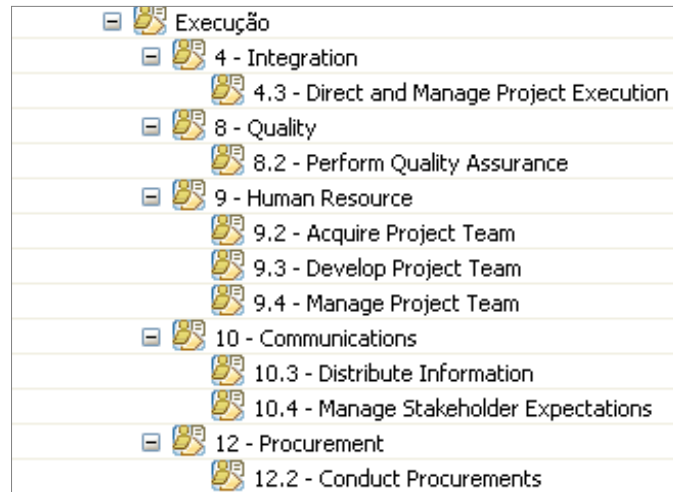


Figura 59 – Representação do Grupo de Processos de Execução no EPF Composer, segundo o PMBOK. Fonte: PMIA, 2008.

No Grupo de Monitoração e Controle estão os processos necessários para acompanhar a execução, rever o planejamento e criar regras e alterações para garantir a realização do projeto. O objetivo geral é evitar ter apenas uma avaliação no final do projeto quando não haverá mais a possibilidade de adaptar o processo às situações que tenham surgido. A Figura 60 mostra a sua implementação no ZEP Framework.

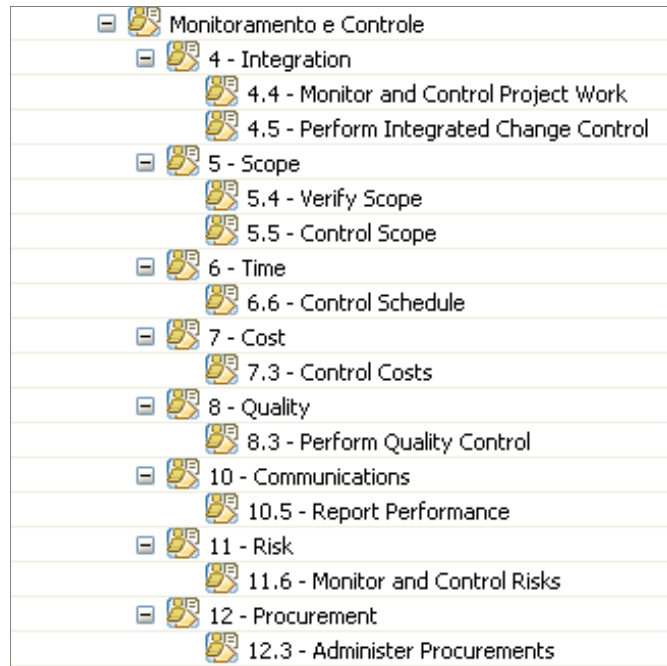


Figura 60 – Representação do Grupo de Processos de Monitoração e Controle no EPF Composer, segundo o PMBOK. Fonte: PMIa, 2008.

No Grupo de Finalização estão os processos para terminar formalmente o projeto ou a fase. Envolve a aceitação do produto ou serviço pelo Solicitante e a assinatura de um documento que declara que os produtos ou serviços foram entregues. A Figura 61 mostra a sua implementação no ZEP Framework.

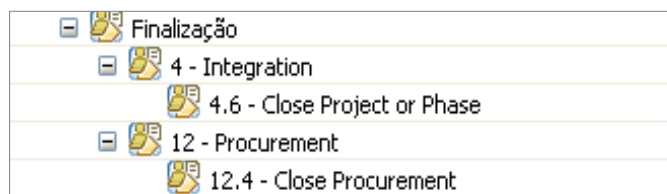


Figura 61 – Representação do Grupo de Processos de Finalização no EPF Composer, segundo o PMBOK. Fonte: PMIa, 2008.

No ZEP Framework vamos considerar o envolvimento de Partes Interessadas (*stakeholders*) como sendo um projeto paralelo a um Projeto de Desenvolvimento de Software e vamos chamá-lo de Projeto das Partes Interessadas.

Os Grupos de Processo do PMBOK representam uma forma genérica que pode ser aplicada a qualquer projeto. Para nossos fins é necessário especificar com mais detalhes os aspectos específicos do envolvimento de *stakeholders*. Um exemplo de planejamento para este envolvimento é mostrado na Figura 62.



Figura 62 – A chave para o planejamento de envolvimento.

Fonte: VICTORIA, 2005b, p. 6.

Este método, criado pelo Departamento de Sustentabilidade e Ambiente do Estado de Victoria, na Austrália, trata dos grupos de processo referentes ao planejamento do envolvimento de *stakeholders*. A implementação da Figura 62, no ZEP Framework, é mostrada na Figura 63.

A partir da inclusão de diferentes abordagens para o envolvimento de *stakeholders* e de uma estrutura geral de projetos (do PMBOK) é possível criar um modelo que atenda a uma determinada situação em uma organização.

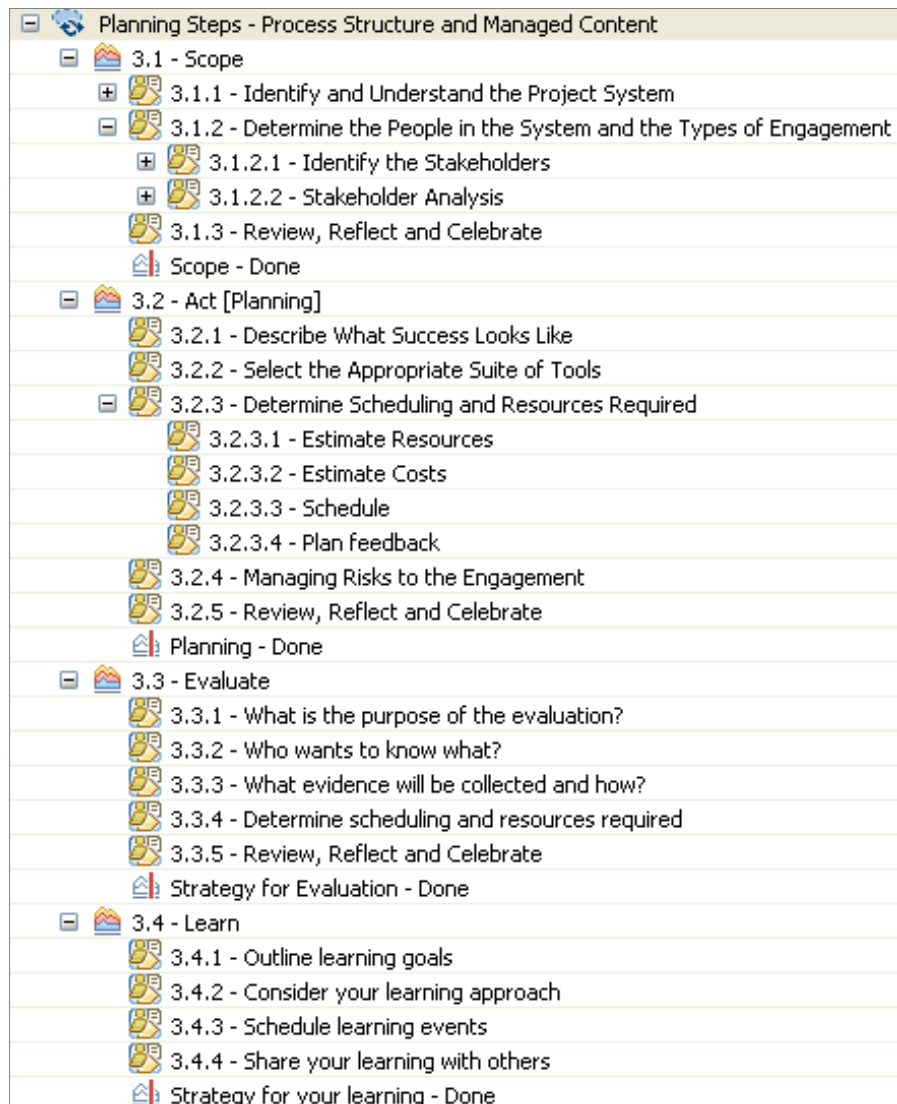


Figura 63 – Implementação do Engagement Workbook.

Fonte: VICTORIA, 2005b, no EPF Composer.

Um outro exemplo de método, incorporado ao ZEP Framework é o The Stakeholder Engagement Manual da UNEP (United Nations Environment Programme), SRA (Stakeholder Research Associates) e AccountAbility (UNEP, 2005). A Figura 64 mostra a estrutura geral do método. O objetivo deste método também corresponde ao Grupo de Processos de Planejamento do PMBOK.



Figura 64 – Os cinco estágios (*stages*) do Envolvimento de *Stakeholders*.

Fonte: UNEP, 2005, p. 21.

Estes estágios foram implementados no ZEP Framework, como pode ser visto na Figura 65.

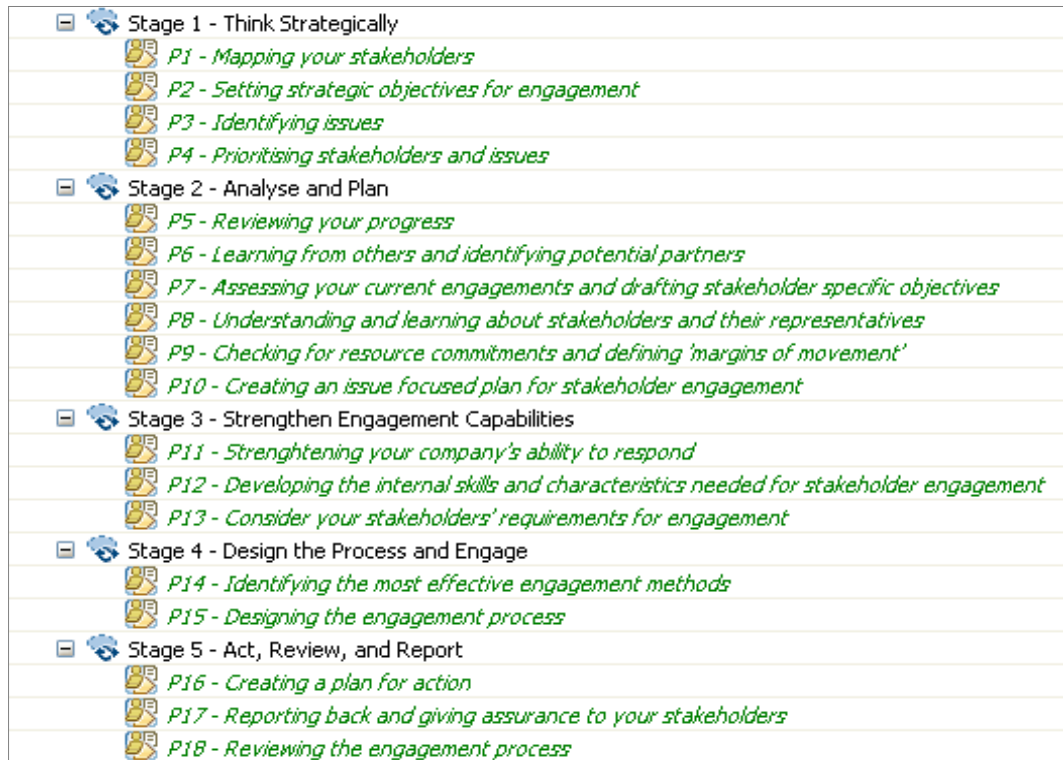


Figura 65 – Os cinco estágios (*stages*) do Manual de Envolvimento de *Stakeholders*.

Fonte: UNEP, 2005, no EPF Composer.

Outro exemplo ainda é o de Babiuch et al. (Babiuch et al., 1994). Apenas as três primeiras fases indicadas na Figura 66 são explicitadas no texto. Estas três fases estão contidas na fase de *Planning* do PMBOK.

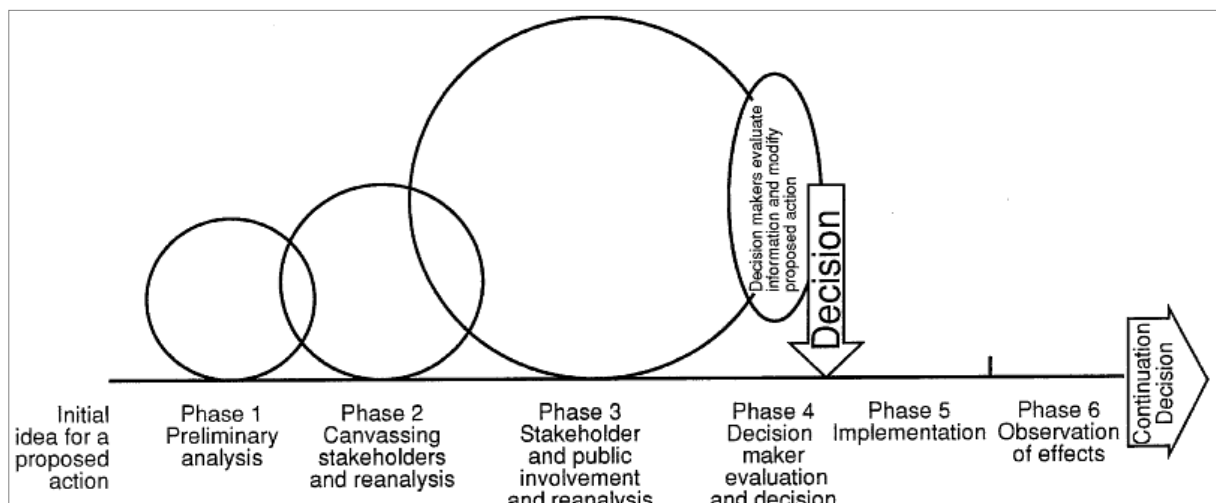


Figura 66 – Fases em um processo ideal de análise e decisão sobre uma ação propostas.

Fonte: BABIUCH et al., 1994, p. 5.

O critério utilizado por Babiuch et al. para a definição das fases foi o tipo de envolvimento de *stakeholders*: na primeira fase não há nenhum contato, na segunda fase o contato é feito apenas com poucos representantes e na terceira ocorre a participação do público em geral (se for necessário) A sua representação, usando o EPF Composer, é mostrada na Figura 67.

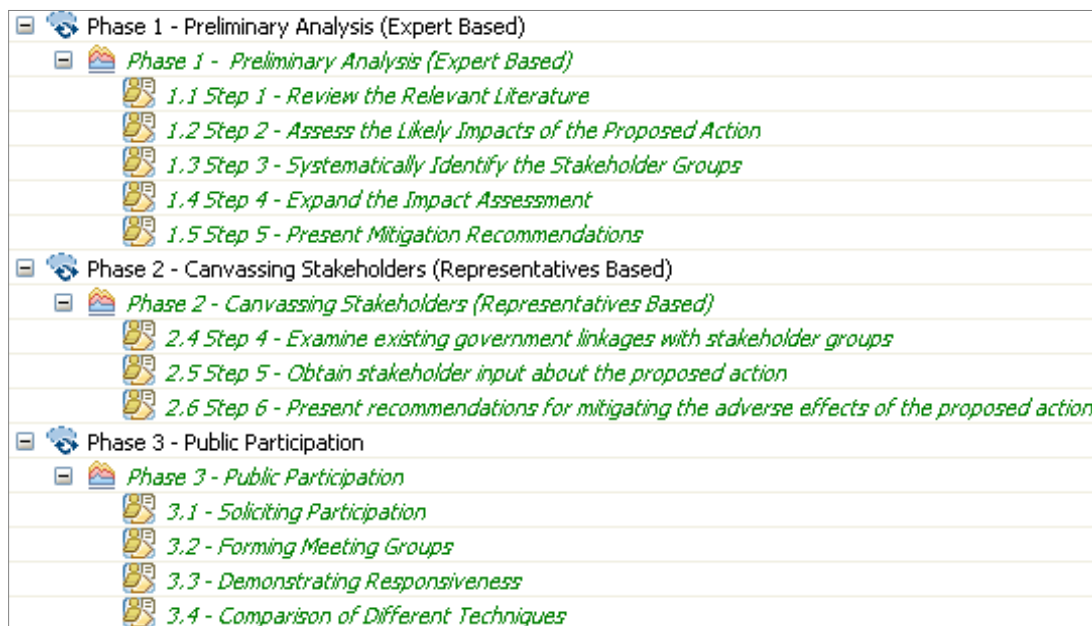


Figura 67 – Representação de método de Babiuch et al. no EPF Composer

O Projeto das Partes Interessadas terá uma primeira atividade que corresponde aos processos de iniciação do Projeto de Desenvolvimento de Software. Como um dos objetivos deste grupo de processos é o Desenvolvimento do Termo de Abertura do Projeto, o Projeto das Partes Interessadas contribuirá com uma lista preliminar de partes interessadas.

Correspondendo às atividades de planejamento do Projeto de Desenvolvimento de Software existirá uma Iteração, no Projeto das Partes Interessadas, que irá se aprofundar na identificação, classificação e planejamento do envolvimento com as partes interessadas. Este planejamento levará em consideração as fases que o Projeto de Desenvolvimento de Software (outro projeto, com execução sobreposta) terá. O principal Artefato criado nesta Iteração indicará, para cada grupo de partes interessadas, e para cada fase do Projeto de Desenvolvimento de Software, qual será o tipo de envolvimento a ser adotado e com quais técnicas. O artefato assumirá a forma de uma tabela com um conteúdo semelhante ao da Tabela 14.

Tabela 14 – Planejamento de envolvimento

	Fase 1	Fase 2	Fase 3
<i>Stakeholder 1</i>	Consultar Técnica 1 Técnica 2	Informar Técnica 4	Informar Técnica 5 Técnica 8
<i>Stakeholder 2</i>	Consultar Técnica 2	Decidir Técnica 9	Informar Técnica 3
<i>Stakeholder 3</i>	Informar Técnica 4 Técnica 3	Informar Técnica 3	Informar Técnica 3
<i>Stakeholder 4</i>	Decidir Técnica 9	Decidir Técnica 7	Informar Técnica 3

A cada fase do Projeto de Desenvolvimento de Software corresponderá uma Iteração do Ciclo Principal (ver Figura 52) do Projeto de Partes Interessadas. Isto significa que haverá um grupo de processos de monitoração e avaliação para cada fase do Projeto de Partes Interessadas em que será feita a avaliação dos resultados do envolvimento. Por exemplo, usando a Tabela 14, é possível verificar se o *Stakeholder 2* realmente participou das decisões na Fase 2. No caso de isto não ter acontecido, este resultado deverá ser levado em conta no Grupo de Processos de Planejamento da fase seguinte (ou constar no relatório final do projeto).

2.2 Conteúdo de Métodos no ZEP Framework

No nível de conformidade 3 (*SPEM Method Content*), visto na Tabela 7 da seção 1.2.4.1, são utilizados apenas os pacotes *Core*, *Managed Content* e *Method Content*. O pacote *Managed Content* contém a descrição textual dos elementos criados a partir do pacote *Method Content*. O pacote *Method Content* permite descrever, com detalhes, elementos que podem ser reutilizados, pois são descritos sem levar em consideração o momento em que são executados dentro de um processo. Um modelo, com os principais elementos de Conteúdo de Método, está na Figura 68.

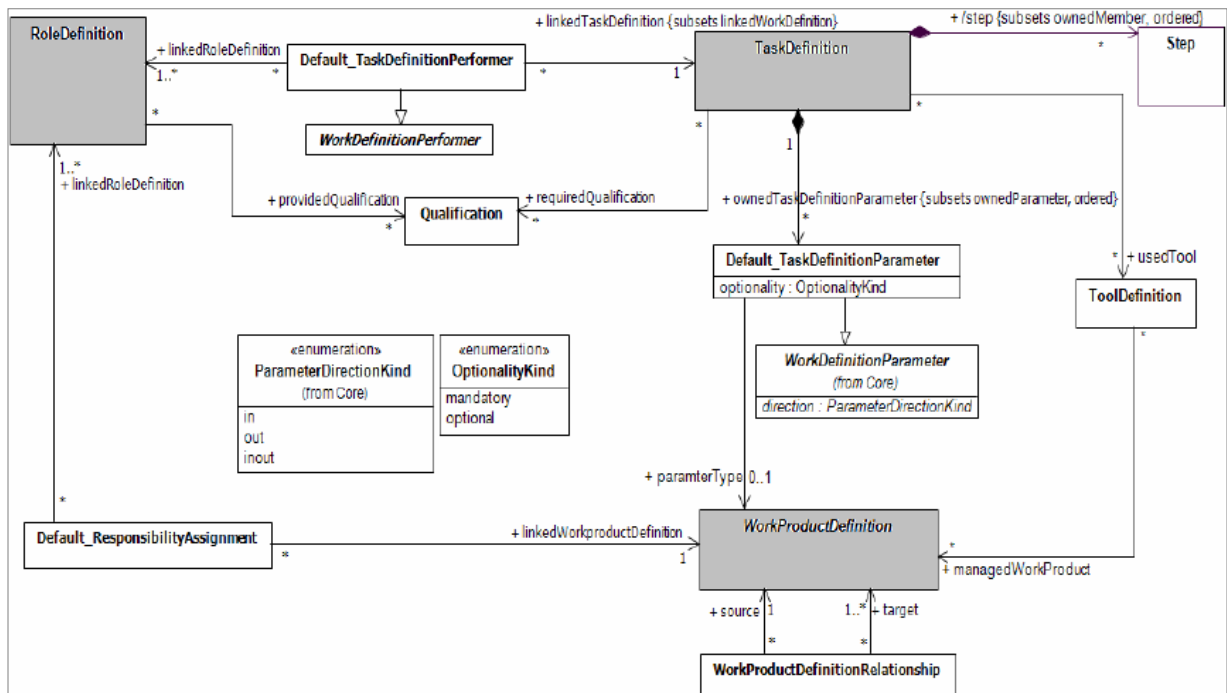


Figura 68 – Principais Elementos de Conteúdo de Método e seus relacionamentos

Fonte: OMG, 2008, p. 83.

Os itens seguintes apresentam a implementação de *Work Products*, *Roles* e *Tasks* no ZEP Framework.

2.2.1 *Work Products* (Produtos de Trabalho) no ZEP Framework

A execução de um trabalho (fase, iteração, atividade, tarefa) produz um efeito. Este efeito é chamado de “*Work Product*” no SPEM e no EPF Composer. Os “*Work Products*” podem ser de três tipos: artefatos (*artifacts*), resultados (*outcomes*) e entregas (*deliverables*).

Um artefato representa um efeito tangível do trabalho. Pode ser um relatório, um programa executável, uma lista de elementos ou qualquer outro efeito que seja objetivamente verificável como existente. É um efeito concreto. Um artefato pode ser formado por outros artefatos criando uma hierarquia de artefatos.

Um resultado (*outcome*) representa um efeito intangível. Pode ser um estado ou situação. Como não é tangível a sua verificação é mais difícil. Por exemplo, a satisfação do usuário é um resultado de um sistema bem feito e entregue no prazo, no entanto, por ser abstrato, precisa de algum critério que sirva como indicador de sua existência. Outros

resultados estão associados a estados mais fáceis de verificar como, por exemplo, os estados de “iniciado”, “em andamento” e “terminado”. Um resultado pode ser formado por outros resultados, criando uma hierarquia de resultados.

Nesta dissertação, um resultado é sempre o efeito de um trabalho. Existem, no entanto, resultados que não dependem apenas do trabalho sob o controle de um gerente de projeto. O efeito pode ter sido produzido por vários trabalhos independentes, alguns deles produzidos por terceiros sem conhecimento do trabalho dos outros. Neste caso a atribuição do resultado a um trabalho específico fica mais difícil. Por exemplo, o bem-estar da população de um país pode ser o resultado de um aumento no nível de renda porém este não é o único nem, talvez, o mais importante resultado que contribuiu para isso. Chamaremos este tipo de resultado de “impacto” e o representaremos, no EPF Composer como um tipo específico de resultado (*outcome*).

O terceiro tipo de efeito é a entrega (*deliverable*). Uma entrega é composta por um conjunto de artefatos e/ou resultados. Geralmente a entrega está associada ao fim de uma fase do projeto ou a algum evento importante. Uma entrega também pode incluir outras entregas. Uma característica importante da entrega é que ela faz sentido para quem a recebe (e não apenas para quem a produz).

Um método pode ser organizado pela ordenação de unidades de trabalho, como por exemplo, uma atividade começar após o término de outra (chamado de *Work Breakdown Structure*, ou *WBS*). Além disso, pode ser organizado pela sequência de efeitos (artefatos, resultados e entregas) sendo chamado, então, de *Product Breakdown Structure*, ou *PBS*. No EPF Composer, um PBS pode ser apresentado através da guia *Product Usage* de um processo.

2.2.2 Role (Papel) no ZEP Framework

A execução de um trabalho (fase, iteração, atividade, tarefa) é realizada por uma pessoa, ou grupo de pessoas, que é considerado como sendo um Ator (*Actor*). Um Ator é um elemento externo ao projeto. Quando o Ator se integra ao projeto assume um Papel (*Role*)

Na descrição dos componentes de método são especificados os *Papéis* associados a trabalhos e efeitos.

O BABoK especifica um conjunto de Papéis (que são chamados de *Stakeholders*) que podem ocorrer em um projeto de Análise de Negócios. A Tabela 15 resume os Papéis apresentados. É importante frisar que a tabela não é exaustiva. Além disso, um Ator pode assumir vários Papéis e um Papel pode ser assumido por vários Atores.

Tabela 15 – Papéis em Análise de Negócios

Papel	Descrição
Analista de Negócios	Participa, por definição, de todas as atividades da análise de negócio.
Cliente	Um cliente está fora da fronteira de uma determinada organização. Usam os produtos e/ou serviços da organização
Especialista do Domínio (<i>Domain Subject Matter Expert</i>) ou (<i>Domain SME</i>)	Possui conhecimentos profundos de um assunto relacionado à necessidade do negócio ou ao escopo da solução.
Usuário Final	Interagem diretamente com a solução. Usado no contexto de desenvolvimento de <i>software</i> .
Especialista de Implementação (<i>Implementation Subject Matter Expert</i>) ou (<i>Implementation SME</i>)	Responsável pela implementação da solução. Pode ser: <ul style="list-style-type: none"> • Desenvolvedor/Engenheiro de Software • Profissional de Mudança Organizacional • Arquiteto de Sistemas • Instrutores • Profissional de Usabilidade
Gerente de Projeto	Responsável por gerenciar o trabalho necessário para entregar a solução.
Testador	Determina como verificar se a solução atende aos requisitos.
Regulador	Responsável pela definição (e cobrança) de padrões.
Patrocinador	Responsável por iniciar o esforço, autorizar o trabalho e controlar o orçamento.
Fornecedor	Um fornecedor está fora da fronteira de uma determinada organização. Fornece produtos e/ou serviços.

Fonte: IIBA, 2009.

2.2.3 Tasks (Tarefas) no ZEP Framework

Para cada tarefa foram definidas várias informações que permitem que o usuário do *framework* saiba como usar a tarefa e que o Engenheiro de Métodos (a pessoa responsável pela criação do método, a partir do *framework*) tenha parâmetros para a seleção do tarefa mais apropriada para cada situação. Os parágrafos seguintes esclarecem a finalidade e o modo de uso de cada campo de informação (em negrito) que pode constar para uma tarefa no *Framework*.

Uma tarefa pode ser conhecida, na literatura, por outros nomes. Isto pode levar o usuário a se questionar se o nome da tarefa é equivalente a outra com a qual esteja mais acostumado. Ao indicar explicitamente, no campo **Outros nomes**, as alternativas existentes, esta dúvida não ocorrerá.

A **Descrição** fornece uma visão geral sobre a tarefa. Informações específicas sobre a tarefa são apresentadas nos outros itens que descrevem a tarefa. O objetivo da **Descrição** é permitir ao leitor decidir se deve continuar a ler as outras informações. No EPF Composer é conhecido como *Main Description*.

As tarefas sempre possuem **Objetivos**. Este item procura dizer da forma mais sucinta possível quais os objetivos da tarefa. No EPF Composer é conhecido como *Purpose*.

O efeito gerado pela execução de uma tarefa pode ser de três tipos diferentes: artefato (*artifact*), resultado (*outcome*) ou entrega (*deliverable*). Um **Resultado** (*outcome*) é um efeito intangível da execução. Um Resultado pode ser composto de outros Resultados. No EPF Composer, o Resultado é um tipo de Produto de Trabalho (*Work product*) chamado de *Outcome*..

Quando o efeito da execução é tangível ele é incluído no campo **Artefatos**. Um Artefato pode ser composto por outros Artefatos. Neste campo indicamos apenas o artefato que agrega os outros. No EPF Composer, o Artefato é um tipo de Produto de Trabalho (*Work product*) chamado de *Artefact*.

O terceiro tipo de efeito é a **Entrega** (*deliverable*). Uma Entrega representa um conjunto de efeitos, isto é, um conjunto com zero ou mais Artefatos, Resultados e outras Entregas. A Entrega normalmente ocorre na última tarefa de uma fase ou sub-fase.

Uma tarefa pode gerar vários tipos de efeitos (Artefatos, Resultados e Entregas). Os **Usos/aspectos positivos** esclarecem em que situações a tarefa (e seus efeitos) podem ser úteis. Como uma tarefa também pode ter aspectos negativos (no campo **Considerações**

especiais/aspectos negativos), é importante enfatizar aqui quais são os aspectos positivos para que a decisão sobre a realização da tarefa possa ser feita de forma mais clara. Os usos são descritos aqui, assim como as contribuições (aspectos positivos) que a execução da tarefa pode fornecer. No EPF Composer, estas informações são incluídas, juntamente com as “**Considerações especiais/aspectos negativos**” no item *Key considerations*.

Para a realização de uma tarefa geralmente são necessários recursos materiais. As informações no campo “**Recursos materiais necessários**” podem servir para a preparação da tarefa e também como critério para a sua escolha. Além disso ajudam na determinação do custo total estimado da tarefa. No EPF Composer, estas informações são incluídas como um tipo de *Guidance*, o *Supporting material*

As tarefas são executadas por pessoas. As pessoas estão associadas a papéis que podem desempenhar. Um papel (*role*) pode ser definido como “um conjunto de habilidades, competências e responsabilidades” (OMG, 2008, p. 87). O campo “**Recursos humanos (roles)**” indica os papéis necessários para a execução da tarefa. No EPF Composer, estas informações são conhecidas como *Roles* e são descritas em um outro tipo de componente. Na descrição da tarefa é feita apenas uma referência aos Papéis necessários.

Um papel indica o conjunto de habilidades necessárias para uma tarefa, porém ainda é necessário saber quantas pessoas com este Papel serão necessárias. No campo “**Quantidade de pessoas necessárias**” é indicado o número de pessoas que irão trabalhar para que a tarefa seja realizada. No EPF Composer, estas informações são incluídas como um tipo de *Guidance*, o *Estimation considerations*

O número de pessoas que irá participar da tarefa (sem considerar as pessoas incluídas no item anterior) são indicadas no campo “**Quantidade de pessoas alcançadas**”. No EPF Composer, essa informação é incluída como um tipo de *Guidance*, o *Estimation considerations*

A quantidade de **Tempo necessário (cronômetro)** indica qual deverá ser, de fato, o tempo utilizado na tarefa. Por exemplo, uma tarefa que possa ser realizada em 3 horas (cronômetro) pode levar 3 dias (calendário) para chegar ao fim, se as pessoas só puderem trabalhar uma hora por dia na tarefa. No EPF Composer, esta informação é incluída como um tipo de *Guidance*, o *Estimation considerations*.

O campo “**Tempo necessário (calendário)**” indica a quantidade de tempo total decorrido para realizar uma tarefa. Por exemplo, uma tarefa que possa ser realizada em 3 horas (cronômetro) pode levar 3 dias (calendário) para chegar ao fim. No EPF Composer, essa informação é incluída como um tipo de *Guidance*, o *Estimation considerations*.

As habilidades necessárias para a execução da tarefa são especificadas no campo “**Nível de habilidade/suporte necessário**”. Elas devem ser indicadas na forma “Papel (habilidade e nível)”. Por exemplo, “Entrevistador (Alta capacidade de comunicação)”. No EPF Composer, essas informações são incluídas como complemento da descrição de um Papel (*Role*), no subitem *Skills*

O **Custo** financeiro da tarefa é um dos principais fatores a serem considerados no momento da escolha das tarefas e técnicas a serem selecionadas para o método. Como o valor da moeda pode variar com o tempo e com o tamanho da organização, deve ser indicado qualitativamente como (1) muito alto, (2) alto, (3) médio, (4) baixo ou (5) muito baixo. No EPF Composer, essas informações são incluídas como um tipo de *Guidance*, o *Estimation considerations*.

O processo de classificação de um *stakeholder* serve para indicar o nível de participação que ele terá. O campo **Nível de participação (IAP2)** indica que a Tarefa pode ser utilizada por *stakeholders* com este nível de participação. Existem muitas formas de classificar o nível de participação dos *stakeholders*. O modelo do IAP2 (visto anteriormente) é uma das formas mais difundidas de se realizar esta classificação. No EPF Composer, essas informações são incluídas como um tipo de *Guidance*, o *Estimation considerations*.

Se as tarefas tiverem um **Nível de inovação** muito alto isto irá acarretar na necessidade de treinamento para as pessoas responsáveis pela tarefa e, algumas vezes, para todos os participantes. O nível deve ser indicado qualitativamente como sendo (1) muito alto, (2) alto, (3) médio, (4) baixo ou (5) muito baixo. No EPF Composer, estas informações são incluídas como um tipo de *Guidance*, o *Estimation considerations*.

A sequência, passo a passo, das **Etapas** a serem seguidas para a realização da tarefa também precisa ser indicada. No EPF Composer, essas informações são conhecidas como *Steps*.

Finalmente, as **Referências** indicam material bibliográfico suplementar sobre a tarefa. No EPF Composer, estas informações podem ser incluídas como *Guidance* do tipo *White paper*, mesmo que seja um livro ou folheto, ou como *Supporting material*, embora aqui, estejamos usando *Supporting material* apenas para recursos materiais.

O Apêndice A contém vários exemplos de componentes de métodos para tarefas.

2.3 Processo de seleção de componentes

O processo de seleção leva em consideração os atributos do componente e os atributos da situação. Os atributos da situação foram vistos no item 1.2.3. Os atributos utilizados em uma determinada situação devem ser determinados a partir do conjunto da Tabela 16.

Tabela 16 – Atributos de situação

Atributo	Descrição
Características do sistema	Nível de reutilização estimado
	Tamanho estimado
	Complexidade do sistema
Características da iniciativa	Categoria do projeto
	Tipo do contrato (<i>outsourcing</i> ou não)
	Restrições de tempo
	Restrições de custo
	Restrições de recursos humanos (quantidade e habilidades)
Características da organização	Conhecimento do domínio pelos <i>stakeholders</i>
	Conhecimento do domínio pelos analistas de negócio
	Relacionamento Solicitante/Solicitado (interno ou externo)

2.4 Considerações

A Tabela 17 (semelhante à Tabela 9) compara a terminologia de alguns modelos de Componentes de Método com o ZEP Framework. Na Figura 50 mostramos que o ZEP Framework é uma aplicação do meta-modelo do SPEM, portanto os tipos de componentes são os mesmos. Existem algumas diferenças de terminologia em relação ao modelo de Firesmith, o que é mostrado na Tabela 17.

Tabela 17 - Componentes do EPF Composer, PMBok e BABoK

	EPF Composer (*)	MFESA (FIRESMITH et al, 2009)	ZEP Framework
Processo	O quê (What)	Tarefa, atividade, <i>capability-pattern</i> , processo	Unidades de Trabalho (<i>Work Units</i>)
	Entrada/Saída (input/output)	Entrada (<i>input</i>), saída (<i>output</i>)	Unidades de Trabalho (<i>Work Units</i>)
	Como (How)	Orientação (<i>guideline</i>), passos	Tarefa, atividade, <i>capability-pattern</i> , processo
	Quando (When)	Fase, iteração, atividade	Entrada (<i>input</i>), saída (<i>output</i>)
	Quem (Who)	Papel (<i>role</i>)	Orientação (<i>guideline</i>), passos
	Reusabilidade e montagem	Descritor de componente	Fase, iteração, atividade
Produto		Artefatos (<i>Artifacts</i>), resultados (<i>outomes</i>), entregas (<i>deliverables</i>)	Papel (<i>role</i>)
		Produtos do Trabalho (<i>Work Products</i>)	Descritor de componente
			Artefatos (<i>Artifacts</i>), resultados (<i>outomes</i>), entregas (<i>deliverables</i>)

(*) EPF Composer possui outros elementos, além dos que foram incluídos aqui.

Neste capítulo definimos um *framework* para a criação de métodos. Estes métodos serão usados para a criação de Modelos de Envolvimento de *Stakeholders*. O ZEP Framework foi baseado em uma combinação dos princípios da Engenharia Situacional de Métodos e de *Design Science*. Usamos a ferramenta EPF Composer, que é baseada no meta-modelo SPEM

O próximo capítulo examina alguns cenários para ilustrar a utilização do ZEP Framework.

3 CENÁRIOS DE APLICAÇÃO DO *FRAMEWORK*

Neste capítulo são apresentados cenários para ilustrar a aplicação do ZEP Framework em situações típicas. Existem três papéis associados à utilização do ZEP Framework: (1) o responsável pela criação e evolução dos componentes de métodos, (2) o Engenheiro de Métodos, responsável pela criação de métodos a partir dos componentes e (3) os usuários do método criado e publicado (geralmente desenvolvedores e gerentes de projeto). Neste capítulo vamos considerar o ponto de vista do Engenheiro de Métodos.

Em todos os cenários vamos considerar uma organização governamental de nível estadual que é responsável pelo desenvolvimento de sistemas de informática para o Estado. O PMBOK considera várias estruturas organizacionais no que diz respeito ao gerenciamento de projetos: funcional, matricial fraca, matricial balanceada, matricial forte e orientada para projetos. A organização considerada nos cenários utiliza uma estrutura de gerenciamento de projetos matricial balanceada.

Nos cenários, será apresentado um processo sequencial para a criação dos métodos. No entanto, deve-se ter em mente que as etapas utilizadas são apenas sugestões, pois em cada situação o seu uso pode ser conveniente ou não. Além disso, o processo de criação do Modelo de Envolvimento de *Stakeholders* é iterativo, pois novos *stakeholders* podem surgir a qualquer momento. Recomenda-se que periodicamente seja feita uma revisão para redefinir quem são os *stakeholders*, sua importância, seus relacionamentos e demais elementos do Modelo.

A criação da política de envolvimento de *stakeholders* terá como ponto de partida o início de um projeto para resolver algum problema ou aproveitar uma oportunidade. O evento que leva à criação do projeto é o recebimento de uma solicitação formal para a prestação de um serviço. Estamos supondo, portanto, que já existe uma certa limitação de escopo pois a criação do Modelo de Envolvimento de *Stakeholders* está sendo considerada como parte de um projeto mais abrangente do qual o Modelo será parte.

A forma mais comum é atender a uma solicitação formal para a criação de uma solução de software. As solicitações para desenvolvimento de software podem ser enviados ao Executivo-chefe da organização ou a um Diretor ou Assessor a quem tenha sido delegada a responsabilidade pelo atendimento de solicitações. Um desses, após uma avaliação subjetiva, o envia a um Gerente funcional. Esse, por sua vez, designa alguém para atuar como gerente

do projeto. A equipe do projeto pode contar com pessoas subordinadas ao mesmo gerente funcional do gerente do projeto ou a algum outro gerente funcional. A alocação de recursos é negociada pelos gerentes funcionais.

O primeiro cenário mostra uma situação em que se conhece muito pouco sobre o domínio da aplicação (Turismo) e seus *stakeholders*. Esta é uma situação em que vários componentes serão utilizados para permitir a identificação, classificação e planejamento do envolvimento com os *stakeholders*.

No segundo cenário, a solicitação é feita por outra organização, porém no mesmo domínio (Turismo). Trata-se neste caso de uma prefeitura que deseja criar informações turísticas para o sítio da prefeitura.

No terceiro cenário voltamos à mesma organização do cenário 1 porém para o desenvolvimento de outra aplicação. Esta nova aplicação também está no domínio de Turismo.

Em todos os casos, o objetivo final é criar um artefato: o Modelo de Envolvimento dos *Stakeholders*.

A aplicação do meta-método depende da obtenção das informações necessárias para o processo de seleção dos componentes.. Essas informações podem ser divididas em dois grupos: (1) informações necessárias para começar a fase preliminar e (2) informações obtidas na fase preliminar que serão utilizadas na fase de planejamento para gerar o Modelo de Envolvimento.

3.1 Evento inicial do processo (comum a todos os cenários)

Nos cenários que serão considerados, a solicitação formal para o desenvolvimento de qualquer serviço deve seguir um padrão estabelecido pelo órgão de informática.

Este documento precisa conter, no mínimo, as seguintes informações:

- Identificação do solicitante. A identificação é composta por nome, cargo, nome do órgão, vinculação com órgãos de nível superior, endereço, telefone e e-mail da pessoa que está fazendo a solicitação. Opcionalmente pode ser indicada, também, uma pessoa para servir de contato sobre a solicitação (qualificada, também, com as informações mencionadas anteriormente).

- Identificação do pedido. Devem ser informados o tipo da aplicação (por exemplo, *web* ou *desktop*) e o número estimado de pessoas que usarão a aplicação.
- Identificação da razão do pedido. O pedido deve estar associado a algum programa ou projeto da organização ou a um de seus objetivos.

O recebimento desta solicitação leva a uma avaliação inicial cuja finalidade é decidir sobre (1) dar início à fase de iniciação do projeto, como foi descrito anteriormente, ou (2) rejeitar o pedido.

3.2 Cenário 1: Sistema para órgão estadual de Turismo

O órgão estadual de turismo enviou uma solicitação formal para o desenvolvimento de um sítio na *Web* sobre o Programa de Regionalização do Turismo no estado.

O primeiro passo é analisar o pedido original, como foi descrito anteriormente no item 4.1, para obter três informações iniciais: quem fez o pedido (Parte Interessada: Patrocinador), o que foi pedido (escopo provisório: um sítio na *Web*) e o por quê (Modelo de Motivação: metas e objetivos). Neste cenário a solicitação é aprovada e tem início a execução do grupo de processos de iniciação.

3.2.1 Iniciação

O objetivo dos processos de iniciação é a criação do Termo de Abertura de Projeto (TAP). A Figura 69 mostra a Estrutura Analítica de Processo para a Fase de Iniciação.

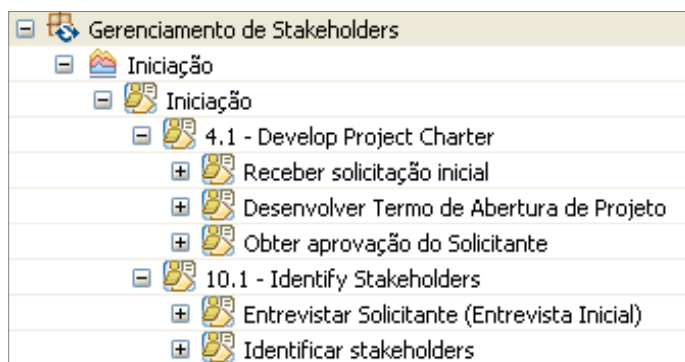


Figura 69 – Estrutura Analítica de Processos para o Grupo de Processos de Iniciação

A Figura 70 mostra o Diagrama de Atividades correspondente à Atividade 4.1 (*Develop Project Charter* (TAP: Termo de Abertura de Projeto)) da Figura 69.

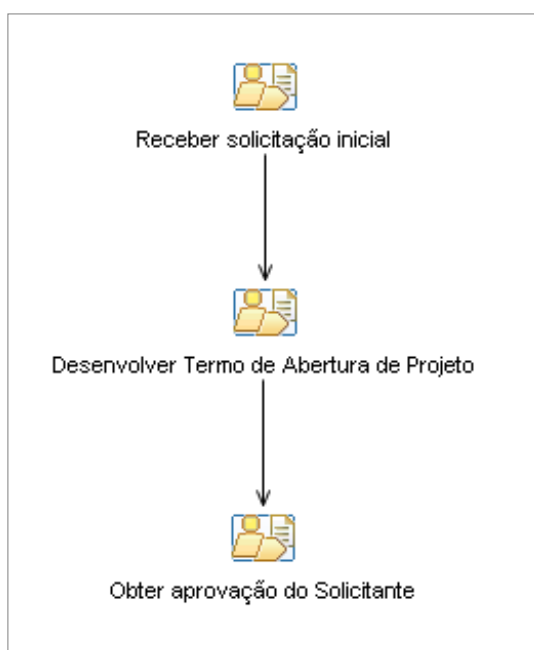


Figura 70 – Diagrama de Atividades para Criação de TAP (Termo de Abertura de Projeto)

Na etapa de iniciação, como pode ser visto na Figura 69, já tem início o processo de identificação dos *stakeholders*.

3.2.2 Planejamento

O objetivo principal da etapa de planejamento é a criação do Modelo de Envolvimento de *Stakeholders*.

Serão considerados três níveis de *stakeholders*: corporativo, negócio e projeto. O nível corporativo corresponde, neste cenário, aos governos federal e estadual. O nível de negócio corresponde a Turismo e o nível de projeto corresponde ao pedido inicial, um sítio sobre o Programa de Regionalização do Turismo.

A primeira decisão a ser tomada é quanto ao escopo do projeto, isto é, se, neste cenário, será necessária a criação de modelos para os três níveis de *stakeholders*. Como não existe, na organização solicitada, nenhuma pessoa com conhecimentos sobre os níveis mencionados anteriormente, deverão ser desenvolvidos modelos para os três níveis. Isto significa que será necessária a execução de três fases: uma para o nível corporativo, outra para

o nível do negócio e mais uma para o nível do projeto.

Neste cenário existe um prazo de duas semanas para a criação do Modelo de Envolvimento de *Stakeholders* e apenas um Analista de Negócios poderá ser alocado.

3.2.2.1 Nível corporativo

A Figura 71 mostra o Padrão de Recurso (*Capability Pattern*) para o nível corporativo. Ele é composto por três atividades que seguem o padrão do *Project Management Institute* (PMI) vistas anteriormente. O Padrão de Recursos (*Capability Patterns*) para os níveis de negócio e de projeto tem uma estrutura semelhante.

Presentation Name	Index	Predecessors	Model Info	Type	Planned
Nível corporativo	0			Capability Pattern	<input checked="" type="checkbox"/>
Planejamento	1			Activity	<input checked="" type="checkbox"/>
Execução	2			Activity	<input checked="" type="checkbox"/>
Monitoramento e controle	3			Activity	<input checked="" type="checkbox"/>

Figura 71 – Padrão de Recurso para o nível corporativo

A atividade de Planejamento, no nível corporativo, deve ser definida. A Figura 72 mostra as atividades possíveis no Padrão de Recurso.

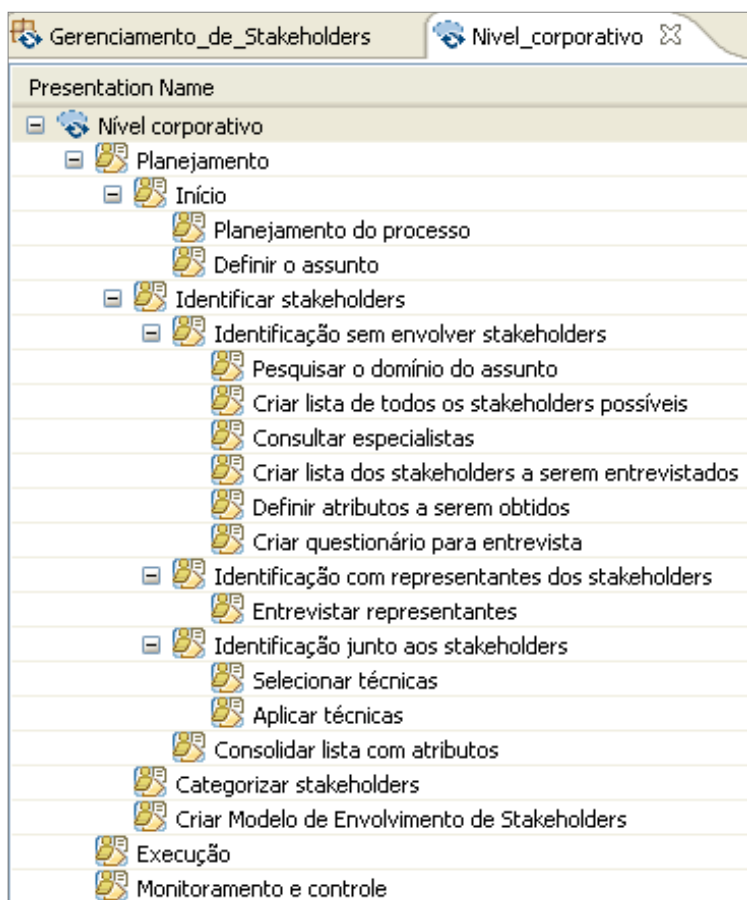


Figura 72 – Padrão de Recurso para atividades de planejamento para o nível corporativo

As atividades de Execução e Monitoramento e Controle seguem o estabelecido no Modelo de Envolvimento de Stakeholders.

3.2.2.2 Nível de negócio (Turismo)

A Figura 73 mostra a atividade de Planejamento para o nível de negócio. Como no nosso cenário o assunto está relacionado com Turismo a organização considerada é o órgão estadual de Turismo.

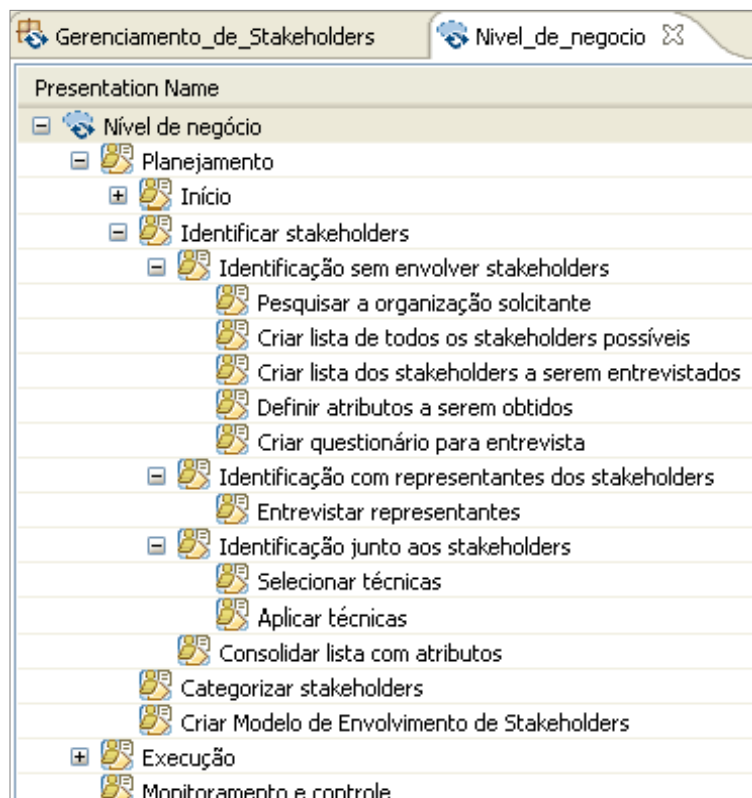


Figura 73 – Atividade de Planejamento para o nível de negócio

3.2.2.3 Nível de projeto (Regionalização do Turismo)

A Figura 74 mostra a atividade de Planejamento para o nível do projeto. No nosso cenário, a solicitação inicial é sobre o Programa de Regionalização do Turismo. Os *stakeholders*, em nível de projeto incluem, entre outros, os Analistas de Negócio, Analistas de Requisitos, Arquitetos de Sistema e os representantes da organização solicitante.

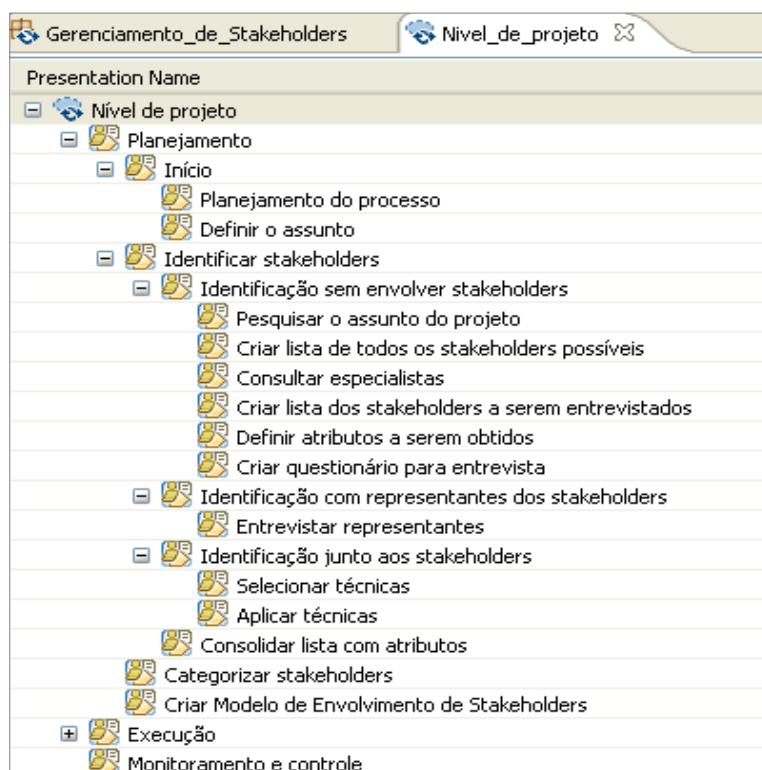


Figura 74 – Atividade de Planejamento para o nível de projeto

3.2.2.4 Criação do método

A Figura 75 mostra o método (chamado de Processo de Entrega ou *Delivery Process* no EPF Composer) que foi criado a partir dos três Padrões de Recurso vistos anteriormente.

Presentation Name	Index	Predecessors	Model Info	Type
Projeto de Regionalizacao do Turismo	0			Delivery Process
Fase Inicial	1			Capability Pattern
Nível corporativo	10			Capability Pattern
Nível de negócio	34			Capability Pattern
Nível de projeto	57			Capability Pattern

Figura 75 – Método para o Projeto de Regionalização do Turismo (resumido)

Na Figura 76 os componentes do método são mostrados com mais detalhes.

Presentation Name	Index
Projeto de Regionalizacao do Turismo	0
Fase Inicial	1
Elaborar documento inicial	2
+ Elaborar documento inicial	3
+ Outra atividade	8
Nível corporativo	10
Planejamento	11
+ Início	12
- Identificar stakeholders	15
+ Identificação sem envolver stakeholders	16
+ Identificação com representantes dos stakeholders	23
+ Identificação junto aos stakeholders	25
Consolidar lista com atributos	28
Categorizar stakeholders	29
Criar Modelo de Envolvimento de Stakeholders	30
+ Execução	31
Monitoramento e controle	33
Nível de negócio	34
Planejamento	35
+ Início	36
- Identificar stakeholders	39
- Identificação sem envolver stakeholders	40
Pesquisar a organização solicitante	41
Criar lista de todos os stakeholders possíveis	42
Criar lista dos stakeholders a serem entrevistados	43
Definir atributos a serem obtidos	44
Criar questionário para entrevista	45
+ Identificação com representantes dos stakeholders	46
+ Identificação junto aos stakeholders	48
Consolidar lista com atributos	51
Categorizar stakeholders	52
Criar Modelo de Envolvimento de Stakeholders	53
+ Execução	54
Monitoramento e controle	56
Nível de projeto	57
Planejamento	58
+ Início	59
- Identificar stakeholders	62
- Identificação sem envolver stakeholders	63
Pesquisar o assunto do projeto	64
Criar lista de todos os stakeholders possíveis	65

Figura 76 – Método para o Projeto de Regionalização do Turismo (expandido)

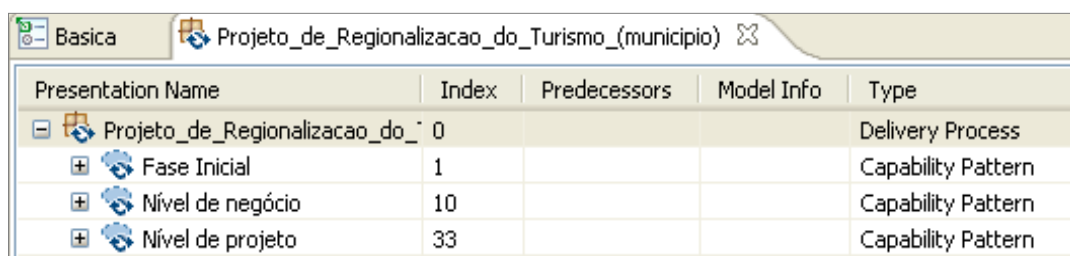
3.3 Cenário 2: Sistema para um órgão municipal de Turismo

O segundo cenário reutiliza muitas informações obtidas no primeiro cenário. Isto demonstra a vantagem de ter realizado o investimento inicial em conhecer os *stakeholders* para posteriormente realizar o trabalho mais rapidamente.

A prefeitura de um município do Estado solicitou que fossem criadas páginas, no sítio da prefeitura, com informações turísticas.

Neste cenário é possível reutilizar o Padrão de Recurso de nível corporativo pois ainda se trata do Turismo para órgão de governo. O Padrão de Recurso, em nível de negócio, no entanto, terá que ser criado pois trata-se de outra unidade de negócio: uma prefeitura.

O método criado terá a forma mostrada na Figura 77.



Presentation Name	Index	Predecessors	Model Info	Type
Projeto de Regionalizacao do_ 0				Delivery Process
+ Fase Inicial	1			Capability Pattern
+ Nível de negócio	10			Capability Pattern
+ Nível de projeto	33			Capability Pattern

Figura 77 – Método para o Projeto de Regionalização do Turismo (município)

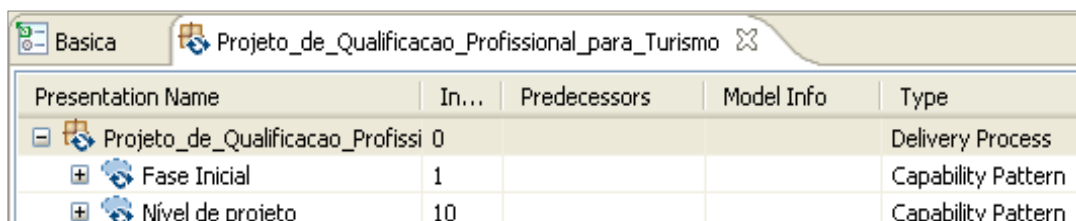
3.4 Cenário 3: Segundo sistema para o órgão estadual de Turismo

O terceiro cenário também é para o órgão estadual de turismo e, portanto, também reutiliza muitas informações obtidas no primeiro cenário. Isto demonstra, novamente, a vantagem de documentar o conhecimento obtido em um projeto para que possa ser utilizado novamente nos projetos posteriores..

O órgão estadual enviou um pedido para a criação de infraestrutura para um dos aspectos do Plano de Regionalização que é a capacitação de mão-de-obra especializada em funções associadas ao turismo.

Neste caso, é necessário criar um método apenas para o nível de projeto pois os *stakeholders* dos níveis corporativo e de negócio já são conhecidos.

A Figura 78 mostra o método criado.



Presentation Name	In...	Predecessors	Model Info	Type
Projeto de Qualificacao Profissi 0				Delivery Process
+ Fase Inicial	1			Capability Pattern
+ Nível de projeto	10			Capability Pattern

Figura 78 – Método para o Projeto de Qualificação Profissional para Turismo

3.5 Considerações

Os cenários ilustram a maneira como a reutilização das informações sobre *stakeholders* pode reduzir o tempo necessário para alcançar o envolvimento dos *stakeholders*. O investimento inicial para obter as informações é justificado, a longo prazo, pela reutilização das informações.

É possível criar mais duas atividades: (1) Validar Informações e (2) Atualizar Informações. Estas duas atividades iriam garantir que as informações estariam sempre corretas.

4 CONCLUSÃO

Um problema recorrente no desenvolvimento de sistemas é obter a participação efetiva de todos os envolvidos. Para obter este envolvimento é necessário mais do que apenas simpatia e boa vontade, é preciso seguir uma orientação sistemática que procure garantir o bom resultado do esforço dispendido. É importante, portanto, seguir um método durante o processo de desenvolvimento de sistemas. Como cada projeto é diferente do outro, surge a necessidade de usar um método adequado para cada um. No entanto criar um método, totalmente novo, consome recursos que poderiam ser melhor utilizados de outra forma. Por outro lado, a simples reutilização de métodos criados anteriormente, não atenderia aos aspectos específicos de cada situação. A solução, adotada pela Engenharia de Métodos Situacionais (e apresentada na seção 1.2) é a divisão de métodos em componentes, que podem ser combinados e/ou alterados para atender às necessidades do momento.

Para obter compatibilidade entre os componentes de métodos é utilizado o padrão SPEM do OMG. Na seção 1.2.4 é apresentado um resumo do padrão, destacando o seu aspecto modular que permite a utilização de níveis de conformidade mais limitados ou mais abrangentes, dependendo da necessidade de cada organização e de cada projeto.

O padrão SPEM foi escolhido, também, por possuir uma ferramenta para a criação de métodos, o EPF Composer, que é apresentado na seção 1.2.5 e utilizado para a construção do ZEP Framework.

O ZEP Framework é um artefato criado para atender às orientações da abordagem de *Design Science* (apresentada na seção 1.1) para o desenvolvimento de pesquisas em Sistemas de Informação. A primeira orientação é atendida pois, o ZEP Framework, instanciado na ferramenta EPF Composer mostrou-se um artefato viável. O planejamento e efetivo envolvimento dos *stakeholders* é um problema relevante para o desenvolvimento de sistemas, o que atende à segunda orientação. A criação de métodos nos cenários do capítulo 3 permitiu fazer uma avaliação do design, atendendo à terceira orientação. A quarta orientação também foi atendida, pois o ZEP Framework é uma clara contribuição, visto não houver outro *framework* criado especificamente para o envolvimento de *stakeholders* no ambiente de desenvolvimento de sistemas. A pesquisa seguiu rigorosos padrões de pesquisa, com indicação clara das fontes e do processo de construção de cenários, atendendo à quinta orientação. A sexta orientação foi atendida pois o ZEP Framework possui a flexibilidade

necessária para permitir melhorias. Finalmente, o resultado da pesquisa foi apresentado nesta dissertação tendo em visto o meio acadêmico, porém, a sua leitura pode ser realizada por profissionais da área sem prejuízo do seu entendimento.

O ZEP Framework foi apresentado no capítulo 2 e exemplos de como criar métodos no capítulo 3.

Quanto às contribuições da dissertação, é possível enumerar as seguintes:

1. A criação de um *framework* para permitir a criação de métodos específicos para cada situação em que haja a necessidade de se criar um Método de Envolvimento de *Stakeholders*.
2. Uma segunda contribuição é a definição de critérios específicos para a seleção e adequação dos componentes de métodos usados em Métodos de Envolvimento de *Stakeholders*.
3. A terceira contribuição é a apresentação de um exemplo prático (o ZEP Framework) de utilização do EPF Composer (e do SPEM) para a definição de um *framework* de componentes para a criação de métodos.

Uma limitação, quanto ao método empregado na dissertação, é que como a pesquisa tratou apenas de cenários em um domínio de aplicação, não é possível fazer generalizações quanto à sua aplicabilidade para outras situações.

Uma segunda limitação é a de o método ter sido aplicado por apenas um pesquisador, o autor da dissertação. Com isso, não se pode garantir que o método irá funcionar independentemente de suas características pessoais.

Um próximo passo, que será realizado em breve, é a aplicação do *framework* e do processo de criação de métodos em situações reais. O método foi construído, desde o início, com a preocupação de ser geral o bastante para ser reutilizado rotineiramente. Pretende-se relatar futuras aplicações através de artigos e apresentações em conferências.

Outra pesquisa a ser feita é a utilização do método por outros analistas de negócios. Como um dos objetivos do trabalho é a criação de um método a ser empregado pelo órgão de informática onde o autor trabalha, esta verificação do método deverá ser realizada em breve e seu resultado também será divulgado através de artigos e apresentações em conferências.

REFERÊNCIAS

ACCOUNTABILITY. *AA1000 Stakeholder Engagement Standard 2011 : Final Exposure Draft*. London. 2011. Disponível em < <http://www.accountability.org/standards/aa1000ses/index.html> >. Acessado em 27 de setembro de 2011.

ALEXANDER, I.; BEUS-DUKIC, L.. *Discovering Requirements – How to Specify Products and Services*. Wiley. Chichester, England. 2009.

ARNSTEIN, S.. *A Ladder of Citizen Participation*. Journal of the American Institute of Planners (JAIP). Vol. 35, No. 4, July, pp. 216-224. 1969.

BECKER, J; JANIESCH, C.; PFEIFFER, D.. *Reuse Mechanisms in Situational Method Engineering in IFIP International Federation for Information Processing*, 2007, Volume 244/2007, pp 79-93. *Situational Method Engineering: Fundamentals and Experiences*. Springer, 2007

BABIUCH, W; FARHAR, B.. *Stakeholder Analysis Methodologies Resource Book*. National Renewable Energy Laboratory. Golde, Colorado, USA. 1994.

BRASIL, Ministério do Turismo. *Programa de Regionalização do Turismo – Roteiros do Brasil: Diretrizes Políticas*. Brasília: Ministério do Turismo, 2004.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil. Diretrizes Operacionais*. Brasília: Ministério do Turismo, 2004b.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 1 - Sensibilização*. Brasília: Ministério do Turismo, 2004c.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 2 - Mobilização*. Brasília: Ministério do Turismo, 2004c.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 3 – Institucionalização de Instância de Governança Regional*. Brasília: Ministério do Turismo, 2005a.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 4 – Elaboração do Plano Estratégico de Desenvolvimento do Turismo Regional*. Brasília: Ministério do Turismo, 2006a.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 5 – Implementação do Plano Estratégico de Desenvolvimento do Turismo Regional*. Brasília: Ministério do Turismo, 2006b.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 6 – Sistema de Informações Turísticas do Programa*. Brasília: Ministério do Turismo, 2006c.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 7 – Roteirização Turística*. Brasília: Ministério do Turismo, 2005b.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 8 – Promoção e Apoio à Comercialização*. Brasília: Ministério do Turismo, 2006d.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Módulo Operacional 9 – Monitoria e Avaliação do Turismo*. Brasília: Ministério do Turismo, 2007a.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Formação de Redes*. Brasília: Ministério do Turismo, 2005c.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Turismo e Sustentabilidade – Conteúdo Fundamental*. Brasília: Ministério do Turismo, 2006e.

_____. *Programa de Regionalização do Turismo – Roteiros do Brasil: Ação Municipal para a Regionalização do Turismo – Conteúdo Fundamental*. Brasília: Ministério do Turismo, 2007b

_____. *Plano Nacional do Turismo: Uma Viagem de Inclusão 2007-2010*. Brasília: Ministério do Turismo, 2007c.

BROOKS, F. *No Silver Bullet – Essence and Accident in Software Engineering*, IEEE Computer Vol. 20, No.4, pp.10-19. 1987. Also in *The mythical man-month : essays on software engineering*. 20th Anniversary Edition. pp. 179-203. Addison-Wesley. 1995.

CANADA (Health Council of Canada). *Primer on Public Involvement*, Toronto, Health Council of Canada. 2006.

COSENTINO, M.; GAGLIO, S.; HENDERSON-SELLERS, B.; SEIDITA, V. A *Metamodelling-based Approach for Method Fragment Comparison*. Proceedings of the 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 06) 2006. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.8507> >

DONALDSON, T.; PRESTON, L. E.. *The stakeholder theory of the corporation: concepts, evidence, and implications*. Academy of Management Review, Vol. 20, No. 1, p. 65-91. 1995

CREIGHTON, J. L.. *The Public Participation Handbook; Making Better Decisions Through Citizen Involvement*. Jossey-Bass, San Francisco, CA, USA, 2005.

CRONBACH, L. J.; MEEHL, P. E.. Construct Validity in Psychological Tests. *Psychological Bulletin*, Vol. 52, No. 4, pp 281-302, 1955. Disponível em <<http://psychclassics.yorku.ca/Cronbach/construct.htm>>.

ELLIOT, J.; HEESTERBEEK, S.; LUKENSMEYER, C. J.; SLOCUM, N.. *Participatory Methods Toolkit: A practitioner's manual*. King Boudain Foundation, Belgium. 2005.

FIRESMITH, D.; CAPELL, P.; FALKENTHAL, D.; HAMMONS, C. D.; LATIMER, D.; MERENDINO, T.. *The method framework for engineering system architectures*, Auerbach Publications. 2009

FERRATER-MORA, J. - *Diccionario de Filosofia* (5th Edition) Editorial Sudamericana, Buenos Aires, 1964.

FERREIRA, A.B.H. - *Novo Dicionário da Língua portuguesa, Segunda Edição*. Editora Nova Fronteira, Rio de Janeiro. 1986.

FRIEDMAN, A.S. ; MILES, S. - *Stakeholders: Theory and Practice*. Oxford University Press, New York, 2006.

FREEMAN, R. E.. *Strategic Management: A Stakeholder Approach*. Pitman, Boston, 1984.

GIL, A.C. *Como elaborar projetos de pesquisa*. 4^a edição São Paulo, Atlas, 2002.

HARMSSEN, F.; LUBBERS, I.; WIJERS, G.. *Success-driven selection of fragments for situational methods: The S3 model*, in: *Proceedings of the Second International Workshop on Requirements Engineering Requirements Engineering: Foundations of Software Quality*, K. Pohl and P. Peters, eds., Aachener Beitrage zur Informatik, Band 13, p. 104–115. 1995.

HEVNER, A.R.; MARCH, S.T.; PARK, J. *Design Science in Information Systems Research. MIS Quaterly, Vol. 28, No. 1, pp 75-105, March 2004*.

IEEE The Institute of Electrical and Electronics Engineers. *IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*,. New York, New York, USA. 2009

IIBA International Institute of Business Analysis. *A Guide to the Business Analysis Body of Knowledge (BABOK Guide) Version 2.0*. Toronto, Ontario, Canada. 2009

JIANG, L.. *A Framework for the Requirements Engineering Process Development*. Tese de doutorado. Department of Electrical and Computer Engineering, University of Calgary. Calgary, Alberta, Canada. 2005.

JOHNSON, G., SCHOLLES, K., WHITTINGTON, R. *Exploring Corporate Strategy, Eighth Edition*. Pearson Education, Harlow, England. 2008.

ELLIOT, J.; HEESTERBEEK, S.; LUKENSMEYER, C. J.; SLOCUM, N.. *Participatory Methods Toolkit: A practitioner's manual*. King Boudain Foundation, Belgium. 2005.

LEITE, J. C. S. P.; HADAD, G. D. S., DOORN, J. H.; KAPLAN, G. N. . *A Scenario Construction Process*. Requirements Engineering, Vol. 5, pp.38-61. 2000.

MARCH, S. T.; SMITH, G. F.. *Design and natural science research on information technology*. Decision Support Systems, Vol. 15, pp.251-66. 1995.

MARQUES, H. R.; MANFROI, J.; CASTILHO, M. A.; NOAI, M. L. *Metodologia da pesquisa e do trabalho científico*. 2ª edição revisada.. Campo Grande, Mato Grosso do Sul, Editora UCBD, 2006

MITCHELL, R. K.; AGLE, B. R.; WOOD, D. J.. *Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts*. The Academy of Management Review, Vol. 22, No. 4, pp. 853-886. 1997.

MITCHELL, R. K.; AGLE, B. R.; CHRISMAN, J. J.; SPENCE, L. J.. *Toward a Theory of Salience in Family Firms*. Business Ethics Quarterly, Vol. 21, No. 2, pp. 235-55. 2011.

NUSEIBEH, B; EASTERBROOK, S.. *Requirements Engineering: A Roadmap*. Proceedings of International Conference on Software Engineering (ICSE-2000), 4-11 June 2000, Limerick, Ireland, ACM Press. 2000.

OECD. *Engaging Citizens Online for Better Policy-making*. Public Affairs Division, Public Affairs and Communications Directorate, OECD, Paris, 2003.
<<http://www.oecd.org/dataoecd/62/23/2501856.pdf>>

OMG. *Software & Systems Process Engineering Meta-Model Specification (Version 2.0)*. Object Management Group, USA. 2008.

OMG. *OMG Unified Modeling Language (OMG UML), Infrastructure (Version 2.4.1)*. Object Management Group, USA. 2011.

PATTON, M Q.. *Qualitative evaluation and research methods (Third Edition)*. Sage Publications, Newbury Park, California, USA. 2002

PMI.. *A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Third Edition*. Project Management Institute Inc.. Newtown Square, Pennsylvania, USA. 2004a.

PMI.. *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK) – Terceira Edição*. Project Management Institute Inc.. Newtown Square, Pennsylvania, USA. 2004b.

PMI. *A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Fourth*

Edition. Project Management Institute Inc.. Newtown Square, Pennsylvania, USA. 2008a.

PMI. *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK) – Quarta Edição*. Project Management Institute Inc.. Newtown Square, Pennsylvania, USA. 2008b.

RUNESON, P.; HÖST, M.. *Guidelines for conducting and reporting case study research in software engineering*. Empirical Software Engineering. Volume 14, pp 131-164, 2009.

SAVAGE, G. T.; NIX, T. W.; WHITEHEAD, C. J.; BLAIR, J. D.. *Strategies for assessing and managing organizational stakeholders*. Academy of Management Executive, Vol. 5, No. 2. pp 61-75, 1991.

SCHMEER, K.. *Stakeholder Analysis Techniques*. Chapter 2 pp 2-1 to 2-43 in Policy Toolkit for Strengthening Health Sector Reform, Latin America and Caribbean Regional Health Sector Reform Initiative, sem local, 2000. Disponível em <http://www.healthsystems2020.org/content/resource/detail/1004/> >

SEBRAE. *Turismo no Brasil: Termo de Referência para a Atuação do Sistema SEBRAE*. Brasília. 2010. 43 p.

SILVA, E. L.; MENEZES, E.M. *Metodologia da pesquisa e elaboração de dissertação*. 3ª edição revisada e atualizada. Florianópolis, Laboratório de Ensino a Distância da UFSC, 2001. 121 p.

SIMON, H. A.. *The Sciences of the Artificial*. Third edition. Cambridge, Massachusetts, The MIT Press. 1996.

TOGAF. *TOGAF Version 9: The Open Group Architecture Framework (TOGAF)*. The Open Group. 2009. Disponível em < <http://www.opengroup.org/togaf9/downloads.htm> >.

TUFT, B. - *Eclipse Process Framework (EPF) Composer: Installation, Introduction, Tutorial and Manual*. Disponível em < http://www.eclipse.org/epf/general/EPF_Installation_Tutorial_User_Manual.pdf >. Acessado em 10 agosto de 2010.

UNEP (United Nations Environment Programme), SRA (Stakeholder Research Associates) and AccountAbility. *From Words to Action: The Stakeholder Engagement Manual. Volume 2: The Practitioner's Handbook on Stakeholder Engagement*. 2005. Disponível em < <http://www.unep.fr/shared/publications/pdf/WEBx0115xPA-SEhandbookEN.pdf> >.

VICTORIA, STATE GOVERNMENT OF VICTORIA STATE, DEPARTMENT OF SUSTAINABILITY AND ENVIRONMENT – *Effective Engagement: building relationships with community and other stakeholders. Book 1: an introduction to engagement. Version 3* Melbourne, Australia. 2005. Disponível em <<http://www.dse.vic.gov.au/effective-engagement/resources/download-effective-engagement> >. Acessado em 23 de agosto de 2011.

VICTORIA, STATE GOVERNMENT OF VICTORIA STATE, DEPARTMENT OF SUSTAINABILITY AND ENVIRONMENT – *Effective Engagement: building relationships*

with community and other stakeholders. Book 2: the engagement planning workbook. Version 3 Melbourne, Australia. 2005. Disponível em <<http://www.dse.vic.gov.au/effective-engagement/resources/download-effective-engagement>>. Acessado em 23 de agosto de 2011.

VICTORIA, STATE GOVERNMENT OF VICTORIA STATE, DEPARTMENT OF SUSTAINABILITY AND ENVIRONMENT – *Effective Engagement: building relationships with community and other stakeholders. Book 3: the engagement toolkit. Version 3* Melbourne, Australia. 2005. Disponível em <<http://www.dse.vic.gov.au/effective-engagement/resources/download-effective-engagement>>. Acessado em 23 de agosto de 2011.

WALLS, J.; WIDMEYER, G.; EL SAWY, O.. *Building an Information System Design Theory for Vigilant EIS*. Information Systems Research, Vol. 3, No. 1, pp 36-59, 1992

WALLS, J.; WIDMEYER, G.; EL SAWY, O.. *Assessing Information System Design Theory in Perspective: How Useful was our 1992 Initial Rendition?*. Journal of Information Technology Theory and Application (JITTA), Vol.6, No. 2, pp 43-58, 2004

GLOSSÁRIO

Atividade (Activity)

1 - “Uma Atividade é um *Work Breakdown Element* e *Work Definition*. que define uma unidade básica de trabalho dentro de um Processo, assim como um Processo como um todo. Em outras palavras, uma Atividade representa um Processo no SPEM 2.0.” (OMG, 2008, p. 46)

2 - “Atividades são o principal bloco de construção de Processos. Uma atividade é um conjunto de *Work Breakdown Elements* tais como Descritores de Tarefas, Descritores de Papéis, Descritores de Produtos de Trabalho e Marcos. Atividades podem conter outras Atividades”. (TUFT, 2010, p. 270)

Activity Kind

Ver Tipo de Atividade.

Artefact (ou Artifact)

Ver Artefato.

Artefato (Artefact)

Um Artefato é representado por um *Artefact Definition*.

Artefact Definition

Ver Definição de Artefato.

Classificador (Classifier)

“Um Classificador (Classifier) é uma classificação de instâncias – descreve um conjunto de instâncias que tem características em comum” (OMG, 2011, p. 35).

Classifier

Ver Classificador.

Conteúdo de Método (*Method Content*)

“Pacote que define os elementos centrais (*core*) dos métodos tais com Papéis, Tarefas e Definições de Produtos de Trabalho” (OMG, 2008, p. 81). Não contém Processos. Não confundir com Componente de Método.

Componente de Metamétodo (*Metamethod Component*)

“Um tipo de Componente de Método que é parte de um Metamodelo de Engenharia de Processo. (FIRESMITH et al., 2009, p. 55).

Componente de Método

“Componentes de método também são conhecidos como *method fragments* e *method chunks* na comunidade de engenharia de métodos situacionais. O termo componente foi selecionado para enfatizar a relação próxima entre engenharia de métodos e a engenharia baseada em componentes, que é bem conhecida dentro da comunidade de engenharia de arquitetura de sistemas. O termo *chunk* foi rejeitado por ser muito informal e o termo *fragment* foi rejeitado pois implica em uma decomposição destrutiva em vez de uma composição construtiva”, (FIRESMITH, 2009, p. Xxvii)

“Uma parte componente de um método de engenharia. O termo '*engineering method*' é chamado de '*method component*' na Figura [6 da dissertação] e no resto do livro”.(FIRESMITH, 2009, p. 55)

No ZEP Framework, o termo é usado de uma forma mais abrangente e representa qualquer parte de um método completo (*Delivery Process* na terminologia do SPEM)

Componente de Processo

Uma parte componente de um Processo de Engenharia e uma instância de um Componente de Método (*Method Component*). O termo “*engineering process component*” é chamado de Componente de Processo “*processs-component*” na Figura [6 da dissertação]) e no resto do livro. (FIRESMITH et al., 2009, p. 54)

Definição de Artefato (*Artefact Definition*)

“Uma Definição de Artefato (*Artefact Definition*) é um *Work Product Definition* que fornece uma descrição e definição para tipos de Produtos de Trabalho tangíveis. Artefatos podem ser compostos por outros Artefatos”.(OMG, 2008, p. 166)

Definição de Papel (*Role*)

“Uma Definição de Papel (*Role*) é um *Method Content Element* que define um conjunto de habilidades, competências e responsabilidades interrelacionadas. Papéis são usados por Definições de Tarefas para definir quem as executa assim como para definir um conjunto de Definições de Produtos de Trabalho pelos quais são responsáveis” (OMG, 2008, p. 87)..

Definição de Produto de Trabalho (*Work Product Definition*)

“Uma Definição de Produto de Trabalho é um *Method Content Element* que é usado, modificado e produzido por Definições de Tarefas”.(OMG, 2008, p. 92)

Definição de Tarefa (*Task Definition*)

“Uma Definição de Tarefa é um *Method Content Element* e um *Work Definition* que define o trabalho realizado por instâncias de *Role Definition*. Uma Tarefa está associada a Produtos de Trabalho de entrada e saída”. (OMG, 2008, p. 89).

Não confundir com Atividade. Uma Atividade está sempre associada a um Processo (e é, ela própria um Processo) enquanto que uma Tarefa é uma definição de Conteúdo de Método que pode ser reutilizada em várias Atividades (através da classe *Task Use* (Uso de Tarefa)).

Definição de Trabalho (*Work Definition*)

“Uma Definição de Trabalho é um *Classifier* abstrato que generaliza todas as definições de trabalho dentro do SPEM 2.0” (OMG, 2008, p. 38)

Definição de Produto de Trabalho (*Work Product Definition*)

“Uma Definição de Produto de Trabalho é um *Method Content Element* que é usado, modificado e produzido por Definições de Tarefas” (OMG, 2008, p.92).

Deliverable

Ver Entrega.

Delivery Process

Ver Processo de Entrega.

Descritor de Papéis (*Role Descriptor*)

Um Descritor de Papel, no EPF Composer, é a representação de um Papel dentro do contexto de um Processo. Chamado de *Role Use* no SPEM.

Descritor de Produto de Trabalho (*Work Product Descriptor*)

Um Descritor de Produto de Trabalho, no EPF Composer, é a representação de um Produto de Trabalho dentro do contexto de um Processo. Chamado de *Work Product Use* no SPEM.

Descritor de Tarefa (*Task Descriptor*)

Um Descritor de Tarefa, no EPF Composer, é a representação de uma Tarefa dentro do contexto de um Processo. Chamado de *Task Use* no SPEM.

Elemento Estendível (*Extensible Element*)

Um Elemento Estendível (*Extensible Element*) é uma generalização abstrata que representa qualquer classe do SPEM 2.0 para a qual é possível atribuir um Tipo (*Kind*) para suas instâncias, expressando uma qualificação definida pelo usuário (OMG, 2008, p. 36).

Elemento de Processo (*Process Element*)

Um Elemento de Processo (*Process Element*) é um Elemento Estendível (*Extensible Element*) que representa uma generalização abstrata para todos os elementos que fazem parte de um Processo (*Process*) no SPEM 2.0 (OMG, 2008, p. 56).

Elemento de Quebra (*Breakdown Element*)

Um Elemento de Quebra (*Breakdown Element*) é uma generalização abstrata para qualquer tipo de *Process Element* que é parte de uma estrutura de quebra (*breakdown structure*) (OMG, 2008, p. 54).

Elemento de Quebra de Trabalho (*Work Breakdown Element*)

Um Elemento de Quebra de Trabalho é um *Breakdown Element* especial que fornece propriedades específicas para *Breakdown Elements* que representam trabalho. (OMG, 2008, p. 60)

Engineering Method

Ver Método (FIRESMITH).

Engineering Method Component

Ver Componente de Método (FIRESMITH).

Engineering Process

Ver Processo (FIRESMITH).

Engineering Process Component

Ver Componente de Processo (FIRESMITH).

Engineering Process Metamodel

Ver Metamodelo de Componente de Processo (FIRESMITH).

Entrega (Deliverable)

Uma Entrega é um *Work Product Definition* que fornece uma descrição e definição para empacotar outros Produtos de Trabalho e pode ser entregue para alguém interno ou externo. Uma Entrega agrega outros *Work Products*. (OMG, 2008, p. 166)

Extensible Element

Ver Elemento Extendível.

Iteration

Ver Iteração.

Iteração (Iteration)

Uma Iteração agrupa um conjunto de Atividades aninhadas (nested) que podem ser repetidas mais de uma vez (OMG, 2008, p. 156).

Kind

Ver Tipo.

Marco (*Milestone*)

Um Marco é um *Work Breakdown Element* que representa um evento significativo para um projeto de desenvolvimento. (OMG, 2008, p. 55)

Metamethod Component

Ver Componente de Metamétodo.

Metamodelo de Componente de Processo (*Engineering Process Metamodel*)

“Um modelo de Modelo de Engenharia de Processo (engineering process model) que define os tipos abstratos de Componentes de Método e seus relacionamentos. O termo “engineering process metamodel” é chamado de “process metamodel” na Figura [6 da dissertação] e no resto do livro” (FIRESMITH et al., 2009, p. 55).

Method Content

Ver Conteúdo de Método.

Método (*Method*)

“Um Modelo de Processo que documenta sistematicamente a maneira como uma ou mais disciplinas de engenharia devem ser realizadas. O termo “*engineering method*” é chamado de “*as-intended method*” e “*process model*” na Figura [6 da dissertação] e será chamado de “método” no resto do livro. (FIRESMITH et al., 2009, p. 55)

Milestone

Ver Marco.

Modelo de Processo

Ver Método.

Outcome

Ver Resultado.

Papel (Role)

Ver Definição de Papel.

Process

Ver Processo.

Process Element

Ver Elemento de Processo.

Processo (Process)

1 - Um Processo é uma Atividade especial que descreve uma estrutura para tipos específicos de projetos de desenvolvimento ou partes dele. Para realizar um tal projeto, um Processo seria adaptado para a situação específica do projeto ou organização e então “instanciado” pela atribuição de recursos concretos para Usos de Papel, criando várias instâncias de Usos de Produto de Trabalho, etc. (OMG, 2008, p. 156).

2 - [Um Processo pode ser considerado como sendo] (i) Uma sequência de etapas realizada com um propósito; por exemplo, o processo de desenvolvimento de software. (ii) Uma unidade de execução gerenciada pelo escalador (*scheduler*) de um sistema operacional. Ver também: tarefa (*task*), trabalho (*job*). (iii) Realização de operações sobre dados. (IEEE, 1990, p. 57)

3 - Um Processo de Engenharia (*Engineering Process*) é a maneira real como uma ou mais disciplinas são realizadas na prática de um projeto real. O termo “*engineering process*” é chamado de “*as-performed process*” [na Figura 6 da dissertação] e será chamado de “*process*” no resto do livro. (FIRESMITH et al., 2009, p. 54).

Processo de Entrega (*Delivery Process*)

Um Processo de Entrega é um Processo que cobre integralmente um ciclo-de-vida de desenvolvimento do início ao fim. Um Processo de Entrega deve ser usado como modelo (*template*) para planejar e executar um projeto (OMG, 2008, p. 157).

Produto de Trabalho (*Work Product*)

Ver Definição de Produto de Trabalho.

Resultado (*Outcome*)

Um Resultado (Outcome) é um Work Product Definition que fornece uma descrição e definição para Produtos de Trabalho intangíveis (OMG, 2008, p. 165)

Role Use

Ver Uso de Papel.

Role

Ver Papel.

Stakeholder

“Todos os grupos e indivíduos que podem afetar, ou serem afetados, pela realização do propósito da organização” (FREEMAN, 1984, p. 25)

Task Definition

Ver Definição de Tarefa.

Task Use

Ver Uso de Tarefa.

Tipo (*Kind*)

Um Tipo é um Elemento Estendível (*Extensible Element*). Suas instâncias são usadas para qualificar outros Elementos Extensíveis do SPEM 2.0 com um tipo definido pelo usuário. (OMG, 2008, p. 37)

Tipo de Atividade (*Activity Kind*)

Tipos de Atividade fornecem a possibilidade para um Engenheiro de Processos de definir modelos de ciclo-de-vida que usam a terminologia com a qual está acostumado. Por exemplo (...) “Fase” e “Iteração” (...). (OMG, 2008, p. 155).

Uso de Papel

Um Uso de Papel representa um Papel no contexto de uma Atividade específica. Toda estrutura de quebra [*WBS, Work Breakdown Structure*] pode definir relacionamentos diferentes entre Usos de Papéis e Usos de Produto de Trabalho. (OMG, 2008, p.110)

Uso de Produto de Trabalho

Um Uso de Produto de Trabalho representa uma Definição de Produto de Trabalho no contexto de uma Atividade específica. Toda estrutura de quebra [*WBS: Work Breakdown Structure*] pode definir relacionamentos diferentes entre Usos de Produtos de Trabalho e Uso de Papéis.(OMG, 2008, p. 116).

Uso de Tarefa (*Task Use*)

Um Uso de Tarefa (*Task Use*) é um *Method Content* e *Work Breakdown Element* que representa uma [classe] intermediária (*proxy*) para uma *Task Definition* dentro do contexto de uma Atividade específica. (OMG, 2008, p. 112).

Work Breakdown Element

Ver Elemento de Quebra de Trabalho.

Work Product Use

Ver Uso de Produto de Trabalho.

Work Product

Ver Produto de Trabalho.

Work Definition

Ver Definição de Trabalho.

APÊNDICE A – Exemplo de descrição dos componentes do *framework*

Neste capítulo serão apresentados exemplos dos componentes de método que são utilizados para a criação dos métodos específicos para cada situação.

A.1 – Abertura do projeto

Outros nomes:

Project Charter – PMI, 2008

Issues Note – UNDP, 200x

..

Descrição:

A Abertura do Projeto é usada para iniciar formalmente o projeto de criação do Gerenciamento de *Stakeholders*.

Inclui a definição de, pelo menos, três aspectos: (1) quem está envolvido (*stakeholders*), (2) o que deve ser realizado (*scope*) e (3) porque será realizado (*motivation*).

No momento da abertura estes três aspectos são muito gerais e serão detalhados no decorrer do projeto.

Objetivos:

O principal objetivo é formalizar o início do projeto, especificando de forma ampla seus aspectos principais

Resultados (*outcomes*):

Comprometimento dos envolvidos quanto à necessidade de alocar recursos para o projeto..

Comprometimento dos envolvidos quanto aos objetivos do projeto.

Artefatos:

- Documento formal de início do projeto.

Entregas:

- Não se aplica.

Usos/aspectos positivos:

- Tornar claro, de forma preliminar, os objetivos, envolvidos e abrangência do projeto
- Cria o comprometimento com a realização do projeto.

Considerações especiais/aspectos negativos:

- Uma definição preliminar incorreta pode levar ao fracasso do projeto.

Recursos materiais necessários:

- Mecanismo de comunicação formal (como exemplos: Comunicação Interna, Memorando, Ofício)

Recursos humanos (*roles*):

- Analista de negócio.
- *Stakeholder* solicitante (*sponsor*)
- Gerente de projeto
- Responsável pelo atendimento

Pode ser usado para:

- Início formal do projeto.

Quantidade de pessoas necessárias:

- Pequena (1-4 pessoas)

Quantidade de pessoas alcançadas:

- Não se aplica

Tempo necessário (cronômetro):

- Pequeno (< 8 horas)

Tempo necessário (calendário):

- Pequeno (< 1 dia)

Nível de habilidade/suporte necessário:

- Alto (Pessoas com poder para dar início ao projeto)

Custo:

- Pequeno (depende do negócio)

Nível de participação (IAP2)

- Não se aplica

Nível de inovação:

- Pequeno (Tradicional)

Etapas:

1. Patrocinador(*Sponsor*) redige e enviar solicitação de serviço.
2. Responsável pelo Atendimento redige, junto com o futuro Gerente de Projeto, o Documento Inicial do projeto com a possível ajuda de um Analista de Negócio.
3. Responsável pelo Atendimento envia o Documento para o Patrocinador(*Sponsor*)
4. Patrocinador(*Sponsor*) aprova o documento e envia para o Responsável pelo Atendimento.

Referências:

ACCOUNTABILITY, 2011, p. 16-21

PMI, 2008

A.2 – Pesquisa de material secundário**Descrição:**

A Pesquisa de Material Secundário procura descobrir, antes mesmo de entrar em contato com algum *stakeholder*, informações sobre a organização que fez o pedido de serviço e sobre a sua área de atuação.

Estas informações podem ser obtidas em sites na Web, livros, artigos de revistas populares ou científicas e de quaisquer outras fontes que possam ser úteis.

Objetivos:

O principal objetivo é familiarizar o Analista de Negócio com a organização solicitante e com a domínio do problema.

Resultados (*outcomes*):

O Analista de Negócio adquire conhecimentos básicos sobre a área da aplicação e sobre possíveis *stakeholders*.

Artefatos:

- Documentação sobre o domínio de conhecimento
- Documentação sobre a organização solicitante
- Lista de *stakeholders* (papéis genéricos).

Entregas:

- Não se aplica.

Usos/aspectos positivos:

- Permite ao Analista de Negócio um contato inicial com a terminologia do domínio.
- Ajuda a identificar os possíveis papéis (*roles*) dos *stakeholders*.
- Não depende de reuniões e, conseqüentemente, da alocação de tempo do Patrocinador.

Considerações especiais/aspectos negativos:

- É necessário um limite de tempo para que a pesquisa não se estenda muito.
- A escolha das fontes iniciais é muito importante.
- Procurar delimitar o escopo da pesquisa tomando como base a solicitação inicial.

Recursos materiais necessários:

- Internet
- Bibliotecas
- Pesquisas feitas para outras solicitações da mesma área de conhecimento

Recursos humanos (*roles*):

- Analista de Negócio.

Pode ser usado para:

- Familiarização com os problemas da área do Patrocinador
- Familiarização com a terminologia (jargão) do Patrocinador
- Familiarização com a estrutura organizacional (organograma) do Patrocinador.

Quantidade de pessoas necessárias:

- Pequena (1-2 pessoas)

Quantidade de pessoas alcançadas:

- Não se aplica

Tempo necessário (cronômetro):

- Médio (8-32 horas)

Tempo necessário (calendário):

- Pequeno (< 1 semana)

Nível de habilidade/suporte necessário:

- Médio (Habilidade de pesquisa e de redação de relatório)

Custo:

- Pequeno (depende do negócio)

Nível de participação (IAP2)

- Não se aplica

Nível de inovação:

- Pequeno (Tradicional)

Etapas:

1. Pesquisar a organização solicitante na Web..
2. Pesquisar a área de atuação da organização solicitante na Web..
3. Pesquisar o tipo de problema que faz parte da solicitação inicial na Web.

4. Pesquisar o nome da pessoa que fez a solicitação na Web.
5. Complementar as informações com a leitura de artigos e livros relacionados aos resultados dos itens anteriores
6. Registrar o que foi descoberto

Referências:

NREL, p. 23-4

A.3 – Amostragem Bola de Neve (*Snowball Sampling*)**Descrição:**

A Amostragem Bola-de-Neve tem esse nome pois é semelhante a uma bola de neve quando rola montanha abaixo. Alguns *stakeholders* iniciais são contactados e eles indicam outros *stakeholders*. Estes, por sua vez, indicam outros e assim por diante.

O pedido de indicação dos outros *stakeholders* deve esclarecer as características desejadas dos novos *stakeholders* como, por exemplo, nível de conhecimento do domínio, poder de decisão, antagonistas e outros.

Normalmente a indicação é apenas uma pergunta dentro de um conjunto mais amplo.

Objetivos:

O principal objetivo é identificar pessoas com determinadas características que deverão participar do projeto.

Resultados (*outcomes*):

Permite identificar as pessoas com características que ajudarão o projeto, bem como suas características relevantes. Também permite identificar os que são contra o projeto e suas características.

Artefatos:

- Lista de *stakeholders* (nomes e características relevantes)

Entregas:

- Não se aplica.

Usos/aspectos positivos:

- Ajuda a identificar novos *stakeholders* de forma rápida.
- Usa os recursos de redes já existentes.

Considerações especiais/aspectos negativos:

- A escolha dos contatos iniciais é muito importante.

Recursos materiais necessários:

- Telefone
- Meio de transporte até local da entrevista
- Material para registro (papel, caneta, fita de áudio)

Recursos humanos (*roles*):

- Entrevistadores

Pode ser usado para:

- Criar envolvimento das pessoas
- Descobrir habilidades e conhecimentos das pessoas

Quantidade de pessoas necessárias:

- Média (2–12 pessoas)

Quantidade de pessoas alcançadas:

- Grande (> 30)
- Média (11–30)

Tempo necessário (cronômetro):

- Pequeno (5 minutos – 15 minutos, por indicação)
- Médio (15 minutos – 1 hora, por indicação)

Tempo necessário (calendário):

- Pequeno (< 6 semanas)

Nível de habilidade/suporte necessário:

- Médio (Habilidade verbal e para registrar)

Custo:

- Pequeno (depende do negócio)

Nível de participação (IAP2)

- Consultar
- Envolver

Nível de inovação:

- Pequeno (Tradicional)

Etapas:

1. Crie uma lista inicial (no mínimo com o solicitante do projeto).
2. Peça para que indique outros *stakeholders* e suas características relevantes para o projeto..
3. Se o número for suficiente, pare.
4. Se os nomes começarem a se repetir, pare.
5. Se o número não for suficiente volte para a etapa 2.

Referências:

AUSTRALIA, 2005, p. 83

PATTON, - *Qualitative evaluation and research methods (Third Edition)*, p. 237-8,

Sage Publications, Newbury Park, California, USA. 2002

A.4 – Determinar nível de poder (Schmeer)

Descrição:

Um dos principais critérios para classificar os *stakeholders* é o nível de poder que ele possui. Existem muitas maneiras de definir o poder.

Será utilizada a abordagem de Schmeer que usa como fundamento a quantidade de recursos e a capacidade de mobilização destes recursos pelo *stakeholder*

O nível de poder está relacionado ao projeto, ou seja, qual o nível de poder em relação aos objetivos do projeto.

Normalmente o poder é combinado com outros critérios para definir a prioridade e o tipo de envolvimento com o *stakeholder*.

Objetivos:

Associar um nível de poder (muito alto, alto, médio, baixo, muito baixo) a um *stakeholder*. O nível irá determinar, isoladamente ou em conjunto com outros fatores, a prioridade e o tipo de envolvimento com o *stakeholder*.

Resultados (*outcomes*):

Permite identificar um critério de importância (o poder) para um *stakeholder* do projeto.

Artefatos:

- Lista de *stakeholders* (nomes e nível de poder para o projeto)

Entregas:

- Não se aplica.

Usos/aspectos positivos:

- Classifica os *stakeholders*, o que permite determinar, direta ou indiretamente, o nível de envolvimento que terá no projeto..

Considerações especiais/aspectos negativos:

- A classificação envolve a avaliação de um conjunto de critérios, logo a subjetividade da classificação não pode ser evitada totalmente..

Recursos materiais necessários:

- Pesquisas, em fontes secundárias, sobre a quantidade de recursos do *stakeholder*.
- Pesquisas, em fontes secundárias, sobre a capacidade de mobilização de recursos do *stakeholder*.
- Gravador para a entrevista

Recursos humanos (*roles*):

- Analista de Negócios

Pode ser usado para:

- Classificar *stakeholders*.

Quantidade de pessoas necessárias:

- Pequena (1-2 pessoas)
- Média (2–12 pessoas)

Quantidade de pessoas alcançadas:

- Não se aplica.

Tempo necessário (cronômetro):

- Pequeno (5 minutos – 15 minutos, por *stakeholder*)
- Médio (15 minutos – 1 hora, por indicação)

Tempo necessário (calendário):

- Pequeno (< 6 semanas)

Nível de habilidade/suporte necessário:

- Médio (Capacidade de julgamento)

Custo:

- Pequeno (depende do negócio)

Nível de participação (IAP2)

- Consulta

Nível de inovação:

- Pequeno (Tradicional)

Etapas:

1. Obter lista criada anteriormente com dados básicos sobre *stakeholders*.
2. Para cada *stakeholder*, repetir todos os itens seguintes.
3. Verificar o tipo de recursos (humanos, financeiros e outros) que são controlados.
4. Avaliar quantidade de recursos disponíveis (onde
3 significa “muito”,
2 significa “algum” e
1 significa “pouco”).
5. Avaliar capacidade de mobilizar estes recursos (onde
3 significa “*stakeholder* pode tomar decisões sobre o uso dos recursos (...)”
2 significa “*stakeholder* é uma pessoa, entre outras, que pode tomar decisões (...)”
1 significa “*stakeholder* não pode tomar decisões sobre o uso dos recursos.”
6. Calcular a média entre quantidade e capacidade de mobilização (resultado ficará entre 1 e 3)
7. Ajustar, subjetivamente, para determinar um nível entre 1 e 5, onde
1 significa “muito baixo”
2 significa “baixo”
3 significa “médio”
4 significa “alto”
5 significa “muito alto”
8. Atribuir o nível para este *stakeholder*.

Obs.: O item 7 não faz parte do procedimento da referência (SCHMEER)

Referências:

SCHMEER, 2000, p. 2-17

A.5 – Determinar nível de posicionamento frente ao projeto (Schmeer)**Descrição:**

Nem todos os *stakeholders* são favoráveis ao projeto. É necessário portanto determinar qual o posicionamento de cada *stakeholder* quanto ao início e/ou prosseguimento do projeto, isto é, se ele apoia ou se opõe ao projeto.

Objetivos:

Associar um nível de apoio (muito favorável, favorável, neutro, contra, muito contra) a cada *stakeholder*. O nível irá determinar, isoladamente ou em conjunto com outros fatores, a prioridade e o tipo de envolvimento com o *stakeholder*.

Resultados (outcomes):

Permite identificar a posição do *stakeholder* em relação ao projeto.

Artefatos:

- Lista de *stakeholders* (nomes e nível de apoio)

Entregas:

- Não se aplica.

Usos/aspectos positivos:

- Classifica os *stakeholders*, o que permite determinar, direta ou indiretamente, o nível de envolvimento que terá..

Considerações especiais/aspectos negativos:

- A classificação se baseia em um julgamento de valor, logo a subjetividade da classificação não pode ser evitada totalmente.

- Resultado deve ser mantido em sigilo.
- Alguns *stakeholders* podem evitar julgar a posição dos outros.

Recursos materiais necessários:

- Pesquisas sobre o *stakeholder*.

Recursos humanos (*roles*):

- Analista de Negócios

Pode ser usado para:

- Classificar *stakeholders*.

Quantidade de pessoas necessárias:

- Pequena (1-2 pessoas)
- Média (2–12 pessoas)

Quantidade de pessoas alcançadas:

- Todos os *stakeholders*, ou seus representantes.

Tempo necessário (cronômetro):

- Pequeno (1 minutos – 5 minutos, avaliação do *stakeholder* sobre si mesmo)
- Médio (5 minutos – 10 minutos, avaliação de um *stakeholder* sobre outro)

Tempo necessário (calendário):

- Pequeno (< 6 semanas, depende da quantidade de *stakeholders*)

Nível de habilidade/suporte necessário:

- Médio (Capacidade de entrevistar)

Custo:

- Pequeno (depende do negócio)

Nível de participação (IAP2)

- Consulta

Nível de inovação:

- Pequeno (Tradicional)

Etapas:

1. Obter lista criada anteriormente com dados básicos sobre *stakeholders*.
2. Para cada *stakeholder*, repetir todos os itens seguintes.
3. Pedir uma auto-avaliação sobre a própria posição. A posição pode ser muito favorável, favorável, neutra, contra, muito contra)
5 significa “muito favorável”,
4 significa “favorável”,
3 significa “neutra”
2 significa “contra” e
1 significa “muito contra”.
4. Pedir ao *stakeholder* entrevistado que avalie a posição dos outros *stakeholders*, usando os mesmos códigos do item anterior.
5. Calcular o posicionamento final de cada *stakeholder* levando em consideração a sua auto-avaliação e as avaliações feitas sobre ele pelos outros *stakeholders*. Mais detalhes na referência abaixo.

Referências:

SCHMEER, 2000, p. 2-15 a 2-17

APENDICE B – Técnicas de Envolvimento

Técnica	Tipo de envolvimento (IAP2)				
	Informar	Consultar	Envolver	Colaborar	Autorizar
Advisory group (2)					
Appreciative inquiry summit (2)					
Backcasting (1)	✓	✓	{✓}	{✓}	{✓}
Beneficiary assessment (2)					
Brainstorming (1)		✓	{✓}	{✓}	{✓}
Briefings (1,2)	{✓}	✓			
Citizen committees (1)		{✓}			
Citizen juries (1,3)		✓	{✓}	{✓}	
City walks (1)					
Civic journalism (1)	✓	✓			
Community fairs (1)	{✓}				
Community group presentation (2)					
Community indicator (1)			✓	{✓}	✓
Community profiling (1)	✓	{✓}	✓	✓	✓
Computer-based negotiation (2)					
Conference (1)	{✓}	✓	✓	✓	✓
Consensus building (2)					
Consensus conference (1,2,3)		{✓}	✓	✓	
Deliberative opinion polls (1,3)	✓	{✓}			
Delphi study (1,3)		✓	{✓}	✓	✓
Design charrettes (1,2,3)	✓				
Displays and exhibits (1,2)	{✓}	✓			
Electronic democracy (1)		{✓}	✓		
Expert panel (1,3)	✓	✓	{✓}	✓	✓
Feature stories (2)					
Field trips (1,2)	{✓}	✓			
Fishbowl (1)	✓	{✓}	✓		
Focus groups (1,2,3)		✓			
Future search conference (1,2)		✓	{✓}	✓	✓
Groupware (2)					
Information contacts (1)	✓				

Técnica	Tipo de involucramiento (IAP2)				
	Informar	Consultar	Envolver	Colaborar	Autorizar
Information hotline (1,2)	✓				
Information repository (1,2)	✓				
Interactive TV (1,2)	✓	{✓}			
Interactive video display kiosks (1)	✓				
Internet - Converge (2)					
Internet - Diverge (2)					
Interviews (2)					
Key stakeholder interviews (1)		✓			
Kitchen table discussion (1,2) [Coffee klatsch - Creighton]	✓	{✓}	✓	✓	✓
Mass mailing (2)					
Media kits (2)					
Media interviews and talk shows (2)					
Media releases (1,2)	{✓}	✓	✓	✓	✓
Mediation and negotiation (1)			✓	{✓}	✓
Mind mapping (1)	✓	{✓}	✓	✓	✓
Multiattribute Utility Analysis (2)					
Multi Objective Decision Making Support (MODSS) (1)			✓	✓	{✓}
News conference and briefings (2)					
News releases (2)					
Newsletter (2)					
Newspaper inserts (1)	✓	{✓}	✓		
Nominal group (1)			✓	✓	{✓}
Open house (or Open Days, Drop-in Centres) (1,2)	✓	{✓}	✓	✓	
Open space technology (1,2)			✓	✓	{✓}
Paid advertisements (2)					
Panels (2)					
Participant Observation (1)		{✓}	✓	✓	
Participatory Assessment, Monitoring and Evaluation (3)					
Participatory Rural Appraisal (2)					
Participatory Technology Assessment (2)					
Photovoice (1)		{✓}	✓		

Técnica	Tipo de involucramiento (IAP2)				
	Informar	Consultar	Envolver	Colaborar	Autorizar
Planning Cell (3)					
Planning4real (1)			✓	✓	{✓}
Plebiscite (2)					
Poster Competitions (1)	{✓}	✓	✓	{✓}	✓
Printed Information (1)	{✓}	✓			
Prioritisation Matrix (1)		✓	✓	{✓}	✓
Public Conversation (1)	✓	{✓}	✓	✓	
Public Hearings (2)					
Public Involvement Volunteers (1)	✓	✓	{✓}	✓	
Public Meeting (1,2)	✓	{✓}	✓	✓	
Public service announcements (2)					
Questionnaires and Responses (1)	✓	{✓}			
Retreat (2)					
Role Plays (1)	{✓}	✓			
Samoan Circles (1,2)	✓	✓	✓	{✓}	✓
SARAR (2)					
Scenario Building (3)					
Scenario Testing (1)	✓	✓	{✓}	✓	✓
Search Conference (1)		✓	{✓}	✓	
Shopfront (1)	{✓}	✓			
Simulation (Electronically Generated) (1)	✓	✓	{✓}	✓	✓
Symposia (2)					
Sketch Interviews (1)	✓	{✓}	✓	✓	
Snowball Sampling (1)		{✓}	✓		
Speakout (Version 1) (1)	{✓}	✓			
Speakout (Version 2) (1)			✓	✓	{✓}
Stakeholder Analysis (CLIP) (1)	✓	✓	✓	✓	✓
Stakeholder Analysis (Stakeholder Matrix) (1)	✓	✓	✓	✓	✓
Stakeholder Analysis (Venn Diagrams) (1)	✓	✓	✓	✓	✓
Study Circles (1)		✓	{✓}	✓	✓
Submissions (1)	✓	{✓}	✓		
Surveys (1,2)	✓	{✓}			

Técnica	Tipo de envolvimento (IAP2)				
	Informar	Consultar	Envolver	Colaborar	Autorizar
Task Force (2)					
Technical Assistance (1)	✓	✓	{✓}	✓	✓
Technical Reports and Discussion Papers (1)	{✓}	✓			
Technology Festival (3)					
Telephone Trees (1)	{✓}	✓			
Town Meeting (1,2)					
Visioning (1,2)		✓	✓	{✓}	✓
Websites (1)	✓	✓	✓	✓	
Workshops (1,2)		✓	{✓}	✓	✓

Onde:

1 – (VICTORIA, 2005)

2 – (CREIGHTON, 2005)

3 – (ELLIOT et al, 2005)