

Reuso e Modificação em Sistemas de Workflow:

Teoria e Estudo de Casos

Carolina Bertoldo Pacheco

Trabalho Final de Mestrado Profissional

Reuso e Modificação em Sistemas de Workflow: Teoria e Estudo de Casos

Carolina Bertoldo Pacheco

Agosto de 2004

Banca Examinadora:

- Prof. Dr. Jacques Wainer (Orientador)
- Prof^ª. Dr^ª. Maria Beatriz Felgar de Toledo
Instituto de Computação – UNICAMP
- Prof. Dr. José Valdeni de Lima
Instituto de Informática – UFRGS
- Tadeu Cruz
TRCR KNOWLEDGE
- Prof. Dr. Luiz Eduardo Buzato (Suplente)
Instituto de Computação – UNICAMP

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Pacheco, Carolina Bertoldo

P115r Reuso e modificação em sistemas de workflow: teoria e estudo de casos / Carolina Bertoldo Pacheco -- Campinas, [S.P. :s.n.], 2004.

Orientador : Jacques Wainer

Trabalho final (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Grupo de trabalho – Processamento de dados. 2. Política organizacional – Automação. 3. Engenharia de software. 4. Controle de processos. 5. Fluxo de trabalho. I. Wainer, Jacques. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Reuso e Modificação em Sistemas de Workflow: Teoria e Estudo de Casos

Este exemplar corresponde à redação final do Trabalho Final devidamente corrigida e defendida por Carolina Bertoldo Pacheco e aprovada pela Banca Examinadora

Campinas, 30 de agosto de 2004

Prof. Dr. Jacques Wainer
(Orientador)

Trabalho Final apresentado ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Computação na área de Engenharia de Software.

© Carolina Bertoldo Pacheco, 2004.

Todos os direitos reservados.

RESUMO

Workflow é uma tecnologia que tem sido, crescentemente, adotada para coordenação de processos, principalmente, por empresas que necessitam de maior dinamismo em seus processos de negócio. O objetivo principal de sistemas de workflow é capturar a descrição formal dos processos de negócio e dar suporte a automação integral ou parcial destes processos. Assim como os processos de negócio de uma organização, as aplicações de workflow, com o passar do tempo, precisarão ser alteradas e adaptadas a diferentes ambientes, de forma a refletir novas realidades da empresa. Entende-se, portanto, que as soluções de workflow devem possuir ferramentas que facilitem a modificação e a reutilização de workflow existentes para a criação de novos sistemas de workflow.

A necessidade de mudanças está diretamente relacionada à necessidade de reuso e de modificação. Ferramentas que facilitem e promovam estas operações através dos sistemas de workflow são imprescindíveis para agilizar e garantir maior qualidade aos sistemas desenvolvidos. Neste trabalho, faz-se uma análise da teoria existente quanto ao reuso e a modificação de sistemas de workflow e levanta as práticas adotadas por empresas para lidar com mudanças em seus processos de negócio.

ABSTRACT

Workflow is a technology whose use has been increasing to deal with process management, by organizations that need dynamism on their business processes. The main goal of workflow systems is to capture the formal definition of business processes and give support to the full or partial automation of these processes. Business processes as well as workflow applications may need to be modified and adapted to different environments, representing different requirements of the organization. Therefore, is expected that workflow solution present tools to support and facilitate the modification and the reuse of existing workflows to create new ones.

The need of changing directly corresponds to reuse and modification needs. Tools aiming to facilitate those operations throw workflow systems are indispensable to decrease developing time and ensure quality for the workflow systems developed. This study makes an analysis of the theory related to workflow reuse and modifications at the literature and the existing practices adopted by organizations to deal with business processes changes.

DEDICATÓRIA

Dedico este trabalho aos meus pais, Hermes e Maria Teresa, e aos meus irmãos, Juliana e Alex, por todo o amor e incentivo, hoje e sempre...

AGRADECIMENTOS

Agradeço a Deus pela oportunidade deste mestrado e por colocar em minha vida todos os que menciono abaixo..

Aos meus pais, Hermes e Maria Teresa, por todo o apoio, amor e a esperança que sempre me deram... A vocês, gratidão eterna.

Aos meus irmãos, Juliana e Alex, por estarem sempre presentes e me ajudarem a acreditar..

Ao Professor Jacques Wainer, pela orientação deste trabalho.

À IComNet Tecnologia da Informação, em especial ao Sr. Hermes e ao Sr. Adailton, pela compreensão e apoio à realização deste mestrado.

À Procuradoria Regional do Trabalho 15^a. Região, em especial a Dra. Abiael, Dr. Alex e aos meus colegas do departamento, pela enorme compreensão.

Aos colaboradores, que atenciosamente forneceram valiosas informações, sem as quais este trabalho não teria sido possível: Sr. Wanderley Montinho, Sr. José Luis, Sr. Fernando Grecco, Sr. Luis Lopes, Sr. Cristiano Oliveira e por último ao André Garcia, que agradeço também pelo amor, incentivo e companheirismo.

Aos funcionários do Instituto de Computação, em especial à Cláudia.

Aos amigos: André Garcia, André Luis Pacheco Lyra, Angela Berlinck, Carolina Baltar, Daniel Novaes, Lara Lucia Abrahão de Souza Mazon, Julia Paixão, Melissa Vendite e tia Silvia por todo o incentivo e pela torcida..

CONTEÚDO

RESUMO	VII
ABSTRACT	VIII
CONTEÚDO.....	XI
LISTA DE TABELAS.....	XIII
LISTA DE FIGURAS	XIV
CAPÍTULO 1.....	XV
INTRODUÇÃO	1
1.1 ABORDAGEM DA DISSERTAÇÃO	2
1.2 CONTEÚDO DOS CAPÍTULOS	3
CAPÍTULO 2.....	4
WORKFLOW.....	4
2.1 DEFINIÇÃO	4
2.2 COMPONENTES DO WORKFLOW	7
2.3 O MODELO DE REFERÊNCIA DA WfMC.....	8
2.4 MODELAGEM DE SISTEMAS DE WORKFLOW.....	10
2.5 CLASSIFICAÇÃO DE SISTEMAS DE WORKFLOW.....	12
2.6 LINGUAGEM DE MODELAGEM DE WORKFLOWS.....	14
2.6.1 Elementos Gráficos de um Workflow	14
CAPÍTULO 3.....	17
REUSO E MODIFICAÇÃO EM ENGENHARIA DE SOFTWARE.....	17
3.1 REUSO.....	17
3.1.1 Desenvolvimento orientado ao reuso	22
3.1.2 Reuso em orientação a objetos.....	23
3.1.3 Reuso através de padrões de software (Software Patterns)	26
3.2 TIPOS DE REUSO	28
3.3 MODIFICAÇÃO.....	29
3.3.1 Processo de modificação	30
3.3.2 Modificação em sistemas orientados a objeto.....	31
3.3.3 Modificação através de Programação Orientada a Aspectos.....	32
3.4 TIPOS DE MODIFICAÇÃO	34
CAPÍTULO 4.....	35
REUSO E MODIFICAÇÃO EM SISTEMAS DE WORKFLOW.....	35
4.1 INTRODUÇÃO AO REUSO	35
4.2 CONSIDERAÇÕES RELEVANTES AO REUSO DE SISTEMAS DE WORKFLOW	36

4.2.1 Aspectos organizacionais	37
4.2.2 Classificação e Localização de Definições de Workflow	38
4.2.3 Compreensão das Definições de Workflow	41
4.2.4 Gerenciamento de Configurações	43
4.3 PRÁTICAS DE REUSO	45
4.3.1 Reuso de Definições de Workflow	45
4.3.2 Reuso através de Templates.....	52
4.3.3 Reuso através de Sub-workflows	53
4.3.4 Reuso através de princípios de Orientação a Objetos com Micro-workflows.....	54
4.4 INTRODUÇÃO A MODIFICAÇÃO.....	56
4.5 PRÁTICAS DE MODIFICAÇÃO	60
4.5.1 Operações de Modificação em Sistemas de Workflow	60
CAPÍTULO 5.....	63
ESTUDO DE CASOS.....	63
5.1 ESTUDO DE CASO 1	63
5.2 ESTUDO DE CASO 2	74
5.3 ESTUDO DE CASO 3	81
CAPÍTULO 6.....	89
REFERÊNCIAS BIBLIOGRÁFICAS	93

LISTA DE TABELAS

TABELA 2.1 – METAS DE EFICIÊNCIA E O SUPORTE DE SISTEMAS DE WORKFLOW	XX
TABELA 2.2 – RELACIONAMENTO ENTRE CONCEITOS: WORKFLOW X BANCO DE DADOS....	XXV
TABELA 2.3 – CLASSIFICAÇÃO DE ATRIBUTOS DE SISTEMAS DE WORKFLOW [BECKER ET AL., 2002].....	XXVI
TABELA 2.4 – SIMBOLOGIA DOS ELEMENTOS GRÁFICOS DE UM WORKFLOW.....	29
TABELA 3.1 – BENEFÍCIOS DO REUSO DE SOFTWARES	32
TABELA 3.2 – DIFICULDADES DO REUSO DE SOFTWARES	34
TABELA 4.1 – COMPARAÇÃO ENTRE OS TIPOS DE MODIFICAÇÃO	67
TABELA 5.1 - DESCRIÇÃO DAS MODIFICAÇÕES REALIZADAS NO DEPARTAMENTO X.	77
TABELA 5.2 - DESCRIÇÃO DAS MODIFICAÇÕES REALIZADAS NO DEPARTAMENTO Y.	81
TABELA 5.3 - DESCRIÇÃO DAS MODIFICAÇÕES REALIZADAS PARA O FLUXO DA EMPRESA B.	89
TABELA 5.4 - DESCRIÇÃO DAS MODIFICAÇÕES REALIZADAS PARA O FLUXO DA EMPRESA C.	94

LISTA DE FIGURAS

FIGURA 2.1 – MODELO DE REFERENCIA DA WFMC - COMPONENTES E INTERFACES [WFMC, 1995]	XXIII
FIGURA 2.2 – CONCEITOS RELACIONADOS A MODELAGEM DE SISTEMAS DE WORKFLOW..	XXV
FIGURA 3.1 – PROCESSO DE DESENVOLVIMENTO ORIENTADO A REUSO	36
FIGURA 3.2 – GRÁFICO DE HIERARQUIA DE CLASSES.....	38
FIGURA 3.3 – ABSTRAÇÃO DA HIERARQUIA DO COMPONENTE WINDOW	44
FIGURA 4.1 – GRÁFICO DE VERSÃO INDICANDO O FUNCIONAMENTO DO GERENCIAMENTO DE CONFIGURAÇÕES (VERSÃO = MODIFICAÇÃO / VARIAÇÃO = REUSO + MODIFICAÇÃO).....	55
FIGURA 4.2 – PROCESSO DE DESENVOLVIMENTO DE APLICAÇÕES DE WORKFLOW [BARESI ET AL., 1999]	58
FIGURA 4.3 – PROCESSO DE DESENVOLVIMENTO DE APLICAÇÃO DE WORKFLOW [WESKE ET AL., 1999]	59
FIGURA 4.4 – DESENVOLVIMENTO DE MODELOS DE WORKFLOW BASEADOS EM REUSO	61
FIGURA 4.5 – GRÁFICO DE VERSÕES ILUSTRANDO O USO DE TEMPLATES EM UMA ORGANIZAÇÃO.	63
FIGURA 4.6 - MODIFICAÇÃO DA ESTRUTURA DAS ATIVIDADES.	69
FIGURA 4.7 - MODIFICAÇÃO NAS PROPRIEDADES DAS ATIVIDADES.	69
FIGURA 4.8 – TAXONOMIA DAS OPERAÇÕES DE MODIFICAÇÃO EM SISTEMAS DE WORKFLOW [KRADOLFER, 2000].....	71
.....	72
FIGURA 5.1 – REDE DE PETRI IDENTIFICANDO O FLUXO DO DEPARTAMENTO X ANTES DA MODIFICAÇÃO	75
FIGURA 5.2 – REDE DE PETRI IDENTIFICANDO O FLUXO DO DEPARTAMENTO X APÓS A MODIFICAÇÃO. EM AMARELO, ENCONTRA-SE A ATIVIDADE ALTERADA. EM LILÁS, AS ATIVIDADES ADICIONADAS.	76
FIGURA 5.3 – REDE DE PETRI IDENTIFICANDO O FLUXO DO DEPARTAMENTO Y ANTES DA MODIFICAÇÃO.....	80
FIGURA 5.4 - REDE DE PETRI IDENTIFICANDO O FLUXO DO DEPARTAMENTO Y DEPOIS DA MODIFICAÇÃO. EM AMARELO, ENCONTRAM-SE AS ATIVIDADES MODIFICADAS.	81
FIGURA 5.5 – FLUXO PARA CONTROLE DE LANÇAMENTO DO LAYOUT DE EMBALAGENS ANTES DA MODIFICAÇÃO	89
FIGURA 5.6 – INTERFACE DE COLETA DE INFORMAÇÕES PARA O PROGRAMA DE CAPTAÇÃO DE IDÉIAS.....	95

CAPÍTULO 1

INTRODUÇÃO

A economia atual é caracterizada pelo crescimento da competição global, tornando necessário que organizações revejam sua estrutura organizacional bem como a maneira de realizar negócios. Neste contexto, a otimização de processo de negócio, tem freqüentemente sido o foco das atenções de organizações. Diferentes entidades de processamento compõem um processo de negócio. Pessoas e sistemas de software são coordenados para a realização de atividades, no intuito de alcançar objetivos em comum. Diante da constante competição global, muitas organizações precisam operar em ambientes dinâmicos, exigindo que os processos de negócio destas empresas sejam constantemente revisados e otimizados para atender a novas necessidades da empresa e a novos requisitos do mercado.

A utilização adequada da Tecnologia da Informação, como ferramenta de suporte aos processos de negócios, tornou-se um fator crítico de sucesso para organizações. Workflow é uma tecnologia que tem sido crescentemente adotada por empresas para a coordenação de processos, principalmente por empresas que necessitam de maior dinamismo em seus processos de negócio. O objetivo principal de sistemas de workflow é capturar a descrição formal dos processos de negócio e dar suporte a automação destes processos.

Um workflow definido para dar suporte ao processo de negócio de uma organização, certamente, com o passar do tempo, precisará ser alterado e adaptado a novos ambientes, que reflitam a novas realidades da empresa. Portanto, a modificação destes sistemas deve ser uma operação suportada pela tecnologia de workflow. A evolução de um workflow pode incluir a criação de novos modelos, a modificação e a exclusão de modelos existentes. Em analogia a manutenção de software, modificações adaptativas, perfectivas (que visam melhoria), assim como corretivas podem ser identificadas em sistemas de workflow. Quando uma modificação é realizada em um

workflow, efeitos no sistema, assim como em suas instâncias, devem ser considerados.

Novos workflows podem ser necessários para dar suporte a novas aplicações. Similarmente ao reuso em Engenharia de Software, o reuso de workflows visa aumentar a produtividade e a qualidade do desenvolvimento. Quando um novo workflow precisa ser desenvolvido, o reuso de definições de workflows (que compreende a definição formal, o modelo e etc) existentes deve ser considerado. O reuso de workflows, assim como o reuso de softwares, é um processo que consiste na localização, compreensão, adaptação e integração de definições de workflow. Contudo, quanto mais definições de workflow estiverem disponíveis ao reuso, mais difícil torna-se a tarefa de localização de uma definição que satisfaça a necessidade corrente. Portanto, faz-se necessário a utilização de ferramentas que suportem a classificação e a localização de definições reutilizáveis.

Reuso e Modificação são temas bastante discutidos e, de grande interesse à engenharia de software. Tendo isto em mente, resolvemos estudar os efeitos do reuso e da modificação de processos de negócio através da tecnologia de workflow.

1.1 Abordagem da Dissertação

Apesar do reuso e da modificação serem tópicos bastante discutidos e maduros em Engenharia de Software, estudos referentes ao reuso e a modificação de workflows ainda estão sendo aprimorados. Os capítulos iniciais resultam da revisão da teoria existente sobre workflow, reuso e modificação em engenharia de software e, reuso e modificação em sistemas de workflow, respectivamente, e reúnem os aspectos considerados mais relevantes da literatura estudada.

O capítulo de estudo de casos é o resultado de uma coletânea de experiências de empresas e, desta forma, contribui com análise dos conceitos reunidos nos capítulos anteriores, comparando-os com as práticas adotadas por empresas em contato com soluções de workflow.

Este estudo visou conhecer como as práticas de modificação e reuso encontram-se presentes em organizações, a partir da necessidade de criação de novos processos,

do compartilhamento de práticas organizacionais, e da modificação de processos existentes.

1.2 Conteúdo dos Capítulos

Esta dissertação está organizada através dos seguintes capítulos:

Capítulo 1 – INTRODUÇÃO: contém esta introdução

Capítulo 2 – WORKFLOW: descreve uma visão geral da tecnologia de workflow e define os conceitos e terminologias utilizadas no restante da dissertação;

Capítulo 3 – REUSO E MODIFICAÇÃO EM ENGENHARIA DE SOFTWARE: apresenta o conceito de reuso e de modificação adotados na engenharia de software;

Capítulo 4 – REUSO E MODIFICAÇÃO EM SISTEMAS DE WORKFLOW: descreve o conceito de reuso e de modificação em sistemas de workflow, fazendo referência aos conceitos utilizados em engenharia de software;

Capítulo 5 – ESTUDO DE CASOS: apresenta o estudo realizado em empresas que possuem processos de negócio identificados, na tentativa de colher informações sobre as práticas de reuso e modificação adotadas;

Capítulo 6 – CONSIDERAÇÕES FINAIS: conclui a dissertação e apresenta espaços abertos para futuros estudos e pesquisa.

CAPÍTULO 2

WORKFLOW

2.1 Definição

Workflow é definido como sendo a automação de procedimentos, em parte ou integralmente, através do qual documentos, informações ou atividades são passadas de um participante ao outro de acordo com um conjunto definido de regras, para atingir ou contribuir com o alcance dos objetivos de uma organização [WfMC, 1995].

Esta definição faz uso de aspectos freqüentemente mencionados ao discutir-se as características ou o funcionamento de workflows, no entanto, relaciona o conceito de workflow com automação, o que não ocorre necessariamente. Workflows podem ser organizados de forma manual, porém, geralmente, utiliza-se ferramentas (sistemas computacionais) para dar suporte a execução dos processos de negócio da empresa.

A definição de Sarin e Abbot [Sarin et al., 1994] ressalta esta separação conceitual: “A ênfase dos sistemas de workflow está em usar computadores para auxiliar a execução e o gerenciamento de processos de negócio, que são compostos por diversas tarefas individuais, e não no uso de computadores para automatizá-las”.

Sistemas de Workflow auxiliam o gerenciamento dos Processos de Negócio de uma organização controlando a seqüência das atividades e associando os recursos humanos e computacionais necessários para a execução de cada uma destas atividades. O conceito de workflow aparece também, freqüentemente relacionado ao conceito de Processos de Negócio, porém nem todo Processo de Negócio é ou necessita ser implementado através do workflow. Por outro lado, nem todo workflow é necessariamente uma implementação de partes de um Processo de Negócio.

Workflows pertencem a uma área de desenvolvimento de softwares denominada Groupware ou “Atividades colaborativas realizadas por intermédio do computador”. Esta

área engloba outras tecnologias, como: correspondência eletrônica, grupos de discussão, ferramentas de videoconferência, escrita colaborativa, PIM (Personal Information Manager), sistema de controle de documentos, entre outros.

Apesar de estar inserido entre a relação de sistemas que visam facilitar o trabalho cooperativo, sistemas de workflow incorporam a maioria das tecnologias presentes nesta relação. Um sistema moderno de workflow faz integração com, pelo menos, ferramentas para correspondência eletrônica, agenda corporativa e controle de documentos. A proximidade dos sistemas de workflow com as ferramentas colaborativas é também um reflexo da similaridade das principais atuações de ambos os sistemas: comunicação, colaboração e coordenação.

Em adição às definições apresentadas, alguns aspectos geralmente desejados pelos usuários de sistemas de workflow são:

- A troca de documentos, informações e tarefas que possibilite, não somente o relacionamento entre um participante e outro, mas também entre grupos de indivíduos. Sistemas de workflow permitem a distribuição de indivíduos em grupos, de acordo com as regras da organização;
- Monitoramento de aspectos da execução do workflow, sendo capaz de gerar informações para avaliação de progressos em um processo de negócio ou em parte dele.

Porém, enquanto soluções de Groupware focam atividades colaborativas que, em sua grande maioria, são desestruturadas, soluções de workflow visam a coordenação de atividades através de um modelo de processo comum, sem que haja necessariamente a automação das atividades.

O propósito de um sistema de workflow é coordenar todas as entidades envolvidas na execução de um processo (processo de negócio). A coordenação pode ser definida como o gerenciamento das dependências entre as atividades. Sistemas de gerenciamento de workflow podem ser usados para lidar com dois tipos de problemas de coordenação:

- Dependência de dados entre atividades (por exemplo: onde uma atividade depende do resultado de uma ou mais atividades), que é gerenciada através do controle e fluxo de dados;
- Compartilhamento de recursos disponíveis (por exemplo: uma pessoa ou uma determinada máquina envolvida no sistema de workflow, só é capaz de executar uma atividade por vez), que é gerenciado através do controle de cronogramas de recursos;

Através da automação destas funções de coordenação, sistemas de workflow trazem diversos benefícios diretos às organizações. A Tabela 2.1 apresenta estes benefícios [Becker et al., 2002].

Tabela 2.1 – Metas de Eficiência e o suporte de sistemas de Workflow

Metas de Eficiência	Descrição	Atividades Suportadas pelos Sistemas de Workflow
Eficiência do Processo	Otimização de características do processo, como tempo de execução, número de falhas (a serem minimizados) e cumprimento de prazos (a ser maximizado).	Coordenação de atividades através do controle do fluxo, de prazos e etc.
Eficiência dos Recursos	Uso eficiente dos recursos (recursos humanos assim como entidades de processamento – sistemas computacionais) disponíveis para a execução dos processos.	Coordenação de cronogramas de execução e de escalonamento de recursos
Eficiência de Mercado	Posicionamento da organização no relacionamento com os parceiros de mercado. Inclui: previsão confiável de prazos de entrega, comunicação transparente e eficiente com fornecedores e clientes, processos otimizados de aquisição e distribuição.	Interfaces bem definidas para serem utilizadas através da Internet, previsão comportamental /departamental de atividades e resultados através da padronização dos procedimentos.
Eficiência de Delegação	Uso adequado da competência de supervisores (maior alcance de visão do processo como um todo) e de subordinados (conhecimento detalhado das atividades).	Coordenação de designações à funcionários (recursos alocados) e do conceito geral do processo
Eficiência da Motivação	Motivação de funcionários para trabalharem no intuito de atingir os objetivos da organização	Orientação de como executar as atividades ao longo do modelo de workflow, monitoramento da eficiência e dos progressos e prévia explicação e definição das atividades.

Os benefícios trazidos pelo uso de sistemas de workflow tornam-se mais visíveis quando o número de atividades a serem coordenadas aumenta. A utilização de sistemas de workflow facilita a coordenação de atividades. O gerenciamento de diversas atividades simultaneamente torna-se uma atividade mais fácil se ocorrer por intermédio de workflows. A quantidade de tarefas coordenadas varia de acordo com o grau de definição e exigências dos componentes controlados pelos sistemas, assim como o tipo de processo que é controlado.

Um mesmo processo de negócio pode ter um ciclo de vida que dure alguns minutos ou dias (ou até mesmo meses e anos), dependendo da complexidade e da duração das atividades que o compõe. Os sistemas de workflow podem ser implementados através de variadas formas, podem utilizar diferentes tecnologias e infra-estrutura de comunicação. Porém, todos os sistemas de workflow devem apresentar características comuns no intuito de prover interoperabilidade e capacidade de integração entre diferentes produtos do mercado.

2.2 Componentes do Workflow

Workflows consistem de diversos componentes. De acordo com a taxonomia apresentada por Ronni T. Marshak [Marshak, 1994], estes componentes são:

- Tarefas: o sistema de workflow é composto por diferentes tarefas ou atividades, que devem ser cumpridas para se atingir um objetivo de negócio.
- Pessoas: pessoas específicas ou entidades automatizadas (realizando tarefas de pessoas) que desempenham as atividades em uma ordem determinada.
- Ferramentas: o sistema de workflow não desempenha todas as atividades em si. Muitas vezes faz uso de sistemas computacionais específicos e dedicados para executar determinadas tarefas. Como exemplo, uma tarefa exige que um texto seja comentado. Para tanto, o sistema inicializa um editor de texto externo possibilitando que o texto seja mostrado e editado.
- Dados: as informações acessadas pelas ferramentas para realizar as tarefas.

As condições ou regras determinadas por uma organização, são responsáveis pela definição da ordem de execução das tarefas, quem deve executá-las e quais ferramentas estarão disponíveis para realização das mesmas. Marshak [Marshak, 1994] também apresenta um conjunto de atributos de sistemas de workflow, simplificados como um conjunto de três Rs e três Ps:

- Roteamento (**R**outes): roteamento de diversos tipos de objetos de diferentes formas – documentos, formulários, aplicações, informações, etc.
- Regras (**R**ules): determinam qual informação deve ser roteada e para quem.
- Papéis (**R**oles): definição de papéis independentemente de pessoas ou processos específicos. Pessoas podem ou não executar determinados papéis.
- Processos (**P**rocesses): identificação de processos/procedimentos que foram desenvolvidos para satisfazer e contemplar as regras de negócio de uma organização.
- Políticas (**P**olicies): identificação de políticas formais que a empresa possui, que determinam como cada processo de negócio deve ocorrer (regras de negócio da organização).
- Práticas (**P**ractices): compreensão de como as atividades realmente ocorrem no cotidiano da organização, para completar as políticas formais com a realidade de como elas são efetivamente executadas.

Atributos são importantes para a determinação do modo de operação do sistema de workflow, caracterizando-os de acordo com sua estrutura e uso.

2.3 O Modelo de Referência da WfMC

A WfMC (Workflow Management Coalition) é a organização internacional responsável pelo desenvolvimento e promoção de padronizações para sistemas de workflow. Esta organização criou um Modelo de Referência para servir como orientação para a padronização. Descreve os conceitos fundamentais para o gerenciamento de

sistemas de workflow, uma arquitetura modelo e as interfaces entre os componentes desta arquitetura. Este modelo é apresentado pela Figura 2.1.

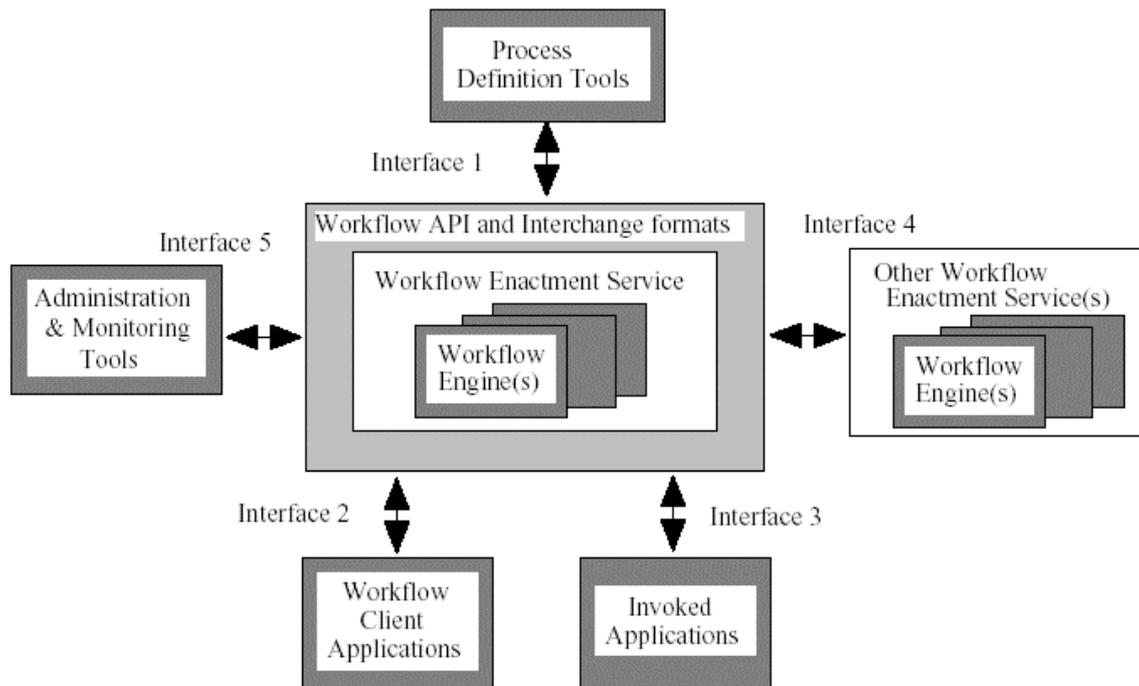


Figura 2.1 – Modelo de Referência da WfMC - Componentes e Interfaces [WfMC, 1995]

A arquitetura do Modelo identifica os principais componentes e interfaces do sistema. A interface que rodeia o *workflow enactment service* é denominada *WAPI (Workflow API) and Interchange Formats*, sendo responsável por oferecer acesso aos serviços do sistema de workflow e por regular as interações entre o software de controle de workflow e os componentes de outros sistemas.

- The Workflow Enactment Service: serviço de software que faz uso de um ou mais mecanismos de workflow para criar, controlar e executar instancias do workflow, interagindo com recursos externos para processar suas atividades. A interação com recursos externos ocorre por intermédio de duas interfaces:
 - Interface 2 – Interface com o Aplicativo do Cliente

- Interface 3 – Interface com Aplicativos invocados pelo workflow, permite que os mecanismos do workflow ativem ferramentas específicas para a execução de atividades.
- The Workflow Engine: mecanismo de software responsável por promover parte (ou todo) do ambiente de execução para uma instancia do workflow. Este serviço de software prove facilitadores para:
 - Interpretar a definição do processo;
 - Controlar instancias de processos (criação / ativação / suspensão / encerramento);
 - Navegar entre as atividades de um processo, que podem invocar operações seqüenciais ou em paralelo, programar o fim de execução de uma atividade, interpretar informações relevantes ao workflow, etc.
 - Registrar entrada e saída de participantes das atividades. O mecanismo de workflow pode controlar a execução de um conjunto de processos ou sub-processos / atividades.
- The Workflow Application Interface & Interchange: como mencionado anteriormente, o WAPI tem sido definido como um núcleo comum de funções de APIs e formatos de intercâmbio de dados.

2.4 Modelagem de Sistemas de Workflow

Modelagem de Workflow denota a criação de uma classe de workflow, que é uma descrição formal de diversos elementos do workflow, como: as atividades que devem ser desempenhadas, a identificação das pessoas ou entidades computacionais que deverão realizar as atividades, as dependências e o relacionamento entre as atividades e outros. A partir de classes de workflow são geradas inúmeras instancias do workflow. O conjunto de definições da classe de workflow por sua vez, origina o Modelo de Workflow [zur Muehlen et al., 1999].

A criação de uma classe de workflow requer um meta-modelo que compreenda um conjunto de conceitos de modelagem. A classe é expressa por uma linguagem de modelagem de workflows, que é uma linguagem formal oferecendo construções que

abranjam os conceitos de modelagem do meta-modelo.

A Figura 2.2 (em UML – Unified Modeling Language), demonstra o relacionamento existente entre vários conceitos relacionados a Modelagem de Sistemas de Workflow.

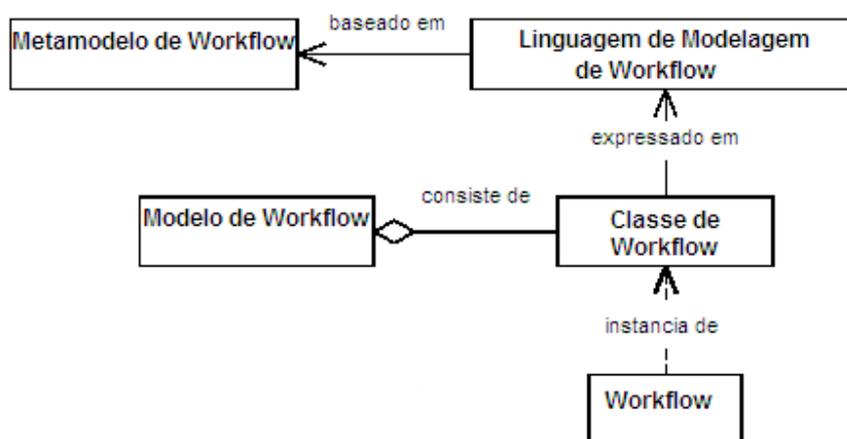


Figura 2.2 – Conceitos relacionados a Modelagem de Sistemas de Workflow

Visando um maior esclarecimento do conceito de Modelagem de Sistemas de Workflow, classes de workflow e Modelos de Workflow, a tabela abaixo mostra os termos que seriam correspondentes utilizando conceitos de tecnologia de banco de dados.

Tabela 2.2 – Relacionamento entre Conceitos: Workflow x Banco de Dados

Gerenciamento de Workflow	Tecnologia de SGBD (Banco de Dados)
Meta-modelo de Workflow	Modelo de Dados
Classe de Workflow	Classes, Tabelas, etc.
Modelo de Workflow	Definição da Base de Dados

2.5 Classificação de Sistemas de Workflow

A classificação de sistemas de workflow em categorias permite que a escolha de um sistema de workflow seja uma tarefa mais fácil ao Usuário. Vendedores de soluções de workflow ao classificarem seus produtos de acordo com categorias do mercado terão, por sua vez, maiores chances de atingir seu público alvo.

Existem diversas classificações de sistemas de workflow, propostas na literatura [Becker et al., 2002]. A Tabela 2.3 mostra uma classificação de sistemas de workflow baseada nos atributos específicos relacionados a implementação dos sistemas. Os participantes de uma aplicação de workflow podem ser pessoas, entidades automatizadas (máquinas) e sistemas computacionais (componentes de software). A estrutura de processos definidos pode ser pré-definida ou flexível (indo de workflows voltados a produção à workflows ad-hoc).

Tabela 2.3 – Classificação de atributos de sistemas de workflow [Becker et al., 2002].

Atributos	Valores possíveis		
Participantes	Pessoas	Máquinas (Hardware)	Software
Estrutura do Processo	Processos e atividades Ad-hoc	Atividade pré-definidas e processos Ad-hoc	Processos pré-definidos
Escopo do Processo	Entre organizações (B2B)	Dentro de uma organização	Entre aplicações de uma organização
Definição dos Dados	Documentos, Objetos		Atributos
Definição da Aplicação	Nível de processo (web services)	Nível de aplicação (programas)	Nível de função (chamadas a métodos)

O escopo do processo automatizado pode ser restrito ou voltado a uma aplicação única, ou estendido ao uso de toda uma organização. A definição dos dados

manipulados pelo sistema de workflow pode ser pouco detalhada (se documentos ou objetos inteiros são passados de uma atividade a outra) ou muito detalhada (se atributos são passados de uma atividade a outra). As aplicações iniciadas pelo sistema de workflow podem ser “grosseiras” (se web services são utilizados em implementações business-to-business) ou “detalhadas” (se chamadas a métodos de componentes de aplicações são utilizados).

Através do atributo “Participantes” é possível distinguir três maiores tipos de processos, que podem ser suportados por soluções de workflow:

- *Processos Organizacionais* são processos de negócio com um alto nível de envolvimento humano. Geralmente, ocorrem em ambientes administrativos de uma organização e consistem em um número de participantes que trabalham de forma autônoma utilizando sistemas que podem ou não estar relacionadas a sistemas de workflow. A estrutura do processo, geralmente, é bem definida (se o processo é bem conhecido, a separação e identificação das atividades leva o processo a um nível maior de definições e detalhamento). Sistemas de workflow são utilizados nesta categoria para a troca de documentos, imagens e etc.
- *Processos de Software* são processos automatizados utilizados através de aplicações de sistemas. Nesta categoria, sistemas de workflow são utilizados para inter-relacionar sistemas isolados, no intuito de realizar troca de informações entre os sistemas, caso a mesma base de dados não seja utilizada. A participação de recursos humanos neste tipo de processo é, geralmente, restrita a inicialização de processos ou análise de resultados.
- *Processos híbridos* combinam as características de processos organizacionais e de software. Neste caso, sistemas de workflow podem funcionar como intermediários entre os participantes humanos e as aplicações da organização, guiando o trabalho dos participantes através das atividades individuais.

2.6 Linguagem de Modelagem de Workflows

A linguagem de modelagem de workflow provê construções concretas para os conceitos do meta-modelo de workflow. Podem ser classificadas em linguagens de modelagem textuais e gráficas. Linguagens de modelagem textuais permitem a representação das classes de workflow na forma de texto. Linguagens de modelagem gráficas, por outro lado, permitem a visualização de determinados aspectos das classes de workflow [zur Muehlen et al., 1999].

Para a especificação de processos de workflow, diversas linguagens são desenvolvidas com o passar do tempo. A maioria dos vendedores de soluções de workflow possui suas linguagens proprietárias para modelagem de modelos de workflow, somente alguns utilizam linguagens existentes para modelagem de processos, como Redes Petri, para a modelagem de componentes de seus produtos.

2.6.1 Elementos Gráficos de um Workflow

Os elementos gráficos de um workflow servem para permitir que Usuários possam executar tarefas, bem como para fornecer informações aos Usuários. Estes elementos podem ser: etapas, ícones, linhas e texto de fluxo [Zar, 2000].

Tabela 2.4 – Simbologia dos elementos gráficos de um workflow

Simbologia	Tipo de Etapa	Significado
	Tarefa	Executar determinada tarefa
	Sub-Fluxo	Provê acesso a outros workflows, em níveis inferiores da hierarquia de workflows.
	Decisão	Permite que o Usuário escolha ou determine o caminho que o fluxo deve seguir, baseado na decisão realizada.
	Atividade	Define atividades que podem ou não estar associadas com o processo em andamento

Etapas: Etapas são as atividades ou unidades de trabalho, que constituem o processo.

- Através de etapas, a maioria das atividades é executada e eventos são iniciados;
- Visualmente as etapas identificam ao usuário, o tipo de atividade a ser realizada: a atividade atual, uma decisão, etc.

Ícones: Ícones podem ser adicionados ao workflow para melhorar a qualidade da definição e compreensão gráfica do modelo.

Linhas: Linhas no workflow são linhas que indicam a seqüência de execução e a conexão entre os outros elementos. Linhas podem ter setas no final, indicando um caminho entre etapas, ou podem ser planas, indicando um agrupamento de elementos do fluxo.

Texto de fluxo: Textos de fluxo são informações de forma textual que são exibidas diretamente no workflow. Textos de fluxo provêem informações importantes como instruções de fluxo, o nome e a versão do projeto, o nome do fluxo e outros.

CAPÍTULO 3

REUSO E MODIFICAÇÃO EM ENGENHARIA DE SOFTWARE

A habilidade de um projeto de software de desenvolvimento posterior, de se beneficiar dos esforços envolvidos na produção de produtos de software previamente projetados, tem sido um objetivo há muito perseguido, por desenvolvedores de softwares [Yao et al., 2004].

Na maioria dos projetos de software, ocorre algum tipo de reuso. Em geral, o reuso acontece informalmente, quando pessoas que trabalham no projeto conhecem componentes ou códigos similares aos exigidos. Desenvolvedores recorrem a estes componentes, incorporando-os aos seus sistemas.

Com o aumento da demanda de softwares, no entanto, elevou-se a necessidade de ferramentas que auxiliassem desenvolvedores de software no projeto, no desenvolvimento e na manutenção dos sistemas de software. O custo relativo a manutenção (modificação) de software é elevado em relação aos custos de engenharia, principalmente por que a maioria do reuso e modificação são realizados manualmente.

3.1 Reuso

Reuso de Software refere-se à utilização de produtos de software, constituídos ao longo do processo de desenvolvimento, em uma situação diferente daquela para a qual foram originalmente produzidos.

O reuso deve ser considerado durante o projeto de engenharia ou de requisitos. O reuso oportuno é possível durante a programação, quando se descobrem componentes que, por acaso, atendam a um requisito desejável. A engenharia de software baseada no reuso é uma abordagem para o desenvolvimento que tenta maximizar o reuso de softwares já existentes. As unidades de software que podem ser reutilizadas são:

- Reuso de conjunto de aplicações: Todo o conjunto de aplicações pode ser reutilizado através da incorporação, sem modificação, em outros sistemas ou pelo desenvolvimento de grupos de aplicações, que podem ser executadas em plataformas diferentes ou ser especializadas para as necessidades de determinadas organizações;
- Reuso de componentes: Os componentes de uma aplicação, que variam em tamanho (incluindo desde subsistemas até objetos isolados podem ser reutilizados).
- Reuso de funções: Os componentes de software que implementam uma única função, podem ser reutilizados.

O reuso de Software, através da reutilização de produtos, de processos e do conhecimento, tem sido identificado por empresas, como uma solução efetiva para alcançar os objetivos principais da Engenharia de Software: desenvolvimento de sistemas de software com qualidade e confiáveis, obedecendo tanto o cronograma físico quanto o financeiro.

De acordo com Pressman [Pressman, 1992] existem duas regras que devem ser consideradas pelos desenvolvedores de software, no intuito de identificar se o processo de desenvolvimento deve ou não contemplar o reuso:

“Se o software existente cumprir os requisitos desejáveis, adquira-o. O custo de aquisição de um software existente quase sempre será menor que o custo para desenvolver um software equivalente. Se o software existente exigir alguma modificação antes que possa ser adequadamente integrado ao sistema, o custo para modificá-lo pode ser maior do que o custo de desenvolvê-lo.”

Uma vantagem óbvia do reuso de softwares é que os custos gerais de desenvolvimento podem ser reduzidos. Contudo, a redução de custos é somente uma vantagem em potencial do reuso. Outras vantagens possíveis, do reuso de softwares podem ser observadas na Tabela 3.1 [Sommerville, 2001].

Tabela 3.1 – Benefícios do reuso de softwares

Maior confiabilidade	Componentes reutilizados a partir de componentes que são utilizados em sistemas que estão em operação, são mais confiáveis que componentes novos.
Redução dos riscos de processo	Ao recorrermos a um componente já existente, as incertezas sobre os custos relacionados ao reuso desse componente são menores do que sobre os custos de desenvolvimento.
Uso efetivo de especialistas	Ao invés de utilizar-se mão de obra especializada para o desenvolvimento do mesmo trabalho, em diferentes projetos, eles podem desenvolver componentes reutilizáveis, que englobam seu conhecimento.
Conformidade com padrões	Padrões, como os de interface com o Usuário, podem ser implementados como um conjunto de componentes-padrão. O uso de interfaces-padrão com o usuário melhora a confiabilidade, uma vez que os usuários possivelmente comentem menos enganos quando utilizam uma interface comum.
Desenvolvimento acelerado	De modo geral, é mais importante fornecer um sistema para o mercado o mais rápido possível do que se prender aos custos gerais de desenvolvimento. O reuso de componentes acelera a produção, por que o tempo de desenvolvimento e de validação deve ser reduzido.

Existem três requisitos fundamentais para o projeto e o desenvolvimento de softwares baseado em reuso:

- 1) Deve ser possível encontrar componentes reutilizáveis apropriados. As organizações precisam de uma base de componentes reutilizáveis classificados e documentados adequadamente, para que a localização de componentes seja fácil.
- 2) O desenvolvedor que irá reutilizar um componente precisa ter certeza de que os componentes se comportarão conforme o esperado e o especificado, e de que são confiáveis. Idealmente, todos os componentes da base de componentes de uma organização, devem estar certificados, a fim de confirmar que atingiram determinado nível de qualidade. Na prática, esta situação não é realista e as pessoas em uma empresa aprendem de maneira informal sobre componentes confiáveis.
- 3) Os componentes devem ter a documentação associada para ajudar o usuário a compreendê-los e adaptá-los a uma nova aplicação. A documentação deve incluir informações sobre onde os componentes foram reutilizados e sobre quaisquer problemas de reuso que tenham sido encontrados.

Existem, porém, alguns custos associados ao reuso, que podem inibir a introdução desse método e fazer com que as reduções no custo total de desenvolvimento com o reuso, sejam menores do que as previstas. A Tabela 3.2 exemplifica as dificuldades do reuso de softwares [Sommerville, 2001].

Tabela 3.2 – *Dificuldades do reuso de softwares*

Aumento nos custos de manutenção	Se o código-fonte do componente não estiver disponível, os custos de manutenção poderão aumentar, uma vez que os elementos reutilizados no sistema podem se tornar crescentemente incompatíveis com as mudanças do sistema.
Falta de ferramentas de apoio	Muitas ferramentas CASE não apóiam o desenvolvimento com reuso. O processo de desenvolvimento de software através destas ferramentas pode não levar em conta o reuso, dificultando a reusabilidade.
Síndrome do “não-foi-inventado-por-mim”	Alguns desenvolvedores preferem criar componentes de software, por acreditar que podem fazer melhor ou por não acreditar na qualidade do trabalho alheio.
Encontrar e adaptar componentes reutilizáveis	Componentes de software devem ser localizados, compreendidos e, algumas vezes, adaptados, a fim de atenderem novas necessidades. Desenvolvedores devem procurar por componentes na biblioteca, antes de considerar o reuso como parte de seu processo de desenvolvimento.
Manutenção de bibliotecas de componentes	As técnicas atuais de classificação e localização de componentes facilitam a operação, porém podem não ser suficientes para tornar simples a operação de recuperação de um componente, em bibliotecas muito grandes. Nem todos os componentes desenvolvidos devem ser armazenados nas bibliotecas. Componentes com maior potencial de reuso e com qualidade comprovada devem ter maiores chances de pertencer a biblioteca.

3.1.1 Desenvolvimento orientado ao reuso

Segundo Sommerville [Sommerville, 2001], embora o estágio inicial de especificação de requisitos e o estágio de validação sejam comparáveis com outros processos de desenvolvimento, os estágios intermediários em um processo de desenvolvimento orientado ao reuso, são diferentes. Para o desenvolvimento orientado ao reuso, os seguintes estágios de desenvolvimento devem ser considerados:

- **Análise de Componentes:** considerando a especificação de requisitos gerada pela fase de análise, é feita uma busca de componentes de software que a satisfaçam. Geralmente, uma especificação não é atendida por somente um componente, que oferece parte da funcionalidade desejada, sendo necessário combiná-lo a outros componentes.
- **Modificação de Requisitos:** Durante este estágio, os requisitos são analisados, utilizando-se as informações sobre os componentes que foram encontrados. Os requisitos podem ser modificados para refletir os componentes disponíveis ou a atividade de análise dos componentes pode ser refeita, no intuito de encontrar novas alternativas.
- **Projeto de sistemas com reuso:** Nesta fase, a infra-estrutura do sistema é projetada ou uma infra-estrutura existente é reutilizada. Os desenvolvedores devem levar em conta os componentes que são reutilizados e devem organizar a infra-estrutura para suportá-los. Um novo componente poderá ser desenvolvido, se os componentes reutilizáveis não estiverem disponíveis.
- **Desenvolvimento e Integração:** O software que não puder ser comprado será desenvolvido, e os componentes serão integrados, a fim de criar um sistema. A integração de sistemas, neste modelo, pode ser parte do processo de desenvolvimento, em vez de uma atividade separada.

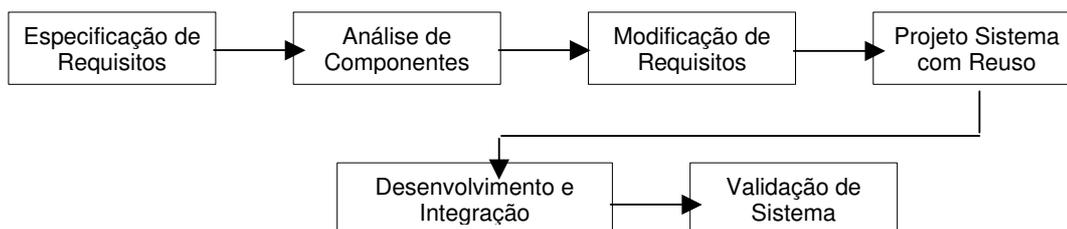


Figura 3.1 – Processo de Desenvolvimento orientado a reuso

Adequações nos requisitos levantados inicialmente são inevitáveis, podendo resultar em um sistema que não atenda às reais necessidades do usuário. Além disto, o controle sobre a evolução do sistema se perde, uma vez que novas versões dos componentes reutilizáveis não estão sob controle da organização que utiliza estes componentes.

3.1.2 Reuso em orientação a objetos

Um sistema de software orientado a objetos é uma coleção de estruturas abstratas de dados, denominadas “classes”. Uma classe é um encapsulamento da especificação do estado de uma estrutura abstrata de dados e suas operações. Uma instanciação, ou instância de classe é um objeto. Podem existir diversos objetos ativos de uma mesma classe, cada instância sendo um objeto diferente.

Objetos executam ações, em resposta a mensagens que lhe são enviadas. Essas mensagens são chamadas de métodos. Ao receber uma mensagem, um objeto executa uma ação de resposta, seja através da mudança de estado e/ou retornando uma mensagem ao objeto que o chamou [Ambler, 1997].

Respostas a mensagens são especificadas através de métodos, que são componentes de uma classe. Métodos são, essencialmente, procedimentos locais à classe. Eles podem ter parâmetros, atribuições a variáveis locais e a variáveis persistentes, e podem enviar outras mensagens.

A Instanciação de uma classe diversas vezes é uma forma de reuso sem modificação (Verbatim Reuse). Este tipo de reuso de entidade de softwares orientados a objetos é bastante parecido ao reuso de softwares de outros paradigmas.

3.1.2.1 Reuso baseado em Herança

O mecanismo de herança em orientação a objetos dá suporte ao ensejo dos desenvolvedores de “Eu quero algo parecido com isto, exceto...” Um desenvolvedor pode modificar uma determinada classe para criar uma classe nova, que se comporte diferentemente de sua classe-pai, em determinados aspectos. A classe original é a superclasse e a nova classe, a sub-classe [Bieman, 1992].

Uma sub-classe pode ser modificada através das seguintes operações:

- Adicionando-se novas variáveis de estado;
- Adicionando-se novos métodos;
- Alterando-se os métodos existentes.

As primeiras duas alterações expandem a especificação da superclasse, sendo caracterizadas modificações de extensão. A terceira modificação altera o comportamento do método da superclasse e pode ser denominada como uma modificação de sobreposição. Ao criar uma sub-classe, o desenvolvedor necessita somente especificar as diferenças que esta classe deverá ter, em relação a super-classe. Claramente, o mecanismo de herança dá suporte ao reuso de classes já existentes.

A hierarquia de classes, superclasses e sub-classes pode ser observada através de um gráfico de hierarquia de classes. A Figura 3.2 mostra um exemplo de gráfico de hierarquia de classes, levando em consideração a estrutura organizacional de uma universidade. Através da figura 3.2, é possível observar que, tanto a classe “Professores” quanto a “Alunos”, herdam dados (como por exemplo: Nome, Idade, Endereço, CPF) e operações (adicionar/excluir pessoas) da superclasse “Pessoa”. As sub-classes reusam estas operações e dados através da herança.

Diversas linguagens de programação orientadas a objetos, como C++, suportam o mecanismo de Herança Múltipla, permitindo que uma classe possua várias superclasses.

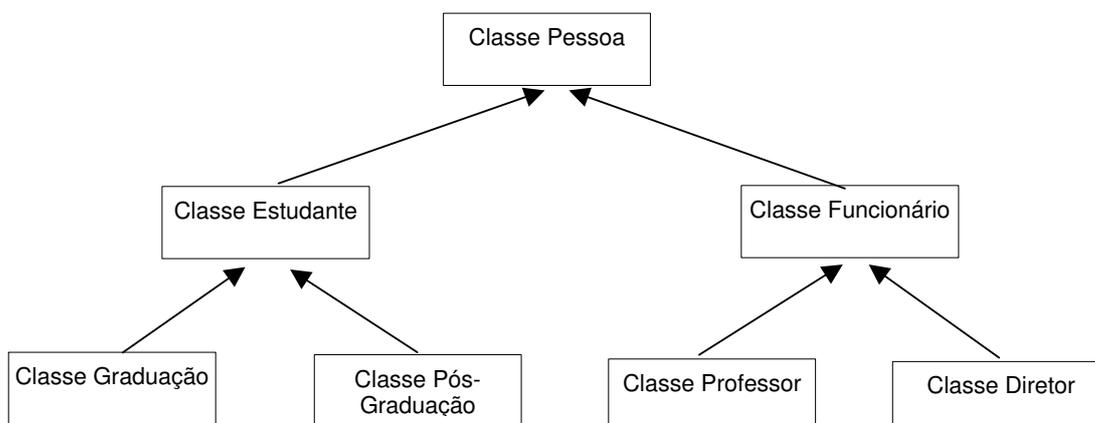


Figura 3.2 – Gráfico de Hierarquia de Classes

Utilizando a idéia de classe, é importante que exista uma implementação na super-classe que possa ser reutilizada em suas sub-classes. O mecanismo de herança também pode ser implementado através do uso de interfaces.

Uma interface é um bloco de código que define um tipo e os métodos e propriedades que esse tipo deve possuir. Classes que desejem ser do tipo definido pela interface devem implementar os métodos dessa interface. A interface não contém nenhum código de implementação, apenas assinaturas de métodos e/ou propriedades que devem ter seu código implementado nas classes que implementarem essa interface.

É possível especular o potencial de reusabilidade de um software. O nível de reusabilidade deve ser relacionado a propriedades internas do software [Bieman, 1992], como:

- Tamanho da classe: uma classe extensa pode ser mais difícil de se compreender, e, portanto, mais difícil de reutilizar;
- Natureza e complexidade do fluxo de controle: uma classe com estruturas de decisão complicadas na implementação de métodos da classe podem ser mais difíceis de serem reutilizadas, especialmente se for necessário modificá-las;

- Natureza e complexidade do fluxo de dados: muitas dependências de dados nos caminhos pelos quais os dados se movimentam, podem dificultar a operação de reuso;
- Tamanho e complexidade das interfaces: interfaces complicadas e grandes dificultam a operação de reuso.

A quantidade de reuso de uma determinada classe está também relacionada a propriedades externas, como a “usabilidade” de uma classe em um domínio de aplicação em específico. Se a funcionalidade de uma classe é freqüentemente necessária, a classe possui maiores chances de ser reutilizada. Outro atributo externo está relacionado à natureza do processo de desenvolvimento de software. Se o reuso for motivado pela direção e/ou suportado por boas ferramentas de classificação e localização de classes, maiores são as chances de reutilização.

3.1.3 Reuso através de padrões de software (Software Patterns)

Padrões (do inglês *Patterns*) são recursos que permitem o compartilhamento do conhecimento e de experiências obtidas através do projeto e desenvolvimento de sistemas computacionais. Desenvolvedores deparam-se freqüentemente com problemas que ocorrem por diversas vezes e que irão ocorrer novamente. A documentação através de padrões de projeto permite o reuso e o compartilhamento de informações que foram aprendidas, sobre a melhor maneira de resolver um problema de projeto em específico. Cada padrão de projeto descreve um problema que ocorre com freqüência no ambiente de desenvolvimento, e aponta uma solução para o problema, de maneira que o mesmo padrão possa ser utilizado muitas vezes [Oliveira, 2004].

Padrões de software não estão relacionados ao uso de uma tecnologia específica. O objetivo da utilização de padrões está em reunir documentações que possam ajudar desenvolvedores a solucionar problemas encontrados durante todas as etapas de desenvolvimento de softwares.

A documentação de padrões é realizada através de formulários bem elaborados, no intuito de definir tópicos como:

- a motivação ou o contexto a que o padrão se aplica;
- os pré-requisitos que devem ser satisfeitos antes de decidir-se pela utilização do padrão;
- a descrição da estrutura do programa que o padrão de projeto irá definir;
- a lista dos participantes necessários para completar o padrão;
- as conseqüências positivas e negativas referentes ao uso do padrão de projeto;
- exemplos.

Existem diversos tipos de padrões para o domínio da engenharia de software, padrão de projeto é um deles. Padrões de Projeto são a descrição de módulos que se comunicam, e que são customizados para solucionar um problema de projeto em um contexto específico.

Raramente um padrão é utilizado isoladamente. Tipicamente, os padrões têm relacionamentos e trabalham em conjunto. Quanto mais familiarizado com os diferentes tipos de padrões de projetos, melhor instruído estará o desenvolvedor, para determinar interações entre esses componentes [Kardell, 1997; Appleton, 2000].

Segue abaixo um *template* de um Padrão de Projeto (*Design Pattern*):

Nome Padrão [*Descreve a essência do padrão em um curto e expressivo nome*].

Objetivo [*Descreve o que o padrão faz*].

Também Conhecido Como [*Lista de sinônimos para o padrão*].

Motivação [*Fornece um exemplo de um problema e como o padrão resolve aquele problema*].

Aplicabilidade [*Lista as situações onde o padrão é aplicado*].

Estrutura [*Conjunto de diagramas de classes e objetos que descrevem o padrão*].

Participantes [*Descreve as classes e objetos que participam do padrão de projeto e suas responsabilidades*].

Colaborações [Descreve como os participantes colaboram para cumprir com suas responsabilidades].

Conseqüências [Descreve os benefícios e os custos da utilização do padrão].

Um padrão de projeto nomeia, abstrai e identifica os aspectos principais de uma estrutura de projeto, tornando-a útil para a criação de projetos reutilizáveis. Identifica módulos (ou classes e suas instancias, em um projeto orientado a objetos) , seus papéis, colaborações e a distribuição de responsabilidades entre os módulos. Cada padrão de projeto identifica um problema ou assunto de projeto (orientado a objetos ou não) em particular.

Para a implementação da técnica de padrões de projeto não é necessária a utilização de ferramentas em específico.

3.2 Tipos de reuso

De acordo com as características do código que está sendo reutilizado, é possível classificá-lo através dos tipos de reuso abaixo [Bieman, 1992]. Os seguintes tipos de reuso podem ocorrer, de acordo com as definições das propriedades do reuso:

- **Reuso Público e Privado:** Reuso Público é definido como a porção do produto que foi construído externamente, ou seja, a entidade que está sendo reutilizada foi desenvolvida em um diferente projeto ou organização. Reuso Privado (ou reuso interno) é definido como a reutilização de módulos que foram desenvolvidos para um produto, neste mesmo produto, porém em diferentes ocasiões;
- **Reuso com e sem Modificação:** o reuso sem modificação ocorre quando o código de uma biblioteca é incorporado ao sistema, sem que haja necessidade de alterá-lo (do inglês: *Verbatim Reuse*). O reuso com modificação, conhecido como *Leveraged Reuse*, ocorre quando se deseja utilizar um pedaço de código já existente, porém é necessário alterá-lo para satisfazer os requisitos do projeto.

- Reuso Direto e Indireto: O reuso direto acontece quando o reuso pode ocorrer sem necessitar de entidades intermediárias. O reuso indireto, em oposição ao reuso direto, é a operação de reuso que ocorre através de entidades intermediárias. Quando o módulo A chama o módulo B, que por sua vez, chama o módulo C, o módulo A esta indiretamente reutilizando o módulo C.

3.3 Modificação

Modificação em engenharia de software é o ato de modificar um sistema ou um componente após sua implementação, para melhorar a performance ou algum outro atributo, ou para adaptar o sistema ou componente para atuar em um ambiente modificado.

A modificação pode ocorrer em um sistema que foi desenvolvido, no intuito de adaptá-lo ou no desenvolvimento de novas aplicações através do reuso. Em engenharia de software, o processo de modificação de um produto de software é denominado manutenção. Manutenibilidade de software é o atributo que caracteriza a facilidade de modificação ou adaptação de um software [Behforooz, 1996]. Muitas vezes, a manutenibilidade é quantificada em termos de tempo médio requerido para a efetivar a revisão do software para eliminar um erro [Schach, 1992].

Assim que um sistema entra em produção, a fase do ciclo de vida que se inicia é a de manutenção. Nesta fase, a correção de um erro ou o melhoramento funcional obriga que a modificação seja analisada, implementada, testada, documentada e integrada. Por este motivo, cada manutenção de software torna-se um processo trabalhoso e delicado, para que o novo código não introduza novos erros e que a documentação do sistema seja atualizada para refletir as modificações realizadas.

Alguns problemas podem dificultar a atividade de manutenção, principalmente para aqueles softwares que atravessam muitos anos em atividade. Esses problemas são:

- Deterioração da documentação: com o passar dos anos, a documentação do software pode não refletir mais a realidade atual do sistema, caso a atualização da mesma não seja fielmente realizada;
- Forte integração de módulos: o módulo a ser modificado pode estar fortemente integrado a outros módulos, dificultando a operação de modificação;
- Alta complexidade algorítmica: quanto mais complexo for módulo a ser modificado, mais difícil torna-se a operação de manutenção;
- Inexistência de controle de configuração do software: sem o controle de configuração do software não é possível distinguir versões do sistema, dificultando a operação de recuperação de uma modificação indevida;
- Inexistência de um ambiente para teste da manutenção: sem um ambiente de testes, a operação de manutenção do software pode ocasionar paradas na operação do sistema, atrapalhando as atividades da empresa;

3.3.1 Processo de modificação

O processo de manutenção visa facilitar a acomodação de mudanças, que são inevitáveis em produtos de software, reduzindo a quantidade de esforço na fase de manutenção.

Este processo deve ser amplo o bastante para suportar manutenção para a manutenção corretiva (correção de erros), para a modificação de uma funcionalidade existente, adaptativa ou a perfectiva (inclusão de funcionalidade). Muitas vezes, diversas ações de mudança são empacotadas juntas em um único bloco, para facilitar a implementação e o controle [Cordeiro, 2000].

O procedimento para conduzir a ação da manutenção deve incluir os seguintes passos:

- Avaliar a requisição da mudança, observando sua real necessidade;

- Avaliar o impacto da mudança dentro do sistema ou em sistemas ao qual está integrado;
- Estimar custo e tempo para implementação;
- Implementar as modificações;
- Testar os módulos modificados;
- Testar a integração do sistema;
- Atualizar a documentação;
- Integrar

3.3.2 Modificação em sistemas orientados a objeto

Sistemas orientados a objetos possuem maior manutenibilidade do que sistemas desenvolvidos através de linguagens estruturadas. Através de sistemas de software estruturados, geralmente, uma pequena modificação pode tornar-se necessária de se realizar em muitos pedaços de código [Kafura, 1996]. Sistemas orientados a objetos tratam de, pelo menos, três dos maiores objetivos da engenharia de software: reusabilidade, extensibilidade e flexibilidade. A propriedade *Reusabilidade* foi discutida anteriormente, no Capítulo 3.1. Neste capítulo, estaremos discutindo sobre a propriedade *Flexibilidade*.

Flexibilidade em sistemas de software, está relacionada a adição, variação ou a modificação que pode ser realizada, sem necessitar que modificar diversos pedaços do código do sistema.

A orientação a objetos contribui para esta propriedade de duas maneiras. A separação da interface de sua implementação é uma delas. Essa separação permite que a interface do usuário permaneça inalterada enquanto sua especificação é modificada, fazendo com que, uma modificação possa ser realizada na implementação (para melhorar a eficiência, a confiabilidade e etc.), sem exigir que modificações sejam realizadas na interface. A segunda maneira está relacionada ao polimorfismo, O polimorfismo permite que, variações e adições sejam realizadas em um conjunto de

classes sobre as quais o polimorfismo aplica-se. Através da Figura 3.3 é possível visualizar este comportamento.

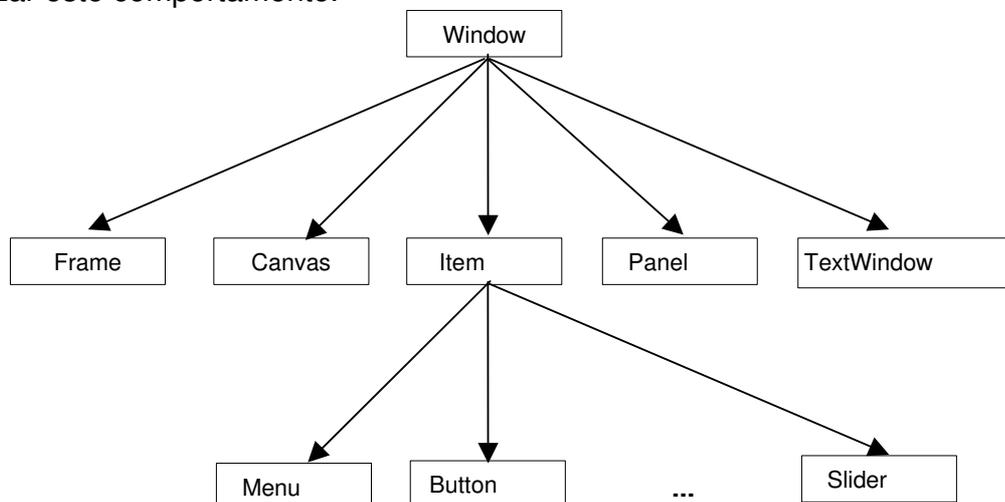


Figura 3.3 – Abstração da hierarquia do Componente Window

Adicionando um novo tipo de interação em **Item** (como por exemplo, *radio buttons*), se **Item** pode ser utilizada por **Panel**, pode parecer necessário alterar o **Panel** para que ele passe a utilizar a nova interação de **Item**, porém, através do polimorfismo, o algoritmo do **Panel** pode relacionar-se somente aos atributos e comportamentos mais generalizados de um objeto (neste caso, o **Item**) e não precisa ter conhecimento do tipo exato do objeto (neste caso, *radio buttons*).

3.3.3 Modificação através de Programação Orientada a Aspectos

A programação orientada a aspectos – Aspects-Oriented Programming ou simplesmente AOP – assim como a programação orientada a objetos introduz um novo paradigma e conjunto de diretrizes para facilitar principalmente a fase de manutenção do desenvolvimento de softwares [Dextra, 2003].

A AOP lida com um problema específico: capturar unidades consistentes de um sistema de software, que as limitações dos modelos de programação tradicionais, forçam a ficar espalhados por diversos pontos do sistema.

A AOP possui alguns conceitos que são fundamentais para sua compreensão:

- *Responsabilidades.* Sistemas de software consistem de um conjunto de “regras de interesse” ou responsabilidades distintas como: responsabilidades funcionais (lógica de negócio) e não funcionais (performance, persistência dos dados, logging, autenticação do usuário, segurança, verificação de erros e etc). Existem também preocupações com o processo de desenvolvimento de software como: a clareza de entendimento, a facilidade de manutenção, a rastreabilidade e etc.
- *Separação de responsabilidades.* A melhor maneira de se projetar um sistema é através da divisão de responsabilidades de tal modo que é possível alterar/re-projetar cada uma delas sem afetar as demais partes do sistema. A eficiência do desenvolvimento aumenta a medida que é possível separar as diferentes responsabilidades em módulos estanques.
- *Responsabilidades transversais.* Em sistemas complexos, sempre existem responsabilidades de interesse comum que são utilizadas por diversos módulos. As responsabilidades não-funcionais geralmente possuem esta característica, mas também algumas funcionais. Estas responsabilidades são difíceis de isolar por que são necessárias em diversas partes do código, resultando em sistemas difíceis de projetar, entender, implementar, manter e evoluir.

O paradigma da AOP consiste na separação das responsabilidades transversais de um sistema em unidades modulares, denominadas “aspectos” e na sua posterior composição junto às classes, formando um sistema único. Os aspectos podem ser inseridos, alterados ou removidos em tempo de compilação. Por estarem em um único bloco de código, sua manutenção é mais simples diminuindo a complexidade do sistema e facilitando o seu entendimento. Além disto, o código das classes fica livre do código relacionado às responsabilidades transversais, facilitando a sua reutilização em diferentes contextos, combinando diferentes aspectos, de acordo com a necessidade da aplicação. A AOP envolve três fases distintas:

- *Fase de Decomposição:* as várias responsabilidades do sistema são identificadas e classificadas como comuns ou transversais.

- *Fase de Implementação*: as responsabilidades são implementadas separadamente em classes (para as responsabilidades comuns) e aspects (para as responsabilidades transversais).
- *Fase de Recomposição*: o sistema é recomposto a partir das responsabilidades implementadas segundo regras de recomposição.

A aplicação de técnicas de programação orientada a objetos para a implementação da responsabilidade transversal resulta em sistemas que são trabalhosos para implementar, difíceis de entender e manter, e também dificultam a evolução do sistema, pois pequenas modificações na especificação podem resultar em intervenções em um grande número de classes.

Utilizando a AOP, pode-se criar implementações que são fáceis de desenhar, entender e manter. Deste modo, AOP traz maior produtividade, qualidade e facilidade para a implementação de novas funcionalidades.

3.4 Tipos de modificação

A literatura classifica a manutenção de software basicamente em três categorias: adaptativa, corretiva e perfectiva [Cappelli et al, 2003].

- *Manutenção Adaptativa*: A manutenção adaptativa realiza modificações no software para que ele possa ser executado em um novo ambiente (nova arquitetura, sistema operacional e etc).
- *Manutenção Corretiva*: A manutenção corretiva trata da modificação do software para melhorias em sua confiabilidade, manutenibilidade, visando corrigir erros residuais observados durante o período de operação do software ou introduzidos durante a própria manutenção.
- *Manutenção Perfectiva*: A manutenção perfectiva ou o melhoramento funcional consiste na modificação para melhorar alguns aspectos do software, de modo a melhorar sua qualidade (melhoria de desempenho, melhoria da interface, a introdução de novas funções e etc).

CAPÍTULO 4

REUSO E MODIFICAÇÃO EM SISTEMAS DE WORKFLOW

4.1 Introdução ao Reuso

Apesar do reuso ser um aspecto relevante no contexto de engenharia de software, estudos visando o reuso de sistemas de workflow não tem apresentado a evolução necessária. O reuso de workflow, consiste no reuso da definição formal do workflow, que descreve as atividades desempenhadas pelo sistema, a atribuição das atividades às entidades de processamento (pessoas e sistemas computacionais) e o relacionamento entre as atividades, como o controle e o fluxo dos dados.

Em analogia ao reuso de softwares, o reuso de workflow é um processo que se baseia no aproveitamento de definições de workflow ou parte delas, para a criação de novos sistemas de workflow. Fazendo-se necessário o desenvolvimento de um novo sistema, ao invés de desenvolvê-lo por completo, este novo workflow é criado utilizando-se como ponto de partida, definições já existentes.

As principais vantagens do reuso em desenvolvimento de software (identificadas na literatura), são: a melhoria da qualidade do software e o aumento da produtividade no desenvolvimento. As mesmas vantagens estão presentes no reuso de workflows:

- O reuso de um workflow existente ao invés da criação integral de um novo modelo, reduz custos de desenvolvimento. Desenvolver módulos de software que serão utilizados uma única vez, geralmente, custa mais caro do que comprá-lo pronto de outra empresa que o desenvolva para ser utilizado em diversas aplicações.
- O reuso também reduz o tempo de desenvolvimento, que na verdade será utilizado somente para a integração do componente reutilizado.

- Workflows que já estiveram em operação são mais confiáveis do que workflows novos, recém desenvolvidos. Porém, quanto mais definições de workflow estiverem disponíveis para o reuso, geralmente, mais difícil torna-se a localização de workflows apropriados para uma situação de reuso em específico. Apesar das vantagens apresentadas, o reuso em sistemas de workflow somente é viável se os esforços necessários para a localização e adaptação de definições existentes, sejam menores que os esforços envolvidos no desenvolvimento completo de novos workflows [Kradolfer, 2000].

Depois que uma definição é escolhida, é importante que o desenvolvedor que a escolheu, compreenda o que ela realmente faz. Informações referentes ao workflow devem estar disponíveis, e de forma suficiente, para esta compreensão. Após verificar se o workflow escolhido é apropriado para o caso em questão (através da compreensão da definição), é necessário avaliar se o workflow atende integralmente as necessidades do novo workflow ou será necessário adaptá-lo para atender aos novos requisitos.

Portanto, para um desenvolvimento voltado ao reuso, algumas definições e preocupações devem estar presentes ao planejar-se o desenvolvimento de um sistema de workflow.

4.2 Considerações relevantes ao Reuso de Sistemas de Workflow

Muitas soluções de workflow não possuem ferramentas facilitadoras das operações de reuso, não oferecendo recursos com vantagens superiores às vantagens apresentadas pelo método de “copy & paste”. Apesar de ser bastante utilizado entre os desenvolvedores que buscam a reutilização de workflows já existentes, este método não permite que desenvolvedores obtenham grandes proveitos de workflows desenvolvidos, para facilitar a operação de reuso.

Para que seja possível reutilizar definições de workflow, assim como componentes de software, diversas considerações devem ser analisadas, no intuito de viabilizá-lo. Este capítulo visa apresentar aspectos relevantes à implementação e compreensão do reuso de sistemas de workflow.

O reuso de workflows compreende as atividades de pesquisa, localização, compreensão, adaptação e de integração de workflows já existentes, para o desenvolvimento de novos workflows. Além disto, algumas práticas devem ser adotadas para facilitar a compreensão e a aceitação do reuso em organizações.

4.2.1 Aspectos organizacionais

Para iniciar a atividade de reuso é fundamental que a gerência da organização esteja ciente dos possíveis ganhos oriundos desta prática, porém, que também tenha ciência das dificuldades iniciais que podem ocorrer, através do processo de introdução do reuso na organização.

Dentre os problemas enfrentados, os mais comuns são:

- Custos iniciais altos, decorrentes da necessidade de investimentos, que devem ser realizados para tornar possível o desenvolvimento *para* o reuso, visando obter ganhos com o desenvolvimento *através* do reuso. O desenvolvimento de modelos de *templates* apropriados, para serem utilizados como ponto de partida a outros projetos, demanda grandes esforços iniciais;
- Falta de compromisso e apoio da gerência. Se a gerência não estiver motivada para o uso e reuso de sistemas de workflow, desenvolvedores e usuários finais dificilmente estarão;
- Falta de confiança em soluções externas. Este problema está relacionado a confiança que usuários terão em soluções feitas por outros usuários, ou seja, na confiança dos usuários na solução de workflow adotada e em definições desenvolvidas *para* o reuso;
- Falta de treinamento para o Reuso. Usuários que estão iniciando um aprendizado sobre sistemas de workflow devem aprender a como tirar proveito de oportunidades disponíveis, e incorporar as práticas de reuso.
- Baixa capacidade relativa à engenharia de software. Organizações que pretendem implementar metodologias de reuso, podem possuir pouca

- experiência em engenharia de software, além de possuir pouco conhecimento em soluções de workflow.

4.2.2 Classificação e Localização de Definições de Workflow

Como mencionado anteriormente, a localização de definições de workflow reutilizáveis, é uma das atividades envolvidas no desenvolvimento de workflows *com* reuso.

A localização de definições de workflow consiste em compreender o que as definições realmente fazem, sem que seja necessário realizar uma análise minuciosa em cada uma delas, visando encontrar definições que possam ser reutilizadas, para um caso em específico. Para que a compreensão de uma definição ocorra, de forma concisa, é necessário oferecer ferramentas que facilitem a pesquisa de definições de workflows reutilizáveis. Para facilitar a localização, técnicas de classificação são adotadas. A seguir, será apresentada uma abordagem de classificação, que lida tanto a abstração concisa de definições, como com mecanismos facilitadores para a pesquisa de definições [Kradolfer, 2000].

A classificação apresentada por [Pietro-Diaz et al., 1987], descreve mecanismos para organizar componentes de software reutilizáveis, para permitir uma pesquisa e localização mais eficiente. Esta organização consiste na escolha de facetas (dimensões em que os componentes de software podem ser agrupados) apresentadas abaixo e classificar a definição de workflow, de acordo com os critérios de cada faceta.

Após a escolha e classificação, a localização de um componente de software dá-se através da pesquisa por palavras-chave, as facetas utilizadas para classificá-lo.

4.2.2.1 Facetas para Componentes de Software

A seguir, duas classificações de componentes de software são apresentadas. A primeira descreve facetas para componentes de softwares orientados a função, enquanto a segunda define facetas voltada para componentes de softwares orientados a objeto. Cada uma das facetas apresentadas será analisada quanto à usabilidade para classificação de definições de workflows.

a) Facetas para Componentes orientados a Função

Em [Prieto-Diaz et al., 1987], a classificação proposta permite descrever as funcionalidades e o ambiente dos componentes de software orientados a função. As funcionalidades dos componentes são descritas através das seguintes facetas:

- *Função*: descrição da operação que é executada pelo componente (exemplo: comparar, adicionar);
- *Objetos*: descrição do tipo dos dados que são manipulados durante a execução do componente (exemplo: valores inteiros, arquivos);
- *Meio*: descrição do meio utilizado para armazenar os dados da aplicação (exemplo: arquivo, base de dados, fita)

Das facetas acima, a mais adequada para a classificação de definições de workflow é a faceta *Função*, já que definições de workflow são consideradas funções de alto-nível. A faceta *Objetos* não é adequada pois as informações referentes aos dados que são manipulados já estão especificados através dos parâmetros de entrada e saída da definição de workflow. Assim como a faceta *Objetos*, a faceta *Meio* não é desejada, pois lida com informações de baixo-nível, tornando-se desinteressante a modelagem e ao reuso de workflow.

O ambiente de execução dos componentes, pode ser descrito através das seguintes facetas:

- *Tipo de Sistema*: descrição do tipo de sistema para o qual o componente foi criado (exemplo: assembler, base de dados relacional);
- *Área Funcional*: descrição da área funcional ao qual o componente é utilizado (exemplo: faturamento, produção);
- *Cenário de uso*: descrição do cenário de uso onde o componente é utilizado (exemplo: loja de roupas, empresa logística)

Das facetas que descrevem o ambiente de execução, as facetas *Área Funcional* e *Cenário de Uso* são proveitosas para a classificação de definições de workflow. A faceta *Tipo de Sistema*, não se aplica diretamente a classificação de definições de

workflow já que, durante a execução de um workflow, uma variedade de sistemas (e humanos) pode ser envolvida. Em diferentes instâncias de uma mesma definição de workflow, diferentes entidades de processamento podem ser envolvidas.

b) Facetas para Componentes Orientados a Objetos

O Projeto REBOOT (REuse Based on Oriented-Objects Techniques), foi elaborado para estudar mecanismos de reuso em desenvolvimento de softwares. Com duração de quatro anos, o projeto deu origem a um livro sobre reuso de softwares e a metodologia REBOOT [Kruke, 1996].

A metodologia REBOOT apresenta uma classificação de componentes orientados a objeto, sugerindo o uso das seguintes facetas:

- *Abstração*: descrição do componente (exemplo: pilha);
- *Operações*: descrição das operações suportadas pelo componente (exemplo: push, pop);
- *Opera-em*: descrição do tipo de dados aos quais o componente atua (exemplo: inteiro);
- *Dependências*: descrição das dependências não-funcionais, limitantes a aplicabilidade do componente.

Devido a natureza orientada a função das definições de workflow, as facetas propostas por Pietro-Diaz e Freeman [Pietro-Diaz et al., 1987] são mais adequadas para classificar definições de workflow do que as facetas apresentadas pelo Projeto REBOOT, que são orientadas a objetos. Mesmo assim, a faceta *Dependências*, pode ser utilizada para capturar relações de dependências entre definições de workflow.

4.2.2.2 Facetas para Definições de Workflow

Baseando-se nas facetas existentes, voltadas à classificação de Componentes de Softwares, Kradofer [Kradofer,2000] propôs as seguintes facetas, voltadas a classificação de definições de workflow:

- *Funcionalidade*: descreve a funcionalidade de um workflow em um nível alto de abstração. A descrição é feita baseando-se no que o workflow faz. O “como é feito” não é discutido por esta faceta;
- *Cenário de Uso*: descreve em que tipo de organização, o workflow pode ser utilizado. Esta faceta é de grande importância quando aplicada em cenários onde definições de workflow de diversas organizações são envolvidas;
- *Área de Negócio Funcional*: descreve as áreas de negócio funcionais, as quais a definição de workflow pode ser aplicada. Quatro grandes áreas funcionais são apontadas como as áreas-mãe, localizadas no topo da hierarquia de áreas, devendo abranger todas as outras áreas. Estas áreas são: manufatura e operação, marketing e vendas, contabilidade e recursos humanos. Cada uma destas áreas é composta por sub-áreas mais especializadas. Uma definição de workflow pode ser utilizada em mais de uma área funcional.
- *Dependência*: esta faceta captura as relações de dependência existentes entre definições de workflow. Uma definição de workflow depende de outra, se declarar em sua definição, um ou mais sub-workflow de outra definição.

4.2.3 Compreensão das Definições de Workflow

Para que um desenvolvedor de sistemas de workflow seja capaz de avaliar se uma definição de workflow é de interesse, e deve ou não ser reutilizada, é necessário que a definição forneça informações suficientes para este julgamento. Além das informações apresentadas pela classificação, existem outras duas fontes de informação presentes nas definições de workflow: o modelo e o histórico de execução do workflow. O Modelo de workflow contém diversas informações referentes à definição de workflow, como parâmetros de entrada e saída utilizados e os sub-workflows. O histórico de execução, por outro lado, armazena informações sobre as instâncias do workflow.

4.2.3.1 Informações contidas no Modelo de Workflow

Diversas informações sobre a definição do workflow podem ser extraídas do modelo de workflow. Cada uma das definições do workflow é decomposta em um nome

textual, descrição de parâmetros de entrada, definições do estado final e dos parâmetros de saída, declaração de sub-workflows e definições de dados e de fluxos de controle. Cada uma destas partes revela informações que podem ajudar o desenvolvedor a ganhar conhecimento sobre o workflow.

A complexidade de sistemas é uma característica relevante ao reuso. Para medir a complexidade de componentes de softwares, diversas métricas são sugeridas na literatura. A medição da complexidade de definições de workflow pode ser realizada considerando-se o número de sub-workflows definidos, e da complexidade dos dados e fluxos de controle apresentados em um modelo de workflow. Um alto número de fluxos de controle (arcos observados no modelo), também indica um alto nível de complexidade, e vice-versa.

4.2.3.2 Informações contidas no Histórico de Execução do Workflow

O histórico de execução de um workflow, representa outra fonte de informações sobre as definições de workflow, pois possui informações referentes às instâncias do workflow. Todas as informações referentes ao Estado de Transição dos workflows são armazenadas como eventos, que possuem diversos atributos [Kradolfer, 2000]. Através do histórico de execução, uma variedade de informações pode ser obtida:

- Identificação de atividades problemáticas: através do histórico de execução é possível observar momentos onde o fluxo está tendo dificuldades de avançar, ou seja, o fluxo tem alto índice de retrocesso, o que pode indicar uma má modelagem ou especificação do workflow;
- Duração: é possível controlar a duração média, mínima e máxima da execução de uma instância de workflow em específico.
- Resultados: o histórico de execução registra os momentos em que a instância atinge o estado final de sua execução, bem como os dados que são produzidos ao atingir-se o estado final. Para cada instância de workflow, pode-se coletar estatísticas sobre os resultados obtidos através de cada um dos estados finais possíveis.

- Caminhos de execução: o caminho de execução de um workflow, como por exemplo, o conjunto de atividades que foi realmente executado, pode ser extraído do histórico de execução da instância de workflow.

4.2.4 Gerenciamento de Configurações

De acordo com Kruke [Kruke, 1996], a utilização de Gerenciamento de Configurações (CM – *Configuration Management*) em sistemas de workflow visa gerenciar versões, revisões e variantes, originadas a partir de criações e modificações realizadas no sistema.

Versões, revisões e variações são descrições que visam separar diferentes modelos, de acordo com o relacionamento existente entre eles. Para que seja possível tirar proveito destas informações, utiliza-se Gerenciamento de Configurações para controlar as diversas versões de um modelo, permitindo que versões mais antigas não sejam sobrepostas por versões mais atuais (caso contrário, para pouco serviriam as informações sobre as versões). Através do Gerenciamento de Configurações, modelos antigos podem ser recuperados, porém enquanto não estão em uso, são ocultados para não poluir o sistema com informações eventualmente desnecessárias.

Uma maneira de exemplificar o uso de versões e variantes em sistemas de workflow pode ser:

Um processo foi modelado de acordo com análises realizadas seis meses atrás. Este modelo funcionou perfeitamente nos primeiros quatro meses, porém alterações no mercado fizeram com que novos métodos internos à organização fossem adotados. Para satisfazer esta nova realidade, decidiu-se que um novo modelo deveria ser desenvolvido. Como muitas das atividades são as mesmas executadas anteriormente, decidiu-se alterar o modelo antigo ao invés de desenvolver por completo um novo modelo. Esta decisão resultou na liberação de uma nova versão, em substituição ao modelo anterior (versão inicial foi *modificada* para atender novos requisitos).

Tempos depois, percebe-se que, para atuar melhor no mercado, são necessários dois modelos para lidar com todos os aspectos da organização.

Como resultado desta necessidade, a versão corrente do modelo foi modificada para atender de forma especializada cada um dos segmentos do mercado, dando origem a duas novas versões, representando duas variações da mesma versão (variações foram criadas *reutilizando* uma versão existente)

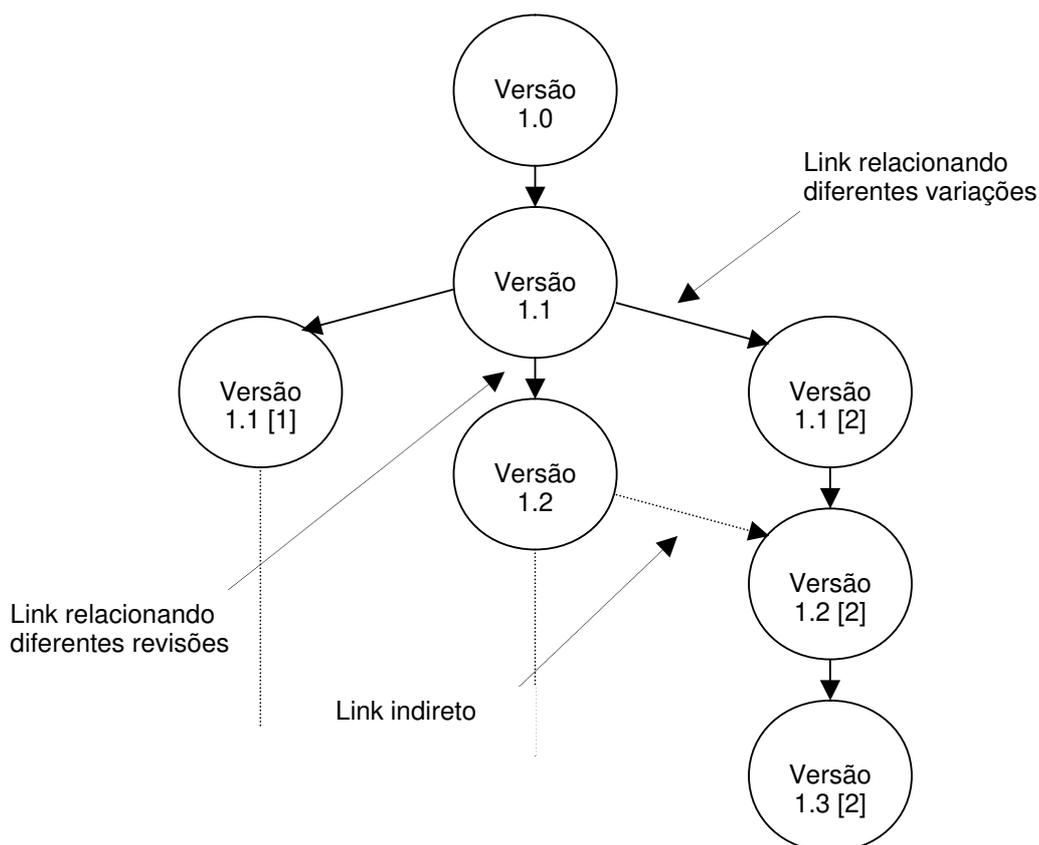


Figura 4.1 – Gráfico de versão indicando o funcionamento do Gerenciamento de Configurações (Versão = Modificação / Variação = Reuso + Modificação).

O Gerenciamento de Configurações gera uma a nova definição de processos, sempre que ocorrem modificações nas definições de processo. A nova versão pode tanto ser uma *revisão*, substituindo a versão anterior, ou pode ser uma *variação*, coexistindo com outras versões. Este assunto é bastante amplo. Mais informações sobre Gerenciamento de Configurações podem ser obtidas em [Kruke, 1996].

4.3 Práticas de Reuso

Visando um maior aproveitamento de workflows, usuários finais necessitam de mecanismos que facilitem o reuso, estejam integrados ao sistema (Capítulo 4.2). Se um usuário tiver que desenvolver um modelo do início, toda vez que desejar um novo modelo, o propósito do uso de sistemas de workflow é disperso.

Nesta monografia, utiliza-se como definição ao Reuso, a habilidade de usar definições de workflow terminadas ou semiterminadas como ponto de partida ao desenvolvimento de novos workflows.

Um aspecto importante do reuso, é a habilidade de coletar informações referentes a experiências passadas, obtidas através da modificação ou da implementação de um workflow. A identificação destas experiências pode dar origem a uma base de conhecimento, que servirá de referência a criação, modificação, reuso e implementação de sistemas de workflow. Para o enriquecimento da base de conhecimento, é importante adicionar além das políticas da organização, as práticas adotadas, discutidas anteriormente no Capítulo 2.2.

Organizações que operam de forma adaptativa ao mercado, precisam adequar-se rapidamente às mudanças do ambiente em que se encontram. Isto envolve alterações na estrutura interna da organização e na maneira como os processos são executados. Em consequência disto, sistemas de workflow utilizados por estas organizações, devem ser capazes de suportar tais mudanças.

Sistemas de workflow voltados à produção enfrentariam dificuldades para adequar-se a estas modificações, já que são otimizados para atuar junto a processos repetitivos e não são projetados para responder a mudanças rápidas. Sistemas de workflow considerados ad-hoc ou flexíveis, são desenvolvidos tendo-se em mente esta possível necessidade de adaptação de organizações, as mudanças do mercado.

4.3.1 Reuso de Definições de Workflow

Para uma nova definição de workflow, o processo de desenvolvimento deve considerar o reuso de definições de workflow já existentes, no intuito de aproveitar as

vantagens relativas à qualidade e produtividade obtidas com o reuso. A seguir, são apresentados dois processos de desenvolvimento de workflow: o primeiro define as etapas do processo para um desenvolvimento de workflow sem considerar o reuso, e o segundo processo, enfatiza o desenvolvimento considerando o reuso de definições de workflows.

4.3.1.1 Processo de Desenvolvimento de Aplicações de Workflow

O desenvolvimento de sistemas de workflow, que consiste na modelagem do workflow e na identificação de suas entidades de processamento, é uma tarefa complexa que exige uma boa estruturação de processos a serem executados. Estes processos definem as atividades que devem ser realizadas e o resultados esperados de cada atividade.

Baresi e outros [Baresi et al., 1999] propõem um processo de desenvolvimento de aplicações de workflow, conforme apresentado na Figura 4.2, que seguem as etapas clássicas do desenvolvimento de software: análise, projeto e implementação:

- *Fase de Análise:* Para a fase de análise, assume-se que descrições dos processos de negócio em UML e objetivos de negócio, já estejam disponíveis. Durante a fase de análise, é determinado se os objetivos de negócio da organização podem ser alcançados através do uso da tecnologia de workflow. Caso seja possível atender as necessidades da organização, é necessário identificar as soluções de workflow do mercado, os papéis a serem desempenhados pelas entidades de processamento, e os objetivos a serem atingidos pelo workflow.
- *Fase de Projeto:* Na fase de projeto, um workflow é modelado, identificando o fluxo normal dos dados, assim como as exceções que podem ocorrer durante a execução do workflow.
- *Fase de Adequação ao WFMS:* Durante a última fase, o workflow desenvolvido é adequado a solução de workflow escolhida. Esta adequação não é uma atividade simples, principalmente em sistemas com muitas atividades.

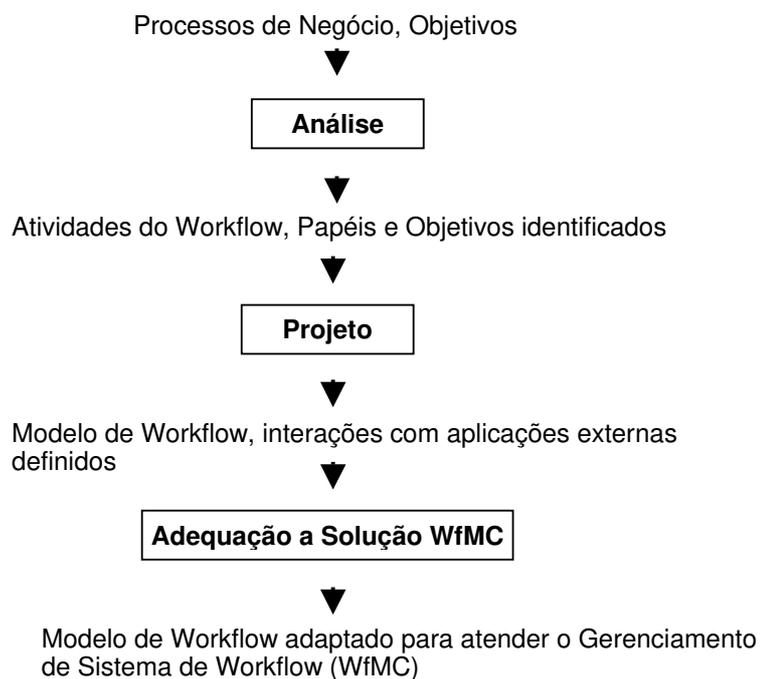


Figura 4.2 – Processo de Desenvolvimento de Aplicações de Workflow [Baresi et al., 1999]

Weske e outros [Weske et al., 1999] propõem um processo de desenvolvimento de aplicações de workflow um pouco mais amplo que o apresentado por Baresi [Baresi et al., 1999]. O processo consiste das seguintes seis fases (referência a Figura 4.3):

- *Fase de Pesquisa:* Nesta fase do desenvolvimento, os objetivos do projeto são identificados e informações para delimitação do escopo do projeto são coletadas, através de entrevistas e análise de documentos. Os processos de negócio que serão gerenciados pelo workflow são estudados e um modelo de processo de negócio é desenvolvido.

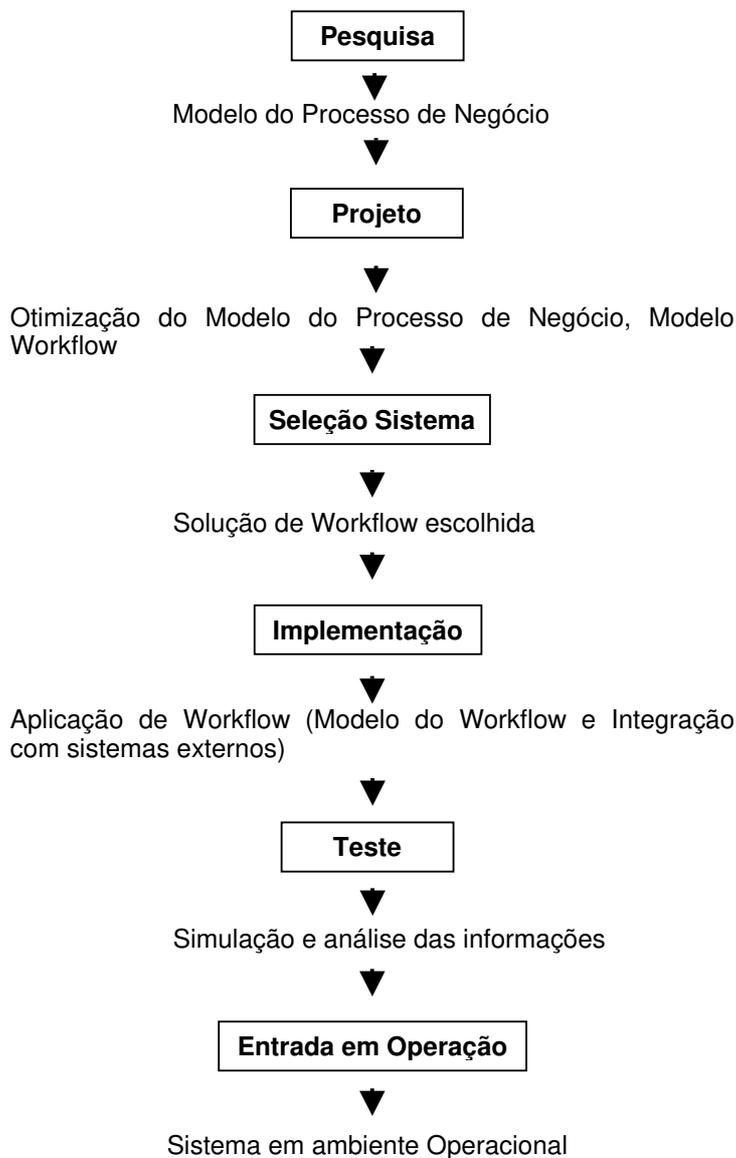


Figura 4.3 – Processo de Desenvolvimento de Aplicação de Workflow [Weske et al., 1999]

- *Fase de Projeto:* Na fase de projeto, o modelo do processo de negócio desenvolvido na fase anterior é analisado e otimizado, levando em consideração os objetivos globais da organização. Após a análise, é realizada a modelagem do workflow, baseando-se no modelo do processo de negócio. Informações técnicas são adicionadas ao modelo do processo de negócio e informações irrelevantes a execução do workflow são descartadas.

- *Fase de Seleção do Sistema:* Na fase de seleção do sistema, uma solução de workflow (WFMS) que satisfaça os requerimentos apontados, é selecionada.
- *Fase de Implementação:* Nesta fase, workflow é implementado utilizando-se a solução escolhida. Aplicações externas são integradas, quando julgadas necessárias.
- *Fase de Teste:* Na fase de teste, a aplicação de workflow é validada. Simulações da execução do sistema de workflow, assim como acompanhamento da execução do sistema em ambiente operacional, são realizadas.
- *Fase Entrada em Operação:* Na fase operacional, o sistema completo é instalado e colocado em prática. A aplicação de workflow é monitorada e as informações geradas e gerenciadas pelo sistema são coletadas e analisadas, visando a melhoria contínua do processo.

Nas duas abordagens apresentadas, o desenvolvimento de aplicações de workflow é feito através de uma fase de projeto, que não considera o reuso de modelagens de workflow já existentes. O processo de desenvolvimento de aplicações de workflow a seguir, prevê o reuso em suas fases de desenvolvimento.

4.3.1.2 Processo de Desenvolvimento Aplicações de Workflows baseado em Reuso

O processo de desenvolvimento de aplicações de workflow, em contraste as abordagens apresentadas, enfatiza o reuso de definições de workflow já existentes. As fases deste processo fazem referências às considerações relevantes ao reuso de workflows, apresentadas no Capítulo 4.2. O processo de desenvolvimento, como ilustrado na Figura 4.4, consiste das seguintes fases:

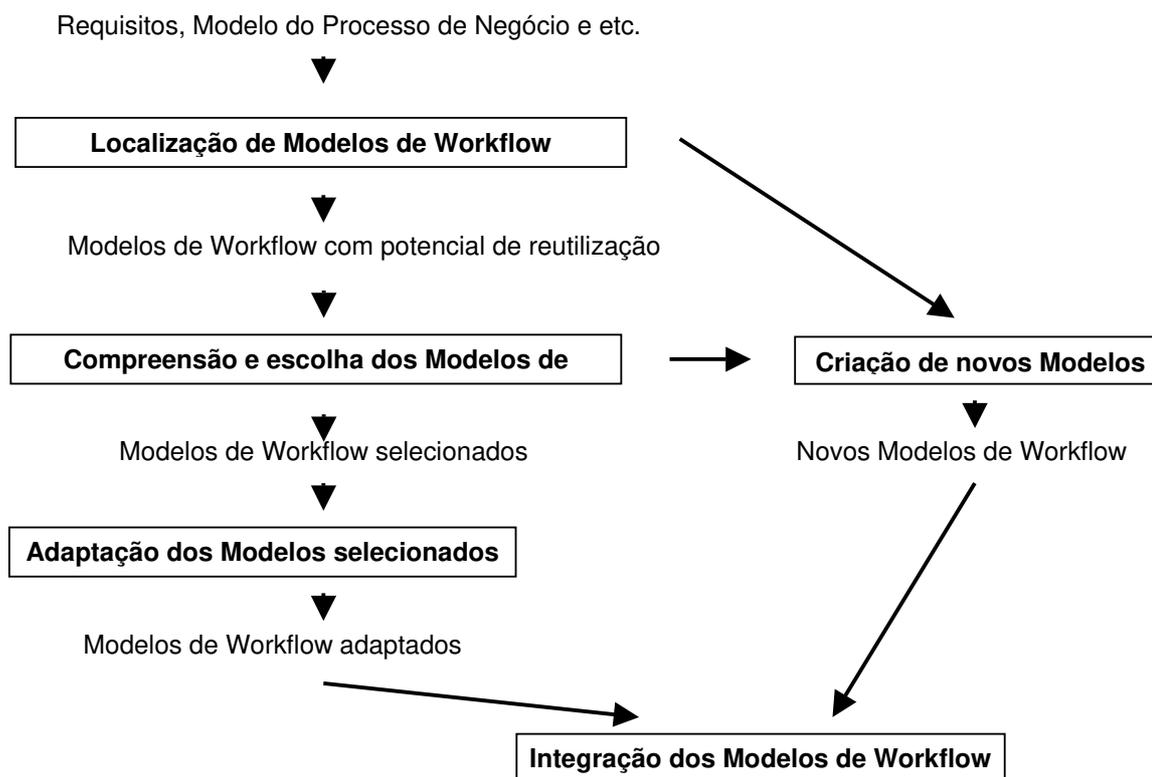


Figura 4.4 – Desenvolvimento de Modelos de Workflow baseados em Reuso

- *Localização de modelagens de workflow reutilizáveis:* antes de iniciar-se o desenvolvimento de aplicações de workflow, uma análise do processo a ser implementado é realizada, resultando em descrições de requisitos do processo e no modelo do processo de negócio. Baseando-se nestas informações, dá-se início ao processo de desenvolvimento, que compreende a localização das modelagens de workflow que podem ser reutilizadas.
- *Compreensão e escolha das modelagens de Workflow:* nesta etapa, o desenvolvedor tenta entender as funcionalidades e o modo de operação das modelagens de workflow localizadas, no intuito de identificar o que realmente pode ser reutilizado na aplicação a ser desenvolvida.

- *Adaptação das modelagens escolhidas:* Parte das modelagens de workflow escolhidas pode não satisfazer integralmente a necessidade da nova aplicação, sendo necessário adaptá-las ao novo uso.
- *Criação de novas modelagens de workflow:* Em adição ao reuso de modelagens existentes, novas modelagens de workflow podem ser necessárias para satisfazer os requisitos do sistema. No entanto, é importante que, para cada nova modelagem, tenha-se considerado anteriormente o possível reuso de modelagens já existentes.
- *Integração das modelagens de workflow:* As modelagens de workflow, incluindo as modelagens existentes que não precisaram ser adaptadas, as modelagens existentes que foram adaptadas e as novas modelagens realizadas, são integradas em um novo workflow.

A abordagem apresentada que enfatiza o reuso de workflow é composta por diversas fases de desenvolvimento, e leva em consideração o cumprimento dos seguintes requisitos:

- *Localização:* a localização de modelagens de workflow apropriadas é um dos maiores problemas para o reuso de workflows. Esta dificuldade pode ser minimizada se o modelo for composto por um número reduzido de atividades, entidades de processamento e etc, permitindo que o desenvolvedor consiga inspecionar cada uma das definições do workflow. Porém, se a quantidade de definições em uma modelagem for grande, o trabalho e inspeção tornam-se inviável. Mecanismos que visem facilitar a localização de modelos devem ser utilizados, como o proposto no Capítulo 4.2.2;
- *Compreensão:* para compreender uma modelagem de workflow, o desenvolvedor pode observar os detalhes da modelagem, como a classificação, os parâmetros e sub-workflows definidos. Além das informações elementares, o modelo assim como seu histórico de execução, possui informações complementares (frequência de reuso de um modelo, duração da execução de uma instância do workflow).

- *Adaptação*: ao adaptar um workflow, um novo workflow pode ser criado. Novas versões, revisões ou variações podem ser criadas. Para mais detalhes, veja o Capítulo 4.2.4;
- *Integração*: o workflow suporta a integração de modelos, permitindo a criação de novos modelos de workflow a partir de modelos já existentes.

4.3.2 Reuso através de *Templates*

Templates são a descrição de um fluxo através de uma representação gráfica demonstrando todos as atividades do fluxo de processo, o relacionamento entre as atividades e as conexões entre o fluxo de processo e entidades externas (aplicativos). Um workflow pode ser descrito por um ou por múltiplos *templates*. *Templates* são criados para serem utilizados como ponto de partida para o desenvolvimento de novos projetos. Através do uso de *Templates*, é possível assegurar que determinadas medidas de controle de qualidade foram levadas em consideração, e fornece um conjunto de configurações e padrões através do qual o projeto deve ser executado [Cappelli et al., 2003].

Como um *template* é uma descrição estática de um fluxo, é possível reutilizá-la constantemente. Um ou mais workflows podem ser criados usando o mesmo *template*, porém cada qual mantém seus próprios dados.

Diversas soluções de workflow utilizam ou recomendam o uso de *templates* como o mecanismo disponível para aplicar-se o reuso de definições de workflow. Desenvolvedores são estimulados a criar *templates* que possam servir como base a novos desenvolvimentos. Para reutilizar um *template*, basta salvá-lo com um novo nome e executar as modificações necessárias sobre o novo *template* [Kruke, 1996].

Porém, modificações em grupos de objetos em um mesmo *template*, não podem ser realizadas de uma única vez. Alterações no novo *template* devem ser realizadas modificando-o objeto por objeto (atividade por atividade).

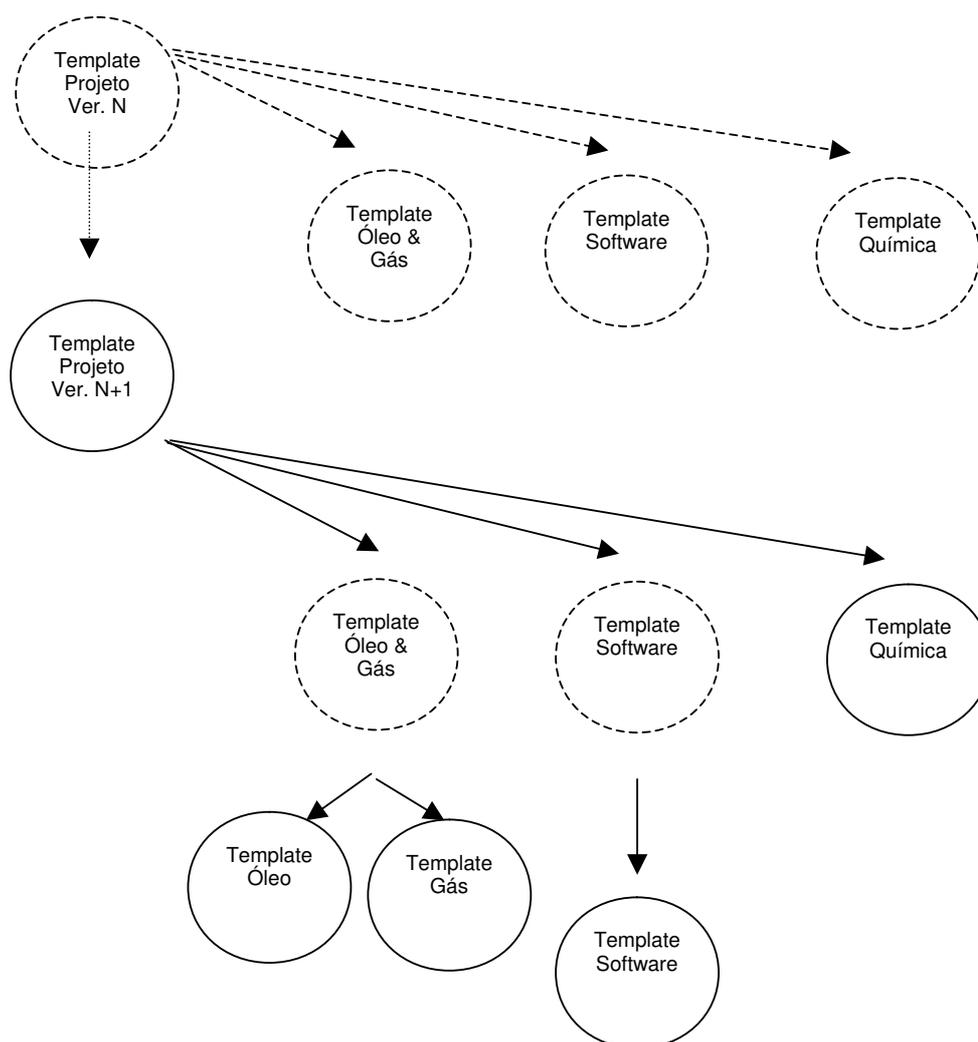


Figura 4.5 – Gráfico de versões ilustrando o uso de templates em uma organização.

Como pode ser observado na figura 4.5, com o passar do tempo e através da experiência adquirida com o uso dos templates mais antigos, templates dão origem a novas revisões e variações.

Existem soluções de workflow que trabalham com o conceito de camadas de templates, permitindo que um processo seja dividido em sessões, onde cada sessão é um template diferente.

4.3.3 Reuso através de Sub-workflows

Sub-workflows segmentam uma definição de workflow (processo) em fluxos menores, dividindo a complexidade do fluxo e dando maior potencial de reuso aos sub-

workflows. Um sistema de workflow pode fazer uso de um ou mais workflows para executar as atividades compreendidas pela definição de um processo. Estas atividades podem ou não ser sub-fluxos (sub-workflows), porém, para o workflow que as invoca, em ambos os casos as atividades são consideradas primitivas ao fluxo principal. Em tempo de execução, o workflow e seus sub-workflows são executados dentro de um mesmo sistema de workflow, compartilhando do mesmo mecanismo de controle de tempo de execução [Krishnan et al., 2001].

4.3.4 Reuso através de princípios de Orientação a Objetos com Micro-workflows

Micro-workflow é uma nova arquitetura de workflow, proposta por Manolescu [Manolescu, 2001], que se baseia no estilo da arquitetura orientada a objetos e permite a desenvolvedores de software a criar e executar workflows fazendo uso de três características de objetos: encapsulamento, herança e polimorfismo. Micro-workflows são voltados a um público alvo diferente do público a que se destinam os sistemas de workflow atuais. Micro-workflows são voltados a desenvolvedores de softwares orientados a objetos. Provêm as funcionalidades básicas de sistemas de workflow, porém deslocam o foco no empacotamento de funcionalidades presentes nos workflows atuais, para atender as seguintes características:

- Permitir que desenvolvedores possam escolher as funcionalidades desejáveis ao sistema;
- Direcionar a funcionalidade a um domínio de aplicação em específico e,
- Estender a arquitetura do sistema com novas características na medida em que for necessário.

As características apresentadas pelos micro-workflows visam dar maior flexibilidade aos sistemas e aumentar o potencial de reuso através das características da orientação a objetos. Micro-workflows possuem uma arquitetura leve, que tratam a funcionalidade principal de sistemas de workflow: permitir que usuários definam e executem workflows. De acordo com Manolescu [Manolescu, 2001], a arquitetura dos sistemas de workflow é pesada, dificultando o reuso, a customização e a integração dos

sistemas de workflow com outros ambientes.

Para solucionar estes problemas, micro-workflows dividem a arquitetura do sistema em pedaços (módulos ou componentes) com diferentes responsabilidades e que tratam de interesses distintos. Como micro-workflows são direcionados a desenvolvedores de sistemas orientados a objetos, a solução provê acesso direto dos desenvolvedores ao processo do modelo (e não á sua interface), para que seja possível definir processos segundo a técnica de orientação a objetos. Ao invés de manipular ícones através de interfaces gráficas, desenvolvedores utilizando micro-workflows definem processos através da composição de objetos.

Ao contrário dos sistemas de workflow que lidam com processos envolvendo pessoas e aplicações, micro-workflows lidam com processos envolvendo objetos. O estágio de execução de um processo que ocorre em sistemas de workflow, é considerado para micro-workflows como o processo de instanciação de um objeto, e a definição de processos corresponde à definição de uma classe. Desenvolvedores são capazes de adicionar novas estruturas de controle de workflow criando sub-classes de classes abstratas. Manolescu considera que, o paralelo traçado entre a instanciação de objetos e a execução de processos, permite uma visualização mais suave entre a implementação da tecnologia de objetos e a tecnologia de workflows.

Micro-workflows não provêm diversas características de sistemas de workflow. Para que desenvolvedores obtenham estas características, é necessário prover outras funcionalidades ao micro-workflow. Por exemplo, o mecanismo de envio de mensagens utilizado por objetos, suportado pelo micro-workflow, não é apropriado para lidar com pessoas. Humanos possuem longos tempos de resposta e contribuem para tornar instâncias de workflow com longa duração. Para adicionar a característica de lidar com pessoas ao processo, o *core* do micro-workflow deve suportar um mecanismo apropriado de invocação a humanos. Da mesma forma, através de técnicas específicas de sistemas orientados a objetos, podem ser adicionadas aos micro-workflows, características avançadas de workflows, como: o componente para monitoramento de sistema de workflow, o de histórico de execução, de persistência de dados, o *worklist*, e outros.

4.4 Introdução a Modificação

O conjunto de uma ou mais regras e procedimentos ou atividades ligadas, realizando coletivamente objetivos de negócio dentro de um contexto organizacional, é uma definição possível à expressão Processos de Negócio.

Uma propriedade importante dos Processos de Negócio é a sua rigidez. O grau de rigidez determina a natureza do processo, de forma que o trabalho a ser desempenhado pode ser completamente conhecido e pré-determinado antes do início do processo ou pode ser imprevisível, onde o trabalho que será necessário não é previamente conhecido, ganhando forma durante o progresso do processo [Sadiq et al., 1998].

Ao discutir-se modificações em Processos de Negócio é importante avaliar quando um processo estrategicamente implementado ou pouco conhecido, pode ser alterado durante seu período de execução. Recomenda-se que, somente modificações previstas inicialmente sejam realizadas em um processo, porém alguns processos de negócio são e precisam ser, ter a flexibilidade de serem a princípio indeterminados. Em uma linha de produção, a flexibilidade no processo pode levar a resultados catastróficos. Neste caso, as etapas são meticulosamente estudadas e definidas, para serem executadas com precisão. O elemento de escolha ou de flexibilidade não é desejado, a não ser que se esteja em fase de modificação do processo para, por exemplo, utilizar-se uma nova tecnologia.

A maioria dos Processos de Negócio encontra-se em um grau de rigidez intermediário, ou seja, processos são definidos antes de serem executados porém necessitam ou permitem flexibilizações. Processos de uma mesma natureza podem possuir diferentes regras e procedimentos em diferentes organizações. Estes processos são passíveis a modificação. Modificações ocorrem devido a três razões principais:

- *Melhoria do Processo*, visando dar continuidade ao processo porém aumentando sua eficiência.

- *Inovação do Processo*, visando realizar o processo de uma forma radicalmente diferente da anterior, envolvendo reengenharia e dando origem a novos conceitos referentes ao Processo de Negócio.
- *Adaptação do Processo*, visando adaptar o processo à mudanças imprevistas.

As modificações resultantes de Melhorias e Inovações são alterações cuja execução são previstas. Já a modificação por adaptação lida com situações imprevisíveis, portanto não podem ser planejadas. A Inovação de Processos introduz modificações nos Processos de Negócio através da introdução de conceitos anteriormente inexistentes. Este tipo de modificação gera discussões para determinar se elementos antigos devem ou não continuar sendo utilizados no novo Processo. As modificações resultantes de Melhoria e Inovação são permanentes, enquanto as modificações realizadas visando atender a adaptações podem ser exigidas por um determinado período de tempo ou ocasião. O Quadro abaixo compara as 3 razões apresentadas, acrescentando-se outros critérios.

Tabela 4.1 – Comparação entre os tipos de modificação

	Melhoria	Inovação	Adaptação
Nível de Modificação	Incremental	Radical	Desconhecido
Ponto de partida	Processo existente	Conceitos / tendências	Processo existente
Frequência da Modificação	Uma única vez / Contínua	Uma única vez	Uma única vez / Diversas vezes
Tempo exigido	Curto	Longo	Curto
Participação	Bottom up	Top down	Bottom up
Risco	Moderado	Alto	Desconhecido
Tipo de Modificação	Cultural	Estrutural	Desconhecido
Planejamento	Planejado	Planejado	Não planejado / Imprevisto
Efeitos da Modificação	Permanentes	Permanentes	Temporários / permanentes

As modificações em Processos de Negócio acarretam diversos efeitos nos sistemas de workflow. Para a compreensão da modificação em sistemas de workflow, é importante conhecer as políticas possíveis de serem adotadas para lidar com estes efeitos:

- *Nivelamento (ocorrências em paralelo)* – Todas as instancias correntes do processo podem ser completadas porém novas instancias devem seguir o novo modelo. Desta forma, a migração ao novo modelo ocorre gradativamente. Situações de Nivelamento ocorrem tanto em modificações visando melhoria quanto a adaptação de processos.
- *Aborto* – instancias em execução devem ser abortadas quando o modelo do processo for modificado
- *Migração* – as modificações afetam todas as instancias correntes, porém precisam ser introduzidas sem que as instancias sejam abortadas ou que haja execução das instancias em paralelo. O maior problema neste caso, ocorre quando instancias atingem um estágio no sistema de workflow, onde as atividades já executadas modificaram os dados de tal forma que as atividades que estão por vir são incapazes de utilizá-los de acordo com a nova especificação. Migração é a política de modificação mais importante, podendo ser requerida para inovação, melhoria e adaptação de processos.
- *Adaptação* – Esta política esta limitada a Adaptação do processo. Inclui casos de erros e exceções, onde o modelo do processo não é alterado, porém instancias precisam ser tratadas diferentemente devido a circunstâncias excepcionais.
- *Criação* – Elaboração de novos processos também é um tipo de modificação do processo. A diferença é que o ponto de partida para a criação não se baseia em modelos pré-existentes, utilizando-se somente de definições de um processo.

A característica comum destas políticas de modificação, com exceção do Nivelamento, é que elas afetam instancias correntes do modelo de processo. Estas

políticas estão relacionadas a modificações dinâmicas em sistemas de workflow, por estarem envolvendo e lidando com instancias do processo. No caso de modificações estáticas, o modelo de workflow é substituído porém não há preocupação com as possíveis instancias correntes do processo.

As figuras abaixo ilustram possíveis modificações em um fluxo previamente definido:

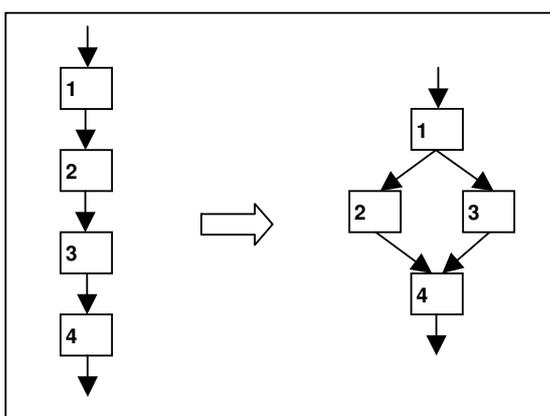


Figura 4.6 - Modificação da Estrutura das atividades.

A etapa 3 tornou-se independente da etapa 2. A etapa 4 pode, após a modificação, ser atingida tanto através da execução da etapa 2 quanto da etapa 3, o que não ocorria no fluxo inicial.

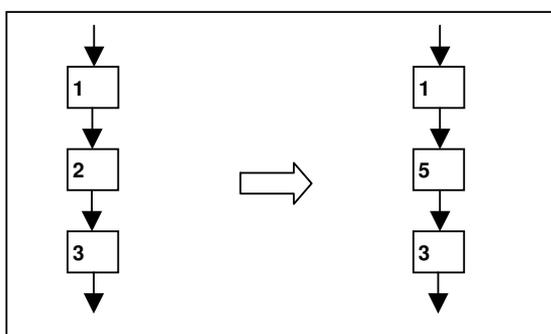


Figura 4.7 - Modificação nas propriedades das Atividades.

Após a modificação, tornou-se necessário que a etapa 5 ocorresse antes de executar-se a etapa 3. Antes da modificação, fazia-se necessário à execução da etapa 2.

Modificações em workflows podem ser permanentes ou temporárias e, de diversas formas. Todas as instâncias correntes podem ou não ser afetadas. A modificação pode afetar diferentes aspectos do modelo de workflow, como por exemplo: a estrutura, os dados, a ordem de execução, e etc.

4.5 Práticas de Modificação

A metodologia de Sadiq [Sadiq et al., 1998], para modificação de Processos de Workflow, segue 3 etapas principais: definição, conformidade e a efetuação da modificação. A etapa de definição descreve como deve ocorrer uma modificação, e envolve a especificação de algumas características:

- *Especificação da Política de Modificação*: modificação “estática” ou dinâmica (políticas: nivelamento, aborto, migração, adaptação ou criação).
- *Especificação das instancias que serão afetadas*: a influencia da modificação sobre as instancias do processo depende da política adotada.
- *Especificação das modificações a serem feitas no Modelo de Processo*: diversos aspectos do processo podem ser alterados durante a modificação. A alteração de um aspecto em particular afeta um elemento correspondente no modelo de workflow. As possíveis operações de modificação do modelo são: adição de novas atividades, exclusão de atividades antigas, alteração da ordem das atividades e modificação da especificação da atividade.

Após definição das especificações, ocorre a etapa de adequação das instancias correntes para que haja conformidade com as novas especificações. Instancias correntes do processo devem ser agrupadas de acordo com seu estágio de execução, caso seja possível realizá-lo de forma automatizada. Caso contrário, as instancias do processo deverão ser tratadas manualmente para que não ocorra a perda de informações.

A ultima etapa consiste na modificação em si, e em seu acompanhamento, até que a alteração atinja o grau desejado.

4.5.1 Operações de Modificação em Sistemas de Workflow

As operações de modificação suportadas pelas soluções de workflow são basicamente:

- operações para adicionar e excluir workflows;

- operações para adicionar e excluir versões, assim como operações para alterar o estado de uma versão;
- operações para alterar o conteúdo de uma versão.

A Figura 4.8 apresenta uma taxonomia das operações de modificação do modelo de workflow, proposta por Krادolder:

- (1) Modificações em uma Definição de Workflow
 - (1.1) Adicionar uma Definição de Workflow
 - (1.2) Excluir uma Definição de Workflow
- (2) Modificações em uma Versão
 - (2.1) Adicionar uma nova Versão
 - (2.1.1) Adicionar uma nova revisão
 - (2.1.2) Adicionar uma nova variação
 - (2.2) Excluir uma Versão
 - (2.3) Modificar o estado de uma Versão
 - (2.3.1) Modificar o estado de uma Versão de Transiente para Liberado
 - (2.3.2) Modificar o estado de uma Versão de Liberado ou Obsoleto para Transiente
 - (2.3.3) Modificar o estado de uma Versão de Liberado para Obsoleto
- (3) Modificações no conteúdo de uma Versão
 - (3.1) Modificações na interface de uma versão
 - (3.1.1) Modificações em definições de parâmetros
 - (3.1.1.1) Adicionar uma nova definição de parâmetro de entrada
 - (3.1.1.2) Adicionar uma nova definição de parâmetro de saída
 - (3.1.1.3) Excluir a definição de um parâmetro de entrada
 - (3.1.1.4) Excluir a definição de um parâmetro de saída
 - (3.1.2) Modificações de estados finais
 - (3.1.2.1) Adicionar um novo estado final a uma atividade
 - (3.1.2.2) Adicionar um novo estado final a uma definição de workflow
 - (3.1.2.3) Excluir um estado final de uma atividade
 - (3.1.2.4) Excluir um estado final de uma definição de workflow
 - (3.2) Modificações em uma definição de workflow
 - (3.2.1) Modificações em declarações de variáveis do workflow
 - (3.2.1.1) Adicionar uma nova declaração de variável
 - (3.2.1.2) Excluir uma declaração de variável
 - (3.2.1.3) Modificar a condição inicial de uma declaração de variável
 - (3.2.2) Modificações na declaração de fluxos de dados
 - (3.2.2.1) Adicionar uma nova definição de fluxo de dados
 - (3.2.2.2) Excluir uma definição de fluxo de dados
 - (3.2.3) Modificar uma condição final

Figura 4.8 – Taxonomia das Operações de Modificação em Sistemas de Workflow [Kradolfer, 2000].

CAPÍTULO 5

ESTUDO DE CASOS

O Estudo de Casos realizado visou identificar as práticas envolvendo modificação e reuso, adotadas por empresas nacionais, multinacionais, governamentais ou privadas, que tenham processos de negócio identificados, estando estes implementados ou não, através de sistemas de workflow.

Como o objetivo deste trabalho é levantar as práticas de reuso e de modificação, procurou-se observá-las em processos tanto dentro, quanto fora de sistemas de workflow. O estudo das práticas adotadas traz informações relevantes quando se compara a teoria com a realidade encontrada nas empresas.

As entrevistas foram realizadas entre novembro de 2003 e abril de 2004. Para a entrevista, foram consultados profissionais responsáveis pelo desenvolvimento e implantação dos sistemas, bem como usuários que acompanharam a definição de processos de trabalho e a adequação dos mesmos a novas realidades das empresas. Paralelamente às entrevistas, foram recolhidos documentos sobre os processos discutidos.

Foi realizado contato com 5 empresas. Os estudo de casos mais interessantes foram adicionados a este trabalho.

5.1 Estudo de Caso 1

A empresa pública A é da administração direta do Estado de São Paulo. É responsável, primordialmente, por controlar a arrecadação de tributos estaduais e controlar as despesas do Estado. Possui, aproximadamente, 8.000 funcionários, distribuídos nas diversas áreas da empresa e em dezesseis unidades administrativas espalhadas em todas as regiões do Estado.

A empresa A possui quatro departamentos onde alguns processos já são

implementados através do workflow. Dois são apresentados nesta monografia: o Departamento X e o Y.

Departamento X

O servidor público, diante da complexa legislação que rege sua vida funcional e salarial, em alguma situação, pode considerar-se prejudicado em relação a algum benefício. Nos casos em que se sente prejudicado, junta-se a outros servidores que estejam na mesma situação e através de um advogado entram com um processo contra o Estado. O juiz, para poder julgar a questão, entre outros elementos necessita de informações de como foram feitos os pagamentos aos servidores. Cada servidor envolvido fornece cópia dos contra-cheques onde os valores foram pagos a menos, segundo seu julgamento. O juiz então, pede que o Estado forneça a sua versão, ou seja, que envie a planilha e os comprovantes do que foi pago.

O Departamento X foi criado para fornecer informações ao juiz, para que o mesmo possa prosseguir com a análise de uma ação recebida. Baseado nestes dados e na sua interpretação da lei, o juiz determina uma *Obrigação de Fazer* (documento relatando o que o Estado deve fazer: geralmente complemento de pagamentos em caso de ganho, informação de desconto das custas do processo em caso de perda).

Descrição do fluxo antes da modificação

O fluxo implantado inicialmente era composto pela seguinte seqüência de atividades:

- 1) Departamento recebe uma ação provinda da justiça solicitando esclarecimentos sobre uma demanda provocada por um servidor.
- 2) O sistema recebe, qualifica e distribui a ação entre uma das equipes pré-definidas.
- 3) Equipe que recebeu a ação coleta informações (contra-cheques, férias e etc) para dar subsídio para o juiz.

- 4) Equipe envia formulário para unidade regional onde o servidor é lotado para ser preenchido com as informações e os documentos solicitados pelo juiz.
- 5) O formulário preenchido e os documentos relevantes são devolvidos pela unidade regional, à equipe, para a verificação da coerência das informações.
- 6) Equipe prepara um documento formal contendo o pedido feito pelo juiz, as respostas anexadas e todo o resultado da investigação realizada, de modo que permita ao juiz avaliar a qualidade das informações recebidas.
- 7) Equipe imprime todo o processo e monta a pasta que será entregue ao juiz.
- 8) A pasta é encaminhada para o diretor do departamento para que seja conferida e assinada
- 9) Depois de assinada, a pasta é encaminhada ao setor de protocolo e enviada ao poder judiciário.

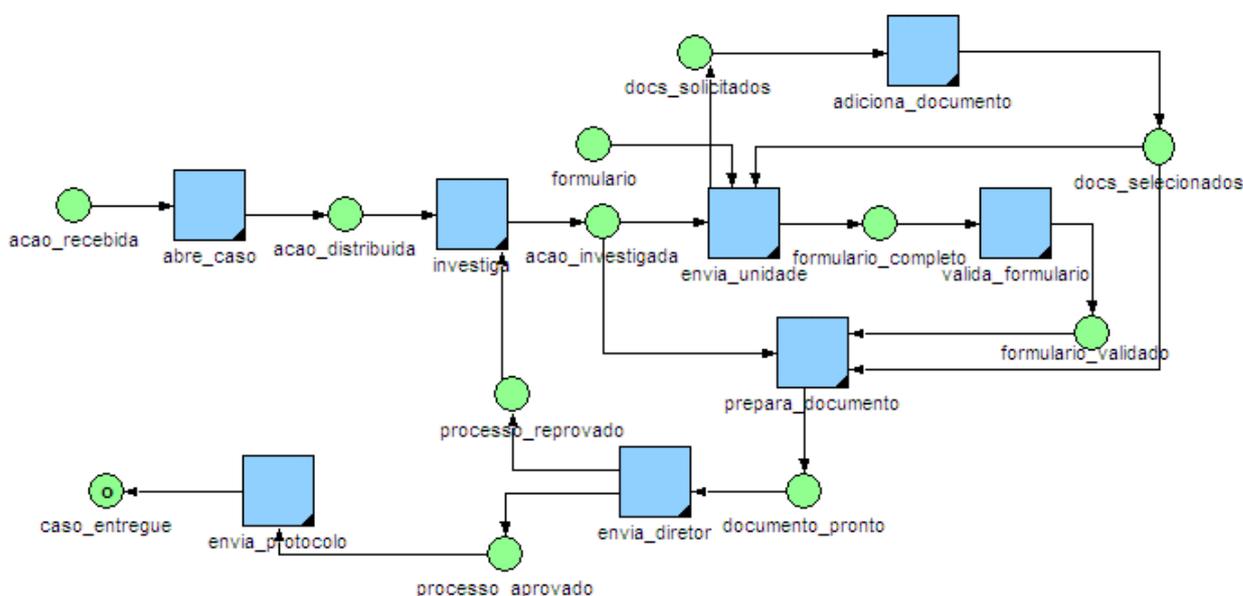


Figura 5.1 – Rede de Petri identificando o fluxo do Departamento X antes da modificação

O sistema de workflow implementado, inicialmente, cobria 60% das ocorrências. Após seis meses de uso, o departamento solicitou que o sistema passasse a atender a

totalidade dos casos, o que não foi possível, tendo em vista, a atuação junto a uma gama de exceções. Ainda assim a solução proposta pela equipe de desenvolvimento foi capaz de atender, aproximadamente, 95% de todas as ocorrências. O novo fluxo visou atender a uma lista imensa de melhorias e sugestões recebidas e passou por uma etapa intensa de testes, sendo avaliado por usuários considerados críticos pela empresa.

Descrição do fluxo após a modificação

Para atender a outras ocorrências, foi necessário que, a etapa de qualificação do Caso (*abre_caso*) fosse alterada, para que o sistema passasse a identificar para qual Entidade Externa o Formulário de recolhimento de informações deve ser enviado. Percebeu-se que, em alguns casos, o formulário deve ser enviado a outras entidades antes de passar pela Unidade Regional do servidor (*envia_rh* e *envia_procuradoria*), e que, em determinados casos, o formulário nem chega a passar pela Unidade, somente por uma das entidades externas.

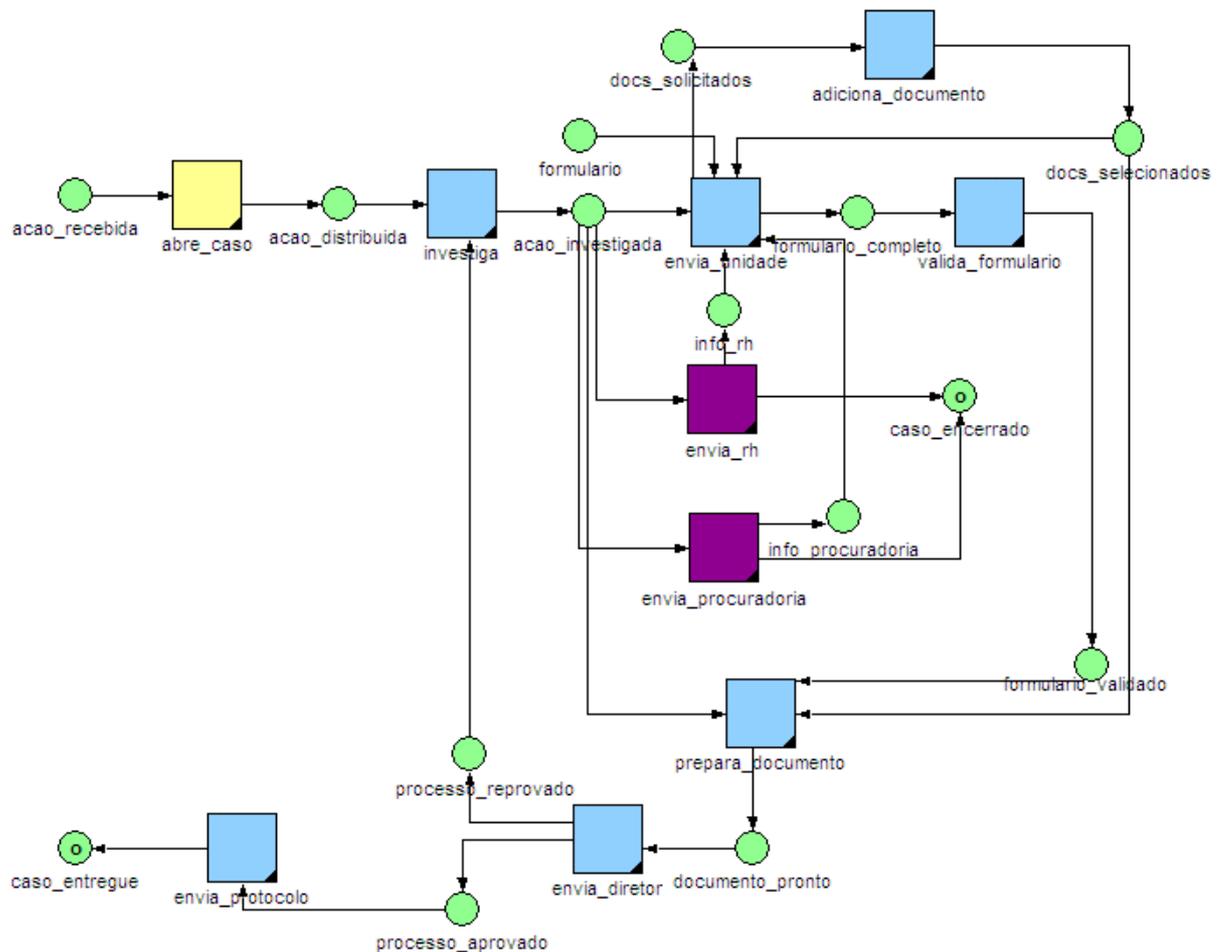


Figura 5.2 – Rede de Petri identificando o fluxo do Departamento X após a modificação. Em amarelo, encontra-se a Atividade alterada. Em lilás, as atividades adicionadas.

Descrição da prática de modificação

Para atender as solicitações de melhoria, o sistema foi modificado de acordo com a tabela abaixo:

Tabela 5.1 - Descrição das modificações realizadas no departamento X.

Componentes	Modificação		Descrição
	SIM	NÃO	
Dados		X	O banco de dados foi completamente reaproveitado, tanto a estrutura como seus Dados.
Interface		X	Todas as interfaces foram reaproveitadas. Foi possível manter as interfaces que já estavam sendo utilizadas no fluxo anterior, sem a necessidade de inserir ou excluir campos. Novas interfaces foram criadas para contemplar as Atividades que foram incluídas no fluxo.
Tecnologia		X	A tecnologia foi mantida: Visual Basic na Capital e ASP na Intranet.
Atividades – Inclusão/Exclusão	X		Foram incluídas novas atividades e um conceito de extra-workflow (tratamento de documentos fora do workflow, porém cujo retorno é esperado para dar continuidade ao fluxo). Em um determinado momento do fluxo, é gerado um Documento, que precisa ser encaminhado a uma das entidades externas. Após o preenchimento e a devolução do Documento, dá-se continuidade ao fluxo.
Atividades - Seqüência	X		Atividades tiveram a ordem de execução alterada.

O objetivo do novo fluxo era atender a uma porcentagem maior das ocorrências do Departamento. Para tanto, o fluxo existente foi mantido, passando a ser o fluxo

principal do novo fluxo. No novo fluxo, etapas foram inseridas para lidar com as outras situações, como fluxos secundários, anexadas ao fluxo principal.

Para a elaboração do novo fluxo, a Empresa em estudo optou por reutilizar o modelo anterior, salvando-o com um novo nome. As modificações apontadas acima foram realizadas sobre o novo fluxo, que na verdade era uma cópia idêntica do fluxo anterior, porém com um outro nome. A solução adotada pela Empresa visou reutilizar a definição de workflow, os aplicativos de software invocados pelo workflow e o Banco de Dados, que já estavam em uso e que deveriam ser mantidos. Na aplicação em Visual Basic, interfaces foram adicionadas. A modificação realizada visou atender a necessidades de adaptação e melhoria aos processos da Organização.

Após a reutilização integral do fluxo anterior, a Empresa percebeu que as atividades que precisavam ser adicionadas ao fluxo (*envia_rh* e *envia_procuradoria*) eram bastante parecidas com a atividade existente *envia_unidade*. Para tentar reaproveitar a atividade, a Empresa optou por retirá-la do fluxo principal e salvá-la como um sub-fluxo. Salvando-a como sub-fluxo, foi possível adotar o mesmo procedimento realizado para o reaproveitamento do fluxo principal: salvar o sub-fluxo criado com o nome das outras atividades, criando então três sub-fluxos idênticos. Estes novos sub-fluxos foram alterados para fazerem a associação a diferentes programas em Visual Basic. Os sub-fluxos foram alterados para fazer a associação conveniente e depois foram incorporados ao fluxo principal.

O fluxo inicialmente implementado permaneceu em operação enquanto o novo fluxo foi desenvolvido. A substituição do fluxo antigo pelo novo ocorreu através da política de migração (Capítulo 4.4). Como o fluxo novo contempla todas as atividades do fluxo anterior, não houve problemas em retirá-lo de operação no momento da substituição.

A ferramenta de workflow utilizada não permite que etapas ou atividades sejam salvas isoladamente em um Modelo. A solução encontrada para reutilizar atividades de um fluxo foi salvá-lo como um sub-fluxo do fluxo principal.

Departamento Y

O regulamento do ICMS do Estado de São Paulo prevê que qualquer contribuinte, cidadão ou entidade de classe possa arguir a empresa A sobre sua interpretação a respeito de pontos específicos deste regulamento.

O Departamento Y tem a função de interpretar os pontos polêmicos do Regulamento do ICMS, em atendimento a diferentes consultas do contribuinte. Para tanto, desenvolve um conjunto de atividades, a saber:

- responder a expedientes gerados por Assistentes Fiscais e Parlamentares a respeito de leis e projetos de lei;
- responder, ainda que de maneira informal, as mensagens eletrônicas (e-mails) enviadas pelos contribuintes;
- redigir decretos, portarias e boletins administrativos e tributários, assim como, a manutenção da própria atualização do Regulamento.

Para dar vazão a estes questionamentos há um rito previsto no Regulamento (RICMS). Este rito, acrescido de um conjunto de procedimentos internos definidos pelo departamento, foi mapeado através da solução de workflow.

Inicialmente foi implementado um fluxo que representava fielmente os processos manuais, porém foi rapidamente abandonado, tendo em vista a percepção dos próprios usuários de que em um processo automatizado, o processo poderia ser muito melhorado.

Descrição do fluxo antes da modificação

O fluxo inicial foi implementado pela seguinte seqüência de atividades:

- 1) Departamento recebe consultas realizadas através do guichê de protocolo, provenientes de contribuintes, cidadãos ou entidades de classe.
- 2) Servidor faz triagem das consultas e as envia para o revisor
- 3) Revisor escolhe um consultor tributário da equipe para responder a consulta

- 4) Consultor tributário responde a consulta
- 5) Revisor verifica a resposta. Se a resposta estiver satisfatória, encaminha-a ao diretor. Caso contrário, solicita ao consultor que a revise.
- 6) Se o diretor estiver satisfeito com a resposta, um e-mail é gerado para a pessoa que fez a pergunta informando que a resposta esta disponível para ser retirada no balcão. Caso contrário, a resposta é devolvida ao revisor, que por sua vez, devolve ao consultor para que a resposta seja revista.

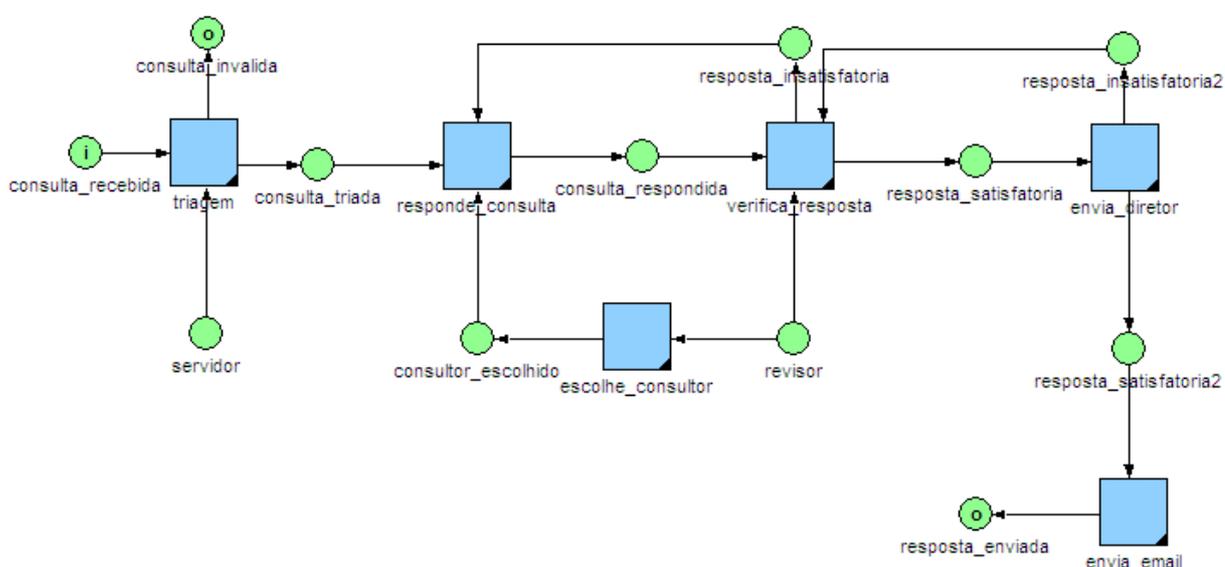


Figura 5.3 – Rede de Petri identificando o fluxo do Departamento Y antes da modificação.

Descrição do fluxo após a modificação

Para atender as melhorias solicitadas, o fluxo sofreu diversas modificações. O servidor passou a enviar a consulta diretamente a um consultor, tornando-o responsável por respondê-la (eliminação da atividade 2). Ao receber a resposta do consultor, o sistema determina qual o revisor mais adequado para verificar a resposta dada, considerando para isso a carga de tarefas atribuídas. Caso o diretor considere a resposta insatisfatória, é possível devolvê-la para o revisor ou, diretamente, para o

consultor que a respondeu. O diretor também passou a ter autonomia para mudar de revisor ou de consultor, caso considere oportuno.

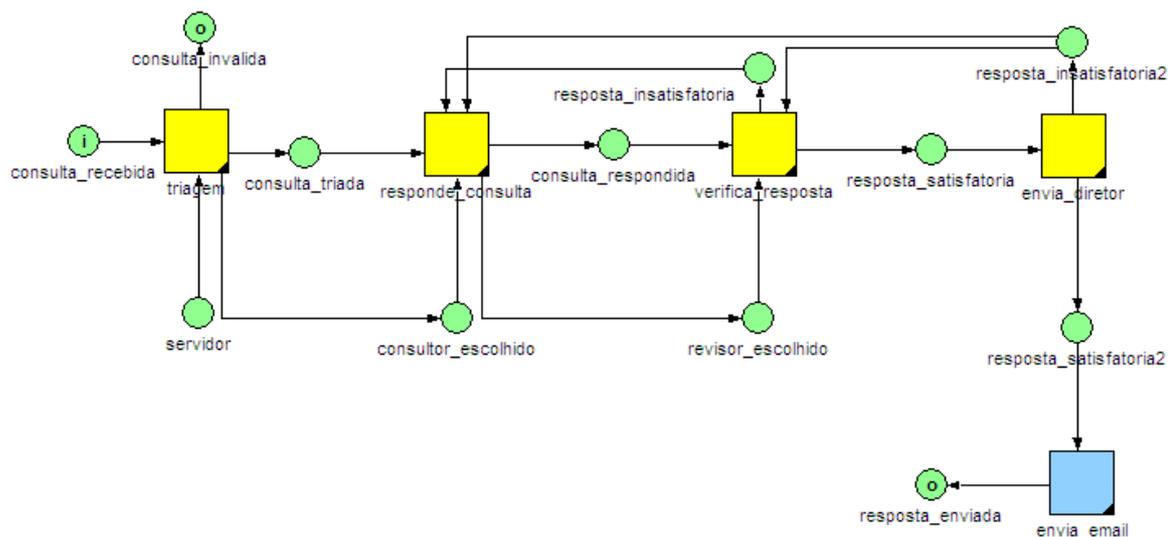


Figura 5.4 - Rede de Petri identificando o fluxo do Departamento Y depois da modificação. Em amarelo, encontram-se as atividades modificadas.

O novo fluxo trouxe agilidade e independência às atividades do Departamento, porém não atendeu completamente sua demanda, durando apenas três dias e encontra-se atualmente em fase de estudo para uma nova modificação.

Descrição da prática de modificação

Para atender a solicitação de alteração, o fluxo inicial foi modificado como segue:

Tabela 5.2 - Descrição das modificações realizadas no departamento Y.

Componentes	Modificação		Descrição
	SIM	NÃO	
Dados	X		O Banco de Dados inicialmente foi estruturado para atender a um tipo de relacionamento. Com as solicitações posteriores, o Banco de Dados não suportou as modificações e teve que ser totalmente refeito para espelhar a nova situação. A estrutura do Banco de Dados foi alterada para satisfazer o novo modelo porém os dados foram preservados.
Interface	X		A interface foi alterada. Em diversas telas foram acrescentados botões e campos de formulários. Novos formulários foram acrescentados .
Tecnologia		X	Manteve-se a mesma tecnologia.
Atividades - Inclusão/Exclusão	X		Ocorreu alteração e exclusão de atividades, que alteraram a ordem do fluxo.
Atividades – Seqüência	X		Houve modificação na seqüência das atividades, pois foi necessário excluir algumas atividades.

O fluxo inicial foi modificado visando melhorar o processo de trabalho do Departamento. A reutilização compreendeu o reaproveitamento de código em Visual Basic do sistema que é invocado pelo workflow. Para algumas atividades do workflow, a empresa utilizou novamente alguns sub-fluxos (sub-workflows) que já haviam sido desenvolvidos, da mesma forma que foi realizado pelo Departamento X. A localização destes sub-workflows exigiu um conhecimento prévio dos sub-fluxos que já haviam sido desenvolvidos. Nenhuma técnica de classificação e localização foi utilizada.

Conclusão

Após a entrevista e revendo as práticas de reuso adotadas, a Empresa A concluiu que inicialmente não foram adotados procedimentos de reaproveitamento durante o desenvolvimento dos sistemas de workflow. Com a necessidade de criação de novos fluxos e de modificação dos fluxos existentes, a empresa percebeu que, diversas atividades eram comuns aos fluxos da empresa e que portanto, mecanismos de reutilização seriam facilitadores do desenvolvimento.

A empresa A não conseguia reaproveitar atividades de um fluxo ao outro copiando uma atividade e colando em outro. Para as atividades que são freqüentemente utilizadas em diversos fluxos, a empresa criou e as adicionou em sub-fluxos. Desta forma, quando uma determinada atividade precisa ser incorporada a um fluxo, a empresa verifica se a atividade já existe em forma de sub-fluxo. Se a atividade existir em um sub-fluxo, o sub-fluxo é incorporado ao fluxo principal.

A utilização de sub-fluxos foi a saída que a empresa A encontrou para aumentar o potencial de reuso de suas atividades. A solução de workflow utilizada não permite que atividades sejam salvas isoladamente. A empresa também reutilizou rotinas de código em Visual Basic. Tanto o reuso através dos sub-fluxos, quanto das rotinas em VB foi possível pois a equipe de desenvolvimento possui conhecimento sobre as experiências passadas. Nenhuma documentação é gerada fora do ambiente do workflow.

Como forma de Documentação, a empresa preservou os Relatórios apresentados pela empresa externa, que contem atas, cronogramas e observações de desenvolvimento.

Segundo a Empresa A, o tempo médio de vida de um fluxo implementado, sem que o haja necessidade de modificações ou de re-criação é de 6 meses.

5.2 Estudo de Caso 2

A empresa B é uma indústria alimentícia multinacional. Presente em mais de 80

países, com 500 fábricas espalhadas pelos cinco continentes, atua em doze segmentos do mercado: leites, cafés, achocolatados, cereais, biscoitos, chocolates e outros. Chegou no Brasil em 1876 e possui hoje 25 fábricas no país e mais de 230 mil representantes de venda. Em território nacional, comercializa mais de 1000 itens sob sua chancela.

Todas as filiais da empresa B utilizam um mesmo sistema para controle de estoque e armazenagem. Porém, para interagir com este sistema, cada filial pode ter autonomia de criar ferramentas de apoio da maneira que lhe for conveniente.

Em 2003, a filial brasileira da empresa B percebeu que precisava de fluxos que interagissem com o sistema de uso mundial para agilizar o processo de Cadastramento de Produtos, Matérias-primas, Representantes de Venda, Clientes e Fornecedores.

Logo após a implementação dos fluxos de apoio (fluxos de cadastro) ao sistema de uso mundial, diversos processos de trabalho da empresa começaram a ser implementados via solução de workflow. Através do trabalho inicial de desenvolvimento dos fluxos de cadastro, foi possível criar templates de fluxos básicos, visando a reutilização para a criação de novos fluxos. A empresa desde o início procurou desenvolver templates, códigos e módulos reutilizáveis (em aplicações invocadas pelo workflow). Criou-se um template básico com as características da empresa, para que pudesse ser reutilizado e pudesse manter uma apresentação uniforme em todas as telas dos sistemas da empresa. Para elaborar as páginas particulares a cada sistema externo, utilizou-se o método "copy & paste" para aproveitar pedaços de código já desenvolvidos e, em seguida adaptou-se o código segundo as regras de negócio necessárias ao sistema.

A empresa B possui disponível para reutilização: templates, interfaces com o usuário e interfaces de comunicação de plataformas. Como todos os sistemas da empresa devem manter comunicação com o sistema de uso global para troca de informações, criou-se uma interface de comunicação que pode ser reaproveitada por todos os sistemas. A adaptação das interfaces de comunicação aos sistemas de workflow exigiu grande esforço de programação, portanto a reutilização do template de integração entre as plataformas foi fundamental para a empresa.

A empresa B conta até os dias atuais, com o apoio de mão de obra terceirizada para a manutenção dos fluxos implementados e para a criação de novos fluxos no workflow.

Lançamento de Layout de Embalagens

Um dos processos de negócio implementados refere-se ao Lançamento de layout de embalagens. Todo o lançamento de produtos envolve a criação de embalagens. Produtos já lançados, muitas vezes possuem suas embalagens alteradas para seguir uma tendência de mercado.

O processo de lançamento de um novo produto envolve diversos departamentos da empresa: Logística, Marketing, Qualidade, Comunicação Gráfica entre outros. A equipe de Marketing apresenta o produto, a finalidade e o segmento consumidor em uma reunião com membros de todos os departamentos, que analisam a proposta e, em caso de aceitação, dá-se início ao processo no workflow. O processo implementado no workflow tem o objetivo de validar o que foi discutido na reunião.

Com o início do fluxo, cada departamento torna-se responsável por descrever as características que o produto deve possuir, para atender aos requisitos e aos padrões exigidos de sua área, como segue:

Departamento de Normas Alimentares: equipe acessa o sistema e preenche formulários para coleta de dados referentes a normas e o formato que a norma deve ser exibido

Departamento de Embalagens: equipe acessa o sistema e fornece informações referentes as medidas que a embalagem deve ter. Este departamento deve considerar a medida da embalagem de uma unidade do produto, assim como as medidas da caixa, contendo mais de um produto.

Departamento de Programação de Produção (Fábrica): não insere informações no fluxo porém deve ser alertado durante o processo para que seja possível controlar a compra de material, principalmente no caso em

que a embalagem é alterada.

Departamento de Marketing: o departamento além de abastecer o sistema com as informações que foram apresentadas na reunião, deve fornecer subsídios que comprovem que a idéia lançada realmente irá satisfazer o público alvo da campanha.

Departamento de Qualidade: o departamento deve informar ao sistema quais são os requisitos de qualidade para que o produto atinja os padrões de qualidade da empresa.

Departamento de Supply-Chain: departamento fornece informações quanto aos possíveis fornecedores e a quantidade de matéria-prima que é recebida por mês, tanto para a elaboração do produto quanto para a criação da embalagem.

Depois que os Departamentos preenchem as informações referentes a sua área, o Departamento de Comunicação Gráfica juntamente com agencias gráficas, desenham a embalagem de acordo com as especificações do fluxo. Nesta etapa, a operação sai do fluxo do workflow e volta somente quando o modelo está pronto. Do workflow gera-se um documento, que é enviado a agencia com as especificações coletadas.

Assim que a agencia devolve o modelo gráfico solicitado, o Departamento de Comunicação Gráfica publica uma imagem do modelo da embalagem, que contém detalhes de cores e até das dobras que a embalagem deverá ter. Após a publicação pelo Departamento de Comunicação Gráfica, todos os integrantes do fluxo são alertados, para que possam validar os requisitos solicitados.

Os departamentos que tiverem alguma objeção quanto ao modelo gráfico apresentado, devem solicitar uma revisão via sistema, e o modelo gráfico volta para ser analisado pela agencia. Assim que a agencia atende as solicitações do Departamento, o modelo é novamente inserido no sistema e todos os Departamentos são novamente alertados.

Descrição do fluxo antes da modificação

- 1) Departamentos envolvidos no processo de lançamento de novas embalagens se reúnem para conhecer as idéias do departamento de marketing
- 2) Após a reunião, o departamento de marketing insere as informações sobre a idéia apresentada no workflow e dá início ao fluxo
- 3) Departamentos envolvidos acessam o sistema e preenchem as informações pertinentes a sua área
- 4) Departamento de comunicação gráfica coleta o documento gerado pelo workflow para levar a agencia gráfica
- 5) Depto de comunicação gráfica juntamente com a agencia gráfica desenham a embalagem segundo as especificações dos departamentos
- 6) Depto de comunicação gráfica publica o modelo da embalagem criada no workflow
- 7) Deptos envolvidos avaliam se o modelo atende as especificações solicitadas. Se satisfizer todos os departamentos, o modelo dá-se como aprovado e encerra-se o fluxo. Caso algum departamento reprove o modelo apresentado, o modelo volta para o depto de comunicação gráfica revisá-lo, juntamente com a agencia gráfica.
- 8) Modelo revisado volta a ser inserido no workflow para ser novamente avaliado
- 9) Workflow avisa a todos os departamentos envolvidos que o modelo revisado esta disponível para nova avaliação

E assim o fluxo permanece, da atividade 7 a 9, até que todos os departamentos tenham aprovado o modelo.

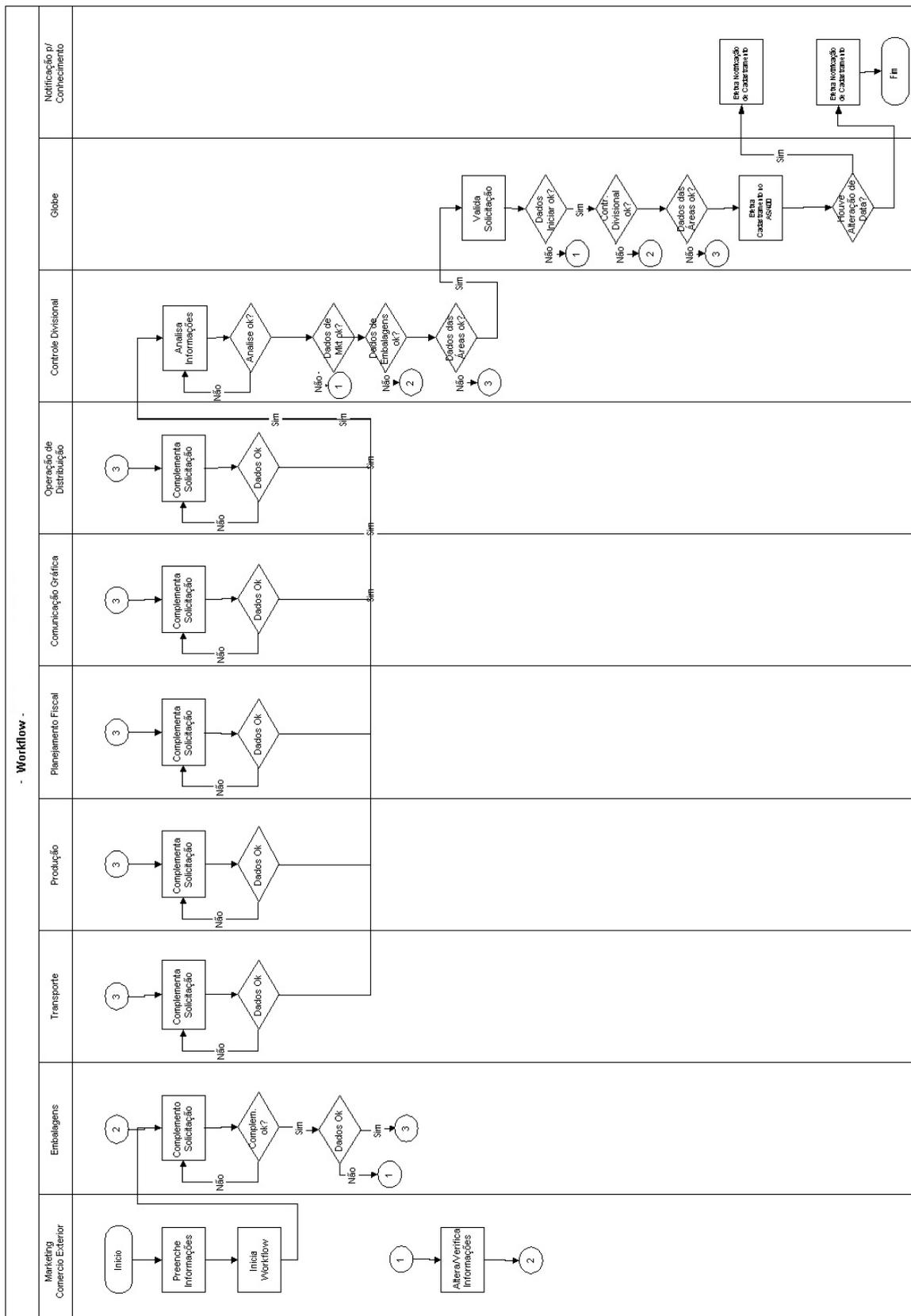


Figura 5.5 – Fluxo para controle de lançamento do layout de embalagens antes da modificação

Figura 5.5 – Fluxo para Controle de Lançamento do Layout de Embalagens implantado inicialmente.

Descrição do fluxo após a modificação

Como são gerados de 200 a 300 fluxos por mês, a Empresa B decidiu alterar o fluxo anterior, para que somente os Departamentos que tenham apresentado alguma objeção fossem alertados quando o modelo revisado fosse publicado, para não confundir os departamentos que já não possuíam pendências em instancias anteriores (alteração na atividade 8 – workflow passou a identificar nesta atividade, quais os departamentos que deveriam ser notificados)

Tendo em mente a modificação do sistema, a empresa B optou também por alterar a forma de comunicação entre a empresa e as agencias gráficas. No novo fluxo, a empresa B passou a se comunicar com as agencias através de sua extranet, que é controlada pelo sistema de workflow, dando maior controle a empresa B, resultando em alteração na atividade 4, já que agora a agencia passou a ser capaz de acessar e fazer parte do workflow.

Para atender as solicitações de melhoria, o sistema foi modificado de acordo com a tabela abaixo:

Tabela 5.3 - Descrição das modificações realizadas para o fluxo da empresa B.

Componentes	Modificação		Descrição
	SIM	NÃO	
Dados		X	A estrutura do banco de dados foi mantida.
Interface	X		Foi criada uma interface para ser acessada pelas agencias, em um ambiente externo a Empresa B.
Tecno[Carol1]logia	X		Foi necessário desenvolver a interface da extranet que é utilizada para o acesso externo das agencias. Para esta participação, utilizou-se uma linguagem de programação voltada para os padrões para desenvolvimento web da empresa.
Atividades – Inclusão/Exclusão	X		Tarefas de Atividades foram alteradas, excluídas e incluídas no novo fluxo.
Atividades - Seqüência		X	Apesar das modificações realizadas nas tarefas as atividades não foram alteradas.

Conclusão

A empresa B considera que, apesar do grande esforço de programação envolvido nas atividades de reutilização, práticas de reuso foram e continuam sendo adotadas pela empresa. Uma mentalidade focada no reuso de templates através do workflow e de código e módulos dos aplicativos externos, foi uma escolha fundamental pois o ciclo de vida dos fluxos de processo são muito curtos e estão constantemente em modificação de melhoria e de adaptação.

Para reutilizar as atividades que estão presentes em praticamente todos os processos da empresa, a empresa B optou por criar templates do modelo de workflow contendo estas atividades. Para reutilizar um template, a empresa salva-o com o nome do fluxo que irá criar e realiza as modificações necessárias sobre o template com novo nome.

A cada modificação do workflow, a empresa gera uma nova versão do sistema. A empresa geralmente procura realizar a migração de modelos de forma gradativa (política de nivelamento), a não ser que a versão corrente esteja trazendo problemas. O gerenciamento de configurações é realizado com o lançamento de versões. Caso a empresa deseje utilizar uma versão antiga do sistema, é possível recuperar a versão desejada.

Para a localização de códigos reutilizáveis, a empresa B não utiliza nenhuma ferramenta de classificação e localização de templates e de módulos. A localização de componentes reutilizáveis ocorre baseada no conhecimento prévio dos desenvolvedores.

5.3 Estudo de Caso 3

A empresa C é uma multinacional que atua em diversos segmentos de mercado, como: construção, transporte, mineração, energia entre outros. A empresa projeta, fabrica e comercializa produtos classificados em duas categorias: máquinas em geral (rodoviárias, de terraplenagem, agrícolas e equipamentos para movimentação de material), e motores/turbinas.

A empresa C está presente em 200 países, com 105 fábricas, 23 centros de distribuição de peças, 14 centros de remanufatura, 15 centros de treinamento de serviço, um centro técnico de pesquisas, e uma rede internacional de 220 revendedores, empresas independentes que possuem 1.844 pontos de vendas com cerca de 89 mil empregados. A instalação da empresa no Brasil ocorreu em 1954. Em 2004, a filial da empresa instalada no país possui cerca de 3.000 funcionários, sendo que 800 atuam no escritório e o restante em operações de fábrica.

A utilização de sistemas de workflow iniciou-se em 2002. A empresa utiliza a solução de workflow adotada pela Corporação. A princípio, houve participação de equipe externa para a implantação do sistema. Desde que os fluxos de processo foram implementados via solução de workflow, não houve necessidade de modificá-los.

Nos últimos anos, a empresa aprimorou diversos processos de negócio que são utilizados fora do sistema de workflow, para obter maior controle sobre os processos.

Programa de Captação de Idéias

A empresa C possui uma política de redução de custos, onde todos os funcionários da empresa são estimulados a apresentar idéias que possam diminuir custos nas diversas áreas da empresa. Todas as idéias são estudadas para verificação de coesão. Se a idéia apresentada pelo funcionário for aprovada, implementada e resultar em reduções de custo para a empresa, o funcionário recebe uma porcentagem da redução obtida, como forma de gratificação.

As sugestões para redução de custo podem ser referentes às diversas áreas da empresa, porém para identificação do tipo de redução que a sugestão pode ocasionar, a sugestão é sempre direcionada a um dos seguintes Grupos:

- Grupo de Certificação: grupo que avalia a gestão dos processos das áreas da empresa. Idéias referentes à modificação de processos de trabalho são enviadas a princípio para este grupo;

- Grupo de 6 Sigma: grupo que organiza times para execução de atividades. Se a idéia exigir trabalho em time para ser avaliada ou envolver modificação de produtos, a idéia é encaminhada a este grupo;

Todas as idéias são registradas através do sistema e são enviadas para um revisor, que identifica a qual grupo deve enviá-la. Assim que um grupo recebe a idéia, pode solicitar que responsáveis das diversas áreas da empresa (departamento de recursos humanos, tecnologia da informação, marketing, engenharia e etc), participem do processo de julgamento e do estudo de viabilidade da idéia.

Depois de analisada, decide-se pela implementação ou não da idéia e o funcionário que a gerou é avisado. Se a empresa C decidir pela implementação da idéia, a gratificação que será atribuída ao funcionário é calculada após um ano que a idéia entrar em operação.

Descrição do fluxo antes da modificação

- 1) Funcionário com uma idéia acessa a intranet da empresa.
- 2) Funcionário identifica-se através de tela de login da intranet.
- 3) Funcionário preenche formulário eletrônico para coleta de informações, para que seja possível descrever a idéia.
- 4) Formulário devidamente preenchido é enviado para um Revisor.
- 5) Revisor identifica o grupo que deve ser notificado e encaminha o formulário ao Grupo.
- 6) Grupo analisa o formulário e solicita participação de responsáveis de uma das áreas da empresa para ajudar a julgar a idéia
- 7) Responsável analisa a idéia e envia resposta ao Funcionário que a enviou.

Com a implantação do sistema, percebeu-se que melhorias poderiam ser realizadas para dar maior agilidade ao processo e controlar melhor os prazos de resposta das áreas e a atuação dos grupos. O processo inicial não permitia que uma idéia recebida por um departamento, pudesse circular por outras áreas da empresa

para complementação de informações, mesmo em casos onde a área havia sido escolhida equivocadamente (somente uma área poderia ser invocada por idéia).

Após 04 meses em funcionamento, o sistema foi modificado. O processo implementado pelo novo sistema funciona através das seguintes atividades:

Descrição do fluxo após a modificação

- 1) Funcionário com idéias de redução de custo acessa a intranet da empresa.
- 2) O sistema identifica o usuário a partir da senha de acesso que foi utilizada para acessar a rede da empresa. Somente caso o computador seja de uso compartilhado, é necessário que o usuário se identifique através da intranet da empresa.
- 3) Funcionário preenche formulário eletrônico para coleta de informações. No formulário, é possível selecionar dentre diversos assuntos, a qual melhor enquadra-se a idéia.
- 4) De acordo com o assunto e a função/cargo do usuário que preencheu o formulário, o sistema calcula a prioridade do assunto e o distribui perante um dos Grupos.
- 5) Grupo solicita participação de responsáveis de uma ou mais áreas da empresa, no intuito de avaliar a idéia.
- 6) Responsável pela área invocada pelo Grupo, preenche o formulário com informações pertinentes a sua área e o devolve ao Grupo.
- 7) Grupo avalia todas as informações coletadas e responde ao funcionário, informando quanto à aceitação ou não da idéia.

Tabela 5.4 - Descrição das modificações realizadas para o fluxo da empresa C.

Componentes	Modificação		Descrição
	SIM	NÃO	
Dados		X	A estrutura do banco de dados alterada para suportar novos relacionamentos e novos campos.
Interface	X		Alguns campos foram adicionados na interface existente. Interfaces novas foram criadas.
Tecno[Carol2]logia	X		Utilizou-se a mesma tecnologia.
Atividades – Inclusão/Exclusão	X		Tarefas de Atividades foram alteradas, excluídas e incluídas no novo sistema.
Atividades - Seqüência		X	Excluindo e alterando as atividades, a seqüência das atividades não pode ser mantida.

A empresa C possui documentos internos que visam instruir a equipe de desenvolvimento, em relação a normas de documentação e de geração de componentes, para que um padrão seja seguido, porém nenhuma ferramenta de classificação e localização é utilizada. Componentes com elevado potencial de reuso são armazenados em uma biblioteca. A localização dos componentes reutilizáveis ocorre baseada no conhecimento prévio da equipe de desenvolvimento.

GOL - Programa de Captação de Idéias

✦ **Dados dos Autores**

Autor 1: 00098583 ANDRE GILHO

 [Nova Pesquisa](#)

✦ **Dados da Idéia**

Seção em que se aplica a idéia :

Descreva o problema ou a situação atual :

Descreva a Idéia :

✦ **Anexos da Idéia**

[Inserir Anexos](#) [Remover Anexos](#)

Figura 5.6 – Interface de coleta de informações para o Programa de Captação de Idéias

Este fluxo não está implementado no sistema de workflow. A modificação ocorreu sobre o sistema desenvolvido pela empresa. Para controle de versões, a empresa utiliza uma ferramenta de CVS. As alterações foram realizadas sobre o sistema existente, dando origem a uma nova versão.

Foi possível reutilizar o componente de autenticação de usuários, que foi desenvolvido para ser incorporado em outros sistemas da empresa. Para localizar este componente, a equipe de desenvolvimento analisou o código de um sistema que possuía esta funcionalidade, para descobrir qual era o componente que realizava esta operação. As interfaces com o usuário foram reutilizadas. A empresa criou uma interface-modelo para o sistema e todas as interfaces foram criadas a partir dela. As

outras modificações envolveram criação e, portanto, novo esforço de desenvolvimento.

Conclusão

A empresa C possui uma preocupação com a padronização de componentes e de documentação gerada pela equipe de desenvolvimento, porém não utiliza métodos para identificar facilmente um componente dentro de sua biblioteca ou repositório.

A modificação realizada no Programa de Captação de Idéias foi perfectiva, visando melhorar a funcionalidade do sistema. Tratando-se de um software desenvolvido pela empresa, observa-se que o uso de interfaces-modelo e de componentes reutilizáveis assemelha-se aos métodos de reuso e modificação que podem ser utilizados em sistemas de workflow.

CAPÍTULO 6

CONSIDERAÇÕES FINAIS

A implementação de sistemas de workflow, como forma de solucionar os mais diversos problemas encontrados em empresas, tem crescido recentemente. A vasta utilização de sistemas de workflow explica a quantidade de soluções de workflow presentes hoje no mercado. No entanto, os sistemas de workflow atuais, são mais adequados a usuários não-técnicos do que a desenvolvedores de software que necessitam de ferramentas para agilizar e dar mais qualidade ao processo de desenvolvimento de workflows.

Sistemas de workflow tradicionais têm como objetivo ser um pacote de funcionalidades voltado a usuários não-técnicos, ou seja, usuários sem conhecimento em programação ou desenvolvimento de sistemas. A rapidez com que empresas precisam otimizar ou expandir seus processos de negócio fez com que, técnicas de reutilização e modificação se tornassem fundamentais para obter maior agilidade e qualidade nos processos de mudança. O público-alvo dos sistemas de workflow está mudando de usuários não-técnicos para usuários que possuem maior conhecimento em formas de agilizar o desenvolvimento e a modificação de processos implementados através do workflow.

Sistemas que focam usuários não-técnicos devem prover ferramentas de suporte como construtores gráficos de processo, projeto de formulários entre outros, tornando visível ao usuário somente informações que o usuário pode entender, escondendo os detalhes que são considerados irrelevantes. Enquanto isto permite que usuários não programadores utilizem sistemas de workflow, por outro lado dificultam operações de reuso e de customização. Em contraste, usuários que necessitam de maior agilidade para o processo de desenvolvimento de workflows, possuem outras necessidades. Esses usuários precisam de ferramentas que são fáceis de entender, reusar e integrar com um grande número de ambientes e aplicações. Sistemas de workflows voltados à

desenvolvedores devem permitir a customização das funcionalidades disponíveis e o reuso de parte das funcionalidades de acordo com a necessidade, em diferentes contextos. Como é possível notar através do Estudo de Casos apresentado, desenvolvedores estão tendo que adotar soluções próprias para lidar com a ausência de suporte, das soluções atuais, ao reuso e a modificação de workflows.

Apesar de encorajados pelos próprios desenvolvedores de soluções de workflow, a utilização de templates e de sub-workflows atende parcialmente as necessidades de reuso. Na maioria das vezes, estes templates e sub-workflows precisam ser reutilizados e depois modificados para atenderem aos requisitos da empresa. Devido a natureza “monolítica” dos sistemas de workflow tradicionais, workflows podem somente ser reutilizados de uma maneira “ou tudo ou nada”. As soluções encontradas pelas empresas entrevistadas visam facilitar esta operação. Empresas encontraram suporte da seguinte forma:

- Utilizando templates: através de templates, é possível criar um modelo padrão, que pode ser salvo com outro nome e então modificado para que, atividades que interessem possam ser reutilizadas e as que não são de interesse, possam ser excluídas ou modificadas no novo modelo;
- Utilizando sub-workflows: com sub-workflows, a empresa percebeu que poderia aumentar seu potencial de reuso, dividindo um fluxo maior e mais complexo, em fluxos pequenos. Estes fluxos pequenos podem então ser adicionados a um fluxo principal. Sub-workflows passaram a ser compostos por até mesmo uma única atividade.

Empresas não têm ferramentas que facilitem o reuso das partes desejadas através workflow, isolando-as de partes que não interessam. Não é possível através da ferramenta de workflow, especificar o que se deseja reutilizar. Da mesma forma, diversas soluções não possuem ferramentas capazes de facilitar a operação de modificação. Como visto no capítulo 4.5.1, dentre as operações de modificação de workflow, não é destacada a opção de modificação da ordem da seqüência das atividades, pratica bastante comum que pode ser observada através das modificações realizadas pelas empresas entrevistadas no Capítulo de estudo de casos.

Percebe-se que, ao comparar-se a literatura de reuso e modificação de sistemas de workflow, o processo de desenvolvimento destes sistemas assemelha-se bastante ao processo de desenvolvimento de softwares discutidos na área de Engenharia de Software, tanto na forma em que os sistemas são projetados, modificados, reutilizados quanto nas propostas de como classificá-los. A literatura trata o reuso e a modificação de workflows como a engenharia de software trara o reuso e a modificação de softwares, dando a impressão que a tecnologia de workflow é uma sub-área da engenharia de software. Não existem conceitos próprios ao reuso e a modificação de workflows.

Sistemas de software que não são desenvolvidos para o reuso dificilmente serão reutilizados. O mesmo ocorre com sistemas de workflow. Ferramentas de classificação e localização de componentes reutilizáveis facilitam a recuperação tanto de módulos de software quanto de sub-workflows, templates ou outros componentes de workflow reutilizáveis.

Porém, o que seria um componente reutilizável em sistemas de workflow? Seria possível reutilizar uma atividade isoladamente, ou o modelo do fluxo sem que ele venha acompanhado do histórico de execução? Esta monografia visou levantar esta entre outras questões. Até o presente momento, portanto, a dúvida permanece. Seria possível reutilizar somente através de templates e sub-workflows?

Diversos autores consideram que o paradigma orientado a objetos não deve ser associado a sistemas de workflow pois workflow são orientados-a-função e não a objetos. Através do paradigma orientado a objetos, a representação do controle do fluxo deixa de ser tão nítida. A decomposição em classes, característica típica da arquitetura orientada a objetos, não enfatiza o controle do fluxo, distribuindo-o entre diferentes objetos, tornando o controle global do fluxo e seu comportamento menos visíveis, indo de encontro com o princípio fundamental dos sistemas de workflow. No entanto, o desafio ao construir sistemas de workflow através de uma arquitetura orientada a objetos, deve recair em prover abstrações que mantenham uma representação explícita do controle do fluxo, sem violar os princípios de um bom projeto orientado a objetos.

É possível observar no estudo de casos, que o tempo de vida dos fluxos (tempo

que os fluxos permanecem sem alteração) é bastante curto, o que deveria chamar a atenção de fornecedores de soluções de workflow. A necessidade de mudança corresponde diretamente à necessidade de reuso e de modificação. Ferramentas facilitadoras destas operações são imprescindíveis para que empresas possam contar com sistemas de workflow para acompanhar as mudanças do mercado.

Diversas soluções de workflow têm dado atenção a estas necessidades. Apesar de não possuírem ferramentas facilitadoras das operações de reuso e de modificação, os fornecedores de soluções tem procurado orientar seus usuários a como proceder, para obter maiores vantagens do reuso, a partir das características disponíveis na solução. Através do Estudo de Casos é possível observar a distância existente entre a teoria e a prática referente ao reuso e a moificação de workflows. Enquanto a teoria preocupa-se com o reuso de definições de workflow, com o gerenciamento de versões e variações, fornecedores de soluções de workflows e empresas voltam-se ao uso de templates e sub-workflows.

Reuso e modificação são temas maduros e de grande interesse à Engenharia de Software. Tendo isto em mente, este trabalho objetivou estudar os efeitos do reuso e da modificação na tecnologia de workflow. Este trabalho também visou comparar os conceitos encontrados na literatura existente sobre reuso e modificação de worklfows com as práticas encontradas nas empresas para enfrentar tais necessidades. No entanto, não foram apontadas soluções para aproximar a literatura à prática apresentada. Estudos complementares visando identificar uma melhor prática de reuso e de modificação, enfocando a teoria discutida, podem trazer grandes benefícios a este mercado emergente.

REFERÊNCIAS BIBLIOGRÁFICAS

[Ambler, 1997]. AMBLER, Scott W. *Análise e Projeto Orientados a Objeto*. COSTA, C. (tradutor). Infobook, 1997.

[Appleton, 2000]. APPLETON, Brad. *Patterns and software: Essentials concepts and terminology*. <http://www.cmcrossroads.com/bradmapp/docs/patterns-intro.html>. Consultado em fevereiro de 2004.

[Baresi et al., 1999]. BARESI, L.; CASATI, F.; CASTANO, S.; FUGINI, M. G.; MIRBEL, I.; PERSINE, B. WIDE. *Workflow Development Methodology*. In: Proceedings Conference on Work Activities Coordination and Collaboration. WACC, 1999.

[Becker et al., 2002]. BECKER, Jörg; zur MUEHLEN, Michael; GILLE, Marc. *Workflow Application Architectures: Classification and Characteristics of Workflow-based Information Systems*. In: FISHER, Layna. (Ed): *Workflow Handbook*, Lighthouse Point, Florida, 2002.

[Behforooz, 1996]. BEHFOROZ, Ali & HUDSON, Frederick. *Software Engineering Fundamentals*. New York. Oxford University Press. 1996.

[Bieman, 1992]. BIEMAN, James M. *Deriving Measures of software reuse in object oriented system*. In: Formal Aspects of Measurement. DENVIR, T; HERMAN, R; WHITTY, R (editors). Spring-Verlag, 1992.

[Cappelli et al., 2003] CAPPELLI, Claudia; BORGES, Marcos R. S.; ARAUJO, Renata. *Qualidade do Processo de Manutenção de Software*. <http://www.uniriotec.br/~spin-rio/arquivos/cits.pdf>. Consultado em abril de 2004.

[Cordeiro, 2000]. CORDEIRO, Marco A. *Manutenibilidade de Software*. <http://www.pr.gov.br/batebyte/edicoes/2000/bb102/manutenibilidade.htm>. Consultado em março de 2004.

[Dextra, 2003]. DOSSO, Luis. *Entendendo a Programação Orientada a Aspectos*. <http://www.dextra.com.br/empresa/boletim/0310-04/04tecnologia.htm>. Consultado em dezembro de 2003.

[Kafura, 1996]. KAFURA, Dennis. *Object-Oriented Software Design and Construction*. <http://people.cs.vt.edu/~kafura/cs2704/basic.concepts.html>.

[Kardell, 1997]. KARDELL, Magnus. *A Classification of Object-Oriented Design Patterns*. Master Thesis. Umea University. <http://www.cs.umu.se/~jubo/ExJobs/MK/patterns.htm>. Consultado em fevereiro de 2004.

[Kradolfer, 2000]. KRADOLFER, Markus. *A workflow meta-model supporting dynamic reuse-based model evolution*. http://www.ifi.unizh.ch/ifiadmin/staff/rofrei/Dissertationen/Jahn_2000/thesis_kradolfer.pdf. Consultado em outubro de 2003

[Krishnan et al., 2001] KRISHNAN, Rupa; MUNAGA, Lalitha; KARLAPALEM, Kamalakar. *XdoC-WFMS: A Framework for Document Centric Workflow Management System*. ER Workshopd, 2001.

[Kruke, 1996]. KRUKE, Vegar. *Reuse in Workflow Modeling*. <http://www.pw.ntnu.no/~crukis/diploma/diploma.html>. Consultado em outubro de 2003.

[Manolescu, 2001]. MANOLESCU, Dragos-Anton. *Micro-Workflow: A workflow architecture supporting compositional object-oriented software development*. Thesis do Doctor of Philosophy in Computer Science. University of Illinois, 2001.

[Marshak, 1994]. MARSHAK, Ronni T. *Workflow White Paper-An overview of workflow software*. Workflow Computing Report, V. 16, n. 10, 1994.

[Oliveira, 2004]. OLIVEIRA, Eric C. M. *Introdução a Design Patterns*. http://www.linhadecodigo.com.br/artigos.asp?id_ac=345 Consultado em 16 de maio de 2004

[Pietro-Diaz et al., 1987]. PIETRO-DIAZ, Rubens & FREEMAN, Peter. *Classifying Software for reusability*. IEEE Software, V. 4, n.1, 1987.

[Pressman, 1992]. PRESSMAN, Roger S. *Software Engineering: A practitioner's*

Approach. 3 ed. McGraw-Hill, 1992.

[Sadiq et al., 1998]. SADIQ, Shazia W. & ORLOWSKA Maria E. *Dynamic Modification of Workflows*. <http://www.dstc.ed.au/praxis/publications/ssadiq-cstr442.1998.PDF>. Consultado em março de 2004.

[Sarin et al., 1994]. SARIN, Sunil K. & ABBOT, Kenneth R. *Experiences with workflow managements: Issues for the next generation*. Proceedings of Computer-Supported Cooperative Workf. Chapel Hill, USA, 1994.

[Schach, 1992]. SCHACH, S. R. *Practical Software Engineering*. IRVING-ACSEN (Ed). Homewood, 1992.

[Sommerville, 2001]. SOMMERVILLE, Ian. *Software Engineering*. 6 ed, Pearson Education Limited, 2001.

[Weske et al., 1999]. WESKE, M.; GOESMANN, T.; HOLTEN, R.; STRIEMER, R. *A reference model for workflow application development processes*. In: Proceedings of Conference on Work Activities Coordination and Collaboration. WACC, 1999.

[WFMC, 1995]. Workflow Management Coalition. *The Workflow Reference Model*, Document Number TCOO-1003, 1995.

[Yao et al., 2004]. YAO, Harning & ETZKORN, Letha. *Towards a Semantic-based approach for software reusable components classification and retrieval*. ACMSE, N.2-3, 2004

[Zar, 2000]. ZAR, David M. *Workflow Basics*.

http://ge.ee.wuste.edu/dzar/tutorials/fx/fx_3/3_basic.html. Consultado em agosto de 2004.

[zur Muehlen et al, 1999]. zur MUEHLEN, Michael & BECKER, Jörg. *Workflow Process Definition Language – Development and Directions of a Meta-Language for Workflow Processes*. In: Bading, L et al. (Ed.): Proceedings of the 1st KnowTech Forum, 1999. [http://www.workflow-research.de/Publications/PDF/MIZU.JOBE-KT\(1999\).pdf](http://www.workflow-research.de/Publications/PDF/MIZU.JOBE-KT(1999).pdf). Consultado em março de 2004.